Multi-Agent Reinforcement Learning based Uplink OFDMA for IEEE 802.11ax Networks

Mingqi Han*, *Student Member*, *IEEE*, Xinghua Sun*, *Member*, *IEEE*, Wen Zhan*, *Member*, *IEEE*, Yayu Gao†, *Member*, *IEEE*, Yuan Jiang*

***Abstract*—In the IEEE 802.11ax Wireless Local Area Networks (WLANs), Orthogonal Frequency Division Multiple Access (OFDMA) has been applied to enable the high-throughput WLAN amendment. However, with the growth of the number of devices, it is difficult for the Access Point (AP) to schedule uplink transmissions, which calls for an efficient access mechanism in the OFDMA uplink system. Based on Multi-Agent Proximal Policy Optimization (MAPPO), we propose a Mean-Field Multi-Agent Proximal Policy Optimization (MFMAPPO) algorithm to improve the throughput and guarantee the fairness. Motivated by the Mean-Field games (MFGs) theory, a novel global state and action design are proposed to ensure the convergence of MFMAPPO in the massive access scenario. The Multi-Critic Single-Policy (MCSP) architecture is deployed in the proposed MFMAPPO so that each agent can learn the optimal channel access strategy to improve the throughput while satisfying fairness requirement. Extensive simulation experiments are performed to show that the MFMAPPO algorithm 1) has low computational complexity that increases linearly with respect to the number of stations 2) achieves nearly optimal throughput and fairness performance in the massive access scenario, 3) can adapt to various diverse and dynamic traffic conditions without retraining, as well as the traffic condition different from training traffic.**

***Index Terms*—Multiple Access, Multi-Agent Reinforcement Learning, Multi-objective Reinforcement Learning, Mean-Field Reinforcement Learning**

I. Introduction

With the development of IEEE 802.11 standard, we have witnessed the global roll-out of Wi-Fi, where the network throughput requirement increases rapidly. Consequently, High Efficiency WLAN (HEW) was proposed in the amendment named IEEE 802.11ax [1], [2] to meet the increasing throughput requirement in dense scenarios, i.e., the stadium or shopping mall with a large number of audiences. In these scenarios, massive stations (STAs) are required to access limited channel resources, and each has high traffic dynamics with personalized service requirement.

In HEW IEEE 802.11ax [1], orthogonal frequency division multiple access (OFDMA) is a critical multiple access technique to enhance the uplink (UL) multi-user (MU) transmission efficiency. With the rapid growth of the number of devices, it is hard for the access point (AP) to acquire buffer status of all devices. Therefore, the Uplink OFDMA Random Access (UORA) has been proposed to allow devices with unknown buffer status to participate in the UL MU transmission via OFDMA [2]. As a random multiple access technique, the UORA enables each STA to randomly access a resource unit (RU) for the UL transmission. However, due to the randomness of accessing RUs, the collision probability is difficult to be reduced, resulting in low access efficiency as $1/e \approx 37\%$ for each RU even with the optimal parameters [3].

Recently, the Reinforcement Learning (RL) technique has achieved tremendous success in the wireless networks. By finding solutions through the experiences from iterations, RL-based algorithms can well adapt to the dynamic traffic environment. Regarding STAs as agents, the multiple access problem can naturally be formulated as a multi-agent problem. Consequently, the Multi-Agent Reinforcement Learning (MARL) technique has been widely adopted in recent literature, which allows devices to learn access strategies cooperatively to avoid collisions between devices and improve the throughput while guaranteeing fairness [4]–[17].

Motivated by this, we aim to propose a MARL based multiple access algorithm to replace the inefficient UORA mechanism. Due to the non-stationary issue introduced by the multi-agent environment, it is hard for each agent to learn an independent efficient access strategy based on its own partial observation. Therefore, it is a common way to improve the MARL performance by centralized training that jointly uses the states and actions of other agents. However, with the growth of the agent scale, the size of state space when using this approach increases rapidly, resulting in high computational complexity for MARL models in the massive access scenario. Therefore, most MARL algorithms consider only a few agents and become difficult to solve in large-scale multi-agent scenarios [8]–[13].

To address this issue, we introduce an action mechanism, with which each agent independently determines its access policy periodically every certain time slots instead of each time slot. Such action mechanism expands the action space to avoid excessive collisions and converging to unexpected access strategies. Based on this action mechanism, a novel global

The work of Xinghua Sun was supported in part by the National Key Research and Development Program of China under Grant 2023YFB2904100, in part by Shenzhen Science and Technology Program (Grant No. ZDSYS20210623091807023), and in part by the Key-Area Research and Development Program of Guangdong Province under Grant 2020B0101120003. The work of Wen Zhan was supported in part by The Shenzhen Science and Technology Program (No.RCBS20210706092408010), in part by National Natural Science Foundation of China under Grant 62001524. The work of Yayu Gao was supported by National Natural Science Foundation of China under Grant 61972172. (Corresponding author: Xinghua Sun)

* M. Han, X. Sun, W. Zhan and Y. Jiang are with the School of Electronics and Communication Engineering, Shenzhen Campus of Sun Yat-sen University, Shenzhen 518107, China (e-mail:hanmq@mail2.sysu.edu.cn; xsunxinghua@mail.sysu.edu.cn; zhanw6@mail.sysu.edu.cn; jiangyuan3@mail.sysu.edu.cn).

† Y. Gao is with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China (e-mail:yayugao@hust.edu.cn).

state is then designed to reduce the size of the global state. Motivated by the Mean-Field Evolutionary (MFE) approach [18], a global state is introduced to consider the "average" action of other agents rather than the individual action of each agent. Such design simplifies the complex interaction among massive agents, and ensures that the proposed algorithm can be applied in the massive access scenario.

In face of personalized service requirements, there always exists tradeoff between throughput and fairness in multiple access scenarios [19]. In particular, high throughput requires STAs with good transmission conditions to occupy channel resources, which jeopardizes transmissions of other STAs and thus the fairness performance. Concerning about both the throughput and the fairness, we aim to maximize the throughput while guaranteeing the fairness among agents. Toward this goal, we apply the Multi-objective Reinforcement Learning (MORL) technique through the Multi-Critic Single Policy (MCSP) architecture.

A. Contributions and Main Results

In this paper, we propose a multi-agent distributed access algorithm based on Multi-Agent Proximal Policy Optimization (MAPPO) [20] named Mean-Field MAPPO (MFMAPPO), aiming at maximizing the throughput while guaranteeing the fairness among STAs in the OFDMA uplink network. Our contributions and main results are highlighted as follows:

- We design a new action mechanism for the proposed MFMAPPO to avoid converging to unexpected local optimum. In this design, we expand the action space over time to tackle the convergence problem caused by the high collision probability with the increase of the number of STAs. Moreover, based on such expanded action space, a novel global state motivated by the MFE is introduced to reduce the input size of global states. By considering the average behavior of other agents instead of their individual behaviors, the computational complexity is greatly reduced, increasing linearly only with the number of agents.
- We further extend the MCSP framework in the MARL scenario to improve both the throughput and the fairness performance of the MFMAPPO [21]. Utilizing the extended MCSP, the MFMAPPO can evaluate value functions of different objectives by independent global states and critic networks, which enables agents to learn better cooperation strategies. Moreover, we adopt the *Pop-Art* layer in the extended MCSP to address the issue of unstable advantage function in multi-objective problem.
- We evaluate the performance of the proposed MFMAPPO algorithm in dynamic and diverse traffic scenarios where the traffic changes dynamically over time and varies across STAs. In both scenarios, the proposed MFMAPPO algorithm can adapt to the traffic variation over time without retraining and achieves nearly optimal performance compared with hybrid UORA (H-UORA) as well as other baseline methods. Moreover, the MFMAPPO has generalization capacity to adapt to traffic conditions that are different from those in training traffic.

B. Related work

The concept of UORA was first proposed in the IEEE 802.11ax to enable random access in the UL MU transmission via OFDMA. In UORA, each STA is informed with the information of RUs via a Trigger Frame for Random access (TF-R). Using the information of TF-R, each STA engages in an OFDMA backoff (OBO) process to compete for access the RU. When traffic loads heavily, UORA is shown to have a maximum normalized throughput similar to that of slotted Aloha due to high probability of collisions [3]. To improve the access efficiency, carrier sensing technique was adopted. In particular, a novel trigger based access mechanism called H-UORA was proposed to reduce the collision possibility by adopting carrier sensing [22]. Instead of attempting to avoid collisions, there are some schemes aiming at resolving collisions, for instance, by using Successive Inference Cancellation (SIC) [23], [24]. Although collision-resolution schemes can significantly improve the access efficiency, they usually come with a high computational complexity for the receiver, and signalling overhead may incur due to the requirement of the channel station information.

Recently, RL-based multiple access algorithms have gained great research interests. In [25], a Deep Q-Network (DQN) algorithm was introduced to optimize the contention window size for networks based on CSMA with Collision Avoidance (CSMA/CA). A deep Q-Learning based algorithm was proposed to control the transmission rates of nodes by adjusting the modulation and coding scheme (MCS) levels in CSMA/CA-based wireless networks [26]. For efficient coexistence of LTE-LAA and WiFi, a RL-enabled Listen-Before-Talk (ReLBT) mechanism aimed to optimize the channel access parameters for LBT in [27]. Instead of designing on the top of current protocols, some studies proposed to replace the inefficient channel access mechanism with RL-based ones [10], [14]. In particular, to address a dynamic spectrum sharing problem, a RL-based algorithm was developed in [14] to select channels and demonstrated a better performance than existing access methods. In [10], the DQN was applied to address the multiple access problem in a time-varying environment.

