Distributed Off-Policy Temporal Difference Learning Using Primal-Dual Method

DONGHWAN LEE1, (Member, IEEE), DO WAN KIM2, AND JIANGHAI HU3, (Member, IEEE)

1Department of Electrical Engineering, KAIST, Daejeon 34141, South Korea
2Department of Electrical Engineering, Hanbat National University, Daejeon 34158, South Korea
3Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47906, USA

Corresponding author: Donghwan Lee (donghwan@kaist.ac.kr)

This work was supported in part by the National Research Foundation under Grant NRF-2019R1F1A1063691, Grant NRF-2021R1H1A3058581, and Grant NRF-2021R1F1A1061613; in part by the National Science Foundation under Grant 2014816 and Grant 2038410; and in part by the BK21 FOUR from the Ministry of Education, Republic of Korea.

ABSTRACT The goal of this paper is to provide theoretical analysis and additional insights on a distributed temporal-difference (TD)-learning algorithm for the multi-agent Markov decision processes (MDPs) via saddle-point viewpoints. The (single-agent) TD-learning is a reinforcement learning (RL) algorithm for evaluating a given policy based on reward feedbacks. In multi-agent settings, multiple RL agents concurrently behave, and each agent receives its local rewards. The goal of each agent is to evaluate a given policy corresponding to the global reward, which is an average of the local rewards by sharing learning parameters through random network communications. In this paper, we propose a distributed TD-learning based on saddle-point frameworks, and provide rigorous analysis of finite-time convergence of the algorithm and its solution based on tools in optimization theory. The results in this paper provide general and unified perspectives of the distributed policy evaluation problem, and theoretically complement the previous works.

INDEX TERMS Reinforcement learning (RL), multi-agent systems, convergence, temporal difference (TD) learning, machine learning, primal-dual method.

I. INTRODUCTION

Temporal difference (TD)-learning [1], [2] is a reinforcement-learning (RL) algorithm to evaluate a given policy without the model knowledge of a Markov decision process (MDP), which is called the policy evaluation problem. In a multi-agent RL setting, multiple RL agents concurrently behave and evaluate a policy based on the global reward, which is a sum of the local rewards, where each agent only receives a local reward. The learning strategies are distributed in the sense that each agent only has access to its local reward which only contains partial information on the global reward.

Recently, significant progresses have been made in distributed RL schemes [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14]. These recent researches apply different approaches and assumptions, and cover different scenarios.

The main paradigm used in most of these methods is based on the averaging consensus algorithm [15], which is popular for solving distributed optimization problems [16], [17], [18], [19], [20], [21]. Among them, our main focus is on distributed off-policy TD-learning algorithms with saddle-point approaches. Here, the term ‘policy’ implies a mapping from the current state to an action, ‘off-policy’ means that the target policy we want to evaluate is different from the behavior policy we apply to collect experiences, and the ‘saddle point’ implies a point on the surface of the graph of a function where its partial derivatives equal to zero but at which the function has neither a maximum nor a minimum value. The saddle-point approaches mean overall frameworks that address the underlying problem as a saddle-point problem, where the saddle-point problem is the problem of finding a saddle-point of a given function, which is an important mathematical subject that has tremendous applications ranging from optimal control [22], mathematical programming [23], [24], to deep learning [25], and...
developing algorithms to solve it have been an active research topic for decades [26].

The main advantages of the saddle-point approaches for TD-learning lie in the following facts: (a) they provide rigorous and unified mathematical structures for analysis of solutions and convergence; (b) analysis tools from optimization perspectives, such as [26], [27], [28], [29], [30], and [31] can be easily applied to prove its convergence; (c) the frameworks offer more algorithmic flexibility such as additional cost constraints and objective, for example, entropic measures and sparsity promoting objectives. The saddle-point viewpoints have been recently applied to TD-learning algorithms for single-agent MDPs [1], [2] and multi-agent MDPs [7], [8], [11], [12], [13]. They address reinforcement learning algorithms with the saddle-point approaches. Although fruitful advances have been made recently, rigorous theoretical analysis of distributed off-policy TD-learning algorithms with saddle-point frameworks has not been fully studied so far.

Motivated by the above discussions, the main goal of this paper is to provide theoretical analysis templates and additional insights for distributed TD-learning algorithms with saddle-point frameworks, which provide general and unified perspectives of the distributed policy evaluation problem. To this end, we propose a new distributed TD-learning algorithm, called a distributed gradient temporal-difference (DGTD) learning, for multi-agent MDPs [32], [33], which generalizes the single-agent GTD [1], [2] to the multi-agent MDPs. We note that this DGTD is developed to provide additional insights and analysis templates rather than improving convergence speed of previous approaches. In this respect, we view the proposed algorithm and its analysis as a theoretical complement of the previous works rather than an improvement over them. Compared to existing results in the literature, the present work systematically provides rigorous analysis of the distributed GTD in a pure saddle-point form, which are lacking in the literature to the authors’ knowledge, and through this, new insights and analysis templates are proposed. They can motivate new algorithm developments, and offer additional insights and understanding on the distributed GTD. Additional features of the proposed algorithm are (a) consideration of random communication networks; (b) consideration of a pure saddle-point formulation, which are lacking in the literature. In the following section, some related works are briefly presented to compare them with the proposed work.

A. RELATED WORKS

1) POLICY OPTIMIZATION PROBLEM

The distributed Q-learning (QD-learning) [3], distributed actor-critic algorithm [4], [5], [7], and distributed fitted Q-learning [4] are studied in multi-agent settings. The work in [6] considers a distributed Q-learning with neural nonlinear function approximation. These existing results apply the averaging consensus technique [16], [17] to learn parameters in distributed manners.

2) POLICY EVALUATION PROBLEM

Distributed TD-learning algorithms are studied in [8], [9], [10], [11], [12], [13], and [14]. The results in [8], [9], and [34] consider central rewards with different assumptions. The result in [10] suggests a distributed TD learning with an averaging consensus steps, and proves its convergence rate. The main difference is that [10] considers an on-policy learning, while this work considers off-policy learning methods. Here, ‘on-policy learning’ means reinforcement learning which is opposite to ‘off-policy learning,’ i.e., on-policy learning implies that the target policy we want to evaluate is identical to the behavior policy we apply to collect experiences. The TD-learning in [11] considers a stochastic primal-dual algorithm for the policy evaluation with stochastic variants of the consensus-based distributed subgradient method akin to [18]. The main difference is that the algorithm in [11] introduces gradient surrogates of the objective function with respect to the local primal and dual variables, and the mixing steps for consensus are applied to both the local parameters and local gradient surrogates. Moreover, [12] provides a new stochastic primal-dual algorithm for the policy evaluation, which is off-policy. The work in [13] develops the so-called homotopy stochastic primal-dual algorithm with local actions and adaptive learning rates. A distributed gradient TD-learning with value functions that lie in reproducing kernel Hilbert spaces is proposed in [14] with its finite-sample analysis.

B. DISCUSSIONS

Among the previous works, approaches closest to the present work are [11], [12], and [13], which are also based on primal-dual approaches. Especially, [12] considers off-policy TD-learning as well, and [11] and [13] can be also potentially extended to the off-policy setting. In particular, [11], [12] provide convergence to solutions of stochastic approximate surrogate objective functions instead of the true solutions. On the other hand, we provide convergence to solutions of the true objective function. Although [13] also addresses convergence to solutions of true objective function, it does not include a rigorous solution analysis including their bounds. Moreover, the consensus constraint is not involved in the saddle-point form, while in our results, the overall problem is blended into a single saddle-point problem. Therefore, the method in [13] cannot enjoy advantages of the proposed pure saddle-point framework such as the generalization to the random communication network scenario. The objective functions are different across the current and previous works as well. A regularization term is used in [11], and the proximal term is introduced in [12] in the saddle-point forms to make the objective functions strongly convex-concave, while in our work, the regularization term is not used in order to obtain an unbiased solution. Besides, additional features of the proposed work are summarized as follows:

1) Most aforementioned distributed TD-learning algorithms assume static network structures except for [10], which addresses deterministic time-varying networks,
and adopts a different assumption on the graph connectivity. On the other hand, we consider stochastic networks.

2) A new saddle-point formulation of the distributed policy evaluation problem is proposed with rigorous analysis. The new formulation is a general and unified framework, which provides new insights, and can potentially open up opportunities to developments of new multi-agent reinforcement learning algorithms.

Preliminary results are included in the conference version [35], which only provides asymptotic convergence. On the other hand, this work provides more rigorous solution analysis and finite-time convergence. In addition, we consider stochastic network communications, and provide a modified algorithm to improve its convergence properties.

II. PRELIMINARIES

A. NOTATION AND TERMINOLOGY

The following notation is adopted: \( \mathbb{R}^n \) denotes the \( n \)-dimensional Euclidean space; \( \mathbb{R}^{n \times m} \) denotes the set of all \( n \times m \) real matrices; \( \mathbb{R}_+ \) and \( \mathbb{R}_+^2 \) denote the sets of nonnegative and positive real numbers, respectively, \( A^T \) denotes the transpose of matrix \( A \); \( I_n \) denotes the \( n \times n \) identity matrix; \( I \) denotes the identity matrix with appropriate dimension; \( \| \cdot \| \) denotes the standard Euclidean norm; \( \| x \|_D := \sqrt{x^T Dx} \) for any positive-definite \( D \); \( \lambda_{\text{min}}(A) \) denotes the minimum eigenvalue of \( A \) for any symmetric matrix \( A \); \( |S| \) denotes the cardinality of the set for any finite set \( S \); \( \mathbb{E}[-] \) denotes the expectation operator; \( \mathbb{P}[-] \) denotes the probability of an event; \( [x]_i \) is the \( i \)-th element for any vector \( x \); \( [P]_{ij} \) indicates the element in \( i \)-th row and \( j \)-th column for any matrix \( P \); if \( z \) is a discrete random variable which has \( n \) values and \( \mu \in \mathbb{R}^n \) is a stochastic vector, then \( z \sim \mu \) stands for \( \mathbb{P}[z = i] = \mu_i \) for all \( i \in \{1, \ldots, n\} \); \( 1_n \in \mathbb{R}^n \) denotes an \( n \)-dimensional vector with all entries equal to one; \( \text{dist}(S, x) \) denotes the standard Euclidean distance of a vector \( x \) from a set \( S \), i.e., \( \text{dist}(S, x) := \inf_{y \in S} \| x - y \|_2 \); for any \( S \subseteq \mathbb{R}^n \), \( \text{diam}(S) := \sup_{x, y \in S} \| x - y \|_2 \) is the diameter of the set \( S \); for a convex closed set \( S \), \( \Gamma_S(x) \) is the projection of \( x \) onto the set \( S \), i.e., \( \Gamma_S(x) := \arg \min_{y \in S} \| x - y \|_2 \); a continuously differentiable function \( f : \mathbb{R}^n \to \mathbb{R} \) is convex if \( f(y) \geq f(x) + \nabla f(x)^T (y - x) \) for any \( x, y \in \mathbb{R}^n \) and \( \rho \)-strongly convex if \( f(y) \geq f(x) + \nabla f(x)^T (y - x) + (\rho/2)\| y - x \|_2^2 \) for any \( x, y \in \mathbb{R}^n \) [24, pp. 691]; \( f(x) \) is a subgradient of a convex function \( f : \mathbb{R}^n \to \mathbb{R} \) at a given point \( x \in \mathbb{R}^n \) when the following relation holds: \( f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}) \leq f(x) \) for all \( x \in \mathbb{R}^n \) [28, pp. 209].

B. GRAPH THEORY

An undirected graph with the node set \( \mathcal{V} \) and the edge set \( \mathcal{E} \subseteq \mathcal{V} \times \mathcal{V} \) is denoted by \( G = (\mathcal{V}, \mathcal{E}) \). We define the neighbor set of node \( i \) as \( N_i := \{ j \in \mathcal{V} : (i, j) \in \mathcal{E} \} \). The adjacency matrix of \( G \) is defined as a matrix \( W \) with \( W_{ij} = 1 \) if and only if \( (i, j) \in \mathcal{E} \). If \( G \) is undirected, then \( W = W^T \). A graph is connected, if there is a path between any pair of vertices.

The graph Laplacian is \( L = H - W \), where \( H \) is a diagonal matrix with \( H_{ii} = |\mathcal{N}_i| \). If the graph is undirected, then \( L \) is symmetric positive semi-definite. It holds that \( L1_{|\mathcal{V}|} = 0 \). If \( G \) is connected, \( 0 \) is a simple eigenvalue of \( L \), i.e., \( 1_{|\mathcal{V}|} \) is the unique eigenvector corresponding to \( 0 \), and the span of \( 1_{|\mathcal{V}|} \) is the null space of \( L \).

C. RANDOM COMMUNICATION NETWORK

We will consider a random communication network model considered in [17]. In this paper, agents communicate with neighboring agents and update their estimates at discrete time instances \( k \in \{0, 1, \ldots \} \) over random time-varying network \( G(k) := (\mathcal{E}(k), \mathcal{V}(k)) \), \( k \in \{1, 2, \ldots \} \). Let \( \mathcal{N}(k) := \{ j \in \mathcal{V}(k) : (i, j) \in \mathcal{E}(k) \} \) be the neighbor set of agent \( i \), \( W(k) \) be the adjacency matrix of \( G(k) \), and \( H(k) \) be a diagonal matrix with \( [H(k)]_{ii} = |\mathcal{N}(k)| \). Then, the graph Laplacian of \( G(k) \) is \( L(k) := H(k) - W(k) \). We assume that \( G(k) \) is a random graph that is independent and identically distributed over time \( k \). A formal definition of the random graph is given below.

Assumption 1: Let \( \mathcal{F} := (\Omega, \mathcal{B}, \mu) \) be a probability space such that \( \Omega \) is the set of all \( |\mathcal{V}| \times |\mathcal{V}| \) adjacency matrices, \( \mathcal{B} \) is the Borel \( \sigma \)-algebra on \( \Omega \) and \( \mu \) is a probability measure on \( \mathcal{B} \). We assume that for all \( k \geq 0 \), the matrix \( W(k) \) is drawn from probability space \( \mathcal{F} \).

Define the expected value of the random matrices \( W(k), H(k), L(k) \), respectively, by

\[
\mathbb{W} := \mathbb{E}[W(k)], \quad \mathbb{H} := \mathbb{E}[H(k)], \quad \mathbb{L} := \mathbb{E}[L(k)] = \mathbb{H} - \mathbb{W},
\]

for all \( k \geq 0 \). An edge set induced by the positive elements of the matrix \( \mathbb{W} \) is \( \mathcal{E} := \{(i, j) \in \mathcal{V} \times \mathcal{V} : [W]_{ij} > 0\} \). Consider the corresponding graph \((\mathcal{E}, \mathcal{V})\), which we refer to as the mean connectivity graph [17]. We consider the following connectivity assumption for the graph.

