Primal-Dual Algorithm for Distributed Reinforcement Learning: Distributed GTD

Donghwan Lee, Hyungjin Yoon, and Naira Hovakimyan

Abstract— **The goal of this paper is to study a distributed version of the gradient temporal-difference (GTD) learning algorithm for multi-agent Markov decision processes (MDPs). The temporal-difference (TD) learning is a reinforcement learning (RL) algorithm which learns an infinite horizon discounted cost function (or value function) for a given fixed policy without the model knowledge. In the distributed RL case each agent receives local reward through a local processing. Information exchange over sparse communication network allows the agents to learn the global value function corresponding to a global reward, which is a sum of local rewards. In this paper, the problem is converted into a constrained convex optimization problem with a consensus constraint. Then, we propose a primal-dual distributed GTD algorithm and prove that it almost surely converges to a set of stationary points of the optimization problem.**

I. INTRODUCTION

The goal of this paper is to study a distributed version of the gradient temporal-difference (GTD) learning algorithm, originally presented in [1], [2], for multi-agent Markov decision processes (MDPs). There are N agents $i \in \{1, \ldots, N\} =: \mathcal{V}$, which do not know the statistics of the state transitions and rewards. Each agent i receives local reward following a given fixed local policy π_i. However, it will be able to learn the global infinite horizon discounted cost function (or value function) corresponding to the reward that is a sum of local rewards through information exchange over a sparse communication network. This paper only focuses on the value evaluation problem with fixed local policies π_i. However, the proposed approach can be extended to actor-critic algorithms, which have multi-agent cooperative control design applications.

A distributed Q-learning (QD-learning) was studied in [3]. The focus of [3] is to learn an optimal Q-factor [4] for a global reward expressed as a sum of local rewards, while each agent is only aware of its local reward. This work therefore addresses the multi-agent optimal policy design problem. If each agent has access to partial states and actions, then the transition model of each agent becomes non-stationary. This is because the state transition model of each agent depends on the other agents' policies. In [3], the authors assumed that each agent observes the global state and

action; therefore, this non-stationary problem does not occur in Q-learning settings. Distributed actor-critic algorithms were explored in [5] with a similar setting. Each agent acquires local observations and rewards, but it tries to learn an optimal policy that maximizes the long-term average of total reward which is a sum of local rewards. It was assumed that each agent's state-action does not change the other agents' transition models. In a more recent work [6], consensus-based actor-critic algorithms were studied, where the authors assumed that the transition model depends on the joint action-states of all agents, and that each agent can observe the entire combination of action-states.

In [7], a distributed policy evaluation was studied with the GTD from [1], [2] combined with consensus steps. The study focused on the scenario that there exists only one global reward, each agnet behaves according to their own behavior policy π_i, and the agents cooperate to learn the value function of the target policy π; thereby, it is a multi-agent off-policy learning scheme. It was also assumed that each agent can only explore a small subset of the MDP states. A consensus-based GTD was also addressed in [8]. The authors considered a problem similar to [7], and the weak convergence of the algorithm was proved. In [9], a gossip-based distributed temporal difference (TD [4]) learning was investigated. Compared to the previous work, the main difference in [9] is that all agents know the global reward, but they have different linear function approximation architectures with different features and parameters of different dimensions. Agents cooperate to find a value function with a linear function approximation consisting of aggregated features of all agents to reduce computational costs. Lastly, the papers [10], [11] addressed distributed consensus-based stochastic gradient optimization algorithms for general convex and non-convex objective functions, respectively. Whenever the learning task can be expressed as a minimization of an objective function, e.g., GTD [1], [2] or the residual method [12], algorithms in [10], [11] can be applied. Besides, [13] studied a distributed Newton method for policy gradient methods.

The main contribution of this paper is the development of a new class of distributed GTD algorithm based on primal-dual iterations as compared to the original one in [2]. The most relevant previous studies which addressed the same problem setting are [5], [6]. Even though [5], [6] studied actor-critic algorithms, if the actor updates are ignored with fixed policies, then they can deal with the same problem as ours. The main difference compared to the previous result is that the proposed algorithm incorporates the consensus task into an equality constraint, while those in [5], [6]

use the averaging consensus steps explicitly. Therefore, our algorithm views the problem as a constrained optimization, and solves it using a primal-dual saddle point algorithm. The proposed method is mainly motivated by [7], where the GTD was interpreted as a primal-dual algorithm using Lagrangian duality theory. The proposed algorithm was also motivated by the continuous-time consensus optimization algorithm from [14]–[16], where the consensus equality constraint was introduced. The recent primal-dual reinforcement learning algorithm from [17] also inspired the development in this paper. We also note a primal-dual variant of the GTD in [18] with proximal operator approaches.

One of the benefits of the proposed scheme is that the consensus and learning tasks are unified into a single ODE. Therefore, the convergence can be proved solely based on the ODE methods [19]–[21], and the proof is relevantly simpler. The second possible advantage is that the proposed algorithm is a stochastic primal-dual method for solving saddle point problems, and hence some analysis tools from optimization perspectives, such as [17], can be applied (for instance, the convergence speed and complexity of the algorithm), and this agenda is briefly discussed at the end of the paper. Full extension in this direction will appear in an extended version of this paper. The third benefit of the approach is that the method can be directly extended to the case when the communication network is stochastic. In addition, the proposed method can be generalized to an actor-critic algorithm and off-policy learning. In this paper, we will focus on a convergence analysis based on the ODE approach [19]–[21]. Several open questions remain. For example, it is not clear if there exists a theoretical guarantee that the proposed algorithm improves previous consensus algorithms [5], [6], [8]. Brief discussions are included in the example section.

II. PRELIMINARIES

A. Notation

The adopted notation is as follows: $\mathbb{R}^n$: n-dimensional Euclidean space; $\mathbb{R}^{n \times m}$: set of all $n \times m$ real matrices; A^T: transpose of matrix A; I_n: $n \times n$ identity matrix; I: identity matrix with an appropriate dimension; $\| \cdot \|$: standard Euclidean norm; for any positive-definite D, $\|x\|_D := \sqrt{x^T D x}$; for a set $\mathcal{S}$, $|\mathcal{S}|$ denotes the cardinality of the set; $\mathbb{E}[\cdot]$: expectation operator; $\mathbb{P}[\cdot]$: probability of an event; for any vector x, $[x]_i$ is its i-th element; for any matrix P, $[P]_{ij}$ indicates its element in i-th row and j-th column; if $\mathbf{z}$ is a discrete random variable which has n values and $\mu \in \mathbb{R}^n$ is a stochastic vector, then $\mathbf{z} \sim \mu$ stands for $\mathbb{P}[\mathbf{z} = i] = [\mu]_i$ for all $i \in \{1, \ldots, n\}$; $\mathbf{1}$ denotes a vector with all entries equal to one; $\text{dist}(\mathcal{S}, x)$: standard Euclidean distance of a vector x from a set $\mathcal{S}$, i.e., $\text{dist}(\mathcal{S}, x) := \inf_{y \in \mathcal{S}} \|x - y\|$; for a convex closed set $\mathcal{S}$, $\Gamma_{\mathcal{S}}(x) := \arg \min_{y \in \mathcal{S}} \|x - y\|$.

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 $\mathcal{G} = (\mathcal{E}, \mathcal{V})$. We define the neighbor set of node i as $\mathcal{N}_i := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. The adjacency matrix of $\mathcal{G}$ is defined as a matrix W with $[W]_{ij} = 1$, if

and only if $(i, j) \in \mathcal{E}$. If $\mathcal{G}$ 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 diagonal with $[H]_{ii} = |\mathcal{N}_i|$. If the graph is undirected, then L is symmetric positive semi-definite. It holds that $L\mathbf{1} = 0$. We put the following assumption on the graph $\mathcal{G}$.