While the aforementioned studies focusing on the Single Agent Reinforcement Learning (SARL) problems, researchers have also adopted the MARL technique in the multiple access scenario to improve various performance metrics through better coordination among agents [4]–[17]. The MARL algorithms can be roughly divided into two categories, including the policy-based and the value-based. For the policy-based MARL, the Multi Agent Deep Deterministic Policy Gradient (MADDPG) and Multi-Agent Deep Stochastic Policy Gradient (MADPSG) algorithms were applied in dynamic spectrum access problem to maximize the average sum event rate, which outperforms the standard multiple access protocols [4]. In a similar vein, a deep actor-critic MARL-based framework was proposed for the dynamic multi-channel access problem [8].

For the value-based MARL, the DQN based Deep Q-learning Spectrum Access (DQSA) was applied to address the dynamic spectrum access problem for network utility maximization in multi-channel wireless networks [7]. Similar

to DQSA, a multiple access scheme based on QMIX was proposed and shown to outperform the Distributed Channel Access (DCA) problem, which outperforms the CSMA/CA and shows robustness in the dynamic network environment [11]. The deep Q-learning was adopted to maximize the throughput in heterogeneous networks and homogeneous networks respectively in [15] and [17]. Moreover, value-based MARL algorithms have been applied in dynamic multi-channel access scenarios to learn the unknown wireless environment and the corresponding channel access strategies in [9]–[13], [16].

In the massive access scenario, MARL-based approaches face several challenges: 1) with the growth of access scale, the computational complexity of MARL models increases rapidly, especially when each agent utilizes the information of other agents [4], [11], [28]–[30]; 2) due to the conflict between access efficiency and fairness issue, it is difficult to guarantee the fairness among STAs while improving the throughput [31]; 3) when each agent needs to access at every slot, the collision probability increases with respect to the number of agents during the exploration of the state-action space. As such, the RL model would converge to unexpected local optimum, which renders it difficult to apply in the massive access scenario [8]–[13]. In the proposed MFMAPPO, we introduce several approaches to address the aforementioned issues.

First, by considering the averaged behaviour of other agents rather than their individual behaviors, the MFE approach can reduce the complexity of the MARL algorithm, which can be applied to the scenario with large number of agents [18], [32]. However, the MFE approach requires a continuous objective function and continuous gradient matrix to update the distribution of strategies, which cannot be applied in the considered multiple access problem with discrete utility, i.e., the number of successful transmissions. In this paper, we design a novel global state motivated by the MFE approach in the proposed MFMAPPO algorithm to reduce the computational complexity and improve convergence performance in the massive access scenario.

Second, to address the multi-objective optimization problem concerning about both the throughput and the fairness, we do not simply summate rewards of different objectives into a joint reward as that in Single-Object Reinforcement Learning (SORL), since such approach will flatten the gradient of critic networks during training, resulting in slower and unstable convergence [31]. Instead, we apply the MCSP to evaluate state values of multiple objectives through different critic networks, which improves the performance of RL in multi-objective optimization problems [21]. Yet, the vanilla MCSP is for the SARL scenario, which is not suitable in the considered multiple access scenario. In the proposed MFMAPPO, we extend the vanilla MCSP into the MARL scenario, and improve its evaluation ability and convergence performance. In particular, we introduce two global states for different objectives to not only provide global information but also further reduce the computational complexity. Then, we also propose separate network architecture instead of the shared network in vanilla MCSP to address the unequal length between global states and local state, which further improves the evaluation ability.

Finally, to deal with the problem of sharing few channel resources among massive STAs, we take the selection of time slot as a part of the action design, in which STAs decide the transmission strategy periodically every certain time slots. Germane to our work, a branching dueling Q-network with such action mechanism was adopted in [16], where the dueling Q-network deterministically selects a set of channel access policies for several consecutive time slots between each decision time to tackle the multi-channel access problem in dynamic network environment. Moreover, a vector Q-learning scheme was proposed to reduce the computational complexity from an exponential increment to a linear one with both the number of STAs and the number of channels [16]. In the proposed MFMAPPO, the computational complexity is further reduced to $\mathcal{O}(N)$ where N denotes the number of STAs.

C. Outline

The remainder of this paper is organized as follows: Section II introduces the Proximal Policy Optimization (PPO) and the CTDE architecture. Section III presents the system model and the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) model formulation in the OFDMA uplink random access scenario. In Section IV, the proposed MFMAPPO algorithm is elaborated. Simulation results are presented in Section V and the conclusion and future of work are provided in Section VI.

II. Reinforcement Learning Preliminaries

A. RL: A PPO-Based Approach

In the SARL, PPO is one of the most popular RL algorithms. In the PPO, there are several main techniques applied in the loss formula to improve the convergence performance [33]. First, the importance sampling is applied to improve the efficiency and stability of the policy gradient estimation. In on-policy RL algorithms, the policy is updated based on the data collected by the same policy. However, due to the stochastic nature of the environment, the same policy can generate different trajectories, leading to a high variance in the policy gradient estimation. In importance sampling, instead of directly updating the old policy $\pi_{\boldsymbol{\theta}_{\text{old}}}\left(a_t \mid s_t\right)$, a new policy $\pi_{\boldsymbol{\theta}}\left(a_t \mid s_t\right)$ will be generated for the update. The ratio between the new policy and old policy $r(\boldsymbol{\theta})$ is defined as

$$r(\boldsymbol{\theta}) \triangleq \frac{\pi_{\boldsymbol{\theta}}\left(\boldsymbol{a_t} \mid \boldsymbol{s_t}\right)}{\pi_{\boldsymbol{\theta}_{\text{old}}}\left(\boldsymbol{a_t} \mid \boldsymbol{s_t}\right)}. \tag{1}$$

Then, the loss can be re-weighted to reduce the variance of policy gradient estimation by applying the ratio $r(\boldsymbol{\theta})$ to the advantage function A_t, which is given by

$$L(\boldsymbol{\theta}) = \widehat{\mathbb{E}}_t\left[r(\boldsymbol{\theta}) A_t\right] + \left[\sigma \frac{1}{B} \sum_{t=1}^{B} S\left[\pi_{\boldsymbol{\theta}}\left(\boldsymbol{a_t} \mid \boldsymbol{s_t}\right)\right]\right], \tag{2}$$

where $S\left[\pi_{\boldsymbol{\theta}}\left(\boldsymbol{a_t} \mid \boldsymbol{s_t}\right)\right]$ is defined as the entropy of the new policy $\pi_{\boldsymbol{\theta}}\left(\boldsymbol{a_t} \mid \boldsymbol{s_t}\right)$ and σ denotes the entropy coefficient. The advantage function A_t represents the advantages of each action

compared with the current policy ϕ in a specific state s_t, which is given by

$$A_t = \sum_{l=1}^{\infty} (\gamma\lambda)^l \left(r_t + \gamma V_\phi\left(\boldsymbol{s_t}\right) - V_\phi\left(\boldsymbol{s_{t+l}}\right)\right) \tag{3}$$

When $r(\boldsymbol{\theta}) \gg 1$, nevertheless, the loss becomes excessively large, which leads to unstable convergence. Therefore, the clipped loss formula is applied in the PPO to address this issue by constraining the scale of loss, i.e.,

$$\begin{aligned} L(\boldsymbol{\theta}) =& \widehat{\mathbb{E}}_t \left[\min\left(r_t(\boldsymbol{\theta})A_t, \operatorname{clip}\left(r_t(\boldsymbol{\theta}), 1-\varepsilon, 1+\varepsilon\right) A_t\right)\right] \\ &+ \left[\sigma \frac{1}{B} \sum_{t=1}^{B} S\left[\pi_{\boldsymbol{\theta}}\left(\boldsymbol{a}_t \mid \boldsymbol{s}_t\right)\right]\right] \end{aligned} \tag{4}$$

where clip is used to circumstance the ratio $r(\boldsymbol{\theta})$ within the bounds of $[1-\varepsilon, 1+\varepsilon]$. If $A_t > 0$, then the maximum value of $\operatorname{clip}\left(r_t(\boldsymbol{\theta}), 1-\varepsilon, 1+\varepsilon\right) A_t$ is given by $(1+\varepsilon)A_t$, otherwise, it is given by $(1-\varepsilon)A_t$.

In this work, we adopt the PPO-based approach to address the multiple access problem. The PPO-based approach can offer better exploration capacity through the clipped surrogate loss and higher learning efficiency through importance sampling. Moreover, PPO can also be extended to MAPPO by introducing a global state to gather information of the environment and other agents, which enables better cooperation among agents. Compared to other policy-based approaches, e.g., MADDPG, with continuous action space, the PPO-based approach with discrete action space is more suitable for the multiple access problem.

B. CTDE

In the MARL algorithm, similar to SARL, each agent interacts with the environment and other agents to maximize its own cumulative rewards based on its local observation. With respect to SARL, the environment in MARL becomes non-stationary and more complex due to the unfixed policy of each agent during training, which makes the MARL problem difficult to solve [34].

The framework of MARL-based algorithms can be roughly divided into two categories, including centralized MARL and decentralized MARL. Considering a cooperative environment, the advantage of the centralized MARL is that it can gather the information of all agents and assign the reward to each agent, which makes MARL models easier to converge and avoids the problem of designing individual reward for each agent. In the centralized MARL, nevertheless, agents needs the information of other agents to make actions, which makes it difficult to be deployed in a distributed manner. For decentralized MARL, though each agent can make action independently, the information of part of environment and other agents is unknown to the agent, which makes it hard to converge to the global optimum.

