Voting-Based Multi-Agent Reinforcement Learning for Intelligent IoT

Yue Xu, *Student Member, IEEE,* Zengde Deng, Mengdi Wang, Wenjun Xu, *Senior Member, IEEE,* Anthony Man-Cho So, *Senior Member, IEEE,* and Shuguang Cui, *Fellow, IEEE,*

***Abstract*—The recent success of single-agent reinforcement learning (RL) in Internet of things (IoT) systems motivates the study of multi-agent reinforcement learning (MARL), which is more challenging but more useful in large-scale IoT. In this paper, we consider a voting-based MARL problem, in which the agents vote to make group decisions and the goal is to maximize the globally averaged returns. To this end, we formulate the MARL problem based on the linear programming form of the policy optimization problem and propose a primal-dual algorithm to obtain the optimal solution. We also propose a voting mechanism through which the distributed learning achieves the same sublinear convergence rate as centralized learning. In other words, the distributed decision making does not slow down the process of achieving global consensus on optimality. Lastly, we verify the convergence of our proposed algorithm with numerical simulations and conduct case studies in practical multi-agent IoT systems.**

***Index Terms*—Multi-agent reinforcement learning, voting mechanism, primal-dual algorithm**

I. Introduction

Reinforcement learning (RL) aims at maximizing a cumulative reward by selecting a sequence of optimal actions to interact with a stochastic *unknown* environment, where the dynamics is usually modeled as a Markov decision process (MDP) [1]. Recently, single-agent RL has been successfully applied to contribute adaptive and autonomous intelligence in many Internet of things (IoT) applications, including smart cellular networks [2–4], smart vehicle networks [5–7], and smart unmanned aerial vehicles (UAV) networks [8–10]. Despite these successes, many recent studies envision that the IoT entities, e.g., smartphones, sensors, and UAVs, will become more decentralized, ad-hoc, and autonomous in nature [11], [12]. This encourages the extension from single-agent RL to multi-agent RL (MARL) to study the smart collaboration among local entities in order to deliver a superior collective intelligence, instead of simply treating them as independent learners. However, MARL is more challenging since each agent interacts with not only the environment but also the other agents.

Although a number of collaborative learning models based on MARL have been recently proposed [13–21], they usually impose a discount factor $\gamma \in (0, 1)$ on the future rewards to render the problem more tractable, e.g., bounding the cumulative reward [22–24]. However, many optimization tasks in the IoT systems, e.g., resource allocation and admission control, are long-run or non-terminating tasks. Existing studies reveal that the RL methods based on discounted MDP may yield a poor performance in the continuing tasks and become computationally challenging when the discount factor is close to one [1], [25–27]. This necessitates the development of MARL models based on the undiscounted average-reward MDP (AMDP) to tackle the continuing optimization tasks in IoT systems. Moreover, existing MARL models usually exhibit a performance degradation compared with their centralized versions [21], [28] and only provide asymptotic convergence to an optimal point [21], [28] or simply give empirical evaluations without theoretical guarantees [16–20]. In contrast, in this paper, we give a sublinear convergence rate and theoretically prove that our proposed MARL model achieves the same convergence rate as centralized learning, which makes it a decent learning paradigm for distributed IoT systems.

Meanwhile, it is critical to specify a proper collaboration protocol in order to promote safe and efficient cooperations in MARL systems. Many existing MARL models are built upon the centralized learning with decentralized execution framework where the agents perform iterative parameter consensus with a centralized server [14], [18], [29], [30]. Moreover, the centralized server is assumed to have access to the behavioral policy or value functions of all distributed agents for model training. However, in many IoT applications (e.g., location services), the privacy-sensitive data (e.g., policy or value functions) should not be logged onto a centralized center due to privacy and security concerns. On the other hand, recent works also propose a number of decentralized solutions which coordinate the agents through iterative parameter consensus among neighboring agents [21–24]. However, this may give rise to massive communication overhead in large-scale IoT

The work was supported in part by National Key R&D Program of China (No. 2018YFB1800802), by Key Area R&D Program of Guangdong Province (No. 2018B030338001), by Natural Science Foundation of China (NSFC-61629101 and NSFC-61771066), and by Guangdong Research Project (No. 2017ZT07X152). (Co-correspondence Authors: Wenjun Xu and Shuguang Cui.)

Yue Xu and Wenjun Xu are with the Key Lab of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing, 100876 China. Yue Xu is also affiliated with the Shenzhen Research Institute of Big Data and School of Science and Engineering, the Chinese University of Hong Kong, Shenzhen, China, 518172. (Email:avexuyue@gmail.com; wjxu@bupt.edu.cn)

Zengde Deng and Anthony Man-Cho So are with the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong. (Email:dengzengde@gmail.com; manchoso@se.cuhk.edu.hk)

Mengdi Wang is with the Department of Electrical Engineering and Center for Statistics and Machine Learning, Princeton University. She is also affiliated with the Department of Operations Research and Financial Engineering and Department of Computer Science. (Email:mengdiw@princeton.edu)

Shuguang Cui is with the Shenzhen Research Institute of Big Data and Future Network of Intelligence Institute (FNii), The Chinese University of Hong Kong, Shenzhen, China, 518172 (Email: shuguangcui@cuhk.edu.cn).

networks. Besides, their convergence depends on the connectivity properties of the networked agents, which can be topology prohibitive in a randomly deployed IoT network. The above issues motivate us to propose a new collaboration protocol for MARL which can coordinate the local entities in a safe and communication-efficient way.

In this paper, we consider a collaborative MARL setting where the agents vote to make group decisions and the aim is to maximize the globally averaged return of all agents in the environment. Our primary interest is to develop a sample-efficient model-free MARL algorithm built upon voting-based coordinations in the context of infinite-horizon AMDP. Particularly, the considered AMDP does not assume the future rewards to be discounted while only needing to satisfy certain fast mixing property. This significantly complicates our analysis when compared with the discounted cases. The main contributions are summarized as follows.

- We formulate the MARL problem in the context of AMDP based on the linear programming form of the policy optimization problem and propose a primal-dual algorithm to obtain the optimal solution.
- We provide the first sublinear convergence rate for solving the MARL problem for infinite-horizon AMDP. The proposed algorithm and theoretical analysis also cover the single-agent RL as a special case, which makes them more general.
- We propose a voting-based collaboration protocol for the proposed MARL algorithm, through which the distributed learning achieves the same sublinear convergence as centralized learning. In other words, the proposed distributed decision-making process does not slow down the process of achieving global optimality. Moreover, the proposed voting-based protocol has superior data privacy and communication-efficiency than existing parameter-consensus-based protocols.

In addition, we also verify the convergence of our proposed algorithm through numerical simulations and conduct a case study in a multi-agent IoT system to justify the learning effectiveness.

The proposed model is promising for solving the long-run or non-terminating optimization tasks in multi-agent IoT systems, where distributed agents vote to determine a joint action, aiming at maximizing the globally averaged return of all agents. For example, the model can be employed to learn the optimal resource (e.g., communication bandwidth and channel) allocation policy for a group of IoT devices to improve the overall capacity; learn the optimal on/off policy for a group of base stations to improve the overall energy efficiency; learn the optimal trajectory planning policy for a group of UAVs to avoid collisions. Moreover, since the distributed agents only need to exchange their vote information for collaboration, without revealing their policy or value functions to each other, the proposed model would be preferable in privacy-sensitive applications, e.g., location services.

The remainder of this paper is organized as follows. Section II reviews the existing works on MARL. Section III introduces the problem formulations. Section IV presents the voting-based multi-agent reinforcement learning algorithm. Section V presents the convergence analysis of our proposed algorithm. Section VI discusses the simulation results. Finally, Section VII concludes the paper.

Notation: For a vector $\boldsymbol{x} \in \mathbb{R}^n$, we denote its i-th component as x_i, its transpose as $\boldsymbol{x}^\top$, and its Euclidean norm as $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$. For a positive number x, we write $\log x$ for its natural logarithm. For a vector $\boldsymbol{e} = (1, \ldots, 1)^\top$, we denote by $\boldsymbol{e}_i$ the vector with its i-th entry equaling 1 and other entries equaling 0. For two probability distributions p, q over a finite set X, we denote their Kullback-Leibler (KL) divergence as $D_{KL}(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$.

II. Related Work

Many existing model-free MARL algorithms are based on the framework of Markov games [31–35] or temporal-difference RL [16–21]. In the context of Markov games, the study of MARL usually models the MARL as stochastic games, such as cooperative games [31], zero-sum stochastic games [32], [36–38], general-sum stochastic games [33], decentralized Q-Learning [35], and the recent mean-field MARL [34]. Alternatively, the study of MARL in the context of temporal-difference RL mainly originates from dynamic programming, which learns by following the Bellman equation, including the ones based on deep neural networks [16–20] and the ones based on linear function approximators [21]. However, first, the above MARL models can only provide asymptotic convergence [21] to an optimal point or simply provide empirical evaluations without theoretical guarantees [16–20]. Second, they are all based on the discounted MDP, instead of the undiscounted AMDP. On the other hand, though average-reward RL has received much attention in recent years, most of them focus on the single-agent cases [25–27], [39], [40]. The research on average-reward MARL still undergoes exploration.

There are two lines of research in existing literature that focus on the saddle-point formulation of RL. One line studies the saddle-point formulation resulted from the fixed-point problem of policy evaluation [22–24], [41], [42], i.e., learning the value function of a fixed policy. Among others, the works [23], [24] provided the sample complexity analysis of policy evaluation in the context of MARL, where the policies of all agents are fixed. The other line, which includes this paper, focuses on the saddle-point formulation resulted from the policy optimization problem [39], [40], where the policy is continuously updated towards the optimal one. This makes the analysis substantially more challenging than that for policy evaluation. In the single-agent setting, our work is closely related to [39]. However, to the best of our knowledge, our work is the first to consider solving a saddle-point policy optimization in the context of MARL, which takes the coordination among multiple agents into account. Moreover, we also provide numerical simulations and case studies to corroborate our theoretical results, while previous works mainly focus on theoretical analysis [39], [40].

Finally, most MARL models are based on the parameter-consensus-based coordination, where the local agents consensus their parameters with a centralized server [14], [18], [29], [30] or their neighboring agents [21–24]. Although many

works adopted voting-based coordination in their proposed learning algorithms [43–45], they are not developed for MARL. A relevant work is [46], which proposed a dedicated majority voting rule to coordinate the MARL agents under discounted MDP, which, however, is a heuristic strategy without theoretical guarantees and may not perform well on non-terminating tasks.

III. Problem Formulation

In this paper, we consider the MARL in the presence of a generative model of the MDP [47–49]. The underlying MDP is unknown but having access to a sampling oracle, which takes an arbitrary state-action pair (i, a) as input and generates the next state j with probability $p_{ij}(a)$, along with an immediate reward for each individual agent. The goal is to find the optimal policy of the unknown AMDP by interacting with the sampling oracle. Such a simulator-defined MDP has been studied by existing literatures in the context of single-agent RL, including the model-based RL [47–49] and model-free RL [50], [51]. In what follows, we first introduce the settings of the multi-agent AMDP and then formulate the multi-agent policy optimization problem as a primal-dual saddle point optimization problem.