Assumption 1: $\mathcal{G}$ is connected.

Under Assumption 1, 0 is a simple eigenvalue of L.

III. REINFORCEMENT LEARNING OVERVIEW

We briefly review basic RL algorithm from [22] with linear function approximation for the single agent case. A Markov decision process 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, $P(s, a, s') := \mathbb{P}[s'|s, a]$ is a tensor that represents the unknown state transition probability from state s to s' given action a, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. The stochastic policy is a mapping $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ representing the probability $\pi(s, a) = \mathbb{P}[a|s]$, $r^\pi(s) : \mathcal{S} \to \mathbb{R}$ is defined as $r^\pi(s) := \mathbb{E}_{a \sim \pi(s)}[r(s, a)]$, P^π denotes the transition matrix whose (s, s') entry is $\mathbb{P}[s'|s] = \sum_{a \in \mathcal{A}} \mathbb{P}[s'|s, a]\pi(s, a)$, and $d : \mathcal{S} \to \mathbb{R}$ denotes the stationary distribution of the observation $s \in \mathcal{S}$. The infinite-horizon discounted value function with policy π and reward r is

$$J^\pi(s) := \mathbb{E}_{\pi, P}\left[\sum_{k=0}^{\infty} \gamma^{k-1} r^\pi(s_k) \middle| s_0 = s\right],$$

where $\mathbb{E}_{\pi, P}$ implies the expectation taken with respect to the state-actor trajectories following the state transition P and policy π. Given pre-selected basis (or feature) functions $\phi_1, \ldots, \phi_q : \mathcal{S} \to \mathbb{R}$, $\Phi \in \mathbb{R}^{|\mathcal{S}| \times q}$ is defined as a full column rank matrix whose i-th row vector is $\begin{bmatrix} \phi_1(i) & \cdots & \phi_q(i) \end{bmatrix}$. The goal of RL with the linear function approximation is to find the weight vector w such that $J_w = \Phi w$ approximates J^π. This is typically done by minimizing the *mean-square Bellman error* loss function [2]

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

where D is a symmetric positive-definite matrix. For online learning, we assume that D is a diagonal matrix with positive diagonal elements $d(s), s \in \mathcal{S}$. The residual method [12] applies the gradient descent type approach $w_{k+1} = w_k - \alpha_k \nabla_w \text{MSBE}(w)(w)$, where $\nabla_w \text{MSBE}(w) = (\gamma P^\pi \Phi - \Phi)^T (r^\pi + \gamma P^\pi \Phi w - \Phi w)$. In the model-free learning, the gradient is replaced with a sample-based stochastic estimate. A drawback of the residual method is that the next observation s' should be sampled twice to obtain an unbiased gradient estimate. In the TD learning [4], [22] with a linear function approximation, the problem is resolved by ignoring the first $\gamma P^\pi \Phi$ in the gradient $\nabla_w \text{MSBE}(w)$: $\nabla_w \text{MSBE}(w) \cong (-\Phi)^T D(r^\pi + \gamma P^\pi \Phi w - \Phi w)$. If the linear function approximation is used, then this algorithm converges to an optimal solution of (1). Compared to the residual method, the double sampling issue does not occur. In the

above two methods, the fixed point problem $r^\pi + \gamma P^\pi \Phi w = \Phi w$ may not have a solution in general because the left-hand side need not lie in the range space of Φ. To address this problem, the GTD in [2] solves instead the minimization of the *mean-square projected Bellman error* loss function

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

where Π is the projection onto the range space of Φ, denoted by $R(\Phi)$: $\Pi(x) := \arg\min_{x' \in R(\Phi)} \|x - x'\|_D^2$. 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 TD learning, the main advantage of GTD [1], [2] algorithms are their off-policy learning abilities.

IV. DISTRIBUTED REINFORCEMENT LEARNING OVERVIEW

Consider N reinforcement learning agents labelled by $i \in \{1, \ldots, N\} =: \mathcal{V}$. A multi-agent Markov decision process is characterized by the tuple $(\{\mathcal{S}_i\}_{i \in \mathcal{V}}, \{\mathcal{A}_i\}_{i \in \mathcal{V}}, P, \{r_i\}_{i \in \mathcal{V}}, \gamma)$, where $\mathcal{S}_i$ is a finite state space (observations) of agent i, $\mathcal{A}_i$ is a finite action space of agent i, $r_i : \mathcal{S}_i \times \mathcal{A}_i \to \mathbb{R}$ is the reward function, $\gamma \in (0, 1)$ is the discount factor, and $P(\bar{s}, \bar{a}, \bar{s}') := \mathbb{P}[\bar{s}'|\bar{s}, \bar{a}]$ represents the unknown transition model of the joint state and action defined as $\bar{s} := (s_1, \ldots, s_N), \bar{a} := (a_1, \ldots, a_N)$, $\bar{\pi}(\bar{s}, \bar{a}) := \prod_{i=1}^N \pi_i(s_i, a_i)$, $\mathcal{S} := \prod_{i=1}^N \mathcal{S}_i$, $\mathcal{A} := \prod_{i=1}^N \mathcal{A}_i$. The stochastic policy of agent i is a mapping $\pi_i : \mathcal{S}_i \times \mathcal{A}_i \to [0, 1]$ representing the probability $\pi_i(s_i, a_i) = \mathbb{P}[a_i|s_i]$, $r_i^{\pi_i} : \mathcal{S}_i \to \mathbb{R}$ is defined as $r_i^{\pi_i}(s_i) := \mathbb{E}_{a_i \sim \pi_i(s_i)}[r_i(s_i, a_i)]$, $P^{\bar{\pi}}$ denotes the transition matrix, whose $(\bar{s}, \bar{s}')$ entry is $\mathbb{P}[\bar{s}'|\bar{s}] = \sum_{\bar{a} \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_N} \mathbb{P}[\bar{s}'|\bar{s}, \bar{a}]\bar{\pi}(\bar{s}, \bar{a})$, $d : \mathcal{S} \to \mathbb{R}$ denotes the stationary distribution of the observation $\bar{s} \in \mathcal{S}$. We assume that each agent can observe the entire joint states $\bar{s}$ and local reward r_i. We consider the following assumption.

Assumption 2: With a fixed policy $\bar{\pi}$, the Markov chain $P^{\bar{\pi}}$ is ergodic with the stationary distribution d with $d(s) > 0, s \in \mathcal{S}$.

Throughout the paper, D is defined as a diagonal matrix with diagonal entries equal to those of d. The goal is to learn an approximate value of the centralized reward $r_c = (r_1^{\pi_1} + \cdots + r_N^{\pi_N})/N$.

Problem 1: The goal of each agent i is to learn an approximate value function of the centralized reward $r_c = (r_1^{\pi_1} + \cdots + r_N^{\pi_N})/N$ without knowledge of its transition model.

Remark 1: Possible scenarios of Problem 1 are summarized as follows. Agents are located in a shared space, can observe the joint states $\bar{s}$ from the environment, but get their own local rewards. Another possibility is that each agent has its own simulation environment and tries to learn the value of their policy π_i for the reward $r_i^{\pi_i}$. However, each agent does not have access to other agents' rewards due to several reasons. For instance, there exists no centralized coordinator; thereby each agent does not know other agents'

rewards. Another possibility is that each agent/coordinator does not want to uncover their own goal or the global goal for security/privacy reasons.

It can be proved that solving Problem 1 is equivalent to solving

$$\min_{w \in C} \sum_{i=1}^N \text{MSPBE}_i(w), \quad (3)$$

where $C \subset \mathbb{R}^q$ is a compact convex set which includes the unique unconstrained global minimum of (3).