The CTDE architecture has been proposed to take advantages of the aforementioned frameworks. In CTDE, the training of the model is done centralized by using the training data collected from virtual environments (through simulations in this work). The centralized training process enables each agent to share information, facilitating the acquisition of a better coordination strategy. Once the model is trained, it is then deployed to each agents for execution. Most of the mainstream MARL methods have adopted the CTDE architecture, e.g., MADDPG, QMIX and Counterfactual Multi-Agent Policy Gradients (COMA) [28]–[30]. Recently, the MAPPO algorithm has been proposed to deal with MARL problems, which is derived from the PPO algorithm [20]. Through the aforementioned CTDE architecture, each MAPPO agent i has its own actor network and critic network to evaluate its state value V^i based on the global information of all agents, and output its own action based on local observation.

III. System Model and Dec-POMDP Formulation

A. System Model

As Fig. 1 illustrates, we consider an OFDMA uplink scenario where N STAs transmit packets to an associated AP in a time-slotted 802.11ax network through M available RUs. Each STA has a buffer to store incoming packets, and each packet is served in a First Come First Serve (FCFS) manner.

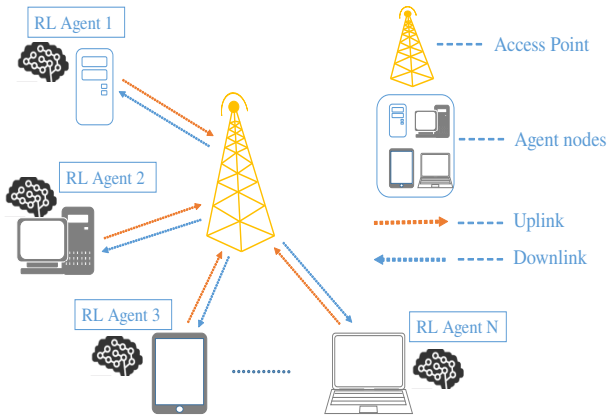


Fig. 1. Multiple STAs share several RUs and transmit in a distributed manner.

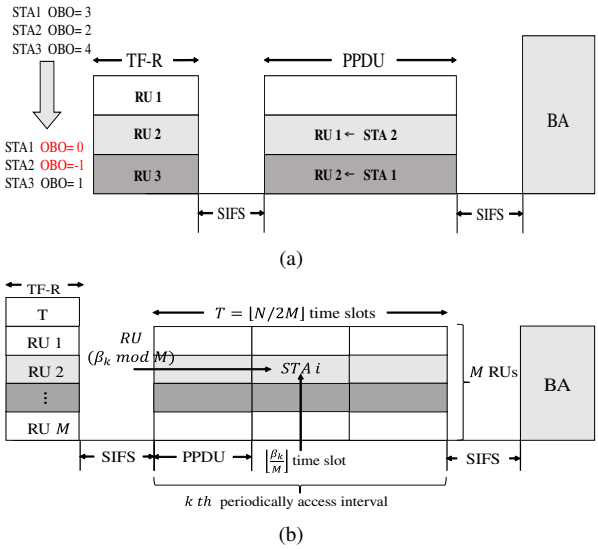


Fig. 2. (a) 802.11ax UL OFDMA-based random access. (b) access procedure based on the proposed MFMAPPO.

In UORA, the AP allows STAs to randomly access the RUs as shown in Fig. 2a. The OBO mechanism is used to reduce the collision probability, which works as follows: the AP first broadcasts the TF-R to each STA, which declares the start of the access and includes information about RUs. Each STA generates a random OBO counter independently, which is then decreased by the number of RUs indicated by the received TF-R. When the OBO counter decreases to a non-positive value, the STA is allowed to transmit a Presentation Protocol Data Unit (PPDU) on one RU after a Short InterFrame Space (SIFS) time. After the transmission, the AP broadcasts a Block Acknowledgement (BA) to each STA to notify them of the transmission results.

However, even with the optimal parameters of OBO mechanism, the maximum access efficiency of UORA is equal to that of slotted Aloha, i.e., $1/e \approx 37\%$. In this work, our goal is to design a distributed multiple access policy based on MARL that achieves better throughput performance and guarantees the fairness among STAs. The access procedure of the proposed MFMAPPO is illustrated in Fig. 2b. In particular, the TF-R is broadcast from AP to each STA to notify the value of T and the informations of M RUs, where T is the number of time slots of each periodical access interval. Each STA then selects which RU and time slot to access at the beginning of periodical access interval. Compared to UORA with which STAs can only choose which RU to access, in the proposed MFMAPPO, each STA needs to further select the time slot to access in each periodical access interval.

B. Dec-PODMP Model Formulation

The OFDMA random access problem can be regarded as the MARL problem where each STA interacts with the environment based on local observation. The MARL problem can be formulated as a Dec-POMDP model, which is given by

$$\langle \mathcal{I}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \mathcal{Z}, \mathcal{O}, N, \gamma \rangle, \tag{5}$$

where $\mathcal{I} = \{1, 2, \ldots, N\}$ is the set comprises all N agents, and $s \in \mathcal{S}$ denotes the state of the environment. At each time step t, agent $i \in \mathcal{I}$ choose an action $a_t^i \in \mathcal{A}$ independently. The environment will change to the next state s_{t+1} according to the transition matrix $\mathcal{P} : \mathcal{S} \times \mathcal{A}^N \times \mathcal{S} \mapsto [0, 1]$ by joint action $\boldsymbol{a} \in \mathcal{A}^n$. Then, each agent will receive a reward based on the reward function: $\mathcal{R} : \mathcal{S} \times \mathcal{A}^N \mapsto \mathbb{R}^N$. Considering a partially observable scenario, each agent has a different local observation $z \in \mathcal{Z}$, where the observation z is decided by observation function $\mathcal{O} : \mathcal{S} \times \mathcal{I} \mapsto \mathcal{Z}$. In the following, the detailed description of the state, action, reward and global state in the MFMAPPO will be presented.

B.1 Action

As Fig. 2b illustrates, instead of deciding the access action at every time slot, we let each agent, i.e., agent $i \in \{1, 2, \ldots, N\}$ decides its action $\boldsymbol{a_k^i}$ at the beginning of periodical access interval k. The number of time slots in each periodical access interval is given by $T = \lfloor \frac{N}{2M} \rfloor$. By implementing this mechanism, we expand the action space, which enables agents to thoroughly explore the state-action space even when the number of RUs is significantly less than the number of STAs. Furthermore, this mechanism prevents frequent collisions between STAs and convergence to an unexpected local optimum.

Given M available RUs, the action of each agent $i \in \{1, 2, \ldots, N\}$ at k-th periodical access interval is defined as a vector in one-hot form with length $MT+1$. Let β_k^i denotes the index of the maximum value in action vector $\boldsymbol{a_k^i}$. For instance, $\beta_k^i = 0$ represents that agent i does not access in this periodical access time interval while $\beta_k^i \in \{1, \ldots, MT\}$ represents that agent i accesses through RU $\left(\beta_k^i \bmod M\right)$ in $\left\lfloor \frac{\beta_k^i}{M} \right\rfloor$-$th$ time slot. As an example, when $M = 2$ and $T = 10$, $\beta_k^i = 13$ represents that agent i will access through RU 1 at 6-th time slot.

B.2 State

In the proposed MFMAPPO, the state $\boldsymbol{s_k^i}$ of agent i at k-th periodical access interval is given by

$$\boldsymbol{s_k^i} = \left\{ o_k^i, D_k^i, D_k^{-i}, l_k^i, \beta_{k-1}^i \right\}. \tag{6}$$

In (6), $o_k^i \in \{-1, 0, 1\}$ denotes the transmission result of agent i at periodical access interval $k-1$, where $o_k^i = 1$ represents a successful access, $o_k^i = 0$ represents that agent i does not access and $o_k^i = -1$ denotes a failed access. v_k^i is defined as the long-term throughput of agent i from the beginning to k-th periodical access interval and v_k^{-i} as the total long-term throughput of all other agents except agent i itself. To mitigate the instability caused by the growing values of v_k^i and v_k^{-i} over time, both v_k^i and v_k^{-i} are normalized as $D_k^i = \frac{v_k^i}{v_k^i + v_k^{-i}}$ and $D_k^{-i} = \frac{v_k^{-i}}{v_k^i + v_k^{-i}}$. Let l_k^i denote the number of periodical access intervals since its last successful access for agent i, which is given by

$$l_{k+1}^i = \begin{cases} 0, & \text{if agent } i \text{ accesses successfully} \\ l_k^i + 1, & \text{otherwise} \end{cases}. \tag{7}$$

We also include β_{k-1}^i in $\boldsymbol{s_k^i}$ to provide history action information, i.e., the selected RU and time slot in the last transmission interval $k-1$. Since o_k^i only indicates whether agent i accesses successfully or not at access interval k, including both β_{k-1}^i and o_k^i does not introduce redundancy.

In the considered multiple access scenario, the local information that can be observed by agent i includes the transmission result of itself, the transmission results of others and the local status of itself. The local state is thus defined as (6) to include all the information, i.e., o_k^i and β_{k-1}^i corresponding to the transmission results of agent i; D_k^{-i} corresponding to the transmission results of others, and D_k^i and l_k^i corresponding to the current status of agent i.

B.3 Reward

As our goal is to maximize the total throughput of the network while guaranteeing the fairness among agents, we formulate two reward functions accordingly. One is the throughput reward $r_{tho,k}$ for maximizing throughput. In particular, we define the throughput reward $r_{tho,k}^i$ of agent i at k-th periodical access interval as

$$r_{tho,k}^i = \begin{cases} 1, & \text{if agent } i \text{ accesses successfully} \\ 0, & \text{if agent } i \text{ does not access} \\ -1, & \text{if agent } i \text{ experiences a collision} \end{cases}. \tag{8}$$

The throughput reward $r^i_{tho,k}$ encourages successful access by assigning positive rewards and punishes collisions with other agents by assigning negative rewards.