Assumption 2 (Mean connectivity): The mean connectivity graph \((\mathcal{E}, \mathcal{V})\) is connected.

Under Assumption 2, 0 is a simple eigenvalue of \( \mathbb{L} \) [36, Lemma 1]. It implies that \( \mathbb{L}1_{|\mathcal{V}|} = 0 \) holds, and later this assumption is used for the consensus of learning parameters.

D. REINFORCEMENT LEARNING OVERVIEW

We briefly review a basic single-agent RL algorithm from [37] with linear function approximation. A Markov decision process (MDP) is characterized by a quadruple \( \mathcal{M} := (\mathcal{S}, \mathcal{A}, P, r, \gamma) \), where \( \mathcal{S} \) is a finite state space (observations in general), \( \mathcal{A} \) is a finite action space, \( P(s, a, s') := \mathbb{P}(s'|s, a) \) represents the (unknown) state transition probability from state \( s \) to \( s' \) given action \( a, \gamma : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1] \), \( \gamma \in (0, 1) \) is the discount factor. If action \( a \) is selected with the current state \( s \), then the state transits to \( s' \) with probability \( P(s, a, s') \) and incurs a random reward \( r(s, a, s') \in [0, \gamma] \) with expectation \( r(s, a, s') \). The stochastic policy is a map \( \pi : \mathcal{S} \times \mathcal{A} \to [0, 1] \) representing the probability \( \pi(s, a) = \mathbb{P}[a|s] \), \( \mathbb{P}^\pi \) denotes the transition matrix.
whose \((s, s')\) entry is \(P[s'|s] = \sum_{a \in A} P(s, a, s')\pi(s, a)\), and
\[d : \mathcal{S} \to \mathbb{R}\] denotes the stationary distribution of the state \(s \in \mathcal{S}\) under the behavior policy \(\beta\). We also define \(r^\pi\) as the expected reward given the policy \(\pi\) and the current state \(s\), i.e., \(r^\pi(s) := \sum_{a \in A} \sum_{s' \in \mathcal{S}} \pi(s, a) P(s, a, s')\pi(s', a, s')\). The infinite-horizon discounted value function with policy \(\pi\) and reward \(\tilde{r}\) is
\[J^\pi(s) := \mathbb{E} \left[ \sum_{k=0}^{\infty} \gamma^k \tilde{r}(s_k, a_k, s_{k+1}) \right] | s_0 = s,\]
where \(\mathbb{E}\) stands for the expectation taken with respect to the state-action trajectories following the state transition \(P^\pi\). Given pre-selected basis (or feature) functions \(\phi_1, \ldots, \phi_d : \mathcal{S} \to \mathbb{R}\), \(\Phi \in \mathbb{R}^{d \times |\mathcal{S}|}\) is defined as a full column rank matrix whose \(s\)-th row vector is \(\phi(s) := [\phi_1(s), \ldots, \phi_d(s)]\). The goal of RL with the linear function approximation is to find the weight vector \(w\) such that \(J(w) := \Phi^T \Phi w\) approximates the true value function \(J^\pi\). This is typically done by minimizing the mean-square Bellman error (MSBE) loss function [2]
\[
\min_{w \in \mathbb{R}^d} \text{MSBE}(w) := \frac{1}{2} \| r^\pi + \gamma P^\pi \Phi w - \Phi w \|_D^2, \tag{1}
\]
where \(\text{MSBE}(w) := \frac{1}{2} \| r^\pi + \gamma P^\pi \Phi w - \Phi w \|_D^2\), \(D\) is a symmetric positive-definite matrix and \(r^\pi \in \mathbb{R}^{|\mathcal{S}|}\) is a vector enumerating all \(r^\pi(s), s \in \mathcal{S}\). For online learning, we assume that \(D\) is a diagonal matrix with positive diagonal elements \(\sigma > 0\). The temporal difference (TD) learning [22], [37] with a linear function approximation is a stochastic gradient descent method with stochastic estimates of the gradient \(\nabla_w \text{MSBE}(w) = (\gamma P^\pi \Phi - \Phi) D (r^\pi + \gamma P^\pi \Phi w - \Phi w)\), which is obtained by dropping \(\gamma P^\pi \Phi\) in \(\nabla_w \text{MSBE}(w)\). If the linear function approximation is used, then this algorithm converges to an optimal solution of (1). The GTD in [2] solves instead the minimization of the mean-square projected Bellman error (MSBE) loss function
\[
\min_{w \in \mathbb{R}^d} \text{MSBE}(w) := \frac{1}{2} \| (r^\pi + \gamma P^\pi \Phi w - \Phi w) \|_D^2, \tag{2}
\]
where \(\Pi\) is the projected onto the range space of \(\Phi\), denoted by \(R(\Phi)\): \(\Pi(x) := \arg \min_{\xi \in \mathcal{R}(\Phi)} \| x - \xi \|_D^2\). The projection can be performed by the matrix multiplication: we write \(\Pi(x) := \Pi x\), where \(\Pi := \Phi (\Phi^T D \Phi)^{-1} \Phi^T D\). Compared to the standard TD learning, the main advantage of the GTD algorithms [1], [2] is their off-policy learning abilities.

Remark 1: Although its direct application to real problems is limited, the policy evaluation problem is a fundamental problem which is a critical building block to develop more practical policy optimization algorithms such as SARSA [38] and actor-critic [39] algorithms.

Note that \(d\) depends on the behavior policy, \(\beta\), while \(P^\pi\) and \(r^\pi\) depend on the target policy \(\pi\) that we want to evaluate. This corresponds to the off-policy learning. The main problem is to obtain samples, \((s, a, \tilde{r}, s')\) under \(\pi\), from the samples under \(\beta\). It can be done by the importance sampling or sub-sampling techniques [1], which will be described soon.

III. DISTRIBUTED REINFORCEMENT LEARNING OVERVIEW

In this section, we introduce the notion of the distributed RL, which will be addressed throughout the paper. Consider \(N\) RL agents labelled by \(i \in \{1, \ldots, N\}\). A multi-agent Markov decision process is characterized by \((\mathcal{S}, \{A_i\}_{i \in \mathcal{V}}, \{P_i\}_{i \in \mathcal{V}}, \{r_i\}_{i \in \mathcal{V}})\), where \(\gamma \in (0, 1)\) is the discount factor, \(\mathcal{S}\) is a finite state space, \(A_i\) is a finite action space of agent \(i, a := (a_1, \ldots, a_N)\) is the joint action, \(A := \prod_{i=1}^N A_i\) is the corresponding joint action space, \(r_i : \mathcal{S} \times A \times \mathcal{S} \to [0, \sigma], \sigma > 0\), is a bounded random reward of agent \(i\) with expectation \(R_i(s, a, s')\), and \(P(s, a, s') := \mathbb{P}[s'|s, a]\) represents the transition model of the state \(s\) with the joint action \(a\) and the corresponding joint action space \(A\). The stochastic policy of agent \(i\) is a mapping \(\pi_i : \mathcal{S} \times A_i \to [0, 1]\) representing the probability \(\pi_i(s, a_i) = \mathbb{P}[a_i|s]\) and the corresponding joint policy is \(\pi(s, a) := \prod_{i=1}^N \pi_i(s, a_i)\). \(P^\pi\) denotes the transition matrix, whose \((s, s')\) entry is \(\mathbb{P}[s'|s, a]\). The stochastic behavior policy of agent \(i\) is a mapping \(\beta_i : \mathcal{S} \times A_i \to [0, 1]\) representing the probability \(\beta_i(s, a_i) = \mathbb{P}[a_i|s]\) and the corresponding joint policy is \(\beta(s, a) := \prod_{i=1}^N \beta_i(s, a_i)\). Then, \(d : \mathcal{S} \to \mathbb{R}\) denotes the stationary state distribution under the behavior policy \(\beta\).

In particular, if the joint action \(a\) is selected with the current state \(s\), then the state transits to \(s'\) with probability \(P(s, a, s')\), and each agent \(i\) observes a random reward \(r_i(s, a, s') \in [0, \sigma]\) with expectation \(R_i(s, a, s')\). We assume that each agent does not have access to other agents’ rewards. For instance, there exists no centralized coordinator; thereby each agent does not know other agents’ rewards. In another example, each agent/coordinator may not want to uncover his/her own goal or the global goal for security/privacy reasons. We denote by \(R_i^\beta(s)\) the expected reward of agent \(i\), given the current state \(s, R_i^\beta(s) := \mathbb{E}[r(s_k, a_k, s_{k+1})|s_k = s, \pi]\). Throughout the paper, a vector enumerating all \(R_i^\beta(s), s \in \mathcal{S}\) is denoted by \(R^\beta_i \in \mathbb{R}^{S}\). In addition, denote by \(P_i(s, a, s')\) the state transition probability of agent \(i\) given joint state \(s\) and joint action \(a\).

This paper, we adopt the standard assumptions: the MDP with given behavior policy, \(\beta\), has a stationary distribution.

Assumption 3: With a fixed policy \(\beta\), the Markov chain \(P^\beta\) is ergodic with the stationary distribution \(d\) with \(d(s) > 0, s \in \mathcal{S}\).

We note that Assumption 3 is a fundamental hypothesis of most TD-learning algorithms [1], [2], [22], [40], [41]. In particular, if the Markov chain is ergodic, it implies that the stationary state distribution
\[
\lim_{k \to \infty} \mathbb{P}[s_k = s|\beta] := d(s), \forall s \in \mathcal{S},
\]
exists and is unique. Without the ergodicity assumption, some notions in the infinite-horizon discounted Markov decision problem are not well defined because of the time-varying nature of the state distribution.
Next, we summarize additional definitions and notations for some important quantities below.

1) $D$ is defined as a diagonal matrix with diagonal entries equal to those of $d$.

2) The central reward is defined as

$$r(s, a, s') := \frac{1}{N} \sum_{i=1}^{N} r_i(s, a, s'),$$

and the corresponding conditional expectation is

$$R^\pi(s) := \mathbb{E}[r(s_k, a_k, s_{k+1})|s_k = s, \pi] = \frac{1}{N} \sum_{i=1}^{N} R^\pi_i(s).$$

3) $J^\pi$ is the infinite-horizon discounted value function corresponding to the policy $\pi$ and the central reward $r = (r_1 + \cdots + r_N)/N$ defined as

$$J^\pi(s) := \mathbb{E} \left[ \sum_{k=0}^{\infty} \gamma^k r(s_k, a_k, s_{k+1}) \right] s_0 = s, \pi,$$

where $\mathbb{E}$ stands for the expectation taken with respect to the state-action trajectories following the state transition $P^\pi$. Note that $J^\pi$ satisfies the Bellman equation

$$J^\pi = \frac{1}{N} \sum_{i=1}^{N} R^\pi_i + \gamma P^\pi J^\pi.$$  

4) We denote $\xi := \min_{s \in \mathcal{S}} d(s)$.

Given pre-selected basis (or feature) functions $\phi_1, \ldots, \phi_q : \mathcal{S} \rightarrow \mathbb{R}$, $\Phi \in \mathbb{R}^{|S| \times q}$ is defined as a full column rank matrix whose $s$-th row vector $\phi(s) := \left[ \phi_1(s), \ldots, \phi_q(s) \right]$. The goal is to learn an approximate value of the centralized reward $\tilde{r} = (r_1 + \cdots + r_N)/N$ as stated below.

**Problem 1 (Multi-agent RL problem (MARLP)):** In the multi-agent RL problem with the linear function approximation, the goal of each agent $i$ is to learn the weight vector $w$ such that $J_w := \Phi w$ approximates the true value function $J^\pi$ corresponding to the centralized reward $r = (r_1 + \cdots + r_N)/N$ by minimizing the mean-square projected Bellman error loss function [2]

$$\min_{w \in \mathbb{R}^q} \text{MSPBE}(w) := \frac{1}{2} \| \Pi (R^\pi + \gamma P^\pi \Phi w) - \Phi w \|^2_D, \tag{3}$$

where $\text{MSPBE}(w) := \frac{1}{2} \| \Pi (R^\pi + \gamma P^\pi \Phi w) - \Phi w \|^2_D$. $\Pi$ is the projection onto the range space of $\Phi$, denoted by $\mathcal{R}(\Phi)$: $\Pi(x) := \arg \min_{x' \in \mathcal{R}(\Phi)} \| x - x' \|^2_D$. $D$ is a symmetric positive-definite matrix and $R^\pi \in \mathbb{R}^{|S|}$ is a vector enumerating all $R^\pi(s), s \in \mathcal{S}$. The projection can be performed by the matrix multiplication: we write $\Pi(x) := \Pi x$, where $\Pi := \Phi (\Phi^T D \Phi)^{-1} \Phi^T D$.

Firstly, we can prove that solving Problem 1 is equivalent to solving the optimization problem

$$\min_{w \in \mathcal{C}} \sum_{i=1}^{N} \text{MSPBE}_i(w), \tag{4}$$

where $\text{MSPBE}_i$ is defined as $\text{MSPBE}_i(w) := \frac{1}{2} \| \Pi (R^\pi_i + \gamma P^\pi \Phi w) - \Phi w \|^2_D$, for all $i \in \{1, 2, \ldots, N\}$, $C \subset \mathbb{R}^q$ is assumed to be a compact convex set which includes an unconstrained global minimum of (4). The following result formally establishes the equivalence between (4) and Problem 1.

**Proposition 1:** The following statements hold:

1) The solution of (4) is identical to the unique solution $w^*$ of the projected Bellman equation

$$\Pi \left( \frac{1}{N} \sum_{i=1}^{N} R^\pi_i + \gamma P^\pi \Phi w^* \right) = \Phi w^*. \tag{5}$$

2) The solution of (4) is given by

$$w^* = (\Phi^T D (I - \gamma P^\pi) \Phi)^{-1} \Phi^T D \frac{1}{N} \sum_{i=1}^{N} R^\pi_i.$$

3) The solution of Problem 1 is identical to the solution of (4).

**Proof:**

1) Since (4) is convex, $w^*$ is an unconstrained global solutions, if and only if

$$\nabla_w \sum_{i=1}^{N} \text{MSPBE}_i(w) = 0 \iff - (\Phi^T D (I - \gamma P^\pi) \Phi)^{-1} \Phi^T D \sum_{i=1}^{N} (R^\pi_i - (I - \gamma P^\pi) \Phi w^*) = 0.$$

Since $\Phi^T D (I - \gamma P^\pi) \Phi$ is nonsingular [22, pp. 300], this implies $(\Phi^T D \Phi)^{-1} \Phi^T D \sum_{i=1}^{N} (R^\pi_i - (I - \gamma P^\pi) \Phi w^*) = 0$. Pre-multiplying the equation by $\Phi$ yields the projected Bellman equation (5).