A. Multi-Agent AMDP

We focus on the infinite-horizon AMDP, which aims at optimizing the average-per-time-step reward over an infinite decision sequence. Existing works on RL usually impose a discount factor $\gamma \in (0, 1)$ on the future rewards to render the problem more tractable; e.g., by making the cumulative reward bounded. However, discounted RL may yield a poor performance over long-run (especially non-terminating) tasks and become computationally challenging when the discount factor is close to one [1], [25–27]. In this paper, we do not assume that the future rewards are discounted. Rather, we assume that the AMDP satisfies certain fast mixing property (given in Sec. V), which significantly complicates our analysis when compared with the discounted cases.

A multi-agent AMDP can be described by the tuple

$$\left(\mathcal{S}, \mathcal{A}, \mathcal{P}, \{\mathcal{R}_m\}_{m=1}^M\right),$$

where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} = \{p_{ij}(a) \mid i, j \in \mathcal{S}, a \in \mathcal{A}\}$ is the collection of state-to-state transition probabilities, and $\{\mathcal{R}_m\}_{m=1}^M$ is the collection of local reward functions with $\mathcal{R}_m = \left\{r_{ij}^m(a) \mid i, j \in \mathcal{S}, a \in \mathcal{A}\right\}$ and M being the number of agents. We consider the setting where the reward functions of the agents may differ from each other and are private to each corresponding agent. We assume that the reward $r_{ij}^m(a)$, where $i, j \in \mathcal{S}$, $a \in \mathcal{A}$, and $m = 1, \ldots, M$, lie in $[0, 1]$. This public state with private reward setting is widely considered in many recent works on collaborative MARL [21], [23], [24]. Moreover, we assume that the multi-agent AMDP is ergodic (i.e., aperiodic and recurrent), so that there is a unique stationary distribution under any stationary policy. The MARL system selects the action to take according to the votes from local agents. Each agent determines its vote individually without communicating with others. In particular, at each time step t, the MARL system works as follows: 1) all agents observe the state $i_t \in \mathcal{S}$; 2) each agent votes for the action a_t to take under i_t; 3) the system executes a_t according to the votes; 4) the system shifts to a new state $i_{t+1} \in \mathcal{S}$ with probability $p_{i_t i_{t+1}}(a_t)$ and returns the rewards $\{r_{i_t i_{t+1}}^m(a_t)\}_{m=1}^M$ to the agents.

B. Multi-Agent Policy Optimization

We denote the global acting policy, which determines the joint action to take, as $\pi^g \in \Xi \subseteq \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, where Ξ consists of non-negative matrices whose (i, a)-th entry $\pi_{i,a}^g$ specifies the probability of taking action a in state i. The multi-agent policy optimization problem aims at improving the global acting policy by maximizing the sum of local average-rewards, i.e.,

$$\max_{\pi^g}\left\{\bar{v}^{\pi^g} = \lim_{T\to\infty} \mathbb{E}^{\pi^g}\left[\frac{1}{T}\sum_{t=1}^{T}\sum_{m=1}^{M} r_{i_t i_{t+1}}^m(a_t)\Big| i_1 = i\right], i \in \mathcal{S}\right\}, \tag{1}$$

where $\mathbb{E}^{\pi^g}[\cdot]$ denotes the expectation over all the state-action trajectories generated by the MARL system when following the acting policy π^g. According to the theory of dynamic programming [52], [53], the value $\bar{v}^*$ is the optimal average reward to problem (1) if and only if it satisfies the following Bellman equation:

$$\begin{aligned}&\bar{v}^* + v^*(i)\\ &= \max_{a\in\mathcal{A}}\left\{\sum_{j\in\mathcal{S}} p_{ij}(a)v^*(j) + \sum_{j\in\mathcal{S}} p_{ij}(a)\sum_{m=1}^{M} r_{ij}^m(a)\right\}, \forall\ i \in \mathcal{S},\end{aligned} \tag{2}$$

where $p_{ij}(a)$ is the transition probability from state i to state j after taking the action a and $\boldsymbol{v}^* \in \mathbb{R}^{|\mathcal{S}|}$ is known as the difference-of-value vector that characterizes the transient effect of each initial state under the optimal policy [39]. Note that there exist infinitely many $\boldsymbol{v}^*$ that satisfy (2); e.g., by adding constant shifts. However, this does not affect our analysis. More detailed descriptions of $\boldsymbol{v}^*$ can be found in [39].

C. Saddle-Point Formulation

The Bellman equation in (2) can be written as the following linear programming problem:

$$\begin{aligned}\min_{\bar{v},\boldsymbol{v}}\quad & \bar{v}\\ \text{s.t.}\quad & \bar{v}\cdot\boldsymbol{e} + (I - P_a)\,\boldsymbol{v} - \sum_{m=1}^{M}\bar{\boldsymbol{r}}_a^m \geq \boldsymbol{0},\quad \forall\ a \in \mathcal{A},\end{aligned} \tag{3}$$

where $P_a \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ is the MDP transition matrix under action a whose (i, j)-th entry is $p_{ij}(a)$ and $\bar{\boldsymbol{r}}_a^m \in \mathbb{R}^{|\mathcal{S}|}$ is the expected state-transition reward under action a with

$\bar{r}^m_{i,a} = \sum_{j\in\mathcal{S}} p_{ij}(a) r^m_{ij}(a),\ \forall i \in \mathcal{S}$. The dual of (3) can be written as

$$\begin{aligned} \max_{\boldsymbol{\mu}} \quad & \sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\sum_{m=1}^{M} \mu_{i,a}\bar{r}^m_{i,a} \\ \text{s.t.} \quad & \sum_{a\in\mathcal{A}} \boldsymbol{\mu}_a^\top (I - P_a) = \mathbf{0}, \\ & \sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}} \mu_{i,a} = 1,\ \mu_{i,a} \geq 0, \end{aligned} \tag{4}$$

where $\boldsymbol{\mu}$ is the dual variable. By linear programming strong duality, if $(\bar{v}^*, \boldsymbol{v}^*)$ and $\boldsymbol{\mu}^*$ are optimal solutions to the primal and dual problems (3) and (4), respectively, then they satisfy the zero complementarity gap condition:

$$\begin{aligned} 0 &= \sum_{a\in\mathcal{A}} (\boldsymbol{\mu}^*_a)^\top \left(\bar{v}^* \cdot \boldsymbol{e} + (I - P_a)\, \boldsymbol{v}^* - \sum_{m=1}^{M} \bar{\boldsymbol{r}}^m_a \right) \\ &= \bar{v}^* + \sum_{a\in\mathcal{A}} (\boldsymbol{\mu}^*_a)^\top \left((I - P_a)\, \boldsymbol{v}^* - \sum_{m=1}^{M} \bar{\boldsymbol{r}}^m_a \right). \end{aligned} \tag{5}$$

Observe that problems (3) and (4) involve rather complicated constraints. Hence, it is common to consider their saddle-point formulation, whose constraints are simpler:

$$\min_{\boldsymbol{v}\in\mathcal{V}} \max_{\boldsymbol{\mu}\in\mathcal{U}} \sum_{a\in\mathcal{A}} \boldsymbol{\mu}_a^\top \left((P_a - I)\boldsymbol{v} + \sum_{m=1}^{M} \bar{\boldsymbol{r}}^m_a \right). \tag{6}$$

Here,

$$\mathcal{V} = \mathbb{R}^{|\mathcal{S}|},\ \mathcal{U} = \left\{ \boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|} \,\middle|\, \sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}} \mu_{i,a} = 1,\ \boldsymbol{\mu} \geq \mathbf{0} \right\}$$

are the primal and dual constraint sets, respectively. Later, we shall focus on multi-agent AMDPs that satisfy certain fast mixing property. This will allow us to use a smaller but still structured primal constraint set $\mathcal{V}$; see Sec. V.

It is known that there is a correspondence between randomized stationary policies and feasible solutions to the dual problem (4) [52]. In particular, given an optimal dual solution $\boldsymbol{\mu}^* \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, the optimal acting policy π^g can be obtained via $\pi^*_{i,a} = \mu^*_{i,a} / \sum_{a\in\mathcal{A}} \mu^*_{i,a}$. Hence, our goal now is to obtain an optimal dual solution $\boldsymbol{\mu}^*$.

IV. Voting-Based Learning Algorithm

In this section, we propose a voting mechanism that specifies how local votes determine the global action. Then, we prove that the voting mechanism yields an equivalence between the update on the global acting policy and that on the distributed voting policies. Consequently, problem (6) can be solved in a distributed manner, and we propose a primal-dual learning algorithm for it.

A. Voting Mechanism

We denote the pair of primal and dual variables corresponding to the global acting policy π^g as $\boldsymbol{v}^g$ and $\boldsymbol{\mu}^g$, respectively. We also introduce a pair of local primal and dual variables corresponding to each local voting π^m $(m = 1, \ldots, M)$ as $\boldsymbol{v}^m$ and $\boldsymbol{\mu}^m$, where $\pi^m \in \Xi \subseteq \mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$ is a randomized stationary policy. Then, the voting mechanism takes the form

$$\mu^{g,t}_{i,a} \propto \prod_{m=1}^{M} \mu^{m,t}_{i,a}.$$

The voting mechanism indeed reveals the relationship between the global acting policy and the local voting policies.

B. Primal-Dual Learning Algorithm

We now develop a primal-dual learning algorithm to solve problem (6) in a distributed manner based on a double-sampling strategy. Recall that we consider the MARL under a generative MDP, where the agents are interacting with a black-box sampling oracle to learn the optimal policy. The sampling oracle works in a similar way as the experience replay used in deep RL models [4], [14], [18], [20]. In practical applications, the sampling oracle or experience replay can be placed in a centralized node which can communicate with the local agents, as in many existing MARL frameworks [14], [18], [29], [30]. However, it only needs to collect the vote information $\mu^m_{i,a}$ from the agents in order to coordinate the sampling during the learning process, instead of performing iterative parameter consensus as existing methods [14], [18], [29], [30]. The detailed procedure is provided in Algorithm 1. In what follows, we first introduce the local dual and primal updates in our algorithm. Then, we prove that the local updates are equivalent to the global updates if the voting mechanism is specified properly.