The fairness reward $r^i_{fai,k}$, on the other hand, is further divided into two parts, i.e., we have $r^i_{fai,k} = r^i_{fai1,k} + r^i_{fai2,k}$, where $r^i_{fai1,k}$ and $r^i_{fai1,k}$ are given by

$$r^i_{fai1,k} = \begin{cases} 1, & \text{if } v^i_k < v^i_{k,mid} \text{ and } \beta^i_k \geq 1 \\ 1, & \text{if } v^i_k \geq v^i_{k,mid} \text{ and } \beta^i_k = 0 \\ -1, & \text{otherwise} \end{cases} \tag{9}$$

and

$$r^i_{fai2,k} = \begin{cases} 1/2, & \text{if } \beta^i_k = 0, \beta^i_{k-1} > 0 \\ 1/2, & \text{if } \beta^i_k > 0, \beta^i_{k-1} = 0 \\ -1/2, & \text{otherwise} \end{cases}, \tag{10}$$

respectively. In (9) and (10), $v^i_{k,mid}$ denotes the median throughput of agents that have the same packet arrival rate as agent i. The first part $r^i_{fai1,k}$ encourages agents with lower throughput to access, while those with higher throughput are discouraged from access. The second part $r^i_{fai2,k}$ imposes a mild penalty on agents that repeatedly choose either to access or not to access during the training process. This promotes fairness among agents by discouraging agents from dominating the channel.

To address the multi-objective optimization problem, we design two different rewards to evaluate corresponding objectives respectively. Rather than combining these competing rewards, which could result in being trapped in suboptimal solutions, we divide the rewards into two parts and adopt the MCSP framework to evaluate them separately [21]. As will be introduced in Section IV, by evaluating different state values using corresponding critic networks, we can reduce the bias and variance of state values estimation, which can improve the convergence performance.

B.4 Global State

In the CTDE architecture, agents can utilize global informations for training. Compared with traditional ways to define the global state by splicing each local state, we design two unique global states including $\boldsymbol{S^i_{tho,k}}$ and $\boldsymbol{S^i_{fai,k}}$ for different objectives and critic networks. Since the total throughput is related only to the agent's access behavior and not to their current status, we only introduce the actions in order to evaluate the throughput performance

$$\boldsymbol{S^i_{tho,k}} \triangleq \left\{ \boldsymbol{s^i_k}, \boldsymbol{a^{-i}_k} \right\}, \tag{11}$$

where $\boldsymbol{a^{-i}_k} = \sum_{j \neq i} \boldsymbol{a^j_k}$ represents the sum of actions in the form of one-hot vector of other STAs. Motivated by the MFE [18], the global state $\boldsymbol{S^i_{tho,k}}$ takes the sum of actions of others into consideration instead of the joint action of all the other agents, in which the dimension becomes independent of the number of agents.

On the other hand, since the fairness among agents only relates to their current throughput, we introduce only the throughput to complete the evaluation of fairness

$$\boldsymbol{S^i_{fai,k}} \triangleq \left\{ \boldsymbol{s^i_k}, v^0_k, v^1_k, \ldots, v^N_k \right\} \tag{12}$$

where v^i_k is the long-term throughput of agent i at k-th periodical access interval.

IV. MFMAPPO Algorithm

Based on the Dec-POMDP formulation, we propose the MFMAPPO algorithm in this section. The overall architecture is shown in Fig. 3, where experience $(\boldsymbol{s}, \boldsymbol{S_{tho}}, \boldsymbol{S_{fai}}, \boldsymbol{a}, \boldsymbol{r_{tho}}, \boldsymbol{r_{fai}})$ of each agent are jointly stored in the memory, which are then utilized for updating the network parameters in centralized training. In the execution stage, only the *Actor* network is required to work. The convergence of centralized training can be improved by utilizing the global information of environment and other agents.

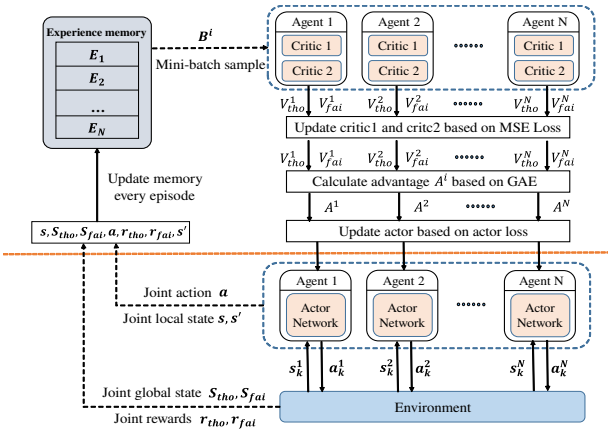


Fig. 3. The procedure of the MFMAPPO based on CTDE architecture: Centralized training is performed based on the experience reported by each agent. In the end of each episode, the *Critic* networks first send the state values of throughput and fairness to *Actor* for updating. Then, both *Critic* and *Actor* update parameters based on the loss function.

A. Loss Function

As Section B.3 presents, we partition the reward into r^i_{tho} and r^i_{fai}, and introduce two different critic networks to evaluate state values of the throughput and the fairness separately. For agent i, we define $\boldsymbol{\phi^i_1}$ and $\boldsymbol{\phi^i_2}$ as the parameters of *Critic 1* and *Critic 2*, respectively, and $\boldsymbol{\theta^i}$ is defined as the parameters of the *Actor*. The loss functions of *Critic 1* and *Critic 2* in the proposed MFMAPPO algorithm have similar forms as those in the MAPPO [20]:, which are given by

$$\begin{cases} L\left(\phi^i_1\right) = \frac{1}{|\boldsymbol{B^i}|} \sum_j \left(\max \left[\left(V^i_{\phi^i_1}\left(\boldsymbol{S^i_{tho,j}}\right) - \hat{R}^i_{tho,j} \right)^2, \right. \right. \\ \left. \left. \left(\text{clip}\left(V^i_{\phi^i_1}\left(\boldsymbol{S^i_{tho,j}}\right), V^i_{\phi^i_1}\left(\boldsymbol{S^i_{tho,j}}\right) - \varepsilon, V^i_{\phi^i_1}\left(\boldsymbol{S^i_{tho,j}}\right) + \varepsilon \right) - \hat{R}^i_{tho,j} \right)^2 \right] \right. , \\ L\left(\phi^i_2\right) = \frac{1}{|\boldsymbol{B^i}|} \sum_j \left(\max \left[\left(V_{\phi^i_2}\left(\boldsymbol{S^i_{fai,j}}\right) - \hat{R}^i_{fai,j} \right)^2, \right. \right. \\ \left. \left. \left(\text{clip}\left(V_{\phi^i_2}\left(\boldsymbol{S^i_{fai,j}}\right), V^i_{\phi^i_2}\left(\boldsymbol{S^i_{fai,j}}\right) - \varepsilon, V^i_{\phi^i_2}\left(\boldsymbol{S^i_{fai,j}}\right) + \varepsilon \right) - \hat{R}^i_{fai,j} \right)^2 \right] \right. \end{cases} \tag{13}$$

where $\boldsymbol{B^i}$ denotes the sampled batch of agent i, i.e., the consecutive experiences (consisting of state, action, reward, next state, and global state) used in each training iteration. j denotes the index of the experiences in the sampled batch $\boldsymbol{B^i}$. $V_{\boldsymbol{\phi^i_1}}\left(\boldsymbol{S^i_{tho,j}}\right)$ and $V_{\boldsymbol{\phi^i_2}}\left(\boldsymbol{S^i_{fai,j}}\right)$ represent the state values of throughput and fairness of agent i based on global state at j-th periodical access interval, respectively, and $\hat{R}^i_{tho,j}$ and

$\hat{R}^i_{fai,j}$ represent the total discounted rewards of $\boldsymbol{r^i_{tho}}$ and $\boldsymbol{r^i_{fai}}$ at periodical access interval j. By calculating the loss of each objective $L(\boldsymbol{\phi^i_1}), L(\boldsymbol{\phi^i_2})$ based on different rewards, the proposed MFMAPPO can have a more accurate evaluation of state values and avoid the problem of flattening the gradient of critic networks.