2) A solution $w^*$ of the projected Bellman equation (5) exists [22, pp. 355]. To prove the second statement, pre-multiply (5) by $\Phi^T D$ to have

$$\Phi^T D \left( \frac{1}{N} \sum_{i=1}^{N} R^\pi_i + \gamma P^\pi \Phi w^* \right) = \Phi^T D w^*,$$

where we use $\Pi := \Phi (\Phi^T D \Phi)^{-1} \Phi^T D$ and $\Phi^T D \Pi = \Phi^T D (\Phi^T D \Phi)^{-1} \Phi^T D = \Phi^T D$. Rearranging terms, we have $\Phi^T D \sum_{i=1}^{N} R^\pi_i = \Phi^T D (I - \gamma P^\pi) \Phi w^*$. Since $\Phi^T D (I - \gamma P^\pi) \Phi$ is nonsingular [22, pp. 300], pre-multiply both sides of the above equation by $(\Phi^T D (I - \gamma P^\pi) \Phi)^{-1}$ to obtain (6). The solution is unique because the objective function in (4) is strongly convex.

3) According to the first statement, the solution of Problem 1 satisfies the projected Bellman equation (5). Moreover, we can easily prove that the solution of the projected Bellman equation (5) is identical to the solution of the optimization in (3). This is because by taking the gradient on (3) with respect to $w$, (5) is obtained. This completes the proof.
Distributed optimization form of MARLP:

\[ \min_{w_i \in \mathcal{C}} \sum_{i=1}^{N} MSPBE_i(w_i) \]

subject to \( w_1 = w_2 = \cdots = w_N \),

where (8) implies the consensus among \( N \) copies of the parameter \( w \). We first establish the equivalence between (4) and (7).

**Proposition 2:** The solution of Problem 1 is identical to that of (7).

**Proof:** In (7), the equality constraint can be merged with the objective by setting \( w_1 = w_2 = \cdots = w_N =: w \). Then, (4) is recovered. Next, according to the third statement of Proposition 1, the solution of (4) is identical to that of Problem 1. This completes the proof. \( \square \)

To make the problem more feasible, we assume that the learning parameters \( w_i \), \( i \in \mathcal{V} \), are exchanged through a random communication network represented by the undirected graph \( \mathcal{G}(k) = (\mathcal{E}(k), \mathcal{V}(k)) \). In Section V, we will make several conversions of (7) to arrive at an optimization form, which can be solved using a primal-dual saddle-point algorithm [26], [28].

**IV. STOCHASTIC PRIMAL-DUAL ALGORITHM FOR SADDLE-POINT PROBLEM**

The proposed RL algorithm is based on a saddle-point problem formulation of the distributed optimization problem (7). In this section, we briefly introduce the definition of the saddle-point problem and a stochastic primal-dual algorithm [26] to find its solution.

**Definition 1 (Saddle-point [28]):** Consider the map \( \mathcal{L} : \mathcal{X} \times \mathcal{W} \rightarrow \mathbb{R} \), where \( \mathcal{X} \) and \( \mathcal{W} \) are compact convex sets. Assume that \( \mathcal{L}(\cdot, w) \) is convex over \( \mathcal{X} \) for all \( w \in \mathcal{W} \) and \( \mathcal{L}(x, \cdot) \) is concave over \( \mathcal{W} \) for all \( x \in \mathcal{X} \). Then, there exists a pair \((x^*, w^*)\) that satisfies

\[ \mathcal{L}(x^*, w) \leq \mathcal{L}(x^*, w^*) \leq \mathcal{L}(x, w^*) \quad \forall (x, w) \in \mathcal{X} \times \mathcal{W}. \]

The pair \((x^*, w^*)\) is called a saddle-point of \( \mathcal{L} \). The saddle-point problem is defined as the problem of finding saddle points \((x^*, w^*)\). It can be also defined as solving

\[ \min_{x \in \mathcal{X}} \max_{w \in \mathcal{W}} \mathcal{L}(x, w) = \max_{w \in \mathcal{W}} \min_{x \in \mathcal{X}} \mathcal{L}(x, w). \]

In our analysis, it will use the notion of approximate saddle-points in a geometric manner. In particular, the concept of the \( \varepsilon \)-saddle set is defined below.

**Definition 2 (\( \varepsilon \)-saddle set):** For any \( \varepsilon \geq 0 \), the \( \varepsilon \)-saddle set is defined as

\[ \mathcal{H}_{\varepsilon} = \{(x^*, w^*) \in \mathcal{X} \times \mathcal{W} : \mathcal{L}(x^*, w) - \mathcal{L}(x^*, w^*) \leq \varepsilon, \forall x \in \mathcal{X}, w \in \mathcal{W} \}. \]

From the definition, it is clear that \( \mathcal{H}_0 \) is the set of all saddle-points. The goal of the saddle-point problem is to find a saddle-point \((x^*, w^*)\) defined in Definition 1 over the set \( \mathcal{X} \times \mathcal{W} \). The stochastic primal-dual saddle-point algorithm in [26] can find a saddle-point when we have access to stochastic gradient estimates of function \( \mathcal{L} \). It executes the following updates:

\[ x_{k+1} = \Gamma_X(x_k - \alpha_k(\mathcal{L}_x(x_k, w_k) + \varepsilon_k)), \]

\[ w_{k+1} = \Gamma_W(w_k + \alpha_k(\mathcal{L}_w(x_k, w_k) + \xi_k)), \]

where \( \mathcal{L}_x(x, w) \) and \( \mathcal{L}_w(x, w) \) are the gradients of \( \mathcal{L}(x, w) \) with respect to \( x \) and \( w \), respectively, and \( \varepsilon_k, \xi_k \) are i.i.d. random variables with zero means. In this paper, the proposed algorithm can be seen as a class of the stochastic primal-dual saddle-point algorithms, and the availability of stochastic gradient estimates is a critical assumption for this purpose. Such stochastic gradient estimates can be obtained using state-action transition samples obtained by interacting with the environment.

To proceed, define the history of the algorithm until time \( k \), \( \mathcal{F}_k := \{\varepsilon_0, \ldots, \varepsilon_{k-1}, \xi_0, \ldots, \xi_{k-1}, x_0, \ldots, x_k, w_0, \ldots, w_k\} \) related to Algorithm 1. In the following result, we provide a finite-time convergence of the primal-dual algorithm with high probabilities.

**Proposition 3:** Assume that there exists a constant \( C > 0 \) such that

\[ \|\mathcal{L}_x(x_k, w_k) + \varepsilon_k\| \leq C, \]

\[ \|\mathcal{L}_w(x_k, w_k) + \xi_k\| \leq C, \]

\[ \text{diam}(\mathcal{X}) \leq C, \quad \text{diam}(\mathcal{W}) \leq C. \]

In addition, we assume that the step-size sequence \((\alpha_k)_{k=0}^\infty\) satisfies \( \alpha_k = \alpha_0 / \sqrt{k + T} \). Let \( \hat{x}_T = \frac{1}{T} \sum_{t=0}^{T-1} x_t \) and \( \hat{w}_T = \frac{1}{T} \sum_{k=0}^{T-1} w_k \) be the averaged dual iterates generated by (9) and (10) with \( T \geq 1 \). Then, for any \( \varepsilon > 0, \delta \in (0, 1) \), if \( T \geq \text{max}\{\Omega_1, \Omega_2\} \), then

\[ \mathbb{P}[\|\hat{x}_T - \hat{w}_T\| \in \mathcal{H}_\varepsilon] \geq 1 - \delta, \]

where \( \Omega_1 := \frac{8C^2((\alpha_0 + 2)C^2 + (\alpha_0 + 4)\delta / 6)}{\varepsilon^2} \ln \left( \frac{1}{\delta} \right) \) and \( \Omega_2 := \frac{4C^4(\varepsilon_0^{-1} + \delta)^2}{\varepsilon^2} \).

The proof of Proposition 3 can be found in Appendix.

**Remark 2:**

1) Convergence of stochastic primal-dual algorithms has been studied in previous works such as [26, Section 3.1]. The analysis given in Proposition 3 is tailored for our purposes, and has not been formally reported in the literature to our knowledge. We note that the analysis given in Proposition 3 can be viewed as a complement rather than replacement of the existing results.

2) We assume i.i.d. samples in Proposition 3. In practice, uniform samples from a sufficiently large replay buffer [42] can sufficiently approximate this scenario. Convergence of stochastic distribution primal-dual
algorithms was studied in [13] with Markovian samplings. We expect that the present work can be potentially extended to Markovian sampling scenario along similar directions, and it remains as a potential future direction.

V. SADDLE-POINT FORMULATION OF MARLP

In the previous section, we introduced the notion of the saddle-point and a stochastic primal-dual algorithm to find it. In this section, we study a saddle-point formulation of the distributed optimization (7) as a next step. Once obtained, the MARLP can be solved by using the stochastic primal-dual algorithm. For notational simplicity, we first introduce stacked vector and matrix notations.

Using those notations, the MSPBE loss function in (7) can be compactly expressed as $\sum_{i=1}^{N} \text{MSPBE}(\omega_i) = \frac{1}{2}(\Phi^T \tilde{D} \Phi - \tilde{B} \tilde{w})^T (\Phi^T \tilde{D} \Phi) - (\Phi^T \tilde{D} \tilde{B} - \tilde{B} \tilde{w})$, where $\otimes$ is the Kronecker product. Note that by the mean connectivity Assumption 2, the consensus constraint (8) can be expressed as $\bar{L} \tilde{w} = 0$, as $\bar{L}$ has a simple eigenvalue 0 with its corresponding eigenvector $\mathbf{1}_{|\mathcal{S}|}$ [36, Lemma 1]. Motivated by the continuous-time consensus optimization algorithms in [29], [30], and [31], we convert the problem (7) into the augmented Lagrangian problem [24, Sec. 4.2]

$$\min_{\tilde{w}, \tilde{\theta}, \tilde{\mu}} \frac{1}{2} (\Phi^T \tilde{D} \Phi - \tilde{B} \tilde{w})^T (\Phi^T \tilde{D} \Phi) - (\Phi^T \tilde{D} \tilde{B} - \tilde{B} \tilde{w}) + \tilde{w}^T \bar{L} \tilde{w}$$

subject to $\bar{L} \tilde{w} = 0,$

where a quadratic penalty term $\tilde{w}^T \bar{L} \tilde{w}$ for the equality constraint $\bar{L} \tilde{w} = 0$ is included. If the model is known, the above problem is an equality constrained quadratic programming problem, which can be solved by means of convex optimization methods [23]. Otherwise, the problem can be still solved using stochastic algorithms with observations. The latter case is our main concern. To develop model-free stochastic algorithms, some issues need to be taken into account. First, to estimate a stochastic estimate of the gradient, we need to assume that at least two independent next state samples can be drawn from any current state, which is impossible in most practical applications. The problem is often called the double sampling problem [22]. Second, the inverse matrix $(\Phi^T \tilde{D} \Phi)^{-1}$ in the objective function (14) needs to be removed. In particular, the main reason we use the linear function approximation is due to the large size of the state-space to the extent that enumerating numbers in the value vector is computationally demanding or even not possible. The computation of the inverse $(\Phi^T \tilde{D} \Phi)^{-1}$ is not possible due to both its computational complexity and the existence of the matrix $\tilde{D}$ including the stationary state distribution, which is assumed to be unknown in most RL settings. In GTD [2], this problem is resolved using a dual problem [8]. Following the same direction, we convert (14) into the equivalent optimization problem

$$\min_{\tilde{\phi}, \tilde{\theta}, \tilde{\mu}} \frac{1}{2} \tilde{\phi}^T (\tilde{D} \Phi)^{-1} \tilde{\phi} + \frac{1}{2} \tilde{\theta}^T \tilde{h}$$

subject to $\tilde{B} \tilde{\theta} - \tilde{L} \tilde{\phi} - \tilde{L} \tilde{\mu} = 0,$

where $\tilde{\phi}, \tilde{\theta}, \tilde{\mu}$ are newly introduced parameters. The next key step is to derive its Lagrangian dual problem [23], which can be obtained using standard approaches [23].

Proposition 4: The Lagrangian dual problem of (15) is given by

$$\min_{\tilde{\phi}, \tilde{\theta}, \tilde{\mu}} \psi(\tilde{\phi}, \tilde{\theta}, \tilde{\mu})$$

subject to $\tilde{B} \tilde{\theta} - \tilde{L} \tilde{\phi} - \tilde{L} \tilde{\mu} = 0,$

where $\psi(\tilde{\phi}, \tilde{\theta}, \tilde{\mu}) := \frac{1}{2} \tilde{\phi}^T (\tilde{D} \Phi)^{-1} \tilde{\phi} - \tilde{\theta}^T (\tilde{D} \Phi)^{-1} \tilde{h} + \frac{1}{2} \tilde{h}^T \tilde{h}.$

Proof: The dual problem can be obtained using standard manipulations as in [23, Ch. 5]. Define the Lagrangian function

$$\mathcal{L}(\tilde{\phi}, \tilde{\theta}, \tilde{\mu}, \tilde{\phi}) = \frac{1}{2} \tilde{\phi}^T (\tilde{D} \Phi)^{-1} \tilde{\phi} - \tilde{\theta}^T (\tilde{D} \Phi)^{-1} \tilde{h} + \frac{1}{2} \tilde{h}^T \tilde{h}.$$

Taking the derivative of $\mathcal{L}(\tilde{\phi}, \tilde{\theta}, \tilde{\mu}, \tilde{\phi})$ with respect to $\tilde{\phi}$ gives

$$\mathcal{L}_\phi(\tilde{\phi}, \tilde{\theta}, \tilde{\mu}, \tilde{\phi}) = \frac{1}{2} (\tilde{D} \Phi)^{-1} \tilde{\phi} - (\tilde{D} \Phi)^{-1} \tilde{h}.$$

Taking the derivative of $\mathcal{L}(\tilde{\phi}, \tilde{\theta}, \tilde{\mu}, \tilde{\phi})$ with respect to $\tilde{\theta}$ gives

$$\mathcal{L}_\theta(\tilde{\phi}, \tilde{\theta}, \tilde{\mu}, \tilde{\phi}) = \frac{1}{2} (\tilde{D} \Phi)^{-1} \tilde{\phi} - (\tilde{D} \Phi)^{-1} \tilde{h}.$$

Taking the derivative of $\mathcal{L}(\tilde{\phi}, \tilde{\theta}, \tilde{\mu}, \tilde{\phi})$ with respect to $\tilde{\mu}$ gives

$$\mathcal{L}_\mu(\tilde{\phi}, \tilde{\theta}, \tilde{\mu}, \tilde{\phi}) = \frac{1}{2} (\tilde{D} \Phi)^{-1} \tilde{\phi} - (\tilde{D} \Phi)^{-1} \tilde{h}.$$

One can observe that the inverse matrix $(\tilde{D} \Phi)^{-1}$ no more appears in the dual problem (16). To solve (16), we again construct the following Lagrangian function of (16) as in [8]:

$$\mathcal{L}(\tilde{\phi}, \tilde{\theta}, \tilde{\mu}, \tilde{\phi}) := \psi(\tilde{\phi}, \tilde{\theta}, \tilde{\mu}) + [\tilde{B} \tilde{\theta} - \tilde{L} \tilde{\phi} - \tilde{L} \tilde{\mu}]^T \tilde{w},$$

where $\tilde{w}$ is the Lagrangian multiplier. We further modify (17) by adding the term $-(\kappa/2)\tilde{w}^T \bar{L} \tilde{w}$:

$$\mathcal{L}(\tilde{\phi}, \tilde{\theta}, \tilde{\mu}, \tilde{\phi}) := \psi(\tilde{\phi}, \tilde{\theta}, \tilde{\mu}) + [\tilde{B} \tilde{\theta} - \tilde{L} \tilde{\phi} - \tilde{L} \tilde{\mu}]^T \tilde{w} - (\kappa/2)\tilde{w}^T \bar{L} \tilde{w},$$

where $\kappa > 0$.
where $\kappa \geq 0$ is a design parameter. Note that the solution of the original problem is not changed for any $\kappa \geq 0$. The term, $-(\kappa/2)\tilde{w}^T \tilde{L}\tilde{w}$, is added to accelerate the convergence in terms of the consensus of $\tilde{w}$.