1) Local Dual Update: We update the local dual variables based on uniform sampling. Specifically, the first state-action pair (i_t, a_t) to update the local dual variables is sampled with uniform probability $p^{\text{dual}}_{i,a} = \frac{1}{|\mathcal{S}|\cdot|\mathcal{A}|}$. The MARL system then shifts to the next state j_t conditioned on (i_t, a_t) and returns the local rewards $\{r^m_{i_t j_t}(a_t)\}_{m=1}^M$ to the agents. The local dual variable $\boldsymbol{\mu}^{m,t}$ of agent m is updated as

$$\mu^{m,t+1}_{i,a} = \begin{cases} \mu^{m,t}_{i,a} \exp\left\{\Delta^{m,t}_{i,a}\right\}, & \text{if } i = i_t,\ a = a_t, \\ \mu^{m,t}_{i,a}, & \text{otherwise}, \end{cases} \tag{7}$$

where

$$\Delta^{m,t}_{i,a} = \beta \left(\frac{\frac{1}{\beta}\log x^t + v^t_j - v^t_i - C}{M} + r^m_{ij}(a) \right) \tag{8}$$

with $(i, a, j) = (i_t, a_t, j_t)$, $\beta > 0$ being the step-size, C being a parameter to be specfied, and

$$x^t = \frac{1}{\sum_{i\in\mathcal{S}, a\in\mathcal{A}} \prod_{m=1}^{M} \mu^{m,t}_{i,a}}. \tag{9}$$

Here, x^t can be viewed as the proportion between the locally recovered partial derivatives and the global true partial derivatives of the minimax objective in (6). It also defines the explicit form of the voting mechanism; see Lemma 1 below. However, it is important to note that we do not need to compute x^t in our algorithm, as it does not influence the sampling in the subsequent primal update step and is used purely for analysis purposes. In other words, one can remove the term of $\log x^t$ from (8) without influencing the learning performance.

2) Local Primal Update: We update the local primal variables based on probability sampling, where the probability is specified by the dual variables. Specifically, the second state-action pair (i_t, a_t) to update the local primal variables is sampled with probability

$$p^{\text{primal}}_{i_t,a_t} = \frac{\prod_{m=1}^{M} \mu^{m,t}_{i_t,a_t}}{\sum_{i\in\mathcal{S},a\in\mathcal{A}} \prod_{m=1}^{M} \mu^{m,t}_{i,a}}. \tag{10}$$

The system then shifts to the next state j_t conditioned on (i_t, a_t), and returns the local rewards to the agents. The local primal variable $\boldsymbol{v}^t$ is updated as

$$\boldsymbol{v}^{t+1} = \Pi_{\mathcal{V}} \left\{ \boldsymbol{v}^t + \boldsymbol{d}^t \right\}, \tag{11}$$

where

$$\boldsymbol{d}^t = \alpha(\boldsymbol{e}_i - \boldsymbol{e}_j) \tag{12}$$

with $(i, j) = (i_t, j_t)$; $\alpha > 0$ is the step-size; $\Pi_{\mathcal{V}}\{\cdot\}$ denotes the projector onto the search space $\mathcal{V}$, which will be defined in Sec. V. Note that the local primal update is identical across the agents. Hence, we use the same notation v^t_i in the primal update for all the agents in the sequel.

3) Communication: The centralized sampling oracle needs to collect the vote information $\mu^m_{i,a}$ to compute the probability $p^{\text{primal}}_{i_t,a_t}$ according to (10) and returns the reward information to the agents. However, note that the vote information $\mu^m_{i,a}$ and the reward information $r^m_{i,j}(a)$ of each agent is a scalar, such that the communication overhead at each learning step of our method only scales as $\mathcal{O}(M)$. In contrast, most existing MARL methods are developed based on parameter consensus, where local agents need to reach consensus on its value or policy function with a centralized center [14], [18], [29], [30] or their nearby agents [21–24]. Since the value or policy function scales as $\mathcal{O}(|\mathcal{S}| \cdot |\mathcal{A}|)$, the communication overhead at each learning step of their models scales as $\mathcal{O}(M \cdot |\mathcal{S}| \cdot |\mathcal{A}|)$. Although this cost can be reduced if they adopt linear or nonlinear function to approximate the value or policy function, it is still related to the size of the function approximators, which can be enormous if they are deep neural networks. Moreover, the exchanged information in our algorithm is the vote information, instead of the privacy-sensitive policy or value information, which can alleviate privacy and security concerns considerably.

4) Equivalent Global Update: We now prove that with a properly specified voting mechanism, the primal-dual updates on the local voting policies are equivalent to the centralized primal-dual updates on the global acting policy.

Lemma 1 (Equivalent Global Update): By specifying the voting mechanism as

$$\mu^{g,t}_{i,a} = x^t \prod_{m=1}^{M} \mu^{m,t}_{i,a}, \tag{13}$$

where x^t is given by (9), the local primal-dual updates (7) and (11) are equivalent to the following global primal-dual updates:

$$\mu^{g,t+1}_{i,a} = x^{t+1} \mu^{g,t}_{i,a} \exp\left\{\Delta^{g,t}_{i,a}\right\}, \ \forall\ i \in \mathcal{S},\ a \in \mathcal{A}, \tag{14a}$$

$$\boldsymbol{v}^{t+1} = \Pi_{\mathcal{V}} \left\{ \boldsymbol{v}^t + \boldsymbol{d}^t \right\}. \tag{14b}$$

Here,

$$\Delta^{g,t}_{i,a} = \beta \left(v^t_j - v^t_i - C + \sum_{m=1}^{M} r^m_{ij}(a) \right) \tag{15}$$

and $\boldsymbol{d}^t = \alpha(\boldsymbol{e}_i - \boldsymbol{e}_j)$, where $(i, a) = (i_t, a_t)$ with probability $\mu^{g,t}_{i_t,a_t}$ and $j = j_t$ is obtained from the system by conditioning on (i_t, a_t). ■

We remark that the global primal-dual updates (14) are conditionally unbiased partial derivatives of the minimax objective given in (6).

Proof. Recall that the local dual variable $\boldsymbol{\mu}^{m,t}$ of agent m is updated by (7). We now prove a recursive relationship between $\mu^{g,t+1}_{i,a}$ and $\mu^{g,t}_{i,a}$ as follows. Given $(i, a) = (i_t, a_t)$, starting from the voting mechanism defined in (13), we have

$$\begin{aligned}
\mu^{g,t+1}_{i,a} &= x^{t+1} \prod_{m=1}^{M} \mu^{m,t+1}_{i,a} \\
&= x^{t+1} \prod_{m=1}^{M} \left(\mu^{m,t}_{i,a} \exp\left\{\Delta^{m,t}_{i,a}\right\} \right) \\
&= x^{t+1} \prod_{m=1}^{M} \mu^{m,t}_{i,a} \exp\left\{ \sum_{m=1}^{M} \Delta^{m,t}_{i,a} \right\} \\
&= x^{t+1} (x^t)^{-1} \mu^{g,t}_{i,a} \exp\left\{ \sum_{m=1}^{M} \Delta^{m,t}_{i,a} \right\} \\
&= x^{t+1} \mu^{g,t}_{i,a} \exp\left\{ \beta \left(v^t_j - v^t_i - C + \sum_{m=1}^{M} r^m_{ij}(a) \right) \right\}.
\end{aligned}$$

Hence, using the definition of $\Delta^{g,t}_{i,a}$ in (15), the local dual update based on $\Delta^{m,t}_{i,a}$ can be equivalently expressed as the global dual update based on $\Delta^{g,t}_{i,a}$, i.e., (14a) holds.

As for the local primal update, since the oracle generates the second sample with probability $p^{\text{primal}}_{i,a}$ given by (10), which is exactly the same as the global dual variable $\mu^{g,t}_{i,a}$ given in (13), the local and global primal updates are identical. ■

Lemma 2 (Unbiasedness): Consider the voting mechanism in Lemma 1. Let $\mathcal{F}_t$ be the filtration at time t, i.e., information about all the state-action pair sampling and state transition right before time t. Then, the dual update weight $\Delta^{g,t}_{i,a}$ is, up to a constant shift, a multiple of the conditional partial derivative of the minimax objective in (6) with respect to $\mu_{i,a}$:

$$\begin{aligned}
&\mathbb{E}\left[\Delta^{g,t}_{i,a} \mid \mathcal{F}_t\right] \\
&= \frac{\beta}{|\mathcal{S}| \cdot |\mathcal{A}|} \left((P_a - I)\boldsymbol{v}^t + \sum_{m=1}^{M} \bar{\boldsymbol{r}}^m_a - C \cdot \boldsymbol{e} \right)_i, \\
&\qquad \forall\ i \in \mathcal{S},\ a \in \mathcal{A}.
\end{aligned}$$

Moreover, the primal update weight d^t_i is a multiple of the conditional partial derivative of the minimax objective in (6) with respect to v_i:

$$\mathbb{E}\left[\boldsymbol{d}^t \mid \mathcal{F}_t\right] = \alpha \sum_{a\in\mathcal{A}} (I - P_a)^\top \boldsymbol{\mu}^{g,t}_a.$$

■

Proof. For arbitrary $i \in \mathcal{S}$ and $a \in \mathcal{A}$, we use (15) to compute

$$\begin{aligned}
&\frac{1}{\beta} \cdot \mathbb{E}\left[\Delta_{i,a}^{g,t} \mid \mathcal{F}_t\right] \\
&= \frac{1}{|\mathcal{S}| \cdot |\mathcal{A}|} \left(\sum_{j \in \mathcal{S}} p_{ij}(a) v_j^t - v_i^t \right) \\
&\quad + \frac{1}{|\mathcal{S}| \cdot |\mathcal{A}|} \left(\sum_{j \in \mathcal{S}} \sum_{m=1}^{M} p_{ij}(a) r_{ij}^m(a) - C \right) \\
&= \frac{1}{|\mathcal{S}| \cdot |\mathcal{A}|} \left((P_a - I)\boldsymbol{v}^t + \sum_{m=1}^{M} \bar{\boldsymbol{r}}_a^m - C \cdot \boldsymbol{e} \right)_i .
\end{aligned}$$

On the other hand, using (12) and the fact that the state-action pair for updating the primal variables is generated with probability $\boldsymbol{\mu}^{g,t}$, we compute, for an arbitrary $i \in \mathcal{S}$,

$$\begin{aligned}
&\mathbb{E}\left[\boldsymbol{d}^t \mid \mathcal{F}_t\right] \\
&= \alpha \left[\sum_{i \in \mathcal{S}} \Pr\left(i_t = i \mid \mathcal{F}_t\right) \boldsymbol{e}_i - \sum_{j \in \mathcal{S}} \Pr\left(j_t = j \mid \mathcal{F}_t\right) \boldsymbol{e}_j \right] \\
&= \alpha \left[\sum_{i \in \mathcal{S}} \sum_{a \in \mathcal{A}} \mu_{i,a}^{g,t} \boldsymbol{e}_i - \sum_{j \in \mathcal{S}} \sum_{i \in \mathcal{S}} \sum_{a \in \mathcal{A}} p_{ij}(a) \mu_{i,a}^{g,t} \boldsymbol{e}_j \right] \\
&= \alpha \sum_{a \in \mathcal{A}} (I - P_a)^\top \boldsymbol{\mu}_a^{g,t}.
\end{aligned}$$

This completes the proof. ∎

V. Theoretical Results

In this section, we present the convergence analysis of Algorithm 1. We start by making the following assumption on the considered multi-agent AMDP. A similar assumption has also been used in [39], [40] for the case of a single-agent RL.