To better utilize the MCSP framework, we denote $A^i_k = \omega_1 A^i_{tho,k} + \omega_2 A^i_{fai,k}$ as the total advantage of agent i at k-th periodical access interval, with ω_1 and ω_2 being coefficient hyper parameters that can be adjusted to balance the objectives of throughput and fairness. Through the General Advantage Estimation (GAE) approach [35], $A^i_{tho,k}$ and $A^i_{fai,k}$ are derived based on rewards $r^i_{tho,k}$ and $r^i_{fai,k}$, and state values $V_{\boldsymbol{\phi^i_1}}$ and $V_{\boldsymbol{\phi^i_2}}$, respectively, which are given by

$$\begin{cases} A^i_{tho,k} = \sum_{l=1}^{\infty} (\gamma\lambda)^l \left(r^i_{tho,k} + \gamma V_{\phi^i_1}\left(S^i_{tho,k+l+1}\right) - V_{\phi^i_1}\left(S^i_{tho,k+l}\right)\right) \\ A^i_{fai,k} = \sum_{l=1}^{\infty} (\gamma\lambda)^l \left(r^i_{fai,k} + \gamma V_{\phi^i_2}\left(S^i_{fai,k+l+1}\right) - V_{\phi^i_2}\left(S^i_{fai,k+l}\right)\right) \\ A^i_k = \omega_1 A^i_{tho,k} + \omega_2 A^i_{fai,k} \end{cases} \tag{14}$$

where λ and γ is the hyper-parameters to control the bias and variance respectively. The calculated advantage function A^i_k is stored in the batch $\boldsymbol{B^i}$, and the loss that the *Actor* of agent i is trained to minimize is given by

$$\begin{aligned} L\left(\boldsymbol{\theta^i}\right) \triangleq & \left[\sigma \frac{1}{|\boldsymbol{B^i}|} \sum_j S\left[\pi_{\boldsymbol{\theta}}\left(\boldsymbol{a^i_j} \mid \boldsymbol{s^i_j}\right)\right]\right] \\ & + \left[\frac{1}{|\boldsymbol{B^i}|} \sum_j \min\left(r\left(\boldsymbol{\theta^i}\right) A^i_j, \text{clip}\left(r\left(\boldsymbol{\theta^i}\right), 1-\epsilon, 1+\epsilon\right) A^i_j\right)\right], \end{aligned} \tag{15}$$

where j denotes the index of experiences in the sampled batch $\boldsymbol{B^i}$. $r(\boldsymbol{\theta^i})$ represents the ratio between the old policy and new policy of agent i, $S\left[\pi_{\boldsymbol{\theta}}\left(\boldsymbol{a_j} \mid \boldsymbol{s_j}\right)\right]$ is defined as the entropy of the new policy $\pi_{\boldsymbol{\theta}}\left(\boldsymbol{a_j} \mid \boldsymbol{s_j}\right)$ and σ denotes the entropy coefficient.

B. Critic and Actor Networks

As Fig. 4 illustrates, each agent has two critic networks and one actor network by extending the MCSP framework in the MARL scenario to evaluate state values of its different objectives separately.

1) Critic Network: To apply the MCSP in the MARL scenario, we introduce global states to provide global information in the centralized training. Moreover, we propose two global states $\boldsymbol{S^i_{tho}}, \boldsymbol{S^i_{fai}}$ as the input of *Critic 1* and *Critic 2* respectively to evaluate the value functions of throughput and fairness. As Fig.4 illustrates, *Critic 1* takes the global state $\boldsymbol{S^i_{tho}}$ as input and outputs the state value of throughput, and *Critic 2* takes the global state $\boldsymbol{S^i_{fai}}$ as input and outputs the state value of fairness. Through the shortened size of $\boldsymbol{S^i_{tho}}$ motivated by the MFE approach, the computational complexity can be greatly reduced.

In particular, the Gated Recurrent Unit (GRU) is applied to record the historical state information of each agent, so that the agent can estimate state values more accurately [36]. The hidden states of GRU $\boldsymbol{h^{(i)}_{0,\phi_1}}$ and $\boldsymbol{h^{(i)}_{0,\phi_2}}$ of *Critic 1* and *Critic 2* are initialized at the beginning of each epoch during training. In the multilayer perceptron (MLP) layer and the full connect (FC) layer, the rectified linear unit (ReLU) is used as the activation function.

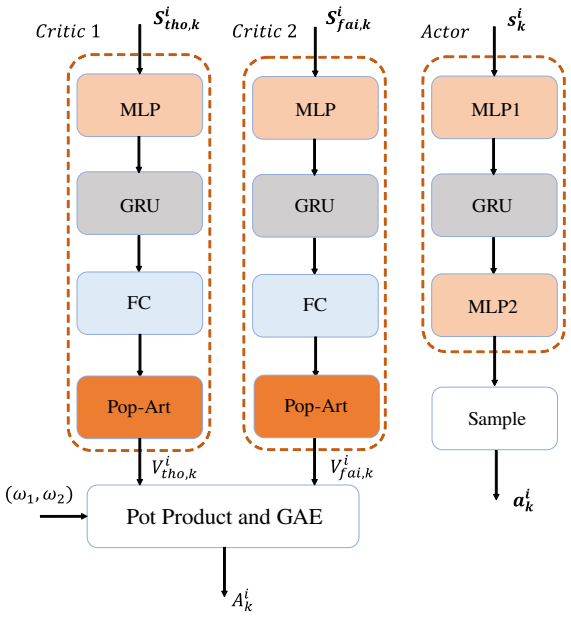


Fig. 4. The extended MCSP architecture in MFMAPPO. MLP represents the multilayer perceptron, GRU represents the gate recurrent unit and FC represents the full-connected layer.

In the MORL scenario, the value functions of different objectives have inconsistent and independent distribution. Since the advantage function takes all value functions into consideration as shown in (14), multiple value functions would lead to an unstable advantage function. To address this issue, we apply the Preserving Outputs Precisely-Adaptive Rescaling Target (Pop-Art) technique in both the *Critic 1* and the *Critic 2* [37]. During training, the Pop-Art layer will adaptively adjust the estimation of the mean value and variance of the state values, which are then used to normalize the state values $V^i_{\boldsymbol{\phi^i_1},k}$ and $V^i_{\boldsymbol{\phi^i_2},k}$ of *Critic 1* and *Critic 2*. By normalizing state values, the stability and convergence of the MFMAPPO can be further improved.

2) Actor Network: As Fig. 4 illustrates, the *Actor* outputs the action $\boldsymbol{a^i_k}$ based on the state $\boldsymbol{s^i_k}$ at k-th periodical access interval. To take consideration of historical states, the GRU is applied in the *Actor* network so that the agent can make more suitable action. The hidden state of GRU $\boldsymbol{h^{(i)}_{0,\theta}}$ of *Actor* are initialized at the beginning of each episode. Here we also use the ReLU as the activation function of MLP1 and MLP2 in *Actor* network. The vector output by MLP2 is the probability distribution of performing each action in the current state $\boldsymbol{s^i_k}$ and then the action $\boldsymbol{a^i_k}$ is selected by sampling the probability distribution.

C. Computational Complexity

This section will present the analysis of the computational complexity of the proposed MFMAPPO algorithm.

For the state space, as the state vector $\boldsymbol{s^i_k}$ presented in (6), it is a vector with fixed length 5 and does not increase with number of STAs N or number of channels M. For the action

space, given the number of time slots in each periodic access interval as $T = \lfloor \frac{N}{2M} \rfloor$, the size of the action can be derived as $\lfloor \frac{N}{2} \rfloor + 1$ as defined in Section B.1, which increases linearly with respect to the number of STAs. For the size of global states, as presented in (11) and (12), the length of global states $\boldsymbol{S^i_{tho,k}}$ and $\boldsymbol{S^i_{fai,k}}$ are given by $\lfloor \frac{N}{2} \rfloor + 6$ and $N + 5$, respectively. By using the global state motivated by the MFE, the size of inputs of both critic networks only increases linearly with respect to the number of STAs and is independent of the number of channels. Moreover, since the dimension of $\boldsymbol{S^i_{tho,k}}$ and $\boldsymbol{S^i_{fai,k}}$ are $\lfloor \frac{N}{2} \rfloor + 6$ and $N + 5$, respectively, and the dimension of their concatenation $\{\boldsymbol{s^i_k}, \boldsymbol{a^{-i}_k}, v^1_k, \ldots, v^N_k\}$ is $\lfloor \frac{N}{2} \rfloor + N + 6$, utilizing two separate global states as the input of different networks instead of taking the concatenation as the input of all networks can further reduce computational complexity by nearly half and enhance evaluation capabilities.

With the above illustration, now let us consider the computational complexity in the training stage, both *Critic 1*, *Critic 2* and *Actor* are required to work. As illustrated in [38], the computational complexity of MLP and GRU increases linearly with the length of input and output. In the proposed MFMAPPO, global states $\boldsymbol{S^i_{tho,k}}, \boldsymbol{S^i_{fai,k}}$ and state $\boldsymbol{s^i_k}$ are the input of *Critic 1*, *Critic 2* and *Actor*, respectively. The output of them are $V^i_{\boldsymbol{\phi^i_1}}$, $V^i_{\boldsymbol{\phi^i_2}}$ and $\boldsymbol{a^i_k}$, which have sizes of 1, 1 and $\lfloor \frac{N}{2} \rfloor + 1$, respectively. Therefore, the computational complexity of *Critic 1*, *Critic 2* and *Actor* are all given by $\mathcal{O}(N)$, with which the computational complexity of the MFMAPPO is given by $\mathcal{O}(N)$ during training.

Considering the computational complexity during execution, only *Actor* is required to work due to the CTDE architecture. As presented above, the overall computational complexity of MFMAPPO is $\mathcal{O}(N)$ during execution, which equals to that of *Actor*. Therefore, the proposed MFMAPPO has a lower computational complexity than other MARL-based multiple access approaches [4], [9]–[11], [16], and can be applied in the massive access scenario.