Since the Lagrangian function (18) is convex-concave, the solutions of the optimization in (16) are identical to solutions $(\tilde{\theta}^*, \tilde{v}^*, \tilde{\mu}^*, \tilde{w}^*)$ of the corresponding saddle-point problem [28]

$$\max \min_{\tilde{\theta}, \tilde{v}, \tilde{\mu}} \mathcal{L}(\tilde{\theta}, \tilde{v}, \tilde{\mu}, \tilde{w}) = \max \min_{\tilde{\theta}, \tilde{v}, \tilde{\mu}} \mathcal{L}(\tilde{\theta}, \tilde{v}, \tilde{\mu}, \tilde{w}).$$

(19)

or equivalently,

$$\mathcal{L}(\tilde{\theta}^*, \tilde{v}^*, \tilde{\mu}^*, \tilde{w}^*) \leq \mathcal{L}(\tilde{\theta}^*, \tilde{v}^*, \tilde{\mu}^*, \tilde{w}^*) \leq \mathcal{L}(\tilde{\theta}, \tilde{v}, \tilde{\mu}, \tilde{w}),$$

(20)

for all $(\tilde{\theta}, \tilde{v}, \tilde{\mu}, \tilde{w})$. Now, the saddle-point problem in (19) can be solved by using the stochastic primal-dual algorithm [26].

### VI. SOLUTION ANALYSIS

In Section III, we derived a saddle-point formulation of the distributed optimization (7). In this section, we analyze the set of saddle-points. In particular, we obtain formulations of the set of saddle-points which solve (19). The formulations will be used in subsequent sections to develop the proposed RL algorithm. According to the standard results in convex optimization [23, Sec. 5.5.3, pp. 243], any saddle-point $(\tilde{\theta}^*, \tilde{v}^*, \tilde{\mu}^*, \tilde{w}^*)$ satisfying (20) must satisfy the following Karush-Kuhn-Tucker (KKT) condition although its converse is not true in general:

$$\begin{align*}
0 &= \nabla_{\tilde{\theta}} \mathcal{L}(\tilde{\theta}^*, \tilde{v}^*, \tilde{\mu}^*, \tilde{w}^*) \\
&= (\phi^T \tilde{D}\phi)\tilde{\theta}^* - \phi^T \tilde{D}\tilde{v}^* + \phi^T \tilde{D}(I_{|N|} - \gamma \tilde{P}^*) \tilde{\mu}^* w^* \\
0 &= \tilde{v}^* - \tilde{L}\tilde{w}^*,
\end{align*}$$

(21)

However, by investigating the KKT points, we can obtain useful information on the saddle-points. We first establish the fact that the set of KKT points corresponds to the set of optimal solutions of the consensus optimization problem (8).

**Proposition 5:** The set of all the KKT points satisfying (21) is given by

$$\mathcal{R} := \{\tilde{\theta}^*\} \times \{\tilde{v}^*\} \times \mathcal{F}^s \times \{1_N \otimes w^*\},$$

where $\tilde{v}^* = 0$, $w^*$ is given in (6) (the unique solution of the projected Bellman equation (5)),

$$\begin{align*}
\tilde{\theta}^* &= (\phi^T \tilde{D}\phi)^{-1} \phi^T \tilde{D}(-\tilde{R}^* + \tilde{P}^* \tilde{\mu}^* w^* - \gamma \tilde{P}^* \tilde{w}^*) \\
&= (\phi^T \tilde{D}\phi)^{-1} \phi^T \tilde{D} \left(-\tilde{R}^* + 1_N \otimes \frac{1}{N} \sum_{i=1}^{N} R_i^T \right),
\end{align*}$$

and $\mathcal{F}^s$ is the set of all solutions to the linear equation for $\tilde{\mu}$

$$\mathcal{F}^s := \{\tilde{\mu} : \tilde{L}\tilde{\mu} = \phi^T (I_{|N|} - \gamma \tilde{P}^*) \phi \tilde{\theta}^*\}. $$

(22)

**Proof:** The KKT condition in (21) is equivalent to the linear equations:

$$\begin{align*}
\nabla_\tilde{v} \mathcal{L}(\tilde{\theta}^*, \tilde{v}^*, \tilde{\mu}^*, \tilde{w}^*) &= (\phi^T \tilde{D}\phi)\tilde{\theta}^* - \phi^T \tilde{D}\tilde{v}^* + \phi^T \tilde{D}(I_{|N|} - \gamma \tilde{P}^*) \tilde{\mu}^* w^* \\
&= 0, \\
\nabla_{\tilde{\mu}} \mathcal{L}(\tilde{\theta}^*, \tilde{v}^*, \tilde{\mu}^*, \tilde{w}^*) &= \tilde{v}^* - \tilde{L}\tilde{w}^* = 0, \\
\nabla_{\tilde{w}} \mathcal{L}(\tilde{\theta}^*, \tilde{v}^*, \tilde{\mu}^*, \tilde{w}^*) &= \tilde{L}\tilde{w}^* \\
&= \tilde{L}\tilde{v}^* + \tilde{L}\tilde{\mu}^* - \phi^T (I_{|N|} - \gamma \tilde{P}^*) \phi \tilde{\theta}^*$
&= 0.
\end{align*}$$

(23)

(24)

(25)

(26)

(27)

Since the mean connectivity graph $(\mathcal{E}, \mathcal{V})$ of $\mathcal{G}(k)$ is connected by Assumption 2, the dimension of the null space of $\tilde{L}$ is one. Therefore, span$(\mathbf{1}_{|V|})$ is the null space, and (26) implies the consensus $w^* = w_1^* = \cdots = w_N^*$. Plugging (26) into (25) yields $\tilde{v}^* = 0$. With $\tilde{v}^* = 0$, (27) is simplified to

$$\tilde{L}\tilde{\mu}^* = \phi^T (I_{|N|} - \gamma \tilde{P}^*) \phi \tilde{\theta}^*.$$ 

(28)

In addition, from (24), the stationary point for $\tilde{\theta}$ satisfies

$$\tilde{\theta}^* = (\phi^T \tilde{D}\phi)^{-1} \phi^T \tilde{D} \left(\tilde{R}^* - \phi \tilde{\theta}^* + \gamma \tilde{P}^* \phi \tilde{w}^*\right).$$

(29)

Plugging the above equation into (28) yields

$$\tilde{L}\tilde{\mu}^* = \phi^T (I_{|N|} - \gamma \tilde{P}^*) \phi \tilde{\theta}^* = \phi^T (I_{|N|} - \gamma \tilde{P}^*) \phi \tilde{\theta}^*$$

$$\times \phi^T (I_{|N|} - \gamma \tilde{P}^*) \phi \tilde{\theta}^* \phi^T (I_{|N|} - \gamma \tilde{P}^*) \phi \tilde{\theta}^*.$$

(30)

Multiplying (30) by $(1 \otimes I)T$ on the left results in

$$\begin{align*}
(\phi^T D(I_{|N|} - \gamma \tilde{P}^*) \phi)^T (\phi^T D\phi)^{-1} \phi^T D \\
&\times \left(1/N \sum_{i=1}^{N} R_i^T + \gamma P^* \phi w^* - \phi w^* \right) = 0.
\end{align*}$$

Since $\phi^T D(I - \gamma \tilde{P}^*) \phi$ is nonsingular [22, pp. 300], pre-multiplying both sides of the last equation with $(\phi^T D(I - \gamma \tilde{P}^*) \phi)^{-1}$ results in

$$\begin{align*}
(\phi^T D\phi)^{-1} \phi^T D \\
&\left(1/N \sum_{i=1}^{N} R_i^T + \gamma P^* \phi w^* - \phi w^* \right) = 0.
\end{align*}$$

(31)

Pre-multiplying (31) with $\phi^T$ from left yields the projected Bellman equation in Proposition 1, and $w^*$ is any of its solutions. In particular, multiplying (24) by $(1 \otimes I)T$ from left, a KKT point for $\tilde{w}^*$ is expressed as $\tilde{w}^* = 1 \otimes w^*$ with

$$w^* = (\phi^T D(I - \gamma \tilde{P}^*) \phi)^{-1} \phi^T D$$

$$\times \left(1/N \sum_{i=1}^{N} R_i^T - \pi \left(1/N \sum_{i=1}^{N} R_i^T + \gamma P^* \phi w^* - \phi w^* \right) \right) = (\phi^T D(I_{|N|} - \gamma \tilde{P}^*) \phi)^{-1} \phi^T D \left(1/N \sum_{i=1}^{N} R_i^T \right).$$

(32)
From (30), \( \mu^* \) is any solution of the linear equation (30). Lastly, (31) can be rewritten as

\[
0 = (\Phi^T \bar{D} \Phi)^{-1} \Phi^T \bar{D} \left( I_N \otimes \frac{1}{N} \sum_{i=1}^N R_i^\gamma + \gamma \bar{P}^\gamma \Phi \bar{w}^* - \Phi \bar{w}^* \right).
\]

Subtracting (29) by the last term, we obtain

\[
\bar{\theta}^* = (\Phi^T \bar{D} \Phi)^{-1} \Phi^T \bar{D} \left( -\bar{R}^\gamma + I_N \otimes \frac{1}{N} \sum_{i=1}^N R_i^\gamma \right).
\]

This completes the proof. □

Since the set of saddle-points of \( L \) in (17) is a subset of the KKT points, we can estimate a potential structure of the set of saddle-points.

**Corollary 1:** The set of all the saddle-points, \( \mathcal{H}_0 \), satisfying (20) is given by \( \mathcal{H}_0 := \{ \bar{\theta}^* \times \{ \bar{v}^* \} \times \bar{F}^* \times \{ I_N \otimes w^* \} \} \), where \( \bar{v}^* = 0 \), is the unique solution of the projected Bellman equation (5), \( \bar{F}^* \) is some subset of \( F^* \), \( F^* \) and \( \bar{v}^* \) are defined in Proposition 5.

According to Corollary 1, the set of KKT points corresponding to \( \bar{\theta}, \bar{v}, \bar{w} \) is a singleton \( \{ \bar{\theta}^* \} \times \{ \bar{v}^* \} \times \{ I_N \otimes w^* \} \). Therefore, it is the unique saddle-point corresponding to \( \bar{\theta}, \bar{v}, \bar{w} \). On the other hand, \( \bar{F}^* \) is a set. We can prove that \( \bar{F}^* \) is an affine space.

**Lemma 1:** \( \bar{F}^* \) is an affine space.

**Proof:** By the saddle-point property in (20), \( \bar{F}^* = \bar{F}^* \) if and only if \( L(\bar{\theta}^*, \bar{v}^*, \bar{\mu}^*, \bar{w}^*) = L(\bar{\theta}^*, \bar{v}^*, \bar{\mu}^*, \bar{w}^*) \), \( \forall \bar{\mu} \in \bar{F}^* \), which is equivalent to \( \hat{\bar{\mu}}^T \bar{L} \bar{w}^* = \bar{m}^T L \bar{w}^*, \forall \bar{\mu} \in \bar{F}^* \), proving that \( \bar{F}^* \) is an affine space.

By Lemma 1, we can obtain an explicit formulation of a point in \( \bar{F}^* \).

**Proposition 6:** We have

\[
\bar{\mu}^* = \bar{L}^\dagger \Phi^T (I - \gamma \bar{P}^\gamma)^T \bar{D} \Phi \bar{w}^* \in \bar{F}^*.
\]

**Proof:** Since \( \bar{F}^* \) is the set of solutions of the linear equation \( \bar{L} \bar{\mu} = \Phi^T (I - \gamma \bar{P}^\gamma)^T \bar{D} \Phi \bar{\theta}^* \), \( \bar{F}^* \) is the set of general solutions of the linear equation, which are given by the affine space \( \bar{\mu} = \bar{L}^\dagger \Phi^T (I - \gamma \bar{P}^\gamma)^T \bar{D} \Phi \bar{\theta}^* + (\bar{L} \bar{1} - I) \bar{z} \), where \( \bar{L} \) is a pseudo-inverse of \( \bar{L} \) and \( \bar{z} \in \mathbb{R}^{\|N\| \times 1} \) is arbitrary. In addition, since \( \bar{F}^* \subseteq \bar{F}^* \) and \( \bar{F}^* \) is also affine by Lemma 1, one concludes that \( \bar{\mu} = \bar{L}^\dagger \Phi^T (I - \gamma \bar{P}^\gamma)^T \bar{D} \Phi \bar{w}^* \in \bar{F}^* \). □

For some technical reasons that will become clear later, algorithms to find a solution need to confine the search space of an algorithm to a compact and convex set which include at least one saddle-point in \( \bar{R} \) in Corollary 1. To this end, we compute a bound on at least one saddle-point \( (\bar{\theta}^*, \bar{v}^*, \bar{\mu}^*, \bar{w}^*) \) in the following lemma.