Assumption 1: There exists a constant $t_{\text{mix}}^* > 0$ such that for any stationary policy π^g, we have

$$t_{\text{mix}}^* \geq \min_t \left\{ t \,\middle|\, \|(P^{\pi^g})^t(i,\cdot) - \boldsymbol{\nu}^{\pi^g}\|_{TV} \leq \frac{1}{4},\ \forall\ i \in \mathcal{S} \right\},$$

where $\|\cdot\|_{TV}$ is the total variation and $P^{\pi^g}(i,j) = \sum_{a \in \mathcal{A}} \pi_{i,a}^g p_{ij}(a)$. ∎

The above assumption requires the multi-agent AMDP to be sufficiently rapidly mixing, with the parameter t_{mix}^* characterizing how fast the multi-agent AMDP reaches its stationary distribution from any state under any acting policy [39]. In particular, t_{mix}^* controls the distance between any stationary policy and the optimal policy under the considered multi-agent AMDP. It has been shown in [39] that under Assumption 1, an optimal difference-of-value vector $\boldsymbol{v}^*$ satisfying $\|\boldsymbol{v}\|_\infty \leq 2t_{\text{mix}}^*$ exists.

Based on the above discussion, we can use the following smaller constraint set $\mathcal{V}$ for the global primal variable $\boldsymbol{v}$:

$$\mathcal{V} = \left\{ \boldsymbol{v} \in \mathbb{R}^{|\mathcal{S}|} \,\middle|\, \|\boldsymbol{v}\|_\infty \leq 2t_{\text{mix}}^* \right\}.$$

Now, we are ready to establish the convergence of our proposed Algorithm 1.

Algorithm 1 Voting-Based MARL

1: **Initialization:** MARL tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \{\mathcal{R}_m\}_{m=1}^M)$, time horizon T, parameters

$$\begin{aligned}
\alpha &= (4t_{\text{mix}}^* + M)\sqrt{\frac{|\mathcal{S}|}{|\mathcal{A}|} \cdot \frac{\log(|\mathcal{S}| \cdot |\mathcal{A}|)}{2T}}, \\
\beta &= \frac{1}{4t_{\text{mix}}^* + M}\sqrt{\frac{|\mathcal{S}| \cdot |\mathcal{A}| \cdot \log(|\mathcal{S}| \cdot |\mathcal{A}|)}{2T}}, \\
C &= 4t_{\text{mix}}^* + M.
\end{aligned}$$

2: Set $\boldsymbol{v} = \boldsymbol{0} \in \mathbb{R}^{|\mathcal{S}|}$ and $\mu_{i,a}^{m,0} = \frac{1}{|\mathcal{S}| \cdot |\mathcal{A}|},\ \forall\ i \in \mathcal{S},\ a \in \mathcal{A}$.
3: **Iteration:**
4: **for** $t = 1, 2, \ldots, T$ **do**
5: The system samples (i_t, a_t) with probability $p_{i_t,a_t}^{\text{dual}}$.
6: The system shifts to next state j_t conditioned on (i_t, a_t) and generates the local rewards $\left\{r_{i_t j_t}^m(a_t)\right\}_{m=1}^M$.
7: **for** $m = 1, 2, \ldots, M$ **do**
8: The agent m updates its local dual variable according to (7).
9: **end for**
10: The system collects the updated $\mu_{i,a}^{m,t+1}$ from local agents and samples (i_t, a_t) with probability $p_{i_t,a_t}^{\text{primal}}$.
11: The system shifts to next state j_t conditioned on (i_t, a_t) and generates the local rewards $\left\{r_{i_t j_t}^m(a_t)\right\}_{m=1}^M$which are returned to the agents.
12: **for** $m = 1, 2, \ldots, M$ **do**
13: The agent m updates its local primal variable according to (11).
14: **end for**
15: **end for**
16: Set $\hat{\mu}_{i,a}^g = \frac{1}{T}\sum_{t=1}^T \prod_{m=1}^M \mu_{i,a}^{m,t},\ \forall\ i \in \mathcal{S},\ a \in \mathcal{A}$.
17: **Return:** $\hat{\pi}_{i,a}^g = \frac{\hat{\mu}_{i,a}^g}{\sum_{a \in \mathcal{A}} \hat{\mu}_{i,a}^g},\ \forall\ i \in \mathcal{S},\ a \in \mathcal{A}$.

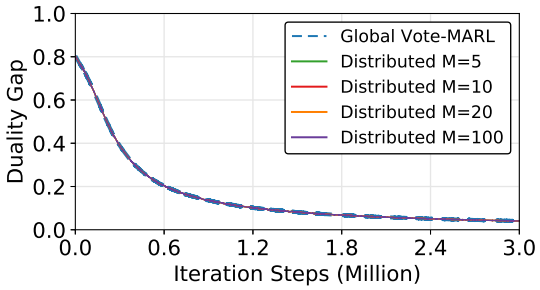
Theorem 1 (Finite-Iteration Duality Gap): Let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \{\mathcal{R}_m\}_{m=1}^M)$ be an arbitrary multi-agent AMDP tuple satisfying Assumption 1. Then, the sequence of iterates generated by Algorithm 1 satisfies

$$\begin{aligned}
&\bar{v}^* + \frac{1}{T}\sum_{t=1}^T \mathbb{E}\left[\sum_{a \in \mathcal{A}} \left((I - P_a)\boldsymbol{v}^* - \sum_{m=1}^M \bar{\boldsymbol{r}}_a^m \right)^\top \boldsymbol{\mu}_a^{g,t} \right] \\
&\leq \tilde{O}\left((4t_{\text{mix}}^* + M)\sqrt{\frac{|\mathcal{S}| \cdot |\mathcal{A}|}{T}} \right),
\end{aligned}$$

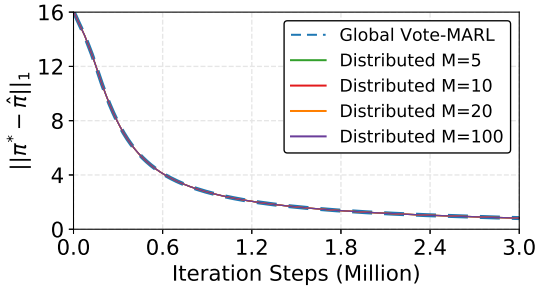
where $\tilde{O}(\cdot)$ hides polylogarithmic factors. ∎

Recall from (5) that the complementarity gap of a pair of optimal solutions to the primal-dual problems (3) and (4) is zero. Hence, Theorem 1 suggests that the iterates $\{\boldsymbol{\mu}^{g,t}\}_{t \geq 0}$ converge to an optimal solution to the dual problem (4) at a sublinear rate. The result also covers the single-agent RL [39] as a special case, which makes our model more general. We defer the proof of Theorem 1 to the appendix.

It is worth pointing out that in our proof, the scalar M in Theorem 1, i.e., the number of agents, comes from the bound of the total reward of all agents $\sum_{m=1}^M r_{ij}^m(a) \in [0, M]$,



(a) Convergence of duality gap



(b) Convergence to the global optimal policy

Fig. 1: Empirical convergence rate of the proposed algorithm. Each generated MDP instance contains $|\mathcal{S}| = 50$ states and $|\mathcal{A}| = 10$ actions at each state. The number of local agents varies from $M = 5$ to $M = 100$ with the total reward of all agents bounded in $[0, 1]$.

$\forall\ i, j \in \mathcal{S},\ a \in \mathcal{A}$. As such, if we consider a normalized reward where $\sum_{m=1}^{M} r_{ij}^m(a) \in [0, 1]$, then the complexity in Theorem 1 will be independent of M.

VI. Numerical Results

In this section, we evaluate the proposed voting-based MARL algorithm through two case studies. In the first case study, we verify the convergence of our proposed algorithm with the generated MDP instances. In the second case study, we exhibit how to apply our proposed algorithm to solve the placement optimization task in a UAV-assisted IoT network, where the ground base stations and the UAV-mounted base station are treated as the IoT devices. The UAV-mounted base station collects vote information from the ground base stations, which is then used to determine the placement of the the UAV-mounted base station to maximize the overall system capacity. Our results show that the distributed decision making does not slow down the process of achieving global consensus on optimality and that voting-based learning is more efficient than letting agents behave individually and selfishly.

A. Empirical Convergence

We generate instances of the multi-agent MDP using a similar setup as in [54]. Specifically, given a state and an action, the multi-agent MDP shifts to the next state assigned from the entire set without replacement. The transition probabilities are generated randomly from $[0, 1]$ and then normalized so that they sum to one. The optimal policy is generated with purposeful behavior by letting the agent favor a single action in each state and assigning it with a higher expected reward in $[0, 1]$.

In Fig. 1, we show the empirical convergence results of 1) the duality gap, i.e., the one given in Theorem 1; 2) the distance between the optimal policy and the learned policy, i.e., $\|\boldsymbol{\pi}^* - \hat{\boldsymbol{\pi}}\|_1$. The convergence curves are averaged over 100 instances. Generally, the empirical convergence rates corroborate the result given in Theorem 1. Besides, we also present 1) the performance change as the number of local agents varies from $M = 5$ to $M = 100$ and 2) the performance of centralized learning, which directly uses the global primal-dual updates to learn the global policy. The result shows that the empirical convergence rates of the centralized case and the distributed case are the same for different numbers of agents M. This indicates that distributed decision making does not slow down the process of achieving global consensus on optimality.

B. Application in Multi-Agent IoT Systems

We now apply the proposed voting-based MARL algorithm to a multi-agent IoT system which contains ground base stations, smartphones, and UAVs. In particular, UAV-assisted wireless communication has recently attracted much attention [9], [55–57], due to that UAV mounted with a mobile base station (UAV-BS) can provide high-speed air-to-ground data access by using the line-of-sight (LoS) communication links. However, obtaining the best performance in an UAV-BS-assisted wireless system highly depends on the placement of the UAV-BS [9], [55], [56]. Here, we consider optimizing the placement of UAV-BS continuously through our proposed voting-based MARL algorithm.

Existing works on the placement optimization of UAV-BS have two major drawbacks. First, many of them do not consider user movements [56], [58–61], but the change of user distribution can largely influence the system performance. Second, many of them determine the optimal placement of UAV-BS by assuming that the performance gain of each ground BS is public information [9], [61], which may be impractical in real-world wireless systems that have mixed wireless operators, infrastructures, and protocols. To overcome these drawbacks, we model the UAV-BS placement optimization as a voting-based MARL problem, where multiple ground BS learn to place the UAV-BS optimally with adaptation to user movements and without the need to share their reward information. The aim is to maximize the global performance gain of all ground BS.