Proposition 1: Solving (3) is equivalent to finding a solution w^* to the projected Bellman equation

$$\Pi\left(\frac{1}{N}\sum_{i=1}^N r_i^{\pi_i} + \gamma P^{\bar{\pi}}\Phi w^*\right) = \Phi w^*. \quad (4)$$

Proof: See Appendix . ∎

Equivalently, the problem can be written by the consensus optimization [23]

$$\min_{w_i \in C} \sum_{i=1}^N \text{MSPBE}_i(w_i) \quad (5)$$

$$\text{subject to} \quad w_1 = w_2 = \cdots = w_N. \quad (6)$$

To make the problem more feasible, we assume that its learning parameter w_i is exchanged via a communication network represented by the undirected graph $\mathcal{G} = (\mathcal{E}, \mathcal{V})$.

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

In this section, we study a distributed GTD algorithm. To this end, we first define several vector and matrix notations to save the space: $\bar{w} := \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix}, \bar{r}^{\bar{\pi}} := \begin{bmatrix} r_1^{\pi_1} \\ \vdots \\ r_N^{\pi_N} \end{bmatrix}, \bar{P}^{\bar{\pi}} := I_N \otimes P^{\bar{\pi}}, \bar{L} := L \otimes I_{|\mathcal{S}|}, \bar{D} := I_N \otimes D, \bar{\Phi} := I_N \otimes \Phi$, and $\bar{B} := \bar{\Phi}^T \bar{D}(I - \gamma \bar{P}^{\bar{\pi}})\bar{\Phi}$. If we consider the loss function in (2), then the sum of loss functions in (5) can be compactly expressed as $\sum_{i=1}^N \text{MSPBE}_i(w_i) = \frac{1}{2}(\bar{\Phi}^T \bar{D}\bar{r}^{\bar{\pi}} - \bar{B}\bar{w})^T(\bar{\Phi}^T \bar{D}\bar{\Phi})^{-1}(\bar{\Phi}^T \bar{D}\bar{r}^{\bar{\pi}} - \bar{B}\bar{w})$. Noting that the consensus constraint (6) can be expressed as

$$\min_{\bar{w}} \frac{1}{2}(\bar{\Phi}^T \bar{D}\bar{r}^\pi - \bar{B}\bar{w})^T(\bar{\Phi}^T \bar{D}\bar{\Phi})^{-1}(\bar{\Phi}^T \bar{D}\bar{r}^\pi - \bar{B}\bar{w})$$
$$\text{subject to} \quad \bar{L}\bar{w} = 0$$

and motivated by [14]–[16], we convert it into the augmented Lagrangian problem [24, sec. 4.2]

$$\min_{\bar{w}} \frac{1}{2}(\bar{\Phi}^T \bar{D}\bar{r}^\pi - \bar{B}\bar{w})^T(\bar{\Phi}^T \bar{D}\bar{\Phi})^{-1}(\bar{\Phi}^T \bar{D}\bar{r}^\pi - \bar{B}\bar{w})$$
$$+ \bar{w}^T \bar{L}\bar{L}\bar{w} \quad (7)$$
$$\text{subject to} \quad \bar{L}\bar{w} = 0.$$

If the system is known, the above problem is an equality constrained quadratic programming problem, which can be solved by means of convex optimization methods [25]. If the model is unknown but observations can be sampled, then the problem can be still solved by using stochastic

optimization techniques. To this end, some issues need to be carefully taken into account. First, the objective function evaluation involves the double sampling problem. Second, the inverse in the objective function may lead to issues in developing algorithms. In GTD [2], this problem is resolved by a decomposition technique. In [7], it was proved that the GTD can be related to the dual problem. Following the same direction, we convert (7) into the equivalent optimization problem

$$\min_{\bar{\varepsilon},\bar{h},\bar{w}} \frac{1}{2}\bar{\varepsilon}^T(\bar{\Phi}^T\bar{D}\bar{\Phi})^{-1}\bar{\varepsilon} + \frac{1}{2}\bar{h}^T\bar{h} \tag{8}$$

$$\text{subject to } \begin{bmatrix} \bar{B} & I & 0 \\ \bar{L} & 0 & -I \\ \bar{L} & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{w} \\ \bar{\varepsilon} \\ \bar{h} \end{bmatrix} + \begin{bmatrix} -\bar{\Phi}^T\bar{D}\bar{r}^{\bar{\pi}} \\ 0 \\ 0 \end{bmatrix} = 0,$$

where $\bar{\varepsilon}, \bar{h}$ are newly introduced parameters. Its Lagrangian dual can be derived by using standard approaches [25].

Proposition 2: The Lagrangian dual problem of (8) is given by

$$\min_{\bar{\theta},\bar{v},\bar{\mu}} \psi(\bar{\theta},\bar{v},\bar{\mu}) \tag{9}$$

$$\text{subject to } \bar{B}^T\bar{\theta} - \bar{L}^T\bar{v} - \bar{L}^T\bar{\mu} = 0,$$

where $\psi(\bar{\theta},\bar{v},\bar{\mu}) := \frac{1}{2}\bar{\theta}^T(\bar{\Phi}^T\bar{D}\bar{\Phi})\bar{\theta} - \bar{\theta}^T\bar{\Phi}^T\bar{D}\bar{r}^{\bar{\pi}} + \frac{1}{2}\bar{v}^T\bar{v}$.

Proof: The dual problem can be obtained by using the standard manipulations in [25, Chap. 5]. ∎

As in [7], we again construct the Lagrangian function of (9), $L(\bar{\theta},\bar{v},\bar{\mu},\bar{w}) := \psi(\bar{\theta},\bar{v},\bar{\mu}) + [\bar{B}^T\bar{\theta} - \bar{L}^T\bar{v} - \bar{L}^T\bar{\mu}]^T\bar{w}$, where $\bar{w}$ is the Lagrangian multiplier. Since (9) satisfies the Slater's condition [25, pp. 226], the strong duality holds, i.e., $\max_{\bar{w}}\min_{\bar{\theta},\bar{v},\bar{\mu}} L(\bar{\theta},\bar{v},\bar{\mu},\bar{w}) = \min_{\bar{\theta},\bar{v},\bar{\mu}}\max_{\bar{w}} L(\bar{\theta},\bar{v},\bar{\mu},\bar{w})$, and the solutions of (9) are identical to solutions $(\bar{\theta}^*,\bar{v}^*,\bar{\mu}^*,\bar{w})$ of the saddle point problem $L(\bar{\theta}^*,\bar{v}^*,\bar{\mu}^*,\bar{w}) \leq L(\bar{\theta}^*,\bar{v}^*,\bar{\mu}^*,\bar{w}^*) \leq L(\bar{\theta},\bar{v},\bar{\mu},\bar{w}^*)$. In addition, the saddle points $(\bar{\theta}^*,\bar{v}^*,\bar{\mu}^*,\bar{w})$ satisfying the saddle point problem are identical to the KKT points $(\bar{\theta}^*,\bar{v}^*,\bar{\mu}^*,\bar{w})$ satisfying

$$0 = \nabla_{\bar{\theta}}L(\bar{\theta}^*,\bar{v}^*,\bar{\mu}^*,\bar{w}^*), \quad 0 = \nabla_{\bar{v}}L(\bar{\theta}^*,\bar{v}^*,\bar{\mu}^*,\bar{w}^*),$$
$$0 = \nabla_{\bar{\mu}}L(\bar{\theta}^*,\bar{v}^*,\bar{\mu}^*,\bar{w}^*), \quad 0 = \nabla_{\bar{w}}L(\bar{\theta}^*,\bar{v}^*,\bar{\mu}^*,\bar{w}^*). \tag{10}$$