**Lemma 2:** \( \bar{w}^*, \bar{v}^* \) and \( \bar{\mu}^* \) satisfy the following bounds:

\[
\| \bar{w}^* \| \leq \frac{1}{1 - \alpha} \sqrt{\frac{\| S \|}{\lambda_{\text{min}}(\Phi^T \Phi)}} \left( \frac{\| \Pi J^\gamma - J^\gamma \|_D + \sigma}{\sqrt{\xi}} \right).
\]

\[
\| \bar{v}^* \| \leq c_{\bar{v}}, \quad \forall c_{\bar{v}} \geq 0,
\]

\[
\| \bar{\mu}^* \| \leq 2\sigma \left( 1 + \frac{N}{\sqrt{\xi \lambda_{\text{min}}(\Phi^T \Phi)}} \right).
\]

where \( \xi := \min_{s \in S} d(s) \) as defined in Section III. Moreover, there exists a \( \bar{\mu}^* \in \bar{F}^* \) such that

\[
\| \bar{\mu}^* \| \leq \| L \|_\infty \| \Phi \|_\infty^2 2\sigma |S|^2 \sqrt{\frac{N}{\xi \lambda_{\text{min}}(\Phi^T \Phi)}}.
\]

For the pseudo-inverse of the graph Laplacian in [43], we can use the expression \( L^\dagger = (L + I_N \xi \sqrt{N})^{-1} - I_N \xi \sqrt{N} \).

**Proof:** To prove Lemma 2, we will first prove a bound on \( w^* \in \bar{W}^* \).

**Claim:** If \( w^* \) is an optimal solution presented in Proposition 1, then

\[
\| w^* \| \leq \frac{1}{1 - \alpha} \sqrt{\frac{\| S \|}{\lambda_{\text{min}}(\Phi^T \Phi)}} \left( \frac{\| \Pi J^\gamma - J^\gamma \|_D + \sigma}{\sqrt{\xi}} \right).
\]

**Proof of Claim:** We first bound the term \( \| \Phi w^* \|_\infty \) as follows:

\[
\| \Phi w^* \|_\infty \leq \| \Phi w^* - J^\gamma + J^\gamma \|_\infty \leq \| \Phi w^* - J^\gamma \|_\infty + \| J^\gamma \|_\infty \leq \| \Phi w^* - J^\gamma \|_2 + \| J^\gamma \|_\infty \leq \frac{1}{\sqrt{\| S \|}} \| \Pi J^\gamma - J^\gamma \|_D + \| J^\gamma \|_\infty \leq \frac{1}{\sqrt{\| S \|}} \| \Pi J^\gamma - J^\gamma \|_2 + \frac{\sigma}{1 - \alpha}.
\]

where the first inequality follows from the triangle inequality, the second inequality uses \( \| \cdot \|_\infty \leq \| \cdot \|_2 \), the third inequality uses \( \sqrt{\| T \|} \leq \sqrt{\| T \|_2} \), the fourth inequality comes from [22, Proposition 6.10], and the last inequality uses the bound on the rewards. On the other hand, its lower bound can be obtained as

\[
\| \Phi w^* \|_\infty \geq \frac{1}{\sqrt{\| S \|}} \| \Phi w^* \|_2
\]

\[
\geq \frac{1}{\sqrt{\| S \|}} \| w^* \|_2 \sqrt{\lambda_{\text{min}}(\Phi^T \Phi)}
\]

\[
\geq \frac{1}{\sqrt{\| S \|}} \| w^* \| \sqrt{\lambda_{\text{min}}(\Phi^T \Phi)}.
\]

where the first inequality comes from \( \| v \| \leq \sqrt{\| T \|} \| v \|_\infty \) for any \( v \in \mathbb{R}^a \) and the second inequality uses \( \| \Phi w^* \|_2 = \sqrt{\| w^* \|^T \Phi \Phi w^*} \geq \sqrt{\lambda_{\text{min}}(\Phi^T \Phi) \| w^* \|^2} \). Combining the two relations completes the proof. □

The first bound easily follows from \( \bar{w}^* = I_N \otimes w^* \) and the **Claim**. Since \( \bar{v}^* = 0 \) from Proposition 5, the second inequality is obvious. For the third bound, we use the expression for \( \bar{\mu}^* \) in Proposition 5 to prove

\[
\| \bar{\mu}^* \| \leq \| \bar{L}^\dagger ( -\bar{R}^\gamma + 1 \otimes \frac{1}{N} \sum_{i=1}^N R_i^\gamma ) \|_\infty
\]

\[
\leq \| \bar{L}^\dagger ( -\bar{R}^\gamma + 1 \otimes \frac{1}{N} \sum_{i=1}^N R_i^\gamma ) \|_2
\]
VOLUME 10, 2022

D. Lee et al.: Distributed Off-Policy Temporal Difference Learning Using Primal-Dual Method

where the first inequality follows from \( \| \cdot \|_\infty \leq \| \cdot \|_2 \), the third inequality follows from the nonexpansive property of the projection (see [22, Proof of Proposition 6.9., pp. 355] for details), and the fact that \( \| v\|_2 \leq \sqrt{N} \| v\|_\infty \) for any \( v \in \mathbb{R}^N \) is used in the fifth inequality. Lower bounds on \( \| \Phi^T \tilde{\theta}^* \|_\infty \) are obtained as

\[
\| \Phi^T \tilde{\theta}^* \|_\infty \geq \frac{1}{\sqrt{N}} \| \Phi^T \tilde{\theta}^* \|_2 \\
\geq \frac{1}{\sqrt{N}} \sqrt{\lambda_{\min}(\Phi^T \Phi)} \| \tilde{\theta}^* \|_2 \\
= \sqrt{\lambda_{\min}(\Phi^T \Phi)} \| \tilde{\theta}^* \|_2 \\
\geq \sqrt{\lambda_{\min}(\Phi^T \Phi)} \| \tilde{\theta}^* \|_\infty.
\]

Combining the two inequalities yields the third bound. For the last inequality, we use Proposition 6 and obtain a bound on \( \| \tilde{\mu} \|_\infty \) as

\[
\| \tilde{\mu} \|_\infty \leq \| \tilde{L}^T \tilde{\mu} \|_\infty \leq \| \tilde{L}^T \|_\infty \| \tilde{\mu} \|_\infty \leq \| |S| \tilde{L}_i^T \|_\infty \| \tilde{\mu} \|_\infty \\
\leq \sqrt{\lambda_{\min}(\Phi^T \Phi)} \| \tilde{\theta}^* \|_\infty.
\]

where the third inequality follows from the fact that absolute values of all elements of \( \| (I - \gamma \tilde{P}^T)^T \|_\infty \) are less than one, and the fourth inequality uses the bounds on \( \| \tilde{\theta}^* \|_\infty \).

In this section, we analyzed the set of saddle-points corresponding to the MARLP. In the next section, we introduce the proposed distributed RL algorithm, which solves the saddle-point problem of the MARLP in (19) by using the stochastic primal-dual algorithm.

VII. PRIMAL-DUAL DISTRIBUTED GTD ALGORITHM (PRIMAL-DUAL DGTD)

In this section, we study a distributed GTD algorithm to solve Problem 1. The main idea is to solve the saddle-point problem of the MARLP in (19) by using the stochastic primal-dual algorithm, where the unbiased stochastic gradient estimates are obtained by using samples of the state, action, and reward. To proceed, we first modify the saddle-point problem of the MARLP in (19) to a constrained saddle-point problem whose domains are confined to compact sets.

Lemma 2 provides rough estimates of the bounds on the sets that include at least one saddle-point of the Lagrangian function (17). Define the cube \( B_\beta := \{ x \in [0,1]^{|S|}: \| x \|_\infty \leq \beta \} \) and \( C_{\tilde{\theta}} = B_{\tilde{c}_k + \tilde{c}_v}, C_{\tilde{\phi}} = B_{\tilde{c}_k + \tilde{c}_v}, C_{\tilde{\psi}} = B_{\tilde{c}_k + \tilde{c}_v} \) for \( \tilde{c}_k, \tilde{c}_v, \tilde{c}_\psi > 0 \). Then, the constraint sets satisfy \( \tilde{\theta}^* \in C_{\tilde{\theta}}, \tilde{\phi}^* \in C_{\tilde{\phi}}, \tilde{\psi}^* \in C_{\tilde{\psi}} \), and \( C_{\tilde{\theta}} \cap C_{\tilde{\phi}} \cap C_{\tilde{\psi}} \neq \emptyset \).

Estimating \( \tilde{c}_k, \tilde{c}_v, \tilde{c}_\psi > 0 \) requires additional analysis or is almost infeasible in most real applications because the model is assumed to be unknown. However, in practice, we can consider sufficiently large parameters \( \tilde{c}_k, \tilde{c}_v, \tilde{c}_\psi > 0 \) so that they include at least one solution. With this respect, we assume that sufficiently large sets \( C_{\tilde{\theta}}, C_{\tilde{\phi}}, C_{\tilde{\psi}} \) satisfy \( C_{\tilde{\theta}} \cap C_{\tilde{\phi}} \cap C_{\tilde{\psi}} \neq \emptyset \). Such an assumption is frequently adopted in the literature when projection steps are required for general stochastic approximation algorithms to guarantee the convergence, e.g. [4, 27, 44]. For simpler analysis, we also assume that the solutions are included in interiors of the compact sets.

Assumption 4: The constraint sets satisfy \( \tilde{\theta}^* \in C_{\tilde{\theta}}, \tilde{\phi}^* \in C_{\tilde{\phi}}, \tilde{\psi}^* \in C_{\tilde{\psi}}, \tilde{\theta}^* \in C_{\tilde{\theta}} \cap C_{\tilde{\phi}} \cap C_{\tilde{\psi}} \).

Under Assumption 4, finding a saddle-point in (19) can be reduced to the constrained saddle-point problem

\[
\min_{\tilde{\theta}, \tilde{\phi}, \tilde{\psi}} \mathcal{L}(\tilde{\theta}, \tilde{\phi}, \tilde{\psi}) \\
subject to \ \tilde{w} \in C_{\tilde{w}}, \ (\tilde{\theta}, \tilde{\phi}, \tilde{\psi}) \in C_{\tilde{\theta}} \times C_{\tilde{\phi}} \times C_{\tilde{\psi}}.
\]

For notational convenience, introduce the notation

\[
\tilde{x} := \begin{pmatrix} \tilde{\theta} \\ \tilde{\phi} \\ \tilde{\psi} \end{pmatrix}, \quad \tilde{x}^* := \begin{pmatrix} \tilde{\theta}^* \\ \tilde{\phi}^* \\ \tilde{\psi}^* \end{pmatrix}, \\
\mathcal{W} := C_{\tilde{w}}, \ \mathcal{X} := C_{\tilde{\theta}} \times C_{\tilde{\phi}} \times C_{\tilde{\psi}}.
\]

Then, the saddle-point problem is \( \min_{\tilde{x} \in \mathcal{X}} \max_{w \in \mathcal{W}} \mathcal{L}(\tilde{x}, \tilde{w}) \). If the gradients of the Lagrangian are available, then the deterministic primal-dual algorithm [28] can be used as follows:

\[
\tilde{x}_{k+1} = \Gamma_{\chi}(\tilde{x}_k - \alpha_k \mathcal{L}_\chi(\tilde{x}_k, \tilde{w}_k)), \quad (32)
\]

\[
\tilde{w}_{k+1} = \Gamma_{\mathcal{Y}}(\tilde{w}_k + \alpha_k \mathcal{L}_w(\tilde{x}_k, \tilde{w}_k)), \quad (33)
\]

In this paper, our problem allows only stochastic gradient estimates of the Lagrangian function: the exact gradients are not available, while only their unbiased stochastic estimations are given. In this case, the stochastic primal-dual algorithm [26] introduced in Section IV can find a solution under certain conditions

\[
\tilde{x}_{k+1} = \Gamma_{\chi}(\tilde{x}_k - \alpha_k \mathcal{L}_\chi(\tilde{x}_k, \tilde{w}_k)), \quad (32)
\]

\[
\tilde{w}_{k+1} = \Gamma_{\mathcal{Y}}(\tilde{w}_k + \alpha_k \mathcal{L}_w(\tilde{x}_k, \tilde{w}_k) + \xi_k)), \quad (33)
\]
where \( \xi_k \) and \( \epsilon_k \) are i.i.d. random variables with zero mean. In our case, stochastic estimates of the Lagrangian function (18) can be obtained by using samples of the state, action, and reward. The overall algorithm is given in Algorithm 1. The algorithm assumes additional partial information sharing among agents, e.g., sharing of learning parameters, through random network communications, where the network structure is represented by a randomly changing undirected graph.

Despite the additional communication model, the algorithm is still distributed in the sense that each agent has a local view of the overall system: it is only accessible to the learning parameter of the neighboring RL agents in the graph. Potential applications are distributed machine learning, distributed resource allocation, and robotics, where the reward information is limited due to physical limitations or privacy constraints.

Some remarks are summarized below:

**Remark 3:**

1) In Algorithm 1, the sets \( C_k, C_{\beta}, C_{\mu}, C_\kappa \) will be defined soon.
2) Similar to GTD algorithms [1], [2], Algorithm 1 is an off-policy algorithm, which means that the behavior policy, \( \beta \), and target policy, \( \pi \), can be different. The behavior policy, \( \beta \), is the policy used to sample transitions, while the target policy, \( \pi \), implies the policy we want to evaluate by predicting the value function \( J^\pi \).
3) An advantage of the off-policy learning is the flexibility in choosing the behavior policy used to sample transitions, which makes additional flexibility and convenience in practical implementation.
4) As can be seen in Algorithm 1, the transitions, \( (s, a, \tilde{r}, s') \), are sampled under the behavior policy \( \beta \), while transitions under the target policy can be obtained by using the importance sampling or subsampling techniques [1], which will be described soon.

**Remark 4:** The proposed work deals with the policy evaluation problem, where we try to evaluate a given fixed policy using the value prediction approaches. The policy evaluation is a fundamental problem, which does not have direct real-world applications. However, it can be applied to the policy optimization problem to find an optimal policy. For example, the proposed model-free policy evaluation algorithm can be used in the actor-critic algorithm [39] and SARSA [38] for policy optimization. In particular, the actor-critic algorithm is a policy optimization algorithm which consists of two parts, 1) policy evaluation and 2) policy improvement. The first part evaluates the current policy, and the second part improves the policy using the value function estimate computed in the first part. The proposed distributed TD learning in this paper can be applied to the first policy evaluation part in the actor-critic algorithm in order to find an optimal policy in a distributed manner. Similarly, SARSA also consists of the two parts, 1) policy evaluation and 2) policy improvement. The main difference between actor-critic and SARSA lies in that actor-critic applies gradient steps for the policy improvement, while SARSA uses a greedy policy. In this case, the distributed RL can be used in resource management problems or multi-robot systems where the local reward cannot be shared with other rewards due to physical or security constraints.