We consider the downlink of a wireless cellular network. As shown in Fig. 2, the 2km×2km area of interest has $M = 20$ regularly deployed ground BS, one UAV-BS flying at 200m to provide air-to-ground communications, and 200 mobile users moving according to the random walk model in [62], each having a constant-bit-rate communication demand. The UAV-BS can move to any one of the aerial locations from a finite set $|\mathcal{A}|$ to provide air-to-ground communication. The user mobility follows the random walk model in [62], where each user moves at an angle uniformly distributed between $[0, 2\pi]$ and a random speed between $[0, c_{\max}]$ with $c_{\max}$ being

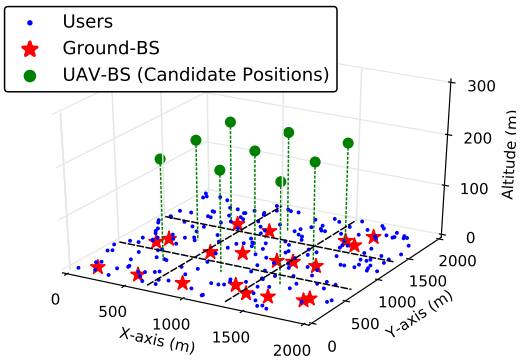


Fig. 2: 3D distribution of the investigated 4km^2 area, which contains $M = 20$ randomly deployed ground BS, one UAV-BS flying among $|\mathcal{A}| = 9$ candidate aerial locations, and 200 mobile users moving by following the random walk model [62]. The action is to move the UAV-BS to any one of the $|\mathcal{A}| = 9$ candidate aerial locations, which is determined by the votes from the ground BS.

TABLE I: Parameters

Parameters	**Values**
CBR (C_u)	128 kbps
Total 2D area	4 km^2
Total bandwidth	20 MHz
Carrier frequency (f_c)	2 GHz
PRB bandwith (B)	180 kHz
Max user velocity ($c_{\max}$)	10 m/s
Ground BS max transmit power (P_m)	46 dBm
UAV-BS max transmit power (P_U)	20 dBm
Additional LoS path loss (η_{LoS})	1 dB
Noise power spectral density (N_0)	-174 dBm/Hz

the maximum moving speed. Table I summarizes the main parameters. The air-to-ground channel and ground-to-ground channel are modeled according to [63] (Sec. II). The load of each base station is defined as the ratio between the required number of PRBs and the total number of available PRBs according to [4] (Sec. II-B).

The learning context is defined as follows. 1) States: We divide the area of interest into 3×3 grids and use the load of each grid to characterize the wireless system status. The load of each grid is indicated by one of two states: a) overloaded, if the users' demand within the grid is higher than the mean demands of all the grids; b) underloaded, otherwise. Since the grids cannot be all overloaded or all underloaded, there are only $|\mathcal{S}| = 510$ states for the wireless system with 9 grids. 2) Actions: The action set $\mathcal{A}$ is defined as the available aerial locations for the placement of the UAV-BS. At each time t, the UAV-BS chooses an action $a_t \in \mathcal{A}$ for placement. 3) Rewards: The reward function is defined with the aim to maximize user throughput. Specifically, we assume that users are always handed over to the BS with the best SINR, so that an increased load at the UAV-BS usually indicates an increased user throughput due to better user SINRs. Hence, we define the reward to be the increased load at the UAV-BS.

We compare the proposed voting-based MARL algorithm with four baselines: 1) the classic Q-learning algorithm [1],

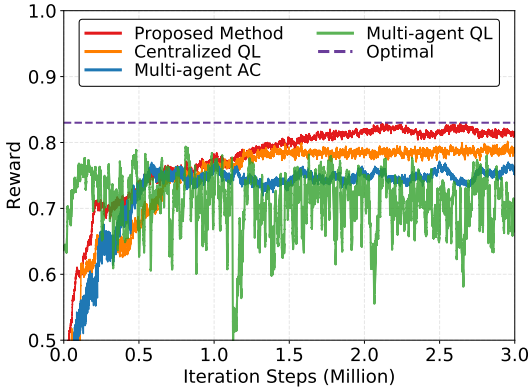


Fig. 3: Rewards for the UAV-BS placement optimization.

which uses centralized Q-learning to learn the optimal UAV placement policy; 2) the multi-agent actor-critic algorithm based on the centralized learning with decentralized execution framework [18], where distributed agents optimize the placement policy jointly by communicating with a centralized center; 3) the multi-agent Q-learning algorithm proposed in [15], where each agent performs independent Q-learning and treats the other agents as part of the environment; 4) the optimal scheme, obtained by assuming that the underlying MDP is known. We refer to them as *centralized QL*, *multi-agent AC*, *multi-agent QL*, and *optimal* for short, respectively. In addition, we adopt the majority voting rule proposed for multi-agent Q-learning in [46] to determine the joint action for both the multi-agent AC algorithm and the multi-agent QL algorithm.

In Fig. 3, we present the averaged rewards over 20 runs. The result shows that the performance of our proposed voting-based MARL algorithm outperforms all the comparing algorithms and is close to the optimal scheme. The discount factor for discounted RL methods is set to be 0.9. The performance gap between our proposed method and the centralized QL indicates that undiscounted RL methods are likely to outperform discounted RL methods in continuing optimization tasks. The performance gap between centralized QL and multi-agent AC/QL indicates that existing MARL algorithms exhibit a performance degradation compared with their centralized versions. In contrast, our proposed MARL algorithm achieves an equivalent performance to its centralized version. In addition, the performance of the multi-agent QL algorithm is the worst and has a large variance. This verifies that specifying a proper collaboration protocol among the distributed agents is critical in MARL in order to improve the learning performance.

We further compare our proposed voting-based scheme with two baselines: 1) the random-voting scheme, where the MARL system randomly chooses one agent to determine the global action per iteration; 2) the greedy scheme, where the MARL system aims at maximizing the cumulative reward of a single agent. Fig. 4 presents the averaged reward of each agent over 20 runs. The rewards of the greedy-maximizing scheme indicate the maximum obtainable reward of each agent, while the rewards of the random-voting scheme indicate the learning effectiveness without the proposed voting mechanism.

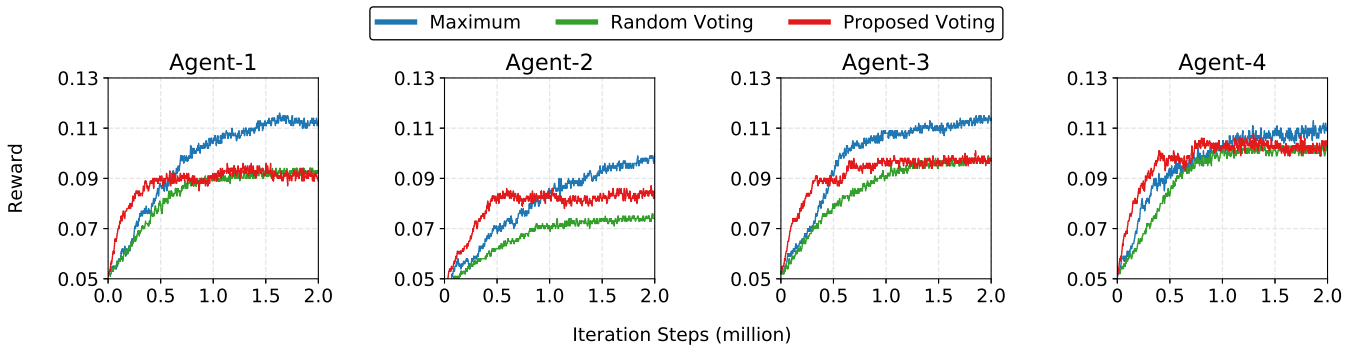


Fig. 4: Local reward of each agent.

The performance of our proposed voting-based scheme lies between the two baselines, which indicates that the agents are *learning to compromise* in order to maximize the cumulative global reward.

VII. Conclusions

In this paper, we considered a collaborative MARL problem, where the agents vote to make group decisions. Specifically, the agents are coordinated to follow the proposed voting mechanism without revealing their own rewards to each other. We gave a saddle-point formulation of the concerned MARL problem and proposed a primal-dual learning algorithm for solving it. We showed that our proposed algorithm achieves the same sublinear convergence rate as centralized learning. Finally, we provided empirical results to demonstrate the learning effectiveness. More interesting applications in the IoT system and the voting mechanism in the context of competitive MARL can be explored in the future.

Appendix
Proof of Theorem 1

Our proof shares a similar spirit as that of Theorem 1 in [39]. However, the analysis in [39] does not readily extend to the case of multi-agent AMDP. As a result, we have to develop a separate new convergence analysis here.

By virtue of Lemma 1, it suffices to study the progress made by the sequences of global dual variables $\{\boldsymbol{\mu}^{g,t}\}_{t\geq 0}$ and global primal variables $\{\boldsymbol{v}^t\}_{t\geq 0}$ in Algorithm 1. We begin with the following lemma, which gives an estimate of the progress of the dual variables in terms of KL-divergence.

Lemma 3 (Dual Improvement in KL-Divergence): The iterates generated by Algorithm 1 will satisfy

$$\begin{aligned}&\mathbb{E}\left[D_{KL}(\boldsymbol{\mu}^{g,*}\|\boldsymbol{\mu}^{g,t+1})\mid\mathcal{F}_t\right]-D_{KL}(\boldsymbol{\mu}^{g,*}\|\boldsymbol{\mu}^{g,t})\\&\leq\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}(\mu_{i,a}^{g,t}-\mu_{i,a}^{g,*})\mathbb{E}\left[\Delta_{i,a}^{g,t}\mid\mathcal{F}_t\right]\\&\quad+\frac{1}{2}\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,t}\mathbb{E}\big[\big(\Delta_{i,a}^{g,t}\big)^2\mid\mathcal{F}_t\big],\end{aligned}\tag{16}$$

for all $t\geq 0$. ■

Proof. By definition, we have

$$\begin{aligned}&D_{KL}(\boldsymbol{\mu}^{g,*}\|\boldsymbol{\mu}^{g,t+1})-D_{KL}(\boldsymbol{\mu}^{g,*}\|\boldsymbol{\mu}^{g,t})\\&=\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,*}\log\frac{\mu_{i,a}^{g,*}}{\mu_{i,a}^{g,t+1}}-\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,*}\log\frac{\mu_{i,a}^{g,*}}{\mu_{i,a}^{g,t}}\\&=\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,*}\log\frac{\mu_{i,a}^{g,t}}{\mu_{i,a}^{g,t+1}}.\end{aligned}$$