It is known in [14], [15] that under a certain set of assumptions the continuous gradient dynamics, $\frac{d\bar{\theta}}{dt} = -\nabla_{\bar{\theta}}L(\bar{\theta},\bar{v},\bar{\mu},\bar{w})$, $\frac{d\bar{v}}{dt} = -\nabla_{\bar{v}}L(\bar{\theta},\bar{v},\bar{\mu},\bar{w})$, $\frac{d\bar{\mu}}{dt} = -\nabla_{\bar{\mu}}L(\bar{\theta},\bar{v},\bar{\mu},\bar{w})$, $\frac{d\bar{w}}{dt} = \nabla_{\bar{w}}L(\bar{\theta},\bar{v},\bar{\mu},\bar{w})$, of the Lagrangian function can solve the saddle point problem. The dynamic systems can be compactly written by the ODE $\dot{x} = -Ax - b$, where

$$A := \begin{bmatrix} \bar{\Phi}^T\bar{D}\bar{\Phi} & 0 & 0 & \bar{\Phi}^T\bar{D}(I-\gamma\bar{P}^\pi)\bar{\Phi} \\ 0 & I & 0 & -\bar{L} \\ 0 & 0 & 0 & -\bar{L} \\ -\bar{\Phi}^T(I-\gamma\bar{P}^\pi)^T\bar{D}\bar{\Phi} & \bar{L} & \bar{L} & 0 \end{bmatrix},$$

$$b := \begin{bmatrix} -\bar{\Phi}^T\bar{D}\bar{r}^{\bar{\pi}} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad x := \begin{bmatrix} \bar{\theta} \\ \bar{v} \\ \bar{\mu} \\ \bar{w} \end{bmatrix}.$$

We first establish the fact that the set of stationary points of the ODE $\dot{x} = -Ax - b$ corresponds to the set of optimal solutions of the consensus optimization problem (6).

Proposition 3: Consider the ODE $\dot{x} = -Ax - b$. The set of stationary points of the ODE is given by $\mathcal{R} := \{\bar{\theta}^*\} \times \{\bar{v}^*\} \times \mathcal{F} \times \{\bar{w}^*\}$, where $\bar{v}^* = 0$, $w^* = w_1^* = \cdots = w_N^*$, w^* is the unique solution of the projected Bellman equation (4), $\bar{\theta}^* = (\bar{\Phi}^T\bar{D}\bar{\Phi})^{-1}\bar{\Phi}^T\bar{D}(\bar{r}^{\bar{\pi}} - \bar{\Phi}\bar{w}^* + \gamma\bar{P}^{\bar{\pi}}\bar{\Phi}\bar{w}^*)$, and $\mathcal{F}$ is the set of all solutions to the linear equation for $\bar{\mu}$

$$\mathcal{F} := \{\bar{\mu} : \bar{L}\bar{\mu} = \bar{\Phi}^T(I-\gamma\bar{P}^{\bar{\pi}})^T\bar{D}\bar{\Phi}\bar{\theta}^*\}. \tag{11}$$

Proof: See Appendix . ∎

From Proposition 3 and Proposition 1, w^* is the optimal solution of (3). In addition, the stationary points in Proposition 3 are the KKT points given in (10). In addition, we can prove that partial coordinates of the set of stationary points in (3) are globally asymptotically stable.

Proposition 4: Consider the ODE $\dot{x} = -Ax - b$. Then, $(\bar{\theta},\bar{v},\bar{w}) \to (\bar{\theta}^*,\bar{v}^*,\bar{w}^*)$ as $t \to \infty$.

Proof: See Appendix . ∎

Based on those observations, one can imagine a stochastic approximation algorithm which can take benefits of the properties of the ODE $\dot{x} = -Ax - b$. In this respect, we propose the distributed GTD (DGTD) in Algorithm 1, where $C_{\bar{\theta}}, C_{\bar{v}}, C_{\bar{\mu}}, C_{\bar{w}}$ are box constraints satisfying the following assumption.

Assumption 3: The constraint sets satisfy $\bar{\theta}^* \in C_{\bar{\theta}}$, $\bar{v}^* \in C_{\bar{v}}$, $\bar{w}^* \in C_{\bar{w}}$, and $C_{\bar{\mu}} \cap \mathcal{F} \neq \emptyset$.

The constraints are added to guarantee the stability and convergence of the algorithm. According to [11, Prop. 4], [20, Appendix E], the corresponding ODE is

$$\dot{x} = \Gamma_{T_C(x)}(-Ax - b), \tag{12}$$

where $\Gamma_{T_C(x)}$ is defined as the projection of x onto the tangent cone $T_C(x)$ [24, pp. 343] of $C := C_{\bar{\theta}} \times C_{\bar{v}} \times C_{\bar{\mu}} \times C_{\bar{w}}$ at x. Due to the additional constraints, the set of stationary points of (12) is a larger set, which includes those of $\dot{x} = -Ax - b$ as a subset. The following results can be directly proved using the definitions of tangent and normal cones [24, pp. 343].

Proposition 5: The set of stationary points of (12) is $\mathcal{P} := \{x \in C : \Gamma_{T_C(x)}(-Ax - b) = 0\} = \{x \in C : -Ax - b \in N_C(x)\}$.

We first establish the convergence of Algorithm 1 to the stationary points of (12) under the standard diminishing step size rule [23]

$$\alpha_k > 0, \forall k \geq 0, \quad \sum_{k=0}^{\infty}\alpha_k = \infty, \quad \sum_{k=0}^{\infty}\alpha_k^2 < \infty. \tag{13}$$

Proposition 6 (Convergence of DGTD): Define

$$\bar{\theta}_k := \begin{bmatrix} \theta_{1,k} \\ \vdots \\ \theta_{N,k} \end{bmatrix}, \bar{v}_k := \begin{bmatrix} v_{1,k} \\ \vdots \\ v_{N,k} \end{bmatrix}, \bar{\mu}_k := \begin{bmatrix} \mu_{1,k} \\ \vdots \\ \mu_{N,k} \end{bmatrix}, \bar{w}_k := \begin{bmatrix} w_{1,k} \\ \vdots \\ w_{N,k} \end{bmatrix},$$

and $\bar{x}_k := \begin{bmatrix} \bar{\theta}_k^T & \bar{v}_k^T & \bar{\mu}_k^T & \bar{w}_k^T \end{bmatrix}^T$ with iterations in Algorithm 1. With the step size rule (13), $\text{dist}(\bar{x}_k, \mathcal{P}) \to 0$ as

$k \to \infty$, where $\mathcal{P} := \{x \in C : -Ax - b \in N_C(x)\}$ with probability one.

Proof: See Appendix . ∎

Algorithm 1 Distributed GTD algorithm (DGTD)

1: Initialize $\{\theta_0^{(i)}\}_{i \in \mathcal{V}}$ and set $k = 0$.
2: **repeat**
3:    $k \leftarrow k + 1$
4:    **for** agent $i \in \{1, \dots, N\}$ **do**
5:       Sample (s, a, s') with $s \sim d_i(s), a \sim \pi_i(a|s), s' \sim p_i(s'|s,a)$ and update parameters according to

$$\theta_{i,k+1/2} = \theta_{i,k} - \alpha_k[\phi\phi^T\theta_{i,k} + \phi\phi^T w_{i,k} \\ - \gamma\phi(\phi')^T w_{i,k} - \phi r_i^{\pi_i}]$$

$$v_{i,k+1/2} = v_{i,k} - \alpha_k \left[v_{i,k} - \left(|\mathcal{N}_i| w_{i,k} - \sum_{j \in \mathcal{N}_i} w_{j,k} \right) \right],$$

$$\mu_{i,k+1/2} = \mu_{i,k} + \alpha_k \left(|\mathcal{N}_i| w_{i,k} - \sum_{j \in \mathcal{N}_i} w_{j,k} \right),$$

$$w_{i,k+1/2} = w_{i,k} - \alpha_k \left(|\mathcal{N}_i| v_{i,k} - \sum_{j \in \mathcal{N}_i} v_{j,k} \right) \\ - \alpha_k \left(|\mathcal{N}_i| \mu_{i,k} - \sum_{j \in \mathcal{N}_i} \mu_{j,k} \right) \\ + \alpha_k(\phi\phi^T\theta_{i,k} - \gamma\phi'\phi^T\theta_{i,k}),$$

where $\mathcal{N}_i$ is the neighborhood of node i on the graph $\mathcal{G}$, $\phi := \phi(s), \phi' := \phi(s'), r_i^{\pi_i} := r_i^{\pi_i}(s)$.
6:       Project the iterates $\theta_{i,k+1} = \Gamma_{C_{\bar{\theta}}}[\theta_{i,k+1/2}]$, $v_{i,k+1} = \Gamma_{C_{\bar{v}}}[v_{i,k+1/2}]$, $\mu_{i,k+1} = \Gamma_{C_{\bar{\mu}}}[\mu_{i,k+1/2}]$, $w_{i,k+1} = \Gamma_{C_{\bar{w}}}[w_{i,k+1/2}]$.
7:    **end for**
8: **until** a certain stopping criterion is satisfied.