D. Algorithm Overview

The overall MFMAPPO procedure is illustrated in Fig. 3. During training, each agent independently decides the action $\boldsymbol{a^i_k}$ based on its local state $\boldsymbol{s^i_k}$ at each periodical transmission interval k, receives the reward $r^i_{tho,k}, r^i_{fai,k}$ and moves to next state $\boldsymbol{s^i_{k+1}}$. Then, the experience $(\boldsymbol{s}, \boldsymbol{S_{tho}}, \boldsymbol{S_{fai}}, \boldsymbol{a}, \boldsymbol{r_{tho}}, \boldsymbol{r_{fai}})$ of all agents will be jointly stored in a joint memory. The global information is extracted into the global states $\boldsymbol{S_{tho}}, \boldsymbol{S_{fai}}$. After collecting experiences, the joint collected experiences are divided into the experience of each agent to form the agent-specific memory $\boldsymbol{E_i}$, including $\boldsymbol{s^i}, \boldsymbol{S^i_{tho}}, \boldsymbol{S^i_{fai}}, \boldsymbol{a^i}, \boldsymbol{r^i_{tho}}$ and $\boldsymbol{r^i_{fai}}$. Subsequently, each agent i starts updating its own MFMAPPO model independently utilizing the sampled batch $\boldsymbol{B^i}$. During parameters updating, each state value $V^i_{\boldsymbol{\phi^i_1}}$ and $V^i_{\boldsymbol{\phi^i_2}}$ of agent i is calculated through different *Critic* networks, and is used to calculate the loss function $L(\boldsymbol{\phi^i_1})$, $L(\boldsymbol{\phi^i_2})$ and $L(\boldsymbol{\theta^i})$, respectively. Then, the *Critic 1*, *Critic 2* and *Actor* networks update their parameters based on the loss function $L(\boldsymbol{\phi^i_1})$, $L(\boldsymbol{\phi^i_2})$ and $L(\boldsymbol{\theta^i})$. Though agents update the model parameters independently, they can utilize the global information including in global states $\boldsymbol{S_{tho}}, \boldsymbol{S_{fai}}$ to learn cooperative transmission strategies. The overall pseudo-code of the proposed MFMAPPO algorithm is summarized in Algorithm 1.

Algorithm 1 MFMAPPO Algorithm

Initialize parameters: $\boldsymbol{\phi^i_1}$, $\boldsymbol{\phi^i_2}$, $\boldsymbol{\theta^i}$, $a^i_0 = 0$, $o^i_0 = 0$, $v^i_0 = 0$, $T = \lfloor \frac{N}{2M} \rfloor$, for $\forall i \in \{1, 2, \ldots, N\}$
Initialize $\boldsymbol{h^{(1)}_{0,\theta}}, \ldots \boldsymbol{h^{(N)}_{0,\theta}}$ of *Actor* for all agents
Initialize $\boldsymbol{h^{(1)}_{0,\phi_1}}, \ldots \boldsymbol{h^{(N)}_{0,\phi_1}}$ of *Critic 1* for all agents
Initialize $\boldsymbol{h^{(1)}_{0,\phi_2}}, \ldots \boldsymbol{h^{(N)}_{0,\phi_2}}$ of *Critic 2* for all agents
while $t < k_{max}T$ **do**
  for agent $i = 1, 2, \ldots, N$ **do**
    if $t \ mod \ T = 0$ **then**
      Calculate state $\boldsymbol{s^i_k}$, where $k = \frac{t}{T}$
      if agent i has packet to transmit **then**
        Generating $\boldsymbol{a^i_k} \leftarrow \pi_{\boldsymbol{\theta^i}}(\boldsymbol{s^i_k})$
      else
        $\beta^i_k \leftarrow 0$, agent i wait
      Execute action $\boldsymbol{a^i_k}$
  for agent $i = 1, 2, \ldots, N$ **do**
    Receive reward $r^i_{tho,k}, r^i_{fai,k}$ and global state $\boldsymbol{S^i_{tho,k}}, \boldsymbol{S^i_{fai,k}}$
    Store experience $\left(\boldsymbol{s^i}, \boldsymbol{S^i_{tho}}, \boldsymbol{S^i_{fai}}, \boldsymbol{a^i}, \boldsymbol{r^i_{tho}}, \boldsymbol{r^i_{fai}}\right)$ as the experience into memory $\boldsymbol{E_i}$
for agent $i = 1, 2, \ldots, N$ **do**
  Randomly sample experiences $\boldsymbol{B^i}$ from the memory D
  for b in sampled batch $\boldsymbol{B^i}$ **do**
    Calculate discounted reward $\hat{R}_{tho,k}$, $\hat{R}_{fai,k}$ based on $r_{tho,b}$, $r_{fai,k}$
    Output normalized state values $V_{\boldsymbol{\phi^i_1}}\left(\boldsymbol{S^i_{tho,k}}\right)$, $V_{\boldsymbol{\phi^i_2}}\left(\boldsymbol{S^i_{fai,k}}\right)$ through each critic using Pop-Art
    Compute loss $L(\boldsymbol{\phi^i_1})$ and $L(\boldsymbol{\phi^i_2})$ of critic net- works
    Compute advantage $A^i_{tho,b}$ and $A^i_{fai,k}$ based on the GAE
    Compute actor loss $L(\boldsymbol{\theta^i})$ based on advantages $A^i_{tho,b}$, $A^i_{fai,k}$
    Update $\boldsymbol{\phi^i_1}$, $\boldsymbol{\phi^i_2}$ and $\boldsymbol{\theta^i}$ by performing mini-batch gradient descent
  Empty the memory $\boldsymbol{E_i} = \{\}$
Move to next episode

In practice, the trained model of the actor can be piggy-backed in the TF, which is broadcast to each STAs by the AP. In such way, the AP should be aware of the number of associated STAs to determine which model to broadcast. Therefore, the association between AP and STAs is required to distribute the MFMAPPO model. There are various approaches to establish the association between AP and STA, including probe request, passive scanning and active scanning [39], [40]. After the association is done, the number of associated STAs can be readily acquired by the AP. Then, the proposed MFMAPPO model trained in the network of N STAs can be distributed to each STA through the TF. Note that STAs may enter or leave the network dynamically. In this case, the above approach can also be applied since the AP is aware of the number of associated STAs.

V. Simulation Results and Discussions

In this section, we evaluate our proposed MFMAPPO algorithm by providing simulation results under different scenarios. In the following, we first introduce the simulation setup, and then the detailed performance evaluations will be presented under different scenarios.

TABLE I
NETWORK PARAMETERS [1]

Parameters	Value
Duration of SIFS t_{SIFS}	$16\mu s$
Duration of BA t_{BA}	$68\mu s$
Duration of one PPDU t_{PPDU}	$512\mu s$
Duration of one RS slot t_{RS}	$16\mu s$
Duration of one TF-R t_{TF-R}	$100\mu s$
Duration of one time slot t_{slot}	$512\mu s$

TABLE II
HYPER-PARAMETERS OF MFMAPPO

Parameters	Value
Learning rate	$5 \cdot 10^{-4}$
Discounted factor γ	0.98
Experience memory size	400
Training epochs	8
Policy entropy coefficient σ	0.002
GAE factor λ	0.95
Huber loss δ	10
Optimizer	Adam

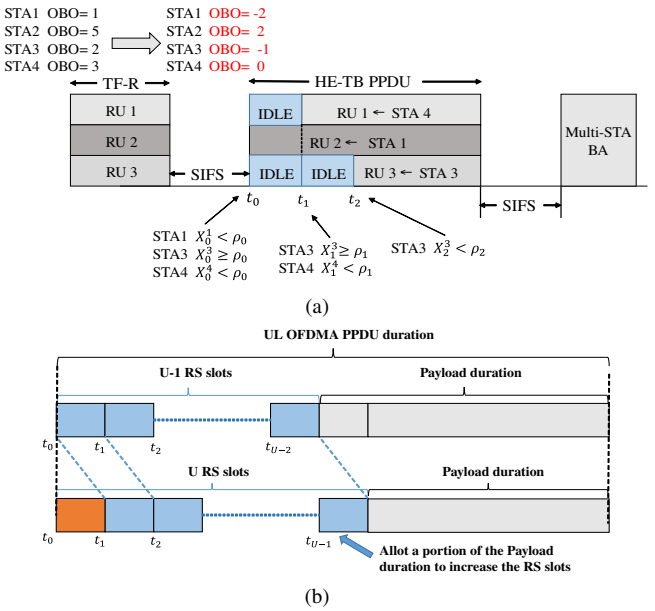


Fig. 5. Illustration of H-UORA. (a) random access mechanism of H-UORA. (b) RU sensing slots U in H-UORA.

A. Simulation Setup

We consider a homogeneous wireless communication network where N STAs deployed with the proposed MFMAPPO algorithm or other multiple access methods access the AP through M RUs. In following simulations, the duration of one time slot is set to be $512\mu s$, which equals to the duration time of one Presentation Protocol Data Unit (PPDU). We assume Bernoulli packet arrival for each STA and the packet arrival rate is denoted as λ. The buffer size of each STA is set to be 10 packets, and packets would be discarded when the buffer is full. The network parameters are summarized in Table I.

To present a comprehensive comparison, we introduce a series of methods as baselines, including IPPO, Random, UORA and H-UORA [22].

1) MFMAPPO: As the architecture of the MFMAPPO algorithm shown in Fig. 4, the number of features of GRU layer in the actor network is 150 and that of GRU in both critics network is 300 when the number of STAs is $N = 150$. The number of hidden layers of MLP is 2 and that of GRU is 1 in both actor network and critic networks. During training, the number of time steps of each GRU in both actor network and critic networks is 2. The number of samples equals to the length of an episode. There is no re-training or parameter update during evaluation. The detailed hyper-parameters of the MFMAPPO algorithm are presented in Table II.

2) IPPO: Independent PPO (IPPO) is introduced as the baseline of single-agent RL methods applied in the problem. For comparison, the actor network in IPPO is the same as that in the MFMAPPO algorithm. Then, the neural network architecture of critic is also the same as that of MFMAPPO while it has only 32 neurons in the GRU layer because there has no global states in the *IPPO* method. The definition of state, action and reward in IPPO are also the same as those in the MFMAPPO while there is no global state in the IPPO.

3) Random: Random method serves as the baseline for the action mechanism of the proposed MFMAPPO algorithm. In Random method, each agent randomly selects an action as introduced in the MFMAPPO at each periodical access interval.