According to (9), (13), and (14a), we have

$$\begin{aligned}\log\mu_{i,a}^{g,t+1}&=\log\frac{\mu_{i,a}^{g,t}\exp\{\Delta_{i,a}^{g,t}\}}{\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,t}\exp\{\Delta_{i,a}^{g,t}\}}\\&=\log\mu_{i,a}^{g,t}+\Delta_{i,a}^{g,t}-\log(Z),\end{aligned}$$

where $Z=\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,t}\exp\{\Delta_{i,a}^{g,t}\}$. It follows that

$$\begin{aligned}&D_{KL}(\boldsymbol{\mu}^{g,*}\|\boldsymbol{\mu}^{g,t+1})-D_{KL}(\boldsymbol{\mu}^{g,*}\|\boldsymbol{\mu}^{g,t})\\&=\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,*}\log\frac{\mu_{i,a}^{g,t}}{\mu_{i,a}^{g,t+1}}\\&=\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,*}\big(\log\mu_{i,a}^{g,t}-\log\mu_{i,a}^{g,t}-\Delta_{i,a}^{g,t}+\log(Z)\big)\\&=\log(Z)-\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,*}\Delta_{i,a}^{g,t}.\end{aligned}$$

Now, for any $\boldsymbol{v}^t\in\mathcal{V}$, we have $\|\boldsymbol{v}^t\|_\infty\leq 2t_{\text{mix}}^*$. Moreover, we have $r_{ij}^m(a)\in[0,1]$ by assumption. Hence, we have

$$v_j^t-v_i^t+\sum_{m=1}^{M}r_{ij}^m(a)\leq 4t_{\text{mix}}^*+M.$$

This, together with the fact that $C=4t_{\text{mix}}^*+M$, implies $\Delta_{i,a}^{g,t}\leq 0,\ \forall\ i\in\mathcal{S},\ a\in\mathcal{A},\ t=0,1,\ldots$. On the other hand,

$$\begin{aligned}&\log(Z)\\&=\log\left(\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,t}\exp\left\{\Delta_{i,a}^{g,t}\right\}\right)\\&\leq\log\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,t}\left(1+\Delta_{i,a}^{g,t}+\frac{1}{2}\big(\Delta_{i,a}^{g,t}\big)^2\right)\qquad(17a)\\&=\log\left(1+\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,t}\Delta_{i,a}^{g,t}+\frac{1}{2}\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,t}\big(\Delta_{i,a}^{g,t}\big)^2\right)\\&\leq\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,t}\Delta_{i,a}^{g,t}+\frac{1}{2}\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,t}\big(\Delta_{i,a}^{g,t}\big)^2,\qquad(17b)\end{aligned}$$

where (17a) uses the fact that $\exp\{x\} \leq 1 + x + \frac{1}{2}x^2$ for $x \leq 0$ and (17b) uses the fact that $\log(1+x) \leq x$ for $x > -1$. Therefore, by combining the above results and taking conditional expectation $\mathbb{E}[\cdot \mid \mathcal{F}_t]$ on both sides, we obtain (16), as desired. ∎

Our strategy now is to bound the two terms on the right-hand side of (16) separately.

Lemma 4: The iterates generated by Algorithm 1 satisfy

$$\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}(\mu_{i,a}^{g,t}-\mu_{i,a}^{g,*})\mathbb{E}\left[\Delta_{i,a}^{g,t}\mid\mathcal{F}_t\right]$$
$$=\frac{\beta}{|\mathcal{S}|\cdot|\mathcal{A}|}\sum_{a\in\mathcal{A}}\left(\boldsymbol{\mu}_a^{g,t}-\boldsymbol{\mu}_a^{g,*}\right)^\top\left((P_a-I)\boldsymbol{v}^t+\sum_{m=1}^{M}\bar{\boldsymbol{r}}_a^m\right)$$

for all $t \geq 0$. ∎

Proof. For arbitrary $i \in \mathcal{S}$ and $a \in \mathcal{A}$, we have

$$\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}(\mu_{i,a}^{g,t}-\mu_{i,a}^{g,*})\mathbb{E}\left[\Delta_{i,a}^{g,t}\mid\mathcal{F}_t\right]$$
$$=\frac{\beta}{|\mathcal{S}|\cdot|\mathcal{A}|}\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}(\mu_{i,a}^{g,t}-\mu_{i,a}^{g,*})\left((P_a-I)\boldsymbol{v}^t+\sum_{m=1}^{M}\bar{\boldsymbol{r}}_a^m\right)_i$$
$$-\frac{C\beta}{|\mathcal{S}|\cdot|\mathcal{A}|}\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}(\mu_{i,a}^{g,t}-\mu_{i,a}^{g,*}) \tag{18}$$
$$=\frac{\beta}{|\mathcal{S}|\cdot|\mathcal{A}|}\sum_{a\in\mathcal{A}}(\boldsymbol{\mu}_a^{g,t}-\boldsymbol{\mu}_a^{g,*})^\top\left((P_a-I)\boldsymbol{v}^t+\sum_{m=1}^{M}\bar{\boldsymbol{r}}_a^m\right), \tag{19}$$

where (18) follows from Lemma 2 and (19) comes from the fact that

$$\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,t}=\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,*}=1.$$

This completes the proof. ∎

Lemma 5: The iterates generated by Algorithm 1 satisfy

$$\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,t}\mathbb{E}\left[\left(\Delta_{i,a}^{g,t}\right)^2\mid\mathcal{F}_t\right]\leq\frac{4\beta^2}{|\mathcal{S}|\cdot|\mathcal{A}|}\left(4t_{\text{mix}}^*+M\right)^2$$

for all $t \geq 0$. ∎

Proof. Using (15), the assumptions that $r_{ij}^m(a) \in [0,1]$ and $\boldsymbol{v}^t \in \mathcal{V}$, and the definition of C, we compute

$$\mathbb{E}\left[\left(\Delta_{i,a}^{g,t}\right)^2\mid\mathcal{F}_t\right]$$
$$=\frac{\beta^2}{|\mathcal{S}|\cdot|\mathcal{A}|}\sum_{j\in\mathcal{S}}p_{ij}(a)\left(v_j^t-v_i^t-C+\sum_{m=1}^{M}r_{ij}^m(a)\right)^2$$
$$\leq\frac{4\beta^2}{|\mathcal{S}|\cdot|\mathcal{A}|}\sum_{j\in\mathcal{S}}p_{ij}(a)\left(4t_{\text{mix}}^*+M\right)^2$$
$$=\frac{4\beta^2}{|\mathcal{S}|\cdot|\mathcal{A}|}\left(4t_{\text{mix}}^*+M\right)^2.$$

Since $\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,t+1}=1$, the result follows. ∎

Next, we give an estimate on the distance of the primal iterate $\boldsymbol{v}^t$ to the optimal primal variable $\boldsymbol{v}^*$.

Lemma 6 (Distance to Primal Optimality): The iterates generated by Algorithm 1 satisfy

$$\mathbb{E}\left[\|\boldsymbol{v}^{t+1}-\boldsymbol{v}^*\|^2\mid\mathcal{F}_t\right]$$
$$\leq\|\boldsymbol{v}^t-\boldsymbol{v}^*\|^2+2\alpha(\boldsymbol{v}^t-\boldsymbol{v}^*)^\top\left(\sum_{a\in\mathcal{A}}(I-P_a)^\top\boldsymbol{\mu}_a^{g,t}\right)+2\alpha^2$$

for all $t \geq 0$. ∎

Proof. We compute

$$\mathbb{E}\left[\|\boldsymbol{v}^{t+1}-\boldsymbol{v}^*\|^2\mid\mathcal{F}_t\right]$$
$$=\mathbb{E}\left[\|\Pi_{\mathcal{V}}\{\boldsymbol{v}^t+\boldsymbol{d}^t\}-\boldsymbol{v}^*\|^2\mid\mathcal{F}_t\right]$$
$$\leq\mathbb{E}\left[\|\boldsymbol{v}^t+\boldsymbol{d}^t-\boldsymbol{v}^*\|^2\mid\mathcal{F}_t\right]$$
$$=\|\boldsymbol{v}^t-\boldsymbol{v}^*\|^2+2(\boldsymbol{v}^t-\boldsymbol{v}^*)^\top\mathbb{E}\left[\boldsymbol{d}^t\mid\mathcal{F}_t\right]+\mathbb{E}\left[\|\boldsymbol{d}^t\|^2\mid\mathcal{F}_t\right],$$

where the inequality follows from the fact that $\boldsymbol{v}^* \in \mathcal{V}$ and the projector $\Pi_{\mathcal{V}}\{\cdot\}$ is non-expansive. By Lemma 2, we have

$$\mathbb{E}\left[\boldsymbol{d}^t\mid\mathcal{F}_t\right]=\alpha\sum_{a\in\mathcal{A}}(I-P_a)^\top\boldsymbol{\mu}_a^{g,t}.$$

Finally, using the definition of $\boldsymbol{d}^t$ in (12), we have $\mathbb{E}\left[\|\boldsymbol{d}^t\|^2\mid\mathcal{F}_t\right]=2\alpha^2$. This completes the proof. ∎

We are now ready to establish the key recursion that will lead to our desired bound on the convergence rate of our proposed Algorithm 1.

Lemma 7: Define

$$V^t=D_{KL}(\boldsymbol{\mu}^{g,*}\|\boldsymbol{\mu}^{g,t})+\frac{1}{2|\mathcal{S}|(4t_{\text{mix}}^*+M)^2}\|\boldsymbol{v}^t-\boldsymbol{v}^*\|^2,$$
$$W^t=\sum_{a\in\mathcal{A}}(\boldsymbol{\mu}_a^{g,t})^\top\left((I-P_a)\boldsymbol{v}^*-\sum_{m=1}^{M}\bar{\boldsymbol{r}}_a^m\right)+\bar{v}^*.$$

The iterates generated by Algorithm 1 satisfy

$$\mathbb{E}\left[V^{t+1}\mid\mathcal{F}_t\right]\leq V^t-\frac{\beta}{|\mathcal{S}|\cdot|\mathcal{A}|}W^t+3\beta^2\cdot\frac{(4t_{\text{mix}}^*+M)^2}{|\mathcal{S}|\cdot|\mathcal{A}|}$$

for all $t \geq 0$. ∎

Proof. Using the results in Lemmas 3–6 and taking $\alpha = \frac{1}{|\mathcal{A}|}(4t_{\text{mix}}^*+M)^2\beta$, we compute

$$\mathbb{E}\left[V^{t+1}\mid\mathcal{F}_t\right]$$
$$\leq V^t+\left(2+\frac{1}{|\mathcal{A}|}\right)\beta^2\cdot\frac{(4t_{\text{mix}}^*+M)^2}{|\mathcal{S}|\cdot|\mathcal{A}|}$$
$$+\frac{\beta}{|\mathcal{S}|\cdot|\mathcal{A}|}\sum_{a\in\mathcal{A}}\left(\boldsymbol{\mu}_a^{g,t}-\boldsymbol{\mu}_a^{g,*}\right)^\top\left((P_a-I)\boldsymbol{v}^t+\sum_{m=1}^{M}\bar{\boldsymbol{r}}_a^m\right)$$
$$+\frac{\beta}{|\mathcal{S}|\cdot|\mathcal{A}|}(\boldsymbol{v}^t-\boldsymbol{v}^*)^\top\left(\sum_{a\in\mathcal{A}}(I-P_a)^\top\boldsymbol{\mu}_a^{g,t}\right).$$