In Line 6, each agent samples the state, action, and the corresponding local reward, Line 8 updates the primal variable

---

**Algorithm 1 Distributed GTD Algorithm**

1: Set \( \kappa \geq 0 \) and the step-size sequence \( \{\alpha_k\}_{k=0}^\infty \).
2: for agent \( i \in \{1, \ldots, N\} \) do
3: \( \text{Initialize} \ (\theta_0^{(i)}, v_0^{(i)}, \mu_0^{(i)}, w_0^{(i)}) \).
4: end for
5: for \( k \in \{0, \ldots, T-1\} \) do
6: for agent \( i \in \{1, \ldots, N\} \) do
7: Sample \((s, a, s') \) with \( s \sim d, a_i \sim \beta_i(\cdot|s), s' \sim P(s, a, \cdot), \tilde{r}_i := \tilde{r}(s, a, s') \).
8: Update primal variables according to
9: \[
\theta_{k+1/2}^{(i)} = \theta_k^{(i)} + \alpha_k \left[ \|N_i(k)\| w_k^{(i)} - \sum_{j \in N_i(k)} w_j^{(i)} \right],
\]
10: \[
v_{k+1/2}^{(i)} = v_k^{(i)} - \alpha_k \left[ \|N_i(k)\| v_k^{(i)} - \sum_{j \in N_i(k)} v_j^{(i)} \right],
\]
11: Project parameters:
12: \[
\theta_{k+1}^{(i)} = \Gamma C_{\beta} \left( \theta_{k+1/2}^{(i)} \right), \quad \mu_{k+1}^{(i)} = \Gamma C_{\mu} \left( \mu_{k+1/2}^{(i)} \right), \quad w_{k+1}^{(i)} = \Gamma C_{w} \left( w_{k+1/2}^{(i)} \right).
\]
13: Output The averaged \( \bar{w}_T^{(i)} = \frac{1}{T} \sum_{k=0}^{T} w_k^{(i)} \), \( i \in \mathcal{V} \), and last, \( w_T^{(i)} \), \( i \in \mathcal{V} \), dual iterates.
according to the stochastic gradient descent step, and Line 9 updates the dual variable by the stochastic gradient ascent step. Line 10 projects the variables to the corresponding compact sets \( C_\theta, C_v, C_{\mu}, C_w \), and Line 13 outputs averaged iterates over the whole iteration steps instead of the final iterates. Note that the averaged dual variables can be computed recursively [22, pp. 181].

**Remark 5:** In this paper, we consider the off-policy learning. In particular, to obtain a stochastic approximation of \( P^\pi \) from the samples under the behavior policy, \( \beta \), we use the fact \[ P[s'|s, \pi] = \sum_{a \in A} P(s, a, s') \pi(s, a) \]
and
\[ R^\pi_i(s) := \sum_{a \in A} \sum_{s' \in S} \pi(s, a) P(s, a, s') R(s, a, s') \]

The next proposition states that the averaged dual variable converges to the set of saddle-points in terms of the \( \epsilon \)-saddle point with a vanishing \( \epsilon \).

**Proposition 7 (Finite-time convergence I):** Consider Algorithm 1, assume that the step-size sequence, \((\alpha_k)_{k=0}^\infty\), satisfies \( \alpha_k = \alpha_0/\sqrt{k+1} \) for some \( \alpha_0 > 0 \), and let
\[
\tilde{x}_k := \begin{bmatrix} \bar{\theta}_k \\ \bar{v}_k \end{bmatrix}, \quad \tilde{\theta}_k := \begin{bmatrix} \bar{\theta}_k^{(1)} \\ \vdots \\ \bar{\theta}_k^{(N)} \end{bmatrix}, \quad \tilde{v}_k := \begin{bmatrix} v_k^{(1)} \\ \vdots \\ v_k^{(N)} \end{bmatrix}, \\
\tilde{\mu}_k := \begin{bmatrix} \bar{\mu}_k^{(1)} \\ \vdots \\ \bar{\mu}_k^{(N)} \end{bmatrix}, \quad \tilde{w}_k := \begin{bmatrix} w_k^{(1)} \\ \vdots \\ w_k^{(N)} \end{bmatrix},
\]
and \( \tilde{x}_T = \frac{1}{T} \sum_{k=0}^{T-1} \tilde{x}_k \) and \( \tilde{w}_T = \frac{1}{T} \sum_{k=0}^{T-1} \tilde{w}_k \) be the averaged dual iterates generated by Algorithm 1 with \( T \geq 1 \). Then, for any \( \epsilon > 0 \), \( \delta \in (0, 1) \), if \( T \geq \max \{ \Omega_1, \Omega_2 \} := \omega(\epsilon, \delta) \), then
\[ P[(\tilde{x}_T, \tilde{w}_T) \in \mathcal{H}_\epsilon] \geq 1 - \delta, \]
where
\[
\Omega_1 := \frac{8C^2((\alpha_0 + 2)^2C^2 + (\alpha_0 + 4)\epsilon/6)}{\epsilon^2}, \\
\Omega_2 := \frac{4C^4(2\alpha_0 + 1 + \alpha_0)^2}{\epsilon^2}.
\]

**Proof:** Since the reward is bounded by \( \sigma \), the stochastic estimates of the gradient are bounded, and the inequalities \( \|L_\epsilon(x_k, w_k) + \xi_k\|_2 \leq C \) and \( \|L_\omega(x_k, w_k) + \xi_k\|_2 \leq C \) are satisfied from some constant \( C > 0 \). Then, the proposition is proved by using Proposition 3.

Proposition 7 provides a convergence of the iterates of Algorithm 1 to the \( \epsilon \)-saddle set, \( \mathcal{H}_\epsilon \), with \( \mathcal{O}(1/\epsilon^2) \) samples (or \( \mathcal{O}(1/\sqrt{T}) \) rate). In particular, the proposed analysis provides a sample complexity in terms of \( \epsilon \), while the other dependencies are unclear due to some unknown parameters such as \( C \). Moreover, the convergence to the \( \epsilon \)-saddle set, \( \mathcal{H}_\epsilon \), roughly implies that each \( w_k^{(i)} \) converges to \( w^* \) which solves the optimization in (3). For the specific \( \mathcal{L} \) for our problem, we can obtain stronger convergence results with convergence rates.

**Proposition 8 (Finite-time convergence II):** Consider Algorithm 1 and the assumptions in Proposition 7. Fix any \( \epsilon > 0 \) and \( \delta \in (0, 1) \). If \( T \geq \omega(\kappa/2, \epsilon, \delta) \), then
\[ P[\tilde{w}_T^T \tilde{L} \tilde{w}_T \leq \epsilon] \geq 1 - \delta, \]
where the function \( \omega : \mathbb{R} \times \mathbb{R} \to \mathbb{R} \) is defined in Proposition 7. Moreover, if
\[ T \geq \omega \left( \frac{\min[\lambda_{\min}(\hat{\Phi}^T \tilde{D} \hat{\Phi}^T \tilde{D} \Phi), 1]}{2\sqrt{\lambda_{\max}(\hat{\Phi}^T \tilde{D} \hat{\Phi}^T \tilde{D} \Phi + I)}}, \epsilon, \delta \right), \]
then
\[ P[\|\tilde{\theta}_T - \tilde{\theta}^*\|^2 + \|\tilde{\nu}_T\|^2 \leq \epsilon] \geq 1 - \delta. \]

**Proof:** The proof is based on Proposition 7, the strong convexity of \( \mathcal{L} \) in some arguments, and the Lipschitz continuity of the gradient of \( \mathcal{L} \). In particular, by Proposition 7, if \( T \geq \omega(\epsilon, \delta) \), then with probability \( 1 - \delta \), \((\tilde{x}_T, \tilde{w}_T) \in \mathcal{H}_\epsilon \), meaning that
\[ \mathcal{L}(\tilde{\theta}_T, \tilde{v}_T, \tilde{\mu}_T, \tilde{w}_T) - \mathcal{L}(\tilde{\theta}, \tilde{v}, \tilde{\mu}, \tilde{w}_T) \leq \epsilon. \]

It is easily proved that \( f \) has a Lipschitz gradient with parameter \( \sqrt{\lambda_{\max}(\hat{\Phi}^T \tilde{D} \hat{\Phi}^T \tilde{D} \Phi + I)} \), i.e.,
\[
\|\nabla f(\tilde{\theta}, \tilde{v}, \tilde{\mu}) - \nabla f(\tilde{\theta}', \tilde{v}', \tilde{\mu}')\|_2 \leq \sqrt{\lambda_{\max}(\hat{\Phi}^T \tilde{D} \hat{\Phi}^T \tilde{D} \Phi + I)} \|\tilde{\theta} - \tilde{\theta}'\| \|	ilde{v} - \tilde{v}'\| \|	ilde{\mu} - \tilde{\mu}'\|_2.
\]

Therefore, using [45, Proposition 6.1.9] and using the fact that \((\tilde{\theta}^*, \tilde{v}^*, \tilde{\mu}^*)\) is a minimizer of \( f \), one concludes
\[
\frac{2\sqrt{\lambda_{\max}(\hat{\Phi}^T \tilde{D} \hat{\Phi}^T \tilde{D} \Phi + I)}}{\sqrt{2\lambda_{\max}(\hat{\Phi}^T \tilde{D} \hat{\Phi}^T \tilde{D} \Phi + I)}} \leq \epsilon.
\]

After algebraic manipulations with (21), we obtain
\[
\frac{\min[\lambda_{\min}(\hat{\Phi}^T \tilde{D} \hat{\Phi}^T \tilde{D} \Phi), 1]}{2\sqrt{\lambda_{\min}(\hat{\Phi}^T \tilde{D} \hat{\Phi}^T \tilde{D} \Phi + I)}} \leq \epsilon.
\]

The second result is obtained by replacing \( \epsilon \) with
\[
\frac{\min[\lambda_{\min}(\hat{\Phi}^T \tilde{D} \hat{\Phi}^T \tilde{D} \Phi), 1]}{2\sqrt{\lambda_{\max}(\hat{\Phi}^T \tilde{D} \hat{\Phi}^T \tilde{D} \Phi + I)}} \epsilon.
\]

\( \square \)
The first result in (34) implies that the iterate, \( \bar{w}_T \), reaches a consensus with at most \( O(1/\varepsilon^2) \) samples or at \( O(1/\sqrt{T}) \) rate. Similarly, (35) implies that the squared norm of the errors of \( \hat{\theta}_T \) and \( \hat{v}_T \), \( \|\hat{\theta}_T - \hat{\theta}^*_T\|_2^2 + \|\hat{v}_T\|_2^2 \), converges at \( O(1/\sqrt{T}) \) rate. However, (34) does not suggest anything about the convergence rate of \( \|\bar{w}_T - \bar{w}^*\|_2^2 \) and \( \|\bar{\mu}_T - \bar{\mu}^*\|_2^2 \). Still, their asymptotic convergence is guaranteed by Proposition 7. The main reason is the lack of the strong convexity with respect to these variables. However, we can resolve this issue with a slight modification of the algorithm by adding the regularization term \( \rho/\bar{\theta} \) to the Lagrangian \( \mathcal{L} \) with a small \( \rho > 0 \) so that \( \mathcal{L}(\theta, \bar{v}, \bar{\mu}) \) is strongly convex in \( \theta \) and strongly concave in \( \bar{\mu} \). In this case, the corresponding saddle-points are slightly altered depending on \( \rho \).

Remark 6: Proposition 7 and Proposition 8 apply the analysis of the primal-dual algorithm in Proposition 3, and exhibit \( O(1/\sqrt{T}) \) convergence rate. The recent primal-dual algorithm in [13] has faster \( O(1/T) \) rate, and can be applied to solve the saddle-point problem in (19).

Remark 7: The last line of Algorithm 1 indicates that both the averaged iterates, \( \bar{w}_T(i) = \frac{1}{T} \sum_{k=0}^{T} w_k(i) \), \( i \in \mathcal{V} \), and the last iterate, \( w_T(i) \), \( i \in \mathcal{V} \), can be used for estimates of the solution. The result in [35] proves the asymptotic convergence of the last iterate of Algorithm 1 by using the stochastic approximation method [46]. For the convergence, the step-size rules should satisfy \( \alpha_k > 0 \) \( \alpha_k \rightarrow 0 \), \( \alpha_k \rightarrow \infty \), \( \sum_{k=0}^{\infty} \alpha_k = \infty \), \( \sum_{k=0}^{\infty} \alpha_k^2 < \infty \), called the Robbins-Monro rule. An example is \( \alpha_k = \alpha_0/(k + \beta) \) with \( \alpha_0, \beta > 0 \). On the other hand, Proposition 7 and Proposition 8 prove the convergence of the averaged iterate of Algorithm 1 with convergence rates by using tools in optimization. The step-size rule is \( \alpha_k = \alpha_0\sqrt{k + \beta} \) with \( \alpha_0, \beta > 0 \), which does not obey the Robbins-Monro rule.

Remark 8: There exist several RLs based on stochastic primal-dual approaches. The GTD can be interpreted as a stochastic primal-dual algorithm by using Lagrangian duality theory [8]. The work in [27] proposes primal-dual reinforcement learning algorithm for the single-agent policy optimization problem, where a linear programming form of the MDP problem is solved. A primal-dual algorithm variant of the GTD is investigated in [47] for a single-agent RL problem.

Remark 9: When nonlinear function approximation, such as the neural network, is used, convergence of TD-learning is not guaranteed in general [41] even for the single-agent case. Similarly, with the nonlinear function approximation, the convergence of Algorithm 1 to a global optimal solution is not guaranteed. In particular, for minimization problems, stochastic gradients converge to a local stationary point [48]. On the other hand, convergence of stochastic primal-dual algorithms to a saddle-point for general non-convex min-max problems is still an open problem [49]. In this respect, the convergence of our algorithm with general nonlinear function approximation is a challenging open question, which needs significant efforts in the future.

VIII. SIMULATION

In this section, we provide simulation studies that illustrate potential applicability of the proposed approach. All numerical examples in the sequel were treated with the help of MATLAB R2020a running on a PC with AMD Ryzen 3.69 GHz CPU, 128 GB RAM.