Although Proposition 6 states that the iterations of Proposition 6 converge to a stationary point of the projected ODE (12), it does not guarantee that they converge to the set of stationary points of the ODE without the projection in Proposition 3. In practice, however, we expect that they may often converge to the set in Proposition 3, if the constraint sets are sufficiently large. On the other hand, if we follow the analysis of the stochastic primal-dual algorithm in [17], we can prove that under certain conditions, the iterations of Algorithm 1 converge to the the stationary points in Proposition 3. The proof is similar to those in [17], and we defer its full analysis to an extended version of this paper.

VI. EXAMPLES

Example 1: Consider a stock market whose price process is approximated by a Markov chain with 100 states $\mathcal{S} := \{\$10, \$20, \dots, \$1000\}$. If an agent buys a stock, then it loses $s \in \mathcal{S}$, and if sells, then it earns $s \in \mathcal{S}$. Define the trading policy $\pi(s; a, b) =$

$\begin{cases} \text{If } a \le s \le b, \text{ then buy a stock} \\ \text{Otherwise, sell a stock} \end{cases}$. There are five trading agents $\mathcal{V} = \{1, 2, \dots, 5\}$ with different private policies $\pi_1(s) = \pi(s; \$10, \$30), \pi_2(s) = \pi(s; \$10, \$40), \pi_3(s) = \pi(s; \$10, \$50), \pi_4(s) = \pi(s; \$10, \$60)$, and $\pi_5(s) = \pi(s; \$10, \$70)$. To determine an investment strategy, each agent is interested in estimating an average of long term discounted profits of all agents as well as its own. When the current state is $s \in \mathcal{S}$, the reward of each agent is $r_i^{\pi_i} = -s$ if $\pi_i = \text{buy}$, and $r_i^{\pi_i} = s$ if $\pi_i = \text{sell}$. For this example, we used Gaussian radial basis functions as features of the linear function approximation with 11 parameters, i.e., $w_i \in \mathbb{R}^{11}$, we considered the discount factor $\gamma = 0.5$, and we used a randomly generated Markov chain for the stock price process model. Using the single agent GTD [2], each agent computed the approximate value functions $J_{w_i^*} = \Phi w_i^*, i \in \mathcal{V}$. The expected profits with the uniform initial state distribution are $\mathbb{E}_{s \sim U(\mathcal{S})}[J_{w_1^*}(s)] = 164.3, \mathbb{E}_{s \sim U(\mathcal{S})}[J_{w_2^*}(s)] = 55.6$, $\mathbb{E}_{s \sim U(\mathcal{S})}[J_{w_3^*}(s)] = -107.5, \mathbb{E}_{s \sim U(\mathcal{S})}[J_{w_4^*}(s)] = -240.4$, and $\mathbb{E}_{s \sim U(\mathcal{S})}[J_{w_5^*}(s)] = -284.4639$, where $U(\mathcal{S})$ is the uniform distribution over the state $\mathcal{S}$. Using the single agent GTD again, the value function $J_{w_c^*} = \Phi w_c^*$ (global value function) corresponding to the central reward $r_c = (r_1 + r_2 + r_3 + r_4 + r_5)/5$ was computed, and the expected profit is $\mathbb{E}_{s \sim U(\mathcal{S})}[J_{w_c^*}(s)] = -82.5$. Since each agent wants to keep its profit secure, agent i can compute its own value function $J_{w_i^*}$ only. However, there are associated agents, which are able to exchange their parameters. The associate relations are depicted in Figure 1. Under this assumption, Algorithm 1
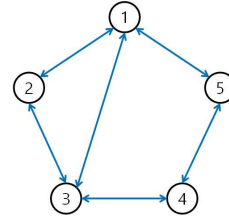

Fig. 1.   Graph describing the associate relations among five trading agents.

was applied with the step size rule $\alpha_k = 10/(k + 1000)$ and without the projections, and each agent computed the global value function estimations $J_{\tilde{w}_i^*} = \Phi \tilde{w}_i^*, i \in \mathcal{V}$. The result of 50000 iterations with a single simulation trajectory is illustrated in Figure 2. Distinguished by different colors, the consensus of 11 parameters of $\tilde{w}_i$ for five agents is shown. The same color is used for each coordinate of all the agents. The expected profits with uniform initial state distribution are $\mathbb{E}_{s \sim U(\mathcal{S})}[J_{\tilde{w}_1^*}(s)] = -83.2, \mathbb{E}_{s \sim U(\mathcal{S})}[J_{\tilde{w}_2^*}(s)] = -85.0$, $\mathbb{E}_{s \sim U(\mathcal{S})}[J_{\tilde{w}_3^*}(s)] = -81.2, \mathbb{E}_{s \sim U(\mathcal{S})}[J_{\tilde{w}_4^*}(s)] = -83.5$, and $\mathbb{E}_{s \sim U(\mathcal{S})}[J_{\tilde{w}_5^*}(s)] = -79.5$. This result demonstrates that each agent successfully estimated the global value function $J_{w_c^*} = \Phi w_c^*$. Even though we are not aware of previous methods on the same topic, we combined the standard consensus method with GTD and compared the result with Figure 2. We observed that the convergence of Figure 2 is usually faster than the standard consensus approach.
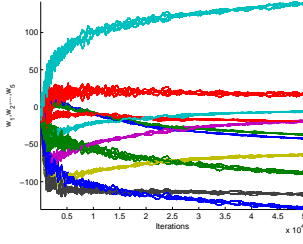
Fig. 2. Example 1: Convergence of 11 parameters (distinguished by different colors) of five agents (not distinguished by colors).

Example 2: Consider a multi-agent Markov decision process with 5 states $\mathcal{S} := \{1, 2, \ldots, 5\}$ and

$$P^{\bar{\pi}} = \begin{bmatrix} 0.2362 & 0.0895 & 0.3536 & 0.1099 & 0.2107 \\ 0.1821 & 0.2719 & 0.1553 & 0.1217 & 0.2689 \\ 0.1999 & 0.0279 & 0.2870 & 0.1628 & 0.3224 \\ 0.1149 & 0.1723 & 0.2726 & 0.3747 & 0.0656 \\ 0.2921 & 0.1719 & 0.0907 & 0.1836 & 0.2618 \end{bmatrix},$$

where the policy $\bar{\pi}$ is not specified. In addition, consider 20 agents $\mathcal{V} = \{1, 2, \ldots, 20\}$. They exchange their parameters through a network described by a star graph, where the agent 1 corresponds to the center node. The reward of each agent i is $r_i^{\pi_i} \equiv i$ for all $i \in \mathcal{V}$. As before, Gaussian radial basis functions are considered as features of the linear function approximation with 3 parameters, i.e., $w_i \in \mathbb{R}^3, i \in \mathcal{V}$. We run Algorithm 1 with the discount factor $\gamma = 0.5$ and the step size rule $\alpha_k = 2/(k + 1000)$. The result
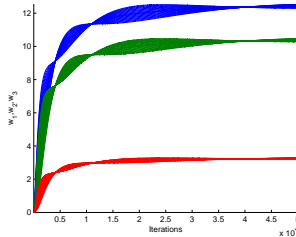

Fig. 3. Example 2: Convergence of 3 parameters (distinguished by different colors) of 20 agents.

with 50000 iterations is illustrated in Figure 3, where the consensus of 3 parameters in $\tilde{w}_i$ is shown. The simulation result demonstrates the validity of the proposed algorithm.