4) UORA: We introduce the UORA method as the baseline of traditional random access methods in the uplink OFDMA scenario. As presented in Fig. 2, each agent attempts to access if the OBO count decreased by the number of RUs M is non-positive. The optimal transmission probability q_m has been given as $q_m = \frac{M}{N}$ in the multiple RUs scenario, where N and M represent the number of STAs and RUs, respectively [3].

5) H-UORA: We introduce the H-UORA as a novel multiple access method based on UORA [22]. This method reduces the collision probability by allowing carrier sensing and retransmissions in multiple time slots within the HE-TB PPDU. As Fig. 5 illustrates, STAs have several chances to detect idle RUs using RU granulated carrier sensing before transmitting. The slot for RU sensing process is called RS slots. As illustrated in Fig. 5a, the OBO counter of STA1, STA2, STA3 and STA4 are decreased to -2, 2, -1 and 0 respectively after received the TF-R. Then, each STA which has a non-positive OBO counter attempts to access RUs. In the beginning of each RS slot $0 \leq u \leq U$, each STA i calculates the transmission probability ρ_u based on the number of available idle RUs and the value of u and generates a random value X_u^i. Then, STAs with $X_u^i \leq \rho_u$ randomly choose an idle RU to access. Other STAs with $X_u^i \geq \rho_u$ keep carrier sensing until the next RS slot. Such procedure repeats until all U RS slots have passed or all STAs with non-positive OBO counter access successfully.

[1]Here, the time duration of the packet payload, i.e., the PPDU, is given instead of the packet size. This is because in this work, we consider the throughput defined as the time proportion for successfully-transmitted payloads. With this regard, when each STA has the same packet transmission rate R (in unit of bit/s), the throughput defined as the proportion of successfully-transmitted payloads can be easily translated into the one defined as the amount of the successfully transmitted data bits per used time (in unit of bit/s), by simply multiplying with R.

B. Performance Metrics

The following metrics are used to evaluate the performance of the proposed MFMAPPO algorithm.

- *JFI:* We introduce Jain Fairness Index (JFI) to evaluate the fairness of STAs with same traffic. The JFI is defined as:

$$JFI = \frac{\left(\sum_{i=1}^{N} v^i\right)^2}{N\sum_{i=1}^{N}\left(v^i\right)^2}, \tag{16}$$

  where N is the number of STAs and v^i is the long-term throughput of STA i and $JFI \in \left[\frac{1}{N}, 1\right]$. When $JFI = 1$, the STAs in the network have absolutely fair throughput. When $JFI = \frac{1}{N}$, one of the agents has completely monopolized the wireless resource, which ruins the fairness. Moreover, when STAs have different packet arrival rate averaged over time in different groups, we demonstrate the average JFI over groups.
- *Throughput:* In scenario *C*, *D* and *E*, we demonstrate the short-term throughput of all agents, which is obtained as the proportion for successfully-transmitted payload over the past 2000 slots. In scenario *F*, *G* and *H*, we demonstrate the long-term throughput of each agent, which is obtained as the proportion for successfully-transmitted payload over the whole simulation.

C. Single-RU

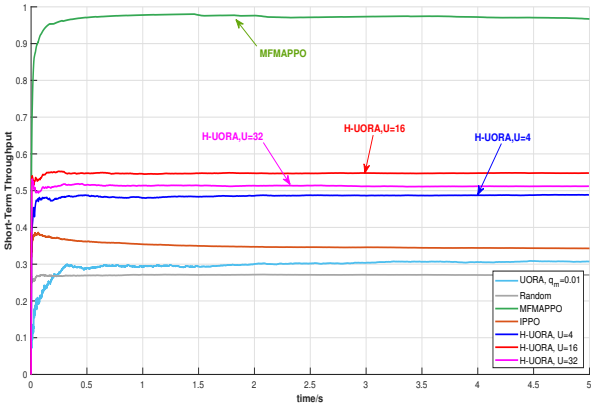


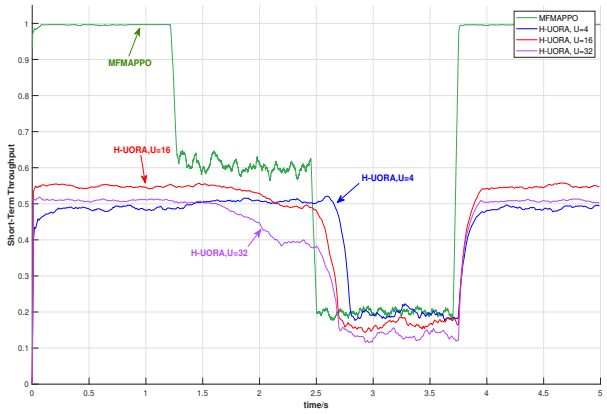
Fig. 6. Performance comparison between each method of dynamic traffic single RU scenario with $N = 100$, $M = 1$ and static packet arrival rate $\lambda = 0.05$.

In this section, we consider only a single RU in the network, i.e., $M = 1$. First, we consider $N = 100$ and the packet arrival rate of STAs is given by $\lambda = 0.05$, and thus the network becomes saturated. As shown in Fig. 6, the throughput performance of the MFMAPPO algorithm is nearly optimal approaching the maximum bound, which is much better than other methods.

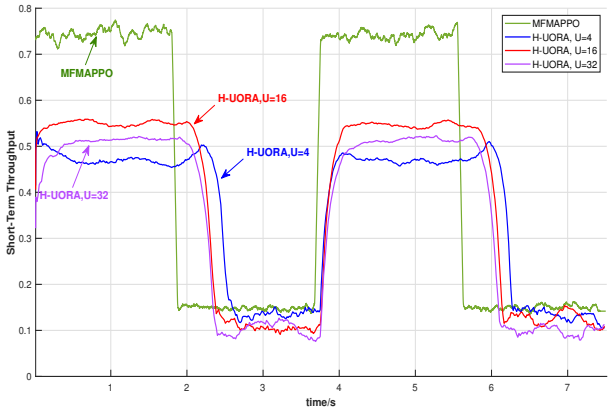
For other methods, it can be seen from Fig. 6 that IPPO can perform better than other methods based on random access including UORA and Random, whose throughput performance is limited by $1/e \approx 37\%$. Moreover, the performance improvement of H-UORA method is limited with the growth of RS slots U as the throughput of H-UORA with $U = 16$ is

TABLE III
THE DETAILED SIMULATION RESULTS FOR SCENARIO.C

Method	Long-Term Throughput	JFI
UORA	0.2972	0.9913
Random	0.2705	0.9992
MFMAPPO	0.9701	0.9955
IPPO	0.3497	0.9995
H-UORA,U=4	0.4847	0.9762
H-UORA,U=16	0.5465	0.9838
H-UORA,U=32	0.5122	0.9728



(a)



(b)

Fig. 7. Performance comparison between each method of different scenarios. (a) $N = 100$ STAs, $M = 1$ RU and dynamic arrival rate $\lambda \in \{1, 0.006, 0.002, 1\}$. (b) $N = 150$ STAs, $M = 1$ RU and dynamic packet arrival rate is $\lambda \in \{0.005, 0.001, 0.005, 0.001\}$.

better than that of $U = 4$ and $U = 32$. Though the increasing of RS slots U can improve the successful access probability by performing more RU sensing slots, the payload duration will be reduced as Fig. 5 illustrates. Table III further presents the fairness performance, and indicates that the proposed MFMAPPO algorithm achieves high JFI as well as other methods, i.e., Random, IPPO, UORA and H-UORA.

Since the proposed MFMAPPO and H-UORA have much better throughput performance than other methods in previous simulations, we only demonstrate comparison of the MFMAPPO with H-UORA in the following. We now consider the scenario that packet arrival rate of each STA λ is dynamic over time.

Fig. 7a shows the performance comparison of total through-

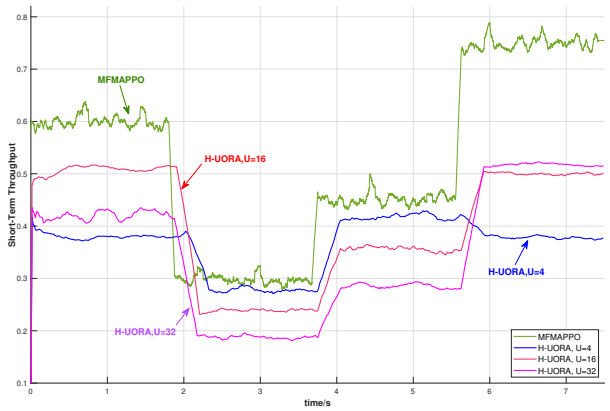


Fig. 8. Performance comparison between each methods of scenario with $N = 150$ STAs, $M = 5$ RUs and dynamic arrival rate $\lambda \in \{0.02, 0.01, 0.015, 0.025\}$.

put when $N = 100$. Here the packet arrival rate $\lambda \in \{1, 0.006, 0.002, 1\}$ indicates that λ changes after same time interval, i.e., λ varies from 1 to 0.006, to 0.002 and to 1 after an equal time interval. As shown in Fig. 7a, the proposed MFMAPPO algorithm can also achieve a nearly optimal throughput performance while the H-UORA can only have half of the throughput of MFMAPPO when the traffic is saturated.

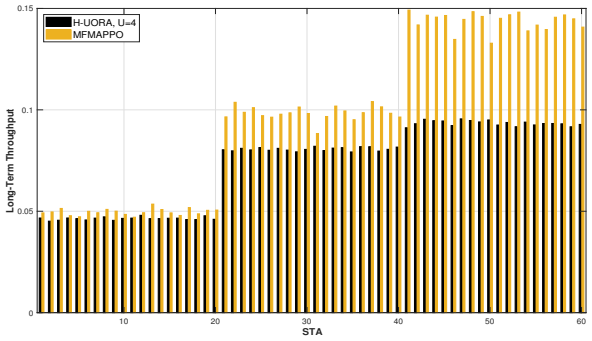
To better simulate the massive access scenario, we further consider a scenario with more STAs, i.e., $N = 150$, and the packet arrival rate $\lambda \in \{0.005, 0.001, 0.005, 0.001\}$. Fig. 7b demonstrates that the MFMAPPO can still maintain the nearly optimal throughput performance without throughput deterioration, which demonstrate the capability of MFMAPPO to deal with the large-scale single-RU access condition.