Example 1: In this example, we provide a comparative analysis using simulations. We consider the Markov chain

\[
P^r = \begin{bmatrix}
0.1 & 0.5 & 0.2 & 0.2 \\
0.5 & 0.0 & 0.1 & 0.4 \\
0.0 & 0.9 & 0.1 & 0.0 \\
0.2 & 0.1 & 0.1 & 0.6
\end{bmatrix},
\]

where \( \pi \) is not explicitly specified, \(|\mathcal{S}| = 4\), \( \gamma = 0.8 \), feature vector \( \phi(s) = \begin{bmatrix} \exp(-s^2) \end{bmatrix} \), local expected reward functions

\[
r_1^r = \begin{bmatrix} 0 & 0 & 0 & 50 \end{bmatrix}, \quad r_2^r = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix},
\]

and the five RL agents over the network given in Figure 1.

FIGURE 1. Network topology of five RL agents.

VOLUME 10, 2022

FIGURE 2. Evolution of iterates of the proposed DGTD (solid lines with different colors for different parameters), Algorithm 1. We use the step-size rule \( \alpha_k = 10/(k + 100) \) and \( \kappa = 1 \).

Figure 2 depicts evolutions of two parameter iterates of the proposed DGTD (different colors for different parameters), Algorithm 1. For the simulation, the step-size is set to be \( \alpha_k = 10/(k + 100) \), \( \kappa = 1 \), and the projection sets \( C_{\hat{\theta}}, C_{\hat{v}}, C_{\hat{\mu}}, C_{\bar{w}} \) are set to be a hypercube with length 10.
These hyperparameters were manually set properly for the proposed algorithm to perform reasonably well. It shows that the parameters of five agents reach a consensus and converge to certain numbers. The results empirically demonstrate the proposed DGTD.

Example 2: Consider a $20 \times 20$ continuous space $X$ with three robots (agent 1 (blue), agent 2 (red), and agent 3 (black)), which patrol the space with identical stochastic motion planning policies $\pi_1 = \pi_2 = \pi_3 = \pi$. We consider a single integrator system for each robot $i$: $x_i(t) = u_i(t)$ with the control policy $u_i(t) = -h(x_i(t) - r_i)$ employed from $[50]$, where $t \in \mathbb{R}_+$ is the continuous time, $h \in \mathbb{R}_{++}$ is a constant, $r_i$ is a randomly chosen point in $X$ with uniform distribution over $X$. Under the control policy $u_i(t) = -h(x_i(t) - r_i)$, $x_i(t)$ globally converges to $r_i$ as $t \to \infty$ $[50, \text{Lemma 1}]$. When $x_i$ is sufficiently close to the destination $r_i$, then it chooses another destination $r_i$ uniformly in $X$, and all agents randomly maneuver the space $X$. The continuous space $X$ is discretized into the $20 \times 20$ grid, which forms the state-space $S$. The collaborative objective of the three robots is to reach the goal regions using individually collected reward by each robot. Note that we only consider the policy evaluation problem for a given fixed policy, and the algorithm cannot find an optimal policy. Nevertheless, the algorithm can be applied to the actor-critic algorithm $[39]$ to find an optimal policy. In particular, the actor-critic algorithm is a policy optimization algorithm which consists of two parts, 1) policy evaluation and 2) policy improvement. The first part evaluates the current policy and the second part improves the policy using the value function computed in the first part. The developed distributed TD learning in this paper can be applied to the first policy evaluation part in the actor-critic algorithm.

We assume that each robot is equipped with a different sensor that can detect different regions, while a pair of robots can exchange their parameters, when the distance between them is less than or equal to 5. We assume that robots do not interfere with each other; thereby we can consider three independent MDPs with identical transition models.

The three regions and robots are depicted in Figure 3, where the blue region is detected only by agent 1 (blue circle), the red region is detected only by agent 2 (red circle), and the black region only by agent 3 (black circle).

For each agent, the detection occurs only if the UAV flies over the region, and a reward $r = 100$ is given in this case. Algorithm 1 is applied with $\gamma = 0.5$ and $\Phi = I|S|$ (tabular representation). We run Algorithm 1 with 50000 iterations, the step-size $\omega_k = 100/(k + 10000)$, and $\kappa = 1$. The projection sets $C_\beta, C_\nu, C_\mu, C_\kappa$ are set to be a hypercube with length 20. The simulation results are shown in Figure 4, where the brighter regions imply that the value is higher than darker regions, and the three bright regions indicate that the agent’s value function has higher values at states in these regions. The results suggest that all agents successfully estimate identical value functions, which are aware of three regions despite of the incomplete sensing abilities. The obtained value function can be used to design a motion planning policy in combination with the actor-critic algorithm.

**FIGURE 3.** Three rectangular regions that can be detected by three different UAVs. The blue region is detected only by agent 1 (blue circle), the red region is detected only by agent 2 (red circle), and the black region only by agent 3 (black circle). For each agent, the detection occurs only if the UAV flies over the region, and a reward $r = 100$ is given in this case.

**FIGURE 4.** Example 2. Value functions of three different agents, where the $x$ and $y$ axis are the state-space, the brighter regions imply that the value is higher than darker regions, and the three bright regions indicate that the agent’s value function has higher values at states in these regions.

IX. CONCLUSION

In this paper, we study a distributed GTD-learning for multi-agent MDPs using a stochastic primal-dual algorithm. Each agent receives local reward through a local processing, while information exchange over random communication networks allows them to learn the global value function corresponding to a sum of local rewards. Possible future research includes its extension to actor-critic and Q-learning algorithms.

X. PROOF OF PROPOSITION 3

In this section, we will provide a proof of Proposition 3. We begin with a basic technical lemma.

**Lemma 3 (Basic Iterate Relations [28]):** Let the sequences $(x_k, w_k)_{k=0}^\infty$ be generated by the stochastic sub-gradient algorithm in (9) and (10). Then, we have:

1) For any $x \in X$ and for all $k \geq 0$,

$$E[\|x_{k+1} - x\|^2 | F_k]$$
\[
\|x_k - x\|^2 + \alpha_k^2 \mathbb{E}[\|\mathcal{L}_\epsilon(x_k, w_k) + \epsilon_k\|^2 | \mathcal{F}_k] \\
-2\alpha_k (\mathcal{L}(x_k, w_k) - \mathcal{L}(x, w_k)).
\]

2) For any \( w \in \mathcal{W} \) and for all \( k \geq 0 \),
\[
\mathbb{E}[\|w_{k+1} - w\|^2 | \mathcal{F}_k] \\
\leq \|w_k - w\|^2 + \alpha_k^2 \mathbb{E}[\|\mathcal{L}_\epsilon(x_k, w_k) + \epsilon_k\|^2 | \mathcal{F}_k] \\
+ 2\alpha_k (\mathcal{L}(x_k, w_k) - \mathcal{L}(x, w_k)).
\]

**Proof:** The result can be obtained by the iterate relations in [28, Lemma 3.1] and taking the expectations. \( \square \)

**Lemma 4 (Bernstein Inequality for Martingales [51]):** Let \((\mathcal{M}_T)_{T=0}^{\infty}\) be a square integrable martingale such that \( M_0 = 0 \). Assume that \( \Delta \mathcal{M}_T \leq b, \forall T \geq 1 \) with probability one, where \( b > 0 \) is a real number and \( \Delta \mathcal{M}_T \) is the martingale difference defined as \( \Delta \mathcal{M}_T := \mathcal{M}_T - \mathcal{M}_{T-1}, T \geq 1 \). Then, for any \( \varepsilon \in [0, b] \) and \( a > 0 \),
\[
\mathbb{P}\left[ \frac{1}{T} \mathcal{M}_T \geq \varepsilon; \frac{1}{T} (\mathcal{M})_T \leq a \right] \leq \exp\left(-\frac{T \varepsilon^2}{2(a + b \varepsilon / 3)}\right),
\]
where
\[
(\mathcal{M})_T := \sum_{t=0}^{T-1} \mathbb{E}[\Delta \mathcal{M}_t^2 | \mathcal{F}_t].
\]

For any \( x \in \mathcal{X} \) and \( y \in \mathcal{Y} \), define
\[
\mathcal{E}_k^{(1)}(x) := \|x_k - x\|^2, \\
\mathcal{E}_k^{(2)}(w) := \|w_k - w\|^2
\]
and
\[
H_k(x) := \frac{1}{2\alpha_k} (\mathcal{E}_k^{(1)}(x) - \mathbb{E}[\mathcal{E}_k^{(1)}(x) | \mathcal{F}_k]), \\
R_k(w) := \frac{1}{2\alpha_k} (\mathcal{E}_k^{(2)}(w) - \mathbb{E}[\mathcal{E}_k^{(2)}(w) | \mathcal{F}_k]).
\]

We use \( \mathbb{E}[\|\mathcal{L}_\epsilon(x_k, w_k) + \epsilon_k\|^2 | \mathcal{F}_k] \leq C^2 \) and rearrange terms in Lemma 3 to have
\[
\mathcal{L}(x_k, w_k) - \mathcal{L}(x, w_k) \\
\leq \frac{1}{2\alpha_k} (\mathcal{E}_k^{(1)}(x) - \mathbb{E}[\mathcal{E}_k^{(1)}(x) | \mathcal{F}_k]) + \frac{\alpha_k}{2} C^2, \\
\forall x \in \mathbb{R}^{[S|A]}, \mathbb{R}^{[S]}, \\
- \frac{1}{2\alpha_k} (\mathcal{E}_k^{(2)}(w) - \mathbb{E}[\mathcal{E}_k^{(2)}(w) | \mathcal{F}_k]) - \frac{\alpha_k C^2}{2} \\
= R_k(w) \\
\leq \mathcal{L}(x_k, w_k) - \mathcal{L}(x_k, w), \quad \forall w \in \mathbb{R}^{[S|A]} \times \mathbb{R}^{[S|A]}. 
\]

Adding these relations over \( k = 0, \ldots, T - 1 \), dividing by \( T \), and rearranging terms, we have
\[
\frac{1}{T} \sum_{k=0}^{T-1} R_k(w) - \frac{1}{T} \sum_{k=0}^{T-1} \frac{\alpha_k}{2} C^2 \\
\leq \frac{1}{T} \sum_{k=0}^{T-1} (\mathcal{L}(x_k, w_k) - \mathcal{L}(x_k, w)), \quad \forall w \in \mathcal{W}. 
\]

Similarly, we have from (37)
\[
\frac{1}{T} \sum_{k=0}^{T-1} (\mathcal{L}(x_k, w_k) - \mathcal{L}(x_k, w_k)) \\
\leq \frac{1}{T} \sum_{k=0}^{T-1} H_k(x) + \frac{1}{T} \sum_{k=0}^{T-1} \frac{\alpha_k}{2} C^2, \quad \forall x \in \mathcal{X}. 
\]

Multiplying both sides of (39) by \(-1\) and adding it with (40) yields
\[
\frac{1}{T} \sum_{k=0}^{T-1} (\mathcal{L}(x_k, w_k) - \mathcal{L}(x_k, w_k)) \\
\leq \frac{1}{T} \sum_{k=0}^{T-1} R_k(w) + \frac{1}{T} \sum_{k=0}^{T-1} H_k(x) + \frac{C^2}{T} \sum_{k=0}^{T-1} \alpha_k.
\]

Using the convexity of \( \mathcal{L} \) with respect to the first argument and the concavity of \( \mathcal{L} \) with respect to the second argument, it follows from the last inequality that
\[
\mathcal{L}(\hat{x}_T, w) - \mathcal{L}(x, \hat{w}_T) \\
\leq \frac{1}{T} \Phi_1(x) + \frac{1}{T} \Phi_2(y) + \frac{1}{T} \mathcal{M}_T + \frac{C^2}{T} \sum_{k=0}^{T-1} \alpha_k, \\
\forall x \in \mathcal{X}, w \in \mathcal{W},
\]
where
\[
\Phi_1(x) := \sum_{k=0}^{T-1} \frac{1}{2\alpha_k} (\mathcal{E}_k^{(1)}(x) - \mathcal{E}_{k+1}^{(1)}(x)), \\
\Phi_2(w) := \sum_{k=0}^{T-1} \frac{1}{2\alpha_k} (\mathcal{E}_k^{(2)}(w) - \mathcal{E}_{k+1}^{(2)}(w)), \\
\mathcal{M}_T := \sum_{k=0}^{T-1} \frac{1}{2\alpha_k} (\mathcal{E}_k^{(1)}(x) + \mathcal{E}_{k+1}^{(1)}(w)) \\
- \mathbb{E}[\mathcal{E}_k^{(1)}(x) | \mathcal{F}_k] - \mathbb{E}[\mathcal{E}_{k+1}^{(2)}(w) | \mathcal{F}_k].
\]

As a next step, we derive bounds on the terms \( \Phi_1(x) \) and \( \Phi_2(w) \). First, \( \Phi_1(x) \) is bounded by using the chains of inequalities
\[
\Phi_1(x) = \sum_{k=0}^{T-1} \frac{1}{2\gamma_k} (\mathcal{E}_k^{(1)}(x) - \mathcal{E}_{k+1}^{(1)}(x)) \\
\leq \sum_{k=0}^{T-1} \frac{1}{2\gamma_k} (\mathcal{E}_k^{(1)}(x) - \mathcal{E}_{k+1}^{(1)}(x)) + \frac{\gamma_k}{\gamma_{k+1}} \mathcal{E}_k^{(1)}(x) \\
= \frac{1}{2} \left( \frac{1}{\gamma_0} \mathcal{E}_0^{(1)}(x) + \sum_{k=0}^{T-1} \left( \frac{1}{\gamma_k} - \frac{1}{\gamma_{k+1}} \right) \mathcal{E}_k^{(1)}(x) \right).
\]
\[
\begin{align*}
\frac{C^2}{2} \left( \frac{1}{y_0} + \sum_{k=0}^{T-1} \left( \frac{1}{y_{k+1}} - \frac{1}{y_k} \right) \right) & = \frac{C^2}{2} \gamma_T, \\
\end{align*}
\]

where (49) is due to \( c^{(1)}(x) \leq C^2 \), \( \forall x \in X \). Similarly, we have \( \Phi_x(w) \leq \frac{C^2}{2} \gamma_T \). Combining the last inequality with (41) yields
\[
\mathcal{L}(\hat{x}_T, w) - \mathcal{L}(x_w) \leq \frac{C^2}{T} \gamma_T + \frac{C^2}{T} \sum_{k=0}^{T-1} \left[ y_k + \frac{1}{T} \mathcal{M}_T \right].
\]

(43)