VII. CONCLUSION

In this paper, we studied a new class of distributed GTD algorithm based on primal-dual iterations, as compared to [2]. The convergence was proved using ODE-based methods. Simulation results demonstrated the applicability of the proposed algorithm. Possible related future research directions include extensions to actor-critic algorithms, off-policy learning cases, and randomly changing networks.

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 *Advances in neural information processing systems*, 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 *Proceedings of the 26th Annual International Conference on Machine Learning*, 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 Transactions on Signal Processing*, vol. 61, no. 7, pp. 1848–1862, 2013.

[4] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific Belmont, MA, 1996.

[5] P. Pennesi and I. C. Paschalidis, "A distributed actor-critic algorithm and applications to mobile sensor network coordination problems," *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 492–497, 2010.

[6] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, "Fully decentralized multi-agent reinforcement learning with networked agents," *arXiv preprint arXiv:1802.08757*, 2018.

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

[8] M. S. Stanković and S. S. Stanković, "Multi-agent temporal-difference learning with linear function approximation: weak convergence under time-varying network topologies," in *American Control Conference (ACC)*, 2016, pp. 167–172.

[9] A. Mathkar and V. S. Borkar, "Distributed reinforcement learning via gossip," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1465–1470, 2017.

[10] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of optimization theory and applications*, vol. 147, no. 3, pp. 516–545, 2010.

[11] P. Bianchi and J. Jakubowicz, "Convergence of a multi-agent projected stochastic gradient algorithm for non-convex optimization," *IEEE Transactions on Automatic Control*, vol. 58, no. 2, pp. 391–405, 2013.

[12] L. Baird, "Residual algorithms: Reinforcement learning with function approximation," in *Machine Learning Proceedings 1995*, 1995, pp. 30–37.

[13] R. Tutunov, H. B. Ammar, and A. Jadbabaie, "An exact distributed newton method for reinforcement learning," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, 2016, pp. 1003–1008.

[14] J. Wang and N. Elia, "Control approach to distributed optimization," in *48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2010, pp. 557–561.

[15] ——, "A control perspective for centralized and distributed convex optimization," in *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011, pp. 3800–3805.

[16] B. Gharesifard and J. Cortés, "Distributed continuous-time convex optimization on weight-balanced digraphs," *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, 2014.

[17] Y. Chen and M. Wang, "Stochastic primal-dual methods and sample complexity of reinforcement learning," *arXiv preprint arXiv:1612.02516*, 2016.

[18] 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.

[19] V. S. Borkar and S. P. Meyn, "The ODE method for convergence of stochastic approximation and reinforcement learning," *SIAM Journal on Control and Optimization*, vol. 38, no. 2, pp. 447–469, 2000.

[20] S. Bhatnagar, H. Prasad, and L. Prashanth, *Stochastic recursive algorithms for optimization: simultaneous perturbation methods*. Springer, 2012, vol. 434.

[21] H. Kushner and G. G. Yin, *Stochastic approximation and recursive algorithms and applications*. Springer Science & Business Media, 2003, vol. 35.

[22] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, 1998.

[23] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[24] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[25] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

Since (3) is strongly convex, its unconstrained global minimum is unique, and it satisfies

$$\nabla_w \sum_{i=1}^{N} \text{MSPBE}_i(w) = -(\Phi^T D(I - \gamma P^{\bar{\pi}})\Phi)^T$$

$$\times (\Phi^T D\Phi)^{-1}\Phi^T D \sum_{i=1}^{N}(r_i^{\pi_i} - (I - \gamma P^{\bar{\pi}})\Phi w) = 0.$$

Since $\Phi^T D(I - \gamma P^{\bar{\pi}})\Phi$ is nonsingular [4, pp. 300], this implies

$$(\Phi^T D\Phi)^{-1}\Phi^T D \sum_{i=1}^{N}(r_i^{\pi_i} - (I - \gamma P^{\bar{\pi}})\Phi w) = 0.$$

Pre-multiplying the equation by Φ yields the desired result.

The proof is based on the analysis of the stochastic recursion

$$x_{k+1} = \Gamma_C(f(x_k) + \varepsilon_k). \tag{14}$$

Define the σ-field $\mathcal{F}_k := \sigma(\varepsilon_0, \ldots, \varepsilon_{k-1}, x_0, \ldots, x_k, \alpha_0, \ldots, \alpha_k)$. According to [11, Prop. 4], [20, Appendix E], the corresponding ODE can be expressed as

$$\dot{x} = \Gamma_{T_C(x)}[f(x)].$$

We consider assumptions listed below.

Assumption 4:
1) The function $f : \mathbb{R}^N \to \mathbb{R}^N$ is continuous.
2) The step sizes satisfy

$$\alpha_k > 0, \forall k \geq 0, \quad \sum_{k=0}^{\infty} \alpha_k = \infty, \quad \alpha_k \to 0 \text{ as } k \to \infty.$$

3) The ODE $\dot{x} = \Gamma_{T_C(x)}[f(x)]$ has a compact subset $\mathcal{P}$ of $\mathbb{R}^N$ as its set of asymptotically stable equilibrium points.

Let $t(k), k \geq 0$ be a sequence of positive real numbers defined according to $t(0) = 0$ and $t(k) = \sum_{j=0}^{k-1} \alpha_j, k \geq 1$. By the step size in Assumption 4, $t(k) \to \infty$ as $k \to \infty$. Define $m(t) := \max\{k|t(k) \leq t\}$. Thus, $m(t) \to \infty$ as $t \to \infty$.

Assumption 5: There exists T such that for all $\delta > 0$

$$\lim_{k \to \infty} \mathbb{P}\left(\sup_{j \geq k} \max_{0 \leq t \leq T} \left\|\sum_{i=m(jT)}^{m(jT+t)-1} \alpha_i \varepsilon_i\right\| \geq \delta\right) = 0. \tag{15}$$

Lemma 1 (Kushner and Clark Theorem [21, Appendix E]): Under Assumption 4 and Assumption 5, for any initial $x(0) \in \mathbb{R}^N$, $x(k) \to \mathcal{P}$ as $k \to \infty$ with probability one.