D. Multi-RU

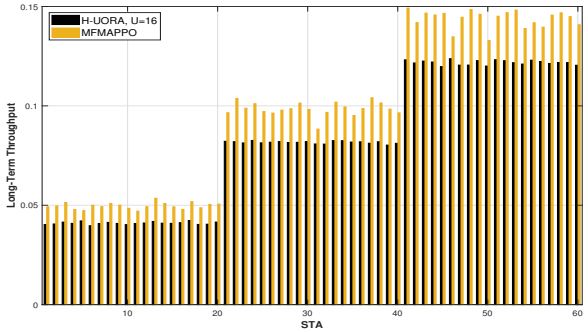
In this section, we further consider the scenario with multiple RUs. The number of RUs and STAs remain unchanged over time, i.e., $M = 5$, $N = 150$. The packet arrival rate $\lambda \in \{0.02, 0.01, 0.015, 0.025\}$, i.e., λ varies from 0.02 to 0.01, to 0.015 and to 0.025 after an equal time interval.

As shown in Fig. 8, the proposed MFMAPPO algorithm can work well in the dynamic traffic scenario with multiple RUs and performs better than H-UORA. By combining Fig. 7a, Fig. 7b and Fig. 8, we can see that a fixed configuration of RS slots U in H-UORA cannot adapt to the dynamic traffic condition. In particular, when the traffic is light, a small U is preferred since a large U leads to unnecessary overhead of carrier sensing. When the traffic becomes heavy, a large U can better alleviate channel contention, and yet a overly-large U may lead to throughput deterioration due to the incurred carrier sensing overhead.

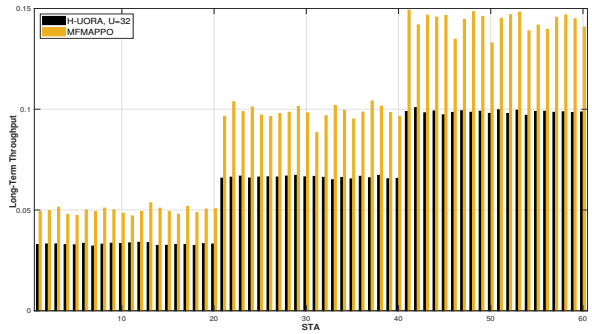
In previous simulations, each STA has the same packet arrival rate. We now consider the condition that STAs have different packet arrival rates. We set $N = 60$ and $M = 10$ in the following simulations and divide all N STAs into three groups as $\boldsymbol{N_1} = \{1, \ldots, 20\}$, $\boldsymbol{N_2} = \{21, \ldots, 40\}$ and $\boldsymbol{N_3} = \{41, \ldots, 60\}$.



(a) $U = 4$ for H-UORA.



(b) $U = 16$ for H-UORA.



(c) $U = 32$ for H-UORA.

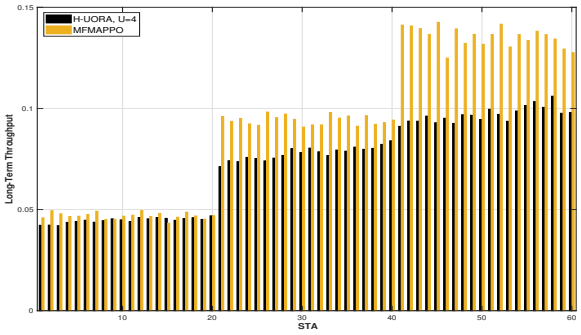
Fig. 9. Performance comparison between H-UORA with various value of U and MFMAPPO with $N = 60$ and $M = 10$. The diverse packet arrival rates are static: $\lambda_i = 0.05, \forall i \in \boldsymbol{N_1}$, $\lambda_i = 0.1, \forall i \in \boldsymbol{N_2}$ and $\lambda_i = 0.15, \forall i \in \boldsymbol{N_3}$.

TABLE IV
THE THROUGHPUT PERFORMANCE OF EACH GROUP WITH STATIC TRAFFIC

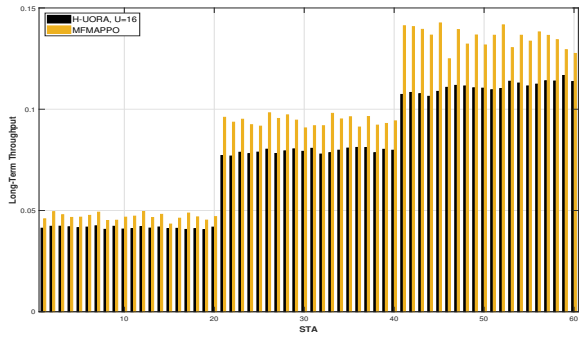
Method	$\boldsymbol{N_1}$	$\boldsymbol{N_2}$	$\boldsymbol{N_3}$
MFMAPPO	0.0497	0.0985	0.1438
H-UORA,U=4	0.0464	0.0807	0.0935
H-UORA,U=16	0.0410	0.0817	0.1218
H-UORA,U=32	0.0330	0.0662	0.0987

The packet arrival rates vary across groups, i.e., $\lambda_i = 0.05, \forall i \in \boldsymbol{N_1}$, $\lambda_i = 0.1, \forall i \in \boldsymbol{N_2}$ and $\lambda_i = 0.15, \forall i \in \boldsymbol{N_3}$. As shown in Fig. 9 and Table IV, for groups $\boldsymbol{N_2}, \boldsymbol{N_3}$ with heavy traffic, the long-term throughput performance of H-UORA with $U = 16$ is better than that with $U = 4$ and $U = 32$, while it is opposite for the group $\boldsymbol{N_1}$ with light traffic. Similarly, we can see from Table IV that H-UORA with a small U is only suitable to light traffic, and that a

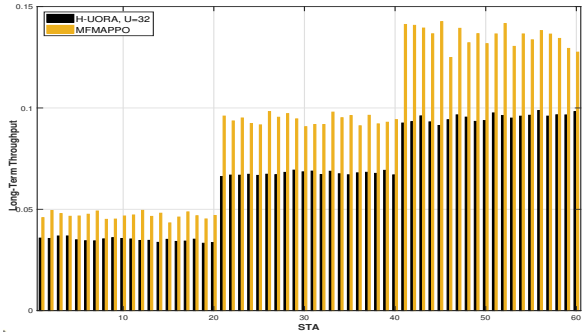
This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2024.3355276



(a) $U = 4$ in dynamic scenario.



(b) $U = 16$ for H-UORA.



(c) $U = 32$ for H-UORA.

Fig. 10. Performance comparison between H-UORA with various value of U and MFMAPPO with $N = 60$ and $M = 10$. The diverse packet arrival rates are dynamic with time-averaged values: $\overline{\lambda}_i = 0.05, \forall i \in \boldsymbol{N_1}$, $\overline{\lambda}_i = 0.1, \forall i \in \boldsymbol{N_2}$, $\overline{\lambda}_i = 0.15, \forall i \in \boldsymbol{N_3}$.

overly-large U leads to throughput deterioration even when traffic is heavy. For the proposed MFMAPPO algorithm, its performance exceeds H-UORA for each group of agents while ensuring the fairness across STAs in each group.

Moving forward, we consider the condition that each STA in the same group has a varied packet arrival rate while the packet arrival rate averaged over time remains identical, i.e., $\overline{\lambda}_i = 0.05, \forall i \in \boldsymbol{N_1}$, $\overline{\lambda}_i = 0.1, \forall i \in \boldsymbol{N_2}$, $\overline{\lambda}_i = 0.15, \forall i \in \boldsymbol{N_3}$. For one STA in the same group, its packet arrival rate has a larger variance over time. As an example, we let $\lambda_1 \in 0.012, 0.05, 0.088$ for STA 1, and $\lambda_{10} \in 0.03, 0.05, 0.07$ for STA 10 which is in the same group with agent 1.

As shown in Fig. 10, although the packet arrival rates $\overline{\lambda}$ averaged over time is identical, the performance of H-UORA is greatly affected by the large variance of the packet arrival rate over time, especially when U is small. When the traffic is heavy, i.e., $\overline{\lambda}_3 = 0.15$ in groups $\boldsymbol{N_3}$, the agents with larger variance have a lower average throughput, which further proves that the fixed configuration of H-UORA cannot adapt to the diverse conditions across STAs well and the greater traffic variance further deteriorates the throughput of H-UORA. In contrast, the proposed MFMAPPO shows a much better adaptation capacity.

Note that the above results of MFMAPPO are obtained by the trained model after convergence. In the following, we present the convergence performance of the proposed MFMAPPO in Fig. 11 by comparing the average rewards obtained by agents in each episode under different scenarios, i.e., the scenarios of Fig. 6, Fig. 7a, Fig. 7b, Fig. 8 and Fig. 9. It can be observed that in the scenarios of Fig. 6, Fig. 7a, Fig. 7b, Fig. 8 and Fig. 9, the MFMAPPO can converge at about $250\text{-}th$, $250\text{-}th$, $200\text{-}th$, $450\text{-}th$ and $200\text{-}th$ episode, respectively. Though the numbers of STAs and packet arrival rates vary across different scenarios, the MFMAPPO can converge in each scenario.