Plugging \( \alpha_k = 0/\sqrt{k+1} \) into the first term, we have
\[
\begin{align*}
\mathcal{L}(\hat{x}_T, w) - \mathcal{L}(x_w) & \leq \frac{C^2}{T} \gamma_T + \frac{C^2}{T} \sum_{k=0}^{T-1} \left[ \alpha_k + \frac{1}{T} \mathcal{M}_T \right] \\
\end{align*}
\]

where (47) follows from the relation \( a - b \leq \frac{a^2}{2} + \frac{|b|^2}{2} - 2d^2 b \) for any vectors \( a, b \). (48) follows from the Cauchy-Schwarz inequality. Moreover, we will find constants \( C^2/2 + 2C^2 \). Combining the last two inequalities, we have \( \mathcal{M}_T \leq a \) holds, where \( a = (\alpha_0 + 2)^2 C^4 \). Using \( \mathcal{L}(\hat{x}_T, w) = \mathcal{L}(x_w) \leq \epsilon \), we can prove that the first inequality holds if
\[
\begin{align*}
T & \geq \frac{4C^4(2\alpha_0 - 1) + \alpha_0^2}{\epsilon^2}.
\end{align*}
\]

To prove the second inequality with high probability, we will use the Bernstein inequality in Lemma 4. To do so, one easily proves that \( \mathcal{L}(\mathcal{M}_T + [\mathcal{F}_T]) = \mathcal{M}_T \), and hence, \( \mathcal{M}_T \geq k \). Moreover, we will find constants \( a > 0 \) and \( b > 0 \) such that \( \mathcal{M}_T = \mathcal{M}_T + - b \) and \( \mathcal{M}_T \leq b \). Noting that \( \mathcal{M}_T = \frac{1}{2\alpha_k}(\mathcal{E}_k^{(1)} - \mathcal{E}_k^{(1)}|[\mathcal{F}_T]) + \frac{1}{2\alpha_k}(\mathcal{E}_k^{(1)}|[\mathcal{F}_T]) \), we obtain the bounds for the first two terms
\[
\begin{align*}
\frac{1}{2\alpha_k}(\mathcal{E}_k^{(1)} + \mathcal{E}_k^{(1)}|[\mathcal{F}_T]) & = \frac{1}{2\alpha_k} \left[ \frac{1}{2\alpha_k} \mathcal{L}_x(x_k, w_k) - \alpha_k \mathcal{E}_k - x^* \right] - \mathcal{E}_{[\mathcal{F}_T]} \\
\end{align*}
\]

(45)
\[ \leq a_k^2 \|L_x(x_k, w_k)\|_2^2 + 2a_k \|L_x(x_k, w_k)\|_2 \|x_k - x^*\|_2 \] 
\[ \leq a_k (a_0 C^2 + 2C^2), \]  
where (51) follows from the relation \( \|a - b\|_2^2 = \|a\|_2^2 + \|b\|_2^2 - 2a^T b \) for any vectors \( a, b \), (52) follows from the Cauchy-Schwarz inequality, (53) is due to the nonexpansive map property of the projection \( \Pi_{\Gamma}(x) = \Gamma(x) \) \[ \leq \|a - b\|_2 \] \( \geq 0 \). Note that for any \( \delta \in (0, 1) \), \[ \exp \left( \frac{T \epsilon^2}{8(a + b \delta/6)} \right) \leq \delta \] holds if and only if \( T \geq \frac{8(a + b \delta/6)}{\epsilon^2} \ln(1/\delta) \). By plugging \( a \) and \( b \) given before into the inequality, it holds that if 
\[ T \geq \frac{8C^2(a_0 + 2)C^2 + (a_0 + 4)\epsilon/6}{\epsilon^2} \ln \left( \frac{1}{\delta} \right), \] then with probability at least \( 1 - \delta \), we have \( \mathcal{M}_T/T \leq \epsilon/2 \). Combined with (45), one concludes that under the conditions in the statement of Proposition 3, with probability at least \( 1 - \delta \), \( L(\hat{x}_T, w) - L(x, \hat{w}_T) \leq \epsilon \) holds. By Definition 2, it implies 
\[ \mathbb{P} \left( \hat{x}_T, \hat{w}_T \right) \in \mathcal{H}_d \geq 1 - \delta. \] This completes the proof.

REFERENCES

[1] R. S. Sutton, H. R. Maei, and C. Szepesvári, “A convergent o(n) temporal-difference algorithm for off-policy learning with linear function approximation,” in Proc. Adv. Neural Inf. Process. Syst., 2009, pp. 1609–1616.
[2] R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora, “Fast gradient-descent methods for temporal-difference learning with linear function approximation,” in Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML), 2009, pp. 993–1000.
[3] S. Kar, J. M. Moura, and H. V. Poor, “QD-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations,” IEEE Trans. Signal Process., vol. 61, no. 7, pp. 1848–1862, May 2013.
[4] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, “Fully decentralized multi-agent reinforcement learning with networked agents,” in Proc. Int. Conf. Mach. Learn. (ICML), 2018, pp. 5872–5881.
[5] W. Suttle, Z. Yang, K. Zhang, Z. Wang, T. Basar, and J. Liu, “A multi-agent off-policy actor-critic algorithm for distributed reinforcement learning,” IPAC-PapersOnline, vol. 53, no. 2, pp. 1549–1554, 2020.
[6] C. Qu, S. Mannor, H. Xu, Y. Qi, L. Song, and J. Xiong, “Value propagation for decentralized networked deep multi-agent reinforcement learning,” in Proc. Adv. Neural Inf. Process. Syst., 2019, pp. 1182–1191.
[7] S. V. Macua, A. Tukiainen, D. García-Ocaña Hernández, D. Baldizo, E. M. de Cote, and S. Zazo, “Diff-DAC: Distributed actor-critic for average multitask deep reinforcement learning,” 2017, arXiv:1710.10363.
[8] S. V. Macua, J. Chen, S. Zazo, and A. H. Sayed, “Distributed policy evaluation under multiple behavior strategies,” IEEE Trans. Autom. Control, vol. 60, no. 5, pp. 1260–1274, May 2015.
[9] M. S. Stankovic and S. S. Stankovic, “Multi-agent temporal-difference learning with linear function approximation: Weak convergence under time-varying network topologies,” in Proc. Amer. Control Conf. (ACC), Jul. 2016, pp. 167–172.
[10] T. Doan, S. Maguluri, and J. Romberg, “Finite-time analysis of distributed TD(0) with linear function approximation on multi-agent reinforcement learning,” in Int. Conf. Mach. Learn., 2019, pp. 1626–1635.
[11] H.-T. Wai, Z. Yang, Z. Wang, and M. Hong, “Multi-agent reinforcement learning via double averaging primal-dual optimization,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 9649–9660.
[12] L. Cassano, K. Yuan, and A. H. Sayed, “Multiagent fully decentralized value function learning with linear convergence rates,” IEEE Trans. Autom. Control, vol. 66, no. 4, pp. 1497–1512, Apr. 2020.
[13] D. Ding, X. Wei, Z. Yang, Z. Wang, and M. R. Jovanović, “Fast multi-agent temporal-difference learning via homotopy stochastic primal-dual optimization,” 2019, arXiv:1908.02805.
[14] P. Heredia and S. Mou, “Finite-sample analysis of multi-agent policy evaluation with kernelized gradient temporal difference,” in Proc. 59th IEEE Conf. Decis. Control (CDC), Dec. 2020, pp. 5647–5652.
[15] A. Jadababaei, J. Lin, and A. Morse, “Coordination of groups of mobile autonomous agents using nearest neighbor rules,” IEEE Trans. Autom. Control, vol. 48, no. 6, pp. 988–1001, Mar. 2003.
[16] A. Redic, A. Ozdaglar, and P. A. Parrilo, “Constrained consensus and optimization in multi-agent networks,” IEEE Trans. Autom. Control, vol. 55, no. 4, pp. 922–938, Apr. 2010.
[17] I. Lobel and A. Ozdaglar, “Distributed subgradient methods for convex optimization over random networks,” IEEE Trans. Autom. Control, vol. 56, no. 6, pp. 1291–1306, Jun. 2011.
[18] G. Qu and N. Li, “Harnessing smoothness to accelerate distributed optimization,” IEEE Trans. Control Netw. Syst., vol. 5, no. 3, pp. 1245–1260, Sep. 2017.
[19] W. Shi, Q. Ling, G. Wu, and W. Yin, “EXTRA: An exact first-order algorithm for decentralized consensus optimization,” SIAM J. Optim., vol. 25, no. 2, pp. 944–966, Apr. 2015.
[20] A. Mohitkar and A. Ribeiro, “DSA: Decentralized double stochastic averaging gradient algorithm,” J. Mach. Learn. Res., vol. 17, no. 1, pp. 1–35, 2016.
[21] J. Lei, H.-F. Chen, and H.-T. Fang, “Primal–dual algorithm for distributed constrained optimization,” Syst. Control Lett., vol. 96, pp. 110–117, Oct. 2016.
[22] D. P. Bertsekas and J. N. Tsitsiklis, Neuro-Dynamic Programming. Belmont, MA, USA: Athena Scientific, 1996.
[23] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[24] D. P. Bertsekas, Nonlinear Programming. Belmont, MA, USA: Athena Scientific, 1999.
[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Proc. Adv. Neural Inf. Process. Syst., vol. 27, 2014, pp. 1–9.
[26] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, “Robust stochastic approximation and random search methods,” IEEE Trans. Autom. Control, vol. 52, no. 7, pp. 1221–1230, Jul. 2007.
[27] S. V. Macua, A. Tukiainen, D. García-Ocaña Hernández, D. Baldizo, E. M. de Cote, and S. Zazo, “Diff-DAC: Distributed actor-critic for average multitask deep reinforcement learning,” 2017, arXiv:1710.10363.
[33] D. Lee, N. He, P. Kamalaruban, and V. Cevher, “Optimization for reinforcement learning: From a single agent to cooperative agents,” IEEE Signal Process. Mag., vol. 37, no. 3, pp. 123–135, May 2020.

[34] X. Sha, J. Zhang, K. You, K. Zhang, and T. Başar, “Fully asynchronous policy evaluation in distributed reinforcement learning over networks,” 2020, arXiv:2003.00433.

[35] D. Lee, H. Yoon, and N. Hovakimyan, “Primal-dual algorithm for distributed reinforcement learning: Distributed GTD,” in Proc. IEEE Conf. Decis. Control (CDC), Dec. 2018, pp. 1967–1972.

[36] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: Algorithms and theory,” IEEE Trans. Autom. Control, vol. 51, no. 3, pp. 401–420, Mar. 2006.

[37] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, Cambridge, MA, USA: MIT Press, 1998.

[38] G. A. Rummery and M. Niranjan, On-Line Q-Learning Using Connectionist Systems, vol. 37. Cambridge, U.K.: Univ. Cambridge, 1994.

[39] V. R. Konda and J. N. Tsitsiklis, “Actor-critic algorithms,” in Proc. Adv. Neural Inf. Process. Syst., 1999, pp. 1008–1014.

[40] S. Bhatnagar, H. Prasad, and L. Prashanth, Stochastic Recursive Algorithms for Optimization: Simultaneous Perturbation Methods, vol. 434. Berlin, Germany: Springer, 2012.

[41] J. N. Tsitsiklis and B. Van Roy, “An analysis of temporal-difference learning with function approximation,” IEEE Trans. Autom. Control, vol. 42, no. 5, pp. 674–690, May 1997.

[42] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, and S. Petersen, “Human-level control through deep reinforcement learning,” Nature, vol. 518, pp. 529–533, Feb. 2015.

[43] A. Ghosh, S. Boyd, and A. Saberi, “Minimizing effective resistance of a graph,” SIAM Rev., vol. 50, no. 1, pp. 57–66, 2008.

[44] H. Maei, C. Szepesvari, S. Bhatnagar, D. Precup, D. Silver, and R. S. Sutton, “Convergent temporal-difference learning with arbitrary smooth function approximation,” in Proc. Adv. Neural Inf. Process. Syst., vol. 22, 2009, pp. 1–9.

[45] D. P. Bertsekas, Convex Optimization Algorithms, Belmont, MA, USA: Athena Scientific, 2015.

[46] H. Kushner and G. G. Yin, Stochastic Approximation and Recursive Algorithms and Applications, vol. 35. Berlin, Germany: Springer, 2003.

[47] S. Mahadevan, B. Liu, P. Thomas, W. Dabney, S. Giguere, N. Jacek, I. Gemp, and J. Liu, “Proximal reinforcement learning: A new theory of sequential decision making in primal-dual spaces,” arXiv preprint arXiv:1405.6757, 2014.

[48] S. Ghadimi and G. H. Lan, “Stochastic first-and zeroth-order methods for nonconvex stochastic programming,” SIAM J. Optim., vol. 23, no. 4, pp. 2343–2398, 2013.

[49] M. Liu, H. Rafique, Q. Lin, and T. Yang, “First-order convergence theory for weakly-convex-weakly-concave min-max problems,” 2018, arXiv:1810.10207.

[50] D. Panagou, M. Turpin, and V. Kumar, “Decentralized goal assignment and trajectory generation in multi-robot networks: A multiple Lyapunov functions approach,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2014, pp. 6757–6762.

[51] B. Bercu, B. Delyon, and E. Rio, Concentration Inequalities for Sums Martingales. Berlin, Germany: Springer, 2015.

DONGHWA LEE (Member, IEEE) received the B.S. degree in electronic engineering from Konkuk University, Seoul, South Korea, in 2008, the M.S. degree in electrical and electronic engineering from Yonsei University, Seoul, in 2010, the M.S. degree in mathematics and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2017. He was a Postdoctoral Research Associate with the Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Champaign, IL, USA, from 2017 to 2019. He is currently an Assistant Professor with the School of Electrical Engineering, KAIST, Daejeon, South Korea. His current research interests include stochastic programming, multi-agent systems, and reinforcement learning.

DO WAN KIM received the B.S., M.S., and Ph.D. degrees from the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea, in 2002, 2004, and 2007, respectively. He worked at the Engineering Research Institute, Yonsei University. He also worked as a Visiting Scholar at the Department of Mechanical Engineering, University of California at Berkeley, Berkeley. In 2009, he worked as a Research Professor at the Department of Electrical and Electronic Engineering, Yonsei University. Since 2010, he has been a Professor with the Department of Electrical Engineering, Hanbat National University, Daejeon, South Korea. His current research interests include analysis and synthesis of nonlinear sampled-data control systems and autonomous unmanned vehicles.

JIANGHAI HU (Member, IEEE) received the B.E. degree in automatic control from Xi’an Jiaotong University, China, in 1994, and the M.A. degree in mathematics and the Ph.D. degree in electrical engineering from the University of California at Berkeley, Berkeley, in 2002 and 2003, respectively. He is currently a Professor with the School of Electrical and Computer Engineering, Purdue University. His research interests include hybrid systems, multi-agent systems, control of systems with uncertainty, and control applications. He is an Associate Editor of SIAM Journal on Control and Optimization and Nonlinear Analysis: Hybrid Systems.

* * *