Proof of Proposition 6: We will check Assumption 4 and Assumption 5 and use Lemma 1 to complete the proof. The ODE in (12) is a projection of an affine map $f(x) = -Ax - b$; therefore, it is obviously continuous. The step size assumption is satisfied by the hypothesis. In addition, the set of stationary points $\mathcal{P} = \{x \in C : \Gamma_{T_C}(-Ax - b) = 0\}$ is compact. This is because $\mathcal{P}$ is expressed as $\mathcal{P} = \{x \in C : -Ax - b \in N_C(x)\}$, where $N_C(x)$ is a convex closed cone,

and its pre-image of an affine map is also closed. Therefore, $\mathcal{P}$ is closed. $\mathcal{P} \subseteq C$, because $\Gamma_{T_C}(-Ax - b) = 0$ only when $x \in C$. Since C is compact and $\mathcal{P}$ is its closed subset, $\mathcal{P}$ is also compact. $\mathcal{P}$ can be also proved to be globally asymptotically stable following analysis given in [11]. For completeness of the presentation, the brief proof is given in Appendix . Next, we will prove Assumption 5. Proposition 6 can be expressed as (14) with $\varepsilon_k = (-\tilde{A}x_k - \tilde{b}) - (-Ax_k - b)$, where $-\tilde{A}x_k - \tilde{b}$ is a stochastic approximate of $-Ax_k - b$ such that $\mathbb{E}[-\tilde{A}x_k - \tilde{b}|\mathcal{F}_k] = -Ax_k - b$. Therefore, $\mathbb{E}[\varepsilon_k|\mathcal{F}_k] = 0$. Define $M_k := \sum_{i=0}^{k-1} \alpha_i \varepsilon_i$. Then, since $\mathbb{E}[M_{k+1}|\mathcal{F}_k] = M_k$, $(M_k)_{k=0}^{\infty}$ is a Martingale sequence. We will prove the sufficient condition for (15).

$$\lim_{k \to \infty} \mathbb{P}\left(\max_{0 \leq t \leq T} \left|\sum_{i=m(kT)}^{m(kT+t)-1} \alpha_i \varepsilon_i\right| \geq \delta\right) = 0.$$

Since $\max_{0 \leq t \leq T} \left\|\sum_{i=m(kT)}^{m(kT+t)-1} \alpha_i \varepsilon_i\right\| \leq \max_{m(kT) \leq m \leq m(kT+T)-1} \left\|\sum_{i=m(kT)}^{t} \alpha_i \varepsilon_i\right\|$, we will consider a more conservative sufficient condition:

$$\lim_{k \to \infty} \mathbb{P}\left(\max_{0 \leq t \leq m(kT+T)-m(kT)-1} \|H_t\| \geq \delta\right) = 0, \tag{16}$$

where $(H_t)_{t=0}^{\infty}$ with $H_t := \sum_{i=m(kT)}^{m(kT)+t} \alpha_i \varepsilon_i$ is a Martingale sequence. Then, by using the Martingale inequality, we have

$$\mathbb{P}\left(\max_{0 \leq t \leq m(kT+T)-m(kT)-1} |H_t| \geq \delta\right)$$

$$\leq \frac{\mathbb{E}\left[\left|\sum_{i=m(kT)}^{m(kT+T)-1} \alpha_i \varepsilon_i\right|^2\right]}{\delta^2} \leq \frac{C^2 \sum_{i=m(kT)}^{\infty} \alpha_i^2}{\delta^2},$$

where we used $\|\varepsilon_i\|^2 \leq C^2$. By the step size rule in (13), $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$ implies that the right-hand side converges to zero as $k \to \infty$. Therefore, we prove (16) and (15). By Lemma 1, we prove that x_k globally converges to the stationary point $\mathcal{H}$ with probability one.

In this section, we will prove the following claim.

Proposition 7: Consider the ODE $\dot{x} = \Gamma_{T_C(x)}(-Ax - b)$ in (12). The set of stationary points $\mathcal{H} = \{x \in C : \Gamma_{T_C}(-Ax - b) = 0\}$ is globally asymptotically stable.

The proof is given in [11], and we provide a brief sketch of the proof.

Proof: If we define the function $V(x) := x^T(Ax + b)$, then the ODE can be represented by $\dot{x} = \Gamma_{T_C(x)}(-\nabla_x V(x))$. Let $V(x)$ be a candidate Lyapunov function for $\mathcal{P}$. Then, its time derivative is expressed as $\dot{V}(x) = \nabla_x V(x)^T \Gamma_{T_C(x)}(-\nabla_x V(x))$. Since $\nabla_x V(x) = \Gamma_{T_C(x)}(\nabla_x V(x)) + \Gamma_{N_C(x)}(\nabla_x V(x))$, and the tangent cone and normal cone are orthogonal, we arrive at $\dot{V}(x) = -\left\|\Gamma_{T_C(x)}(-\nabla_x V(x))\right\|^2$. Therefore, $\mathcal{P} = \{x \in C : \Gamma_{T_C(x)}(-\nabla_x V(x)) = 0\} = \{x \in C : -\nabla_x V(x) \in N_C(x)\}$ is globally asymptotically stable. Since $-\nabla_x V(x) = -Ax - b$, the proof is completed. ∎

We first consider the stationary points of (12) without the projection. They are obtained by solving the linear equation:

$$0 = (\bar{\Phi}^T \bar{D}\bar{\Phi})\bar{\theta} - \bar{\Phi}^T \bar{D}\bar{r}^{\bar{\pi}} + \bar{\Phi}^T \bar{D}(I - \gamma \bar{P}^{\bar{\pi}})\bar{\Phi}\bar{w}, \tag{17}$$

$$0 = \bar{v} - \bar{L}\bar{w}, \tag{18}$$

$$0 = \bar{L}\bar{w}, \tag{19}$$

$$0 = \bar{L}\bar{v} + \bar{L}\bar{\mu} - \bar{\Phi}^T(I - \gamma\bar{P}^{\bar{\pi}})^T\bar{D}\bar{\Phi}\bar{\theta}. \tag{20}$$

Since $\mathcal{G}$ is connected by Assumption 1, the dimension of the null space of L is one. Therefore, $\mathrm{span}(\mathbf{1})$ is the null space. Therefore, (19) implies the consensus $w^* = w_1^* = \cdots = w_N^*$, and plugging (19) into (18) yields $\bar{v}^* = 0$. With $\bar{v}^* = 0$, (20) is simplified to

$$\bar{L}\bar{\mu}^* = \bar{\Phi}^T(I - \gamma\bar{P}^{\bar{\pi}})^T\bar{D}\bar{\Phi}\bar{\theta}^*. \tag{21}$$

In addition, from (17), the stationary point for $\bar{\theta}$ satisfies

$$\bar{\theta}^* = (\bar{\Phi}^T\bar{D}\bar{\Phi})^{-1}\bar{\Phi}^T\bar{D}(\bar{r}^{\bar{\pi}} - \bar{\Phi}\bar{w}^* + \gamma\bar{P}^{\bar{\pi}}\bar{\Phi}\bar{w}^*). \tag{22}$$

Plugging the above equation into (21) yields

$$\begin{aligned}
\bar{L}\bar{\mu} &= \bar{\Phi}^T(I - \gamma\bar{P}^{\bar{\pi}})^T\bar{D}\bar{\Phi}\bar{\theta} \\
&= \bar{\Phi}^T(I - \gamma\bar{P}^{\bar{\pi}})^T\bar{D}\bar{\Phi}(\bar{\Phi}^T\bar{D}\bar{\Phi})^{-1} \\
&\quad \times \bar{\Phi}^T\bar{D}(\bar{r}^{\bar{\pi}} - \bar{\Phi}\bar{w} + \gamma\bar{P}^{\bar{\pi}}\bar{\Phi}\bar{w}).
\end{aligned} \tag{23}$$

Multiplying (23) by $(\mathbf{1} \otimes I)^T$ on the left results in

$$\begin{aligned}
\sum_{i=1}^N ([\Phi - \gamma\bar{P}^{\bar{\pi}}\Phi]^T D\Phi(\Phi^T D\Phi)^{-1} \\
\Phi^T D[-r_i^{\pi_i} + \Phi w^* - \gamma\bar{P}^{\bar{\pi}}\Phi w^*]) = 0,
\end{aligned}$$

which is equivalent to $\sum_{i=1}^N \nabla_w\mathrm{MSPBE}_i(w^*) = 0$. Since the loss functions are strict convex quadratic functions, w^* is the unique global minimum of $\sum_{i=1}^N \mathrm{MSPBE}_i(w)$. From (22), $\bar{\theta}^*$ is also uniquely determined. In particular, multiplying (17) by $(\mathbf{1} \otimes I)^T$ from the left, the unique stationary point for $\bar{w}^*$ is expressed as $\bar{w}^* = \mathbf{1} \otimes w^*$ with

$$\begin{aligned}
w^* = \frac{1}{N}(\Phi^T D(I - \gamma\bar{P}^{\bar{\pi}})\Phi)^{-1}\Phi^T D\left(\sum_{i=1}^N r_i^{\pi_i} - N\Pi\right) \\
\times \left(-\frac{1}{N}\sum_{i=1}^N r_i^{\pi_i} + \Phi w_i^* - \gamma\bar{P}^{\bar{\pi}}\Phi w_i^*\right)\Bigg),
\end{aligned}$$

From (23), stationary $\bar{\mu}^*$ is any solution of the linear equation (23).