Now, observe that

$$\begin{aligned}
&\sum_{a\in\mathcal{A}}\left(\boldsymbol{\mu}_a^{g,t}-\boldsymbol{\mu}_a^{g,*}\right)^\top\left((P_a-I)\boldsymbol{v}^t+\sum_{m=1}^{M}\bar{\boldsymbol{r}}_a^m\right)\\
&\quad+(\boldsymbol{v}^t-\boldsymbol{v}^*)^\top\left(\sum_{a\in\mathcal{A}}(I-P_a)^\top\boldsymbol{\mu}_a^{g,t}\right)\\
&=\sum_{a\in\mathcal{A}}\left(\boldsymbol{\mu}_a^{g,t}-\boldsymbol{\mu}_a^{g,*}\right)^\top\left((P_a-I)\boldsymbol{v}^t+\sum_{m=1}^{M}\bar{\boldsymbol{r}}_a^m\right)\\
&\quad+(\boldsymbol{v}^t-\boldsymbol{v}^*)^\top\left(\sum_{a\in\mathcal{A}}(I-P_a)^\top(\boldsymbol{\mu}_a^{g,t}-\boldsymbol{\mu}_a^{g,*})\right) \qquad (20a)\\
&=\sum_{a\in\mathcal{A}}\left(\boldsymbol{\mu}_a^{g,t}-\boldsymbol{\mu}_a^{g,*}\right)^\top\left((P_a-I)\boldsymbol{v}^*+\sum_{m=1}^{M}\bar{\boldsymbol{r}}_a^m\right)\\
&=\sum_{a\in\mathcal{A}}(\boldsymbol{\mu}_a^{g,t})^\top\left((P_a-I)\boldsymbol{v}^*+\sum_{m=1}^{M}\bar{\boldsymbol{r}}_a^m\right)-\bar{v}^*\sum_{a\in\mathcal{A}}(\boldsymbol{\mu}_a^{g,*})^\top\boldsymbol{e} \qquad (20b)\\
&=\sum_{a\in\mathcal{A}}(\boldsymbol{\mu}_a^{g,t})^\top\left((P_a-I)\boldsymbol{v}^*+\sum_{m=1}^{M}\bar{\boldsymbol{r}}_a^m\right)-\bar{v}^*, \qquad (20c)
\end{aligned}$$

where (20a) and (20c) use the dual feasibility conditions $\sum_{a\in\mathcal{A}}(\boldsymbol{\mu}_a^{g,*})^\top(I-P_a)=\mathbf{0}$ and $\sum_{i\in\mathcal{S}}\sum_{a\in\mathcal{A}}\mu_{i,a}^{g,*}=1$ in (4), respectively; (20b) uses the complementarity condition

$$\mu_{i,a}^{g,*}\left((P_a-I)\boldsymbol{v}^*+\sum_{m=1}^{M}\bar{\boldsymbol{r}}_a^m-\bar{v}^*\cdot\boldsymbol{e}\right)_i=0,\ \forall\, i\in\mathcal{S},\ a\in\mathcal{A}$$

of the linear program (3). Combining the preceding relations, we obtain Lemma 7. ■

Proof of Theorem 1: We claim that

$$V^1\le\log(|\mathcal{S}|\cdot|\mathcal{A}|)+\frac{2(t_{\text{mix}}^*)^2}{(4t_{\text{mix}}^*+M)^2}.$$

To see this, we note that $\mu^{g,1}$ is the uniform distribution and $\boldsymbol{v}^0,\boldsymbol{v}^*\in\mathcal{V}$. Therefore, we have $D_{KL}(\boldsymbol{\mu}^{g,*}\|\boldsymbol{\mu}^{g,1})\le\log(|\mathcal{S}|\cdot|\mathcal{A}|)$ and $\|\boldsymbol{v}^t-\boldsymbol{v}^*\|^2\le 4|\mathcal{S}|(t_{\text{mix}}^*)^2$ for $t=0,1,\ldots$. This yields

$$\begin{aligned}
V^1&\le D_{KL}(\boldsymbol{\mu}^{g,*}\|\boldsymbol{\mu}^{g,1})+\frac{1}{2|\mathcal{S}|(4t_{\text{mix}}^*+M)^2}\|\boldsymbol{v}^1-\boldsymbol{v}^*\|^2\\
&\le\log(|\mathcal{S}|\cdot|\mathcal{A}|)+\frac{2(t_{\text{mix}}^*)^2}{(4t_{\text{mix}}^*+M)^2}.
\end{aligned}$$

Now, we rearrange the terms in Lemma 7 and obtain

$$W^t\le\frac{|\mathcal{S}|\cdot|\mathcal{A}|}{\beta}(V^t-\mathbb{E}\left[V^{t+1}\mid\mathcal{F}_t\right])+3\beta(4t_{\text{mix}}^*+M)^2.$$

Summing over $t=1,\ldots,T$ and taking the expectation, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}W^t\right]\\
&\le\frac{|\mathcal{S}|\cdot|\mathcal{A}|}{\beta}\sum_{t=1}^{T}(\mathbb{E}\left[V^t\right]-\mathbb{E}\left[V^{t+1}\right])+3\beta T(4t_{\text{mix}}^*+M)^2\\
&=\frac{|\mathcal{S}|\cdot|\mathcal{A}|}{\beta}\left(\mathbb{E}\left[V^1\right]-\mathbb{E}\left[V^{T+1}\right]\right)+3\beta T(4t_{\text{mix}}^*+M)^2\\
&\le\frac{|\mathcal{S}|\cdot|\mathcal{A}|}{\beta}\left(\log(|\mathcal{S}|\cdot|\mathcal{A}|)+\frac{2(t_{\text{mix}}^*)^2}{(4t_{\text{mix}}^*+M)^2}\right)\\
&\quad+3\beta T(4t_{\text{mix}}^*+M)^2.
\end{aligned}$$

By taking

$$\beta=\frac{1}{4t_{\text{mix}}^*+M}\sqrt{\frac{|\mathcal{S}|\cdot|\mathcal{A}|\cdot\log(|\mathcal{S}|\cdot|\mathcal{A}|)}{2T}},$$

we obtain

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}W^t\right]=\tilde{O}\left((4t_{\text{mix}}^*+M)\sqrt{\frac{|\mathcal{S}|\cdot|\mathcal{A}|}{T}}\right),$$

as desired. ■

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA, USA: MIT press, 2018.

[2] M. Chu, H. Li, X. Liao, and S. Cui, "Reinforcement learning-based multiaccess control and battery prediction with energy harvesting in IoT systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2009–2020, April 2019.

[3] N. Jiang, Y. Deng, A. Nallanathan, and J. A. Chambers, "Reinforcement learning for real-time optimization in NB-IoT networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1424–1440, June 2019.

[4] Y. Xu, W. Xu, Z. Wang, J. Lin, and S. Cui, "Load balancing for ultra-dense networks: A deep reinforcement learning based approach," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9399–9412, December 2019.

[5] H. Ye, G. Y. Li, and B. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, April 2019.

[6] X. Zhang, M. Peng, S. Yan, and Y. Sun, "Deep reinforcement learning based mode selection and resource allocation for cellular V2X communications," *IEEE Internet Things J.*, December 2019, to appear.

[7] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11 158–11 168, November 2019.

[8] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, Sep. 2018.

[9] R. Ghanavi, E. Kalantari, M. Sabbaghian, H. Yanikomeroglu, and A. Yongacoglu, "Efficient 3D aerial base station placement considering users mobility by reinforcement learning," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Barcelona, Spain, April 2018, pp. 1–6.

[10] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[11] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, Fourthquarter 2019.

[12] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C. Lin, "Edge of things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment," *IEEE Access*, vol. 6, pp. 1706–1717, December 2018.

[13] W. Jiang, G. Feng, S. Qin, T. S. P. Yum, and G. Cao, "Multi-agent reinforcement learning for efficient content caching in mobile D2D networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1610–1622, March 2019.

[14] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, October 2019.

[15] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, February 2020.

[16] J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, Barcelona, Spain, December 2016, pp. 2137–2145.

[17] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *International Conference on Autonomous Agents and Multiagent Systems (AAAMS)*, São Paulo, Brazil, May 2017, pp. 66–83.

[18] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA, December 2017, pp. 6379–6390.

[19] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian, "Deep decentralized multi-task multi-agent reinforcement learning under partial observability," in *International Conference on Machine Learning (ICML)*, Sydney, NSW, Australia, August 2017, pp. 2681–2690.

[20] J. Foerster, N. Nardelli, G. Farquhar, T. Afouras, P. Torr, P. Kohli, and S. Whiteson, "Stabilising experience replay for deep multi-agent reinforcement learning," in *International Conference on Machine Learning (ICML)*, Sydney, Australia, August 2017, pp. 1146–1155.

[21] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *International Conference on Machine Learning (ICML)*, Stockholm, Sweden, July 2018, pp. 9340–9371.

[22] 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.

[23] H.-T. Wai, Z. Yang, Z. Wang, and M. Hong, "Multi-agent reinforcement learning via double averaging primal-dual optimization," in *Advances in Neural Information Processing Systems (NeurIPS)*, Montréal, Canada, December 2018, pp. 9649–9660.

[24] D. Lee, H. Yoon, and N. Hovakimyan, "Primal-dual algorithm for distributed reinforcement learning: distributed GTD," in *IEEE Conference on Decision and Control (CDC)*, Miami Beach, USA, December 2018, pp. 1967–1972.

[25] J. Yang, Y. Li, H. Chen, and J. Li, "Average reward reinforcement learning for semi-Markov decision processes," in *International Conference on Neural Information Processing*, Guangzhou, China, November 2017, pp. 768–777.

[26] S. Yang, Y. Gao, B. An, H. Wang, and X. Chen, "Efficient average reward reinforcement learning using constant shifting values," in *AAAI Conference on Artificial Intelligence (AAAI)*, Phoenix, Arizona, July 2016, p. 2258–2264.

[27] M. Ghavamzadeh and S. Mahadevan, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *J. Mach. Learn. Res.*, vol. 8, no. 1, p. 2629–2669, December 2007.

[28] A. Mathkar and V. S. Borkar, "Distributed reinforcement learning via gossip," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1465–1470, March 2017.

[29] L. Kraemer and B. Banerjee, "Multi-agent reinforcement learning as a rehearsal for decentralized planning," *Neurocomputing*, vol. 190, no. 1, pp. 82–94, May 2016.

[30] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, pp. 1–14, March 2020, to appear.

[31] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Auton. Agent. Multi. Agent. Syst.*, vol. 11, no. 3, pp. 387–434, November 2005.

[32] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *International Conference on Machine Learning (ICML)*, New Brunswick, NJ, USA, July 1994, pp. 157–163.