Define

$$x := \begin{bmatrix} \bar{\theta} \\ \bar{v} \end{bmatrix}, \quad y := \bar{\mu}, \quad z := \bar{w},$$

$$f(x,y) := \frac{1}{2}\bar{\theta}^T(\bar{\Phi}^T\bar{D}\bar{\Phi})\bar{\theta} - \bar{\theta}^T\bar{\Phi}^T\bar{D}\bar{r}^{\bar{\pi}} + \frac{1}{2}\bar{v}^T\bar{v},$$

$$A := \begin{bmatrix} \bar{B}^T & -\bar{L}^T & -\bar{L}^T \end{bmatrix}.$$

Then, the dual problem can be compactly expressed as $\min_{x,y} f(x,y)$ s.t. $A\begin{bmatrix} x \\ y \end{bmatrix} = 0$, and the ODE (12) can be written by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = -\begin{bmatrix} \nabla_x f(x,y) \\ \nabla_y f(x,y) \end{bmatrix} - A^T z, \dot{z} = A\begin{bmatrix} x \\ y \end{bmatrix}.$$

The asymptotic stability analysis is based on the Lyapunov method in the proof of [15, Thm. 2.1]. However, the proof in [15, Thm. 2.1] cannot be directly applied because $f(x,y)$

is not strictly convex in y, which requires an additional analysis. Let (x^*, y^*, z^*) be the stationary point given in Section , and define $(\tilde{x}, \tilde{y}, \tilde{z}) := (x - x^*, y - y^*, z - z^*)$. The corresponding ODE is

$$\begin{aligned}
\frac{d}{dt}\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} &= -\begin{bmatrix} \nabla_x f(x,y) \\ \nabla_y f(x,y) \end{bmatrix} + \begin{bmatrix} \nabla_x f(x^*,y^*) \\ \nabla_y f(x^*,y^*) \end{bmatrix} - A^T\tilde{z}, \\
\frac{d}{dt}\tilde{z} &= A\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix}.
\end{aligned} \tag{24}$$

Consider the quadratic candidate Lyapunov function

$$V(\tilde{x}, \tilde{y}, \tilde{z}) := \frac{1}{2}\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix}^T\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} + \frac{1}{2}\tilde{z}^T\tilde{z},$$

whose time derivative is

$$\frac{d}{dt}V(\tilde{x}, \tilde{y}, \tilde{z}) = -\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix}^T\begin{bmatrix} \nabla_x f(x,y) \\ \nabla_y f(x,y) \end{bmatrix} + \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix}^T\begin{bmatrix} \nabla_x f(x^*,y^*) \\ \nabla_y f(x^*,y^*) \end{bmatrix}.$$

Since f is convex, the gradient satisfies the global under-estimator property

$$\begin{aligned}
f(x',y') \geq & f(x,y) \\
& + \begin{bmatrix} \nabla_x f(x,y) \\ \nabla_y f(x,y) \end{bmatrix}^T\left(\begin{bmatrix} x' \\ y' \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix}\right), \forall\begin{bmatrix} x' \\ y' \end{bmatrix}, \begin{bmatrix} x \\ y \end{bmatrix}.
\end{aligned}$$

Since $f(x,y)$ only depends on x and is strictly convex in x, a strict inequality holds if and only if $x' = x$. Therefore, the following holds:

$$f(x,y) > f(x^*,y^*) + \begin{bmatrix} \nabla_x f(x^*,y^*) \\ \nabla_y f(x^*,y^*) \end{bmatrix}^T\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix},$$

$$f(x^*,y^*) > f(x,y) - \begin{bmatrix} \nabla_x f(x,y) \\ \nabla_y f(x,y) \end{bmatrix}^T\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix},$$

if and only if $\tilde{x} \neq 0$. Adding both sides of the inequalities leads to $\frac{d}{dt}V(\tilde{x}, \tilde{y}, \tilde{z}) < 0$ if and only if $\tilde{x} \neq 0$. Therefore, one concludes that $\bar{\theta} \to \bar{\theta}^*$ and $\bar{v} \to \bar{v}^* = 0$. Since $\frac{d}{dt}V(\tilde{x}, \tilde{y}, \tilde{z}) = 0, \forall(\tilde{x}, \tilde{y}, \tilde{z}) \in \mathcal{G} := \{\tilde{x}, \tilde{y}, \tilde{z} : \tilde{x} = 0\}$, we invoke LaSalle invariant principle to prove that all bounded trajectories converge to the largest invariant set $\mathcal{M}$ such that $\mathcal{M} \subseteq \mathcal{G}$. Now, we focus on the trajectories $(\tilde{x}(t), \tilde{y}(t), \tilde{z}(t)) \in \mathcal{G}$, where the ODE (24) becomes $\frac{d}{dt}\begin{bmatrix} 0 \\ \tilde{y} \end{bmatrix} = -A^T\tilde{z}, \frac{d}{dt}\tilde{z} = A\begin{bmatrix} 0 \\ \tilde{y} \end{bmatrix}$. In particular, we have $0 = \bar{B}(\bar{w} - \bar{w}^*), 0 = \bar{L}(\bar{w} - \bar{w}^*), \frac{d}{dt}\bar{\mu} = \bar{L}(\bar{w} - \bar{w}^*), \frac{d}{dt}\bar{w} = -\bar{L}^T(\bar{\mu} - \bar{\mu}^*)$. Since $\bar{B} := \bar{\Phi}^T\bar{D}(I - \gamma\bar{P}^{\bar{\pi}})\bar{\Phi}$ is nonsingular, we have $\bar{w} = \bar{w}^*, \frac{d}{dt}\bar{\mu} = 0$, and $0 = -\bar{L}^T(\bar{\mu} - \bar{\mu}^*)$. This implies

$$\mathcal{M} = \left\{\begin{bmatrix} \bar{\theta} \\ \bar{v} \\ \bar{\mu} \\ \bar{w} \end{bmatrix} : \begin{bmatrix} \bar{\theta} \\ \bar{v} \\ \bar{\mu} \\ \bar{w} \end{bmatrix} = \begin{bmatrix} \bar{\theta}^* \\ \bar{v}^* \\ \bar{\mu}^* \\ \bar{w}^* \end{bmatrix}, \bar{\mu}^* \in \mathcal{F}\right\},$$

where $\mathcal{F}$ is defined in (11). Therefore, all bounded solutions converge to $\mathcal{M}$. However, the boundedness of the solutions is not guaranteed. By Lyapunov inequality $\frac{d}{dt}V(\tilde{x}, \tilde{y}, \tilde{z}) \leq 0, \forall(\tilde{x}, \tilde{y}, \tilde{z})$, trajectory $(\bar{\theta}, \bar{v}, \bar{w})$ is guaranteed to be bounded, while $\bar{\mu}$ may not because the set of stationary points $\mathcal{F}$

of $\bar{\mu}$ defined in (11) is an unbounded affine space. However, LaSalle's invariance principle can be applied to those bounded partial coordinates. Therefore, we have $(\bar{\theta}, \bar{v}, \bar{w}) \rightarrow (\bar{\theta}^*, \bar{v}^*, \bar{w}^*)$ as $t \rightarrow \infty$ globally. This completes the proof.