[33] ——, "Friend-or-foe Q-learning in general-sum games," in *International Conference on Machine Learning (ICML)*, Williamstown, MA, USA, July 2001, pp. 322–328.

[34] Y. Yang, R. Luo, M. Li, M. Zhou, W. Zhang, and J. Wang, "Mean field multi-agent reinforcement learning," in *International Conference on Machine Learning (ICML)*, Stockholm, Sweden, July 2018, pp. 5571–5580.

[35] G. Arslan and S. Yüksel, "Decentralized Q-learning for stochastic teams and games," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 1545–1558, April 2017.

[36] K. G. Vamvoudakis, H. Modares, B. Kiumarsi, and F. L. Lewis, "Game theory-based control system algorithms with real-time reinforcement learning: How to solve multiplayer games online," *IEEE Control Syst. Mag.*, vol. 37, no. 1, pp. 33–52, February 2017.

[37] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to h-infinity control," *Automatica*, vol. 43, no. 3, p. 473–481, January 2007.

[38] J.-H. Kim and F. L. Lewis, "Model-free h-infinity control design for unknown linear discrete-time systems via Q-learning with LMI," *Automatica*, vol. 46, no. 8, p. 1320–1326, August 2010.

[39] M. Wang, "Primal-dual π learning: Sample complexity and sublinear run time for ergodic Markov decision problems," *arXiv preprint:1710.06100*, October 2017. [Online]. Available: https://arxiv.org/abs/1710.06100

[40] Y. Chen, L. Li, and M. Wang, "Scalable bilinear π learning using state and action features," in *International Conference on Machine Learning (ICML)*, Stockholm, Sweden, July 2018, pp. 834–843.

[41] B. Dai, N. He, Y. Pan, B. Boots, and L. Song, "Learning from conditional distributions via dual embeddings," in *Artificial Intelligence and Statistics (AISTATS)*, Florida, USA, April 2017, pp. 1458–1467.

[42] S. S. Du, J. Chen, L. Li, L. Xiao, and D. Zhou, "Stochastic variance reduction methods for policy evaluation," in *International Conference on Machine Learning (ICML)*, Sydney, Australia, August 2017, pp. 1049–1058.

[43] B. Aygun and A. M. Wyglinski, "A voting-based distributed cooperative spectrum sensing strategy for connected vehicles," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5109–5121, June 2017.

[44] S. M. Nam and T. H. Cho, "Context-aware architecture for probabilistic voting-based filtering scheme in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2751–2763, October 2017.

[45] N. Katenka, E. Levina, and G. Michailidis, "Local vote decision fusion for target detection in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 329–338, January 2008.

[46] I. Partalas, I. Feneris, and I. Vlahavas, "Multi-agent reinforcement learning using strategies and voting," in *IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, Patras, Greece, January 2007, pp. 318–324.

[47] T. G. Dietterich, M. A. Taleghan, and M. Crowley, "PAC optimal planning for invasive species management: Improved exploration for reinforcement learning from simulator-defined MDPs," in *AAAI Conference on Artificial Intelligence (AAAI)*, Bellevue, Washington, July 2013.

[48] M. A. Taleghan, T. G. Dietterich, M. Crowley, K. Hall, and H. J. Albers, "PAC optimal MDP planning with application to invasive species management," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 3877–3903, January 2015.

[49] M. G. Azar, R. Munos, and H. J. Kappen, "Minimax PAC bounds on the sample complexity of reinforcement learning with a generative model," *Mach. Learn.*, vol. 91, no. 3, pp. 325–349, June 2013.

[50] M. Kearns, Y. Mansour, and A. Y. Ng, "A sparse sampling algorithm for near-optimal planning in large Markov decision processes," *Mach. Learn.*, vol. 49, no. 2-3, pp. 193–208, November 2002.

[51] M. J. Kearns and S. P. Singh, "Finite-sample convergence rates for Q-learning and indirect algorithms," in *Advances in Neural Information Processing Systems (NeurIPS)*, Denver, CO, December 1999, pp. 996–1002.

[52] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. Hoboken, New Jersey: John Wiley & Sons, 2014.

[53] D. P. Bertsekas, *Dynamic programming and optimal control*. Belmont, MA, USA: Athena scientific, 2005.

[54] A. Adam and M. White, "Investigating practical linear temporal difference learning," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Singapore, May 2016, pp. 494–502.

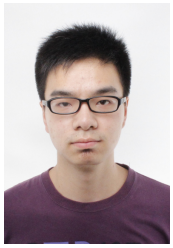
[55] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2334–2360, thirdquarter 2019.

[56] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, "Placement optimization of UAV-mounted mobile base stations," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 604–607, March 2017.

[57] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, May 2017.

[58] Y. Sun, T. Wang, and S. Wang, "Location optimization for unmanned aerial vehicles assisted mobile networks," in *IEEE International Conference on Communications (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.
[59] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Commun. Lett.*, vol. 6, no. 4, pp. 434–437, August 2017.
[60] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular networks," in *IEEE International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–5.
[61] A. Merwaday and I. Guvenc, "UAV assisted heterogeneous networks for public safety communications," in *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, New Orleans, LA, USA, March 2015, pp. 329–334.
[62] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, "A survey of mobility models for ad hoc network research," *Wireless Communications and Mobile Computing*, vol. 2, no. 5, pp. 483–502, August 2002.
[63] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected UAVs: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2125–2140, March 2019.

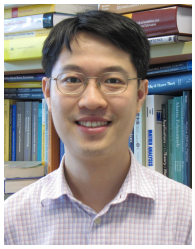
Yue Xu (S'19) received his B.S. and Ph.D. degree from Beijing University of Post and Telecommunication (BUPT) in 2020. He has been a Visiting Researcher with University of California, Davis, USA, The Chinese University of Hong Kong, Shenzhen, China, and Shenzhen Research Institute of Big Data. He is currently a research scientist at Alibaba Group. His research interests include data-driven wireless network management, machine learning, large-scale data analytics and system control.



Zengde Deng is a Ph.D. candidate in the Department of Systems Engineering and Engineering Management at The Chinese University of Hong Kong, Hong Kong, China. He received a B.S degree in statistics from Nankai University, Tianjin, China. His research interests are designing efficient algorithms for convex and nonconvex constrained optimization problems that arise from machine learning, statistics, and signal processing areas.

Mengdi Wang is an associate professor at the Department of Electrical Engineering and Center for Statistics and Machine Learning at Princeton University. She is also affiliated with the Department of Operations Research and Financial Engineering and Department of Computer Science. Her research focuses on data-driven stochastic optimization and applications in machine and reinforcement learning. She received her PhD in Electrical Engineering and Computer Science from Massachusetts Institute of Technology in 2013. At MIT, Mengdi was affiliated with the Laboratory for Information and Decision Systems and was advised by Dimitri P. Bertsekas. Mengdi received the Young Researcher Prize in Continuous Optimization of the Mathematical Optimization Society in 2016 (awarded once every three years), the Princeton SEAS Innovation Award in 2016, the NSF Career Award in 2017, the Google Faculty Award in 2017, and the MIT Tech Review 35-Under-35 Innovation Award (China region) in 2018. She serves as an associate editor for Operations Research and Mathematics of Operations Research, as area chair for ICML, NeurIPS, AISTATS, and is on the editorial board of Journal of Machine Learning Research. Research supported by NSF, AFOSR, Google, Microsoft C3.ai DTI, FinUP.

Wenjun Xu (M'10-SM'18) is a professor and Ph.D. supervisor in School of Information and Communication Engineering at Beijing University of Posts and Telecommunications (BUPT), Beijing, China. He received his B.S. and Ph.D. degrees from BUPT, in 2003 and 2008, respectively. He currently serves as a center director of the Key Laboratory of Universal Wireless Communications, Ministry of Education, P. R. China. He is a senior member of IEEE, and is now an Editor for China Communications. His research interests include AI-driven networks, UAV communications and networks, green communications and networking, and cognitive radio networks.



Anthony Man-Cho So received the BSE degree in Computer Science from Princeton University, Princeton, NJ, USA, with minors in Applied and Computational Mathematics, Engineering and Management Systems, and German Language and Culture. He then received the M.Sc. degree in Computer Science and the Ph.D. degree in Computer Science with a Ph.D. minor in Mathematics from Stanford University, Stanford, CA, USA.

Dr. So joined The Chinese University of Hong Kong (CUHK) in 2007. He is now the Associate Dean of Student Affairs in the Faculty of Engineering, Deputy Master of Morningside College, and Professor in the Department of Systems Engineering and Engineering Management. His research focuses on optimization theory and its applications in various areas of science and engineering, including computational geometry, machine learning, signal processing, and statistics.

Dr. So is appointed as an Outstanding Fellow of the Faculty of Engineering at CUHK in 2019. He has received a number of research and teaching awards, including the 2018 IEEE Signal Processing Society Best Paper Award, the 2015 IEEE Signal Processing Society Signal Processing Magazine Best Paper Award, the 2014 IEEE Communications Society Asia-Pacific Outstanding Paper Award, the 2013 CUHK Vice-Chancellor's Exemplary Teaching Award, and the 2010 Institute for Operations Research and the Management Sciences (INFORMS) Optimization Society Optimization Prize for Young Researchers. He currently serves on the editorial boards of Journal of Global Optimization, Optimization Methods and Software, and SIAM Journal on Optimization.

Shuguang Cui (S'99-M'05-SM'12-F'14) received his Ph.D in Electrical Engineering from Stanford University, California, USA, in 2005. Afterwards, he has been working as assistant, associate, full, Chair Professor in Electrical and Computer Engineering at the Univ. of Arizona, Texas A&M University, UC Davis, and CUHK at Shenzhen respectively. He has also been the Executive Vice Director at Shenzhen Research Institute of Big Data. His current research interests focus on data driven large-scale system control and resource management, large data set analysis, IoT system design, energy harvesting based communication system design, and cognitive network optimization. He was selected as the Thomson Reuters Highly Cited Researcher and listed in the Worlds' Most Influential Scientific Minds by ScienceWatch in 2014. He was the recipient of the IEEE Signal Processing Society 2012 Best Paper Award. He has served as the general co-chair and TPC co-chairs for many IEEE conferences. He has also been serving as the area editor for IEEE Signal Processing Magazine, and associate editors for IEEE Transactions on Big Data, IEEE Transactions on Signal Processing, IEEE JSAC Series on Green Communications and Networking, and IEEE Transactions on Wireless Communications. He has been the elected member for IEEE Signal Processing Society SPCOM Technical Committee (2009 2014) and the elected Chair for IEEE ComSoc Wireless Technical Committee (2017 2018). He is a member of the Steering Committee for IEEE Transactions on Big Data and the Chair of the Steering Committee for IEEE Transactions on Cognitive Communications and Networking. He was also a member of the IEEE ComSoc Emerging Technology Committee. He was elected as an IEEE Fellow in 2013, an IEEE ComSoc Distinguished Lecturer in 2014, and IEEE VT Society Distinguished Lecturer in 2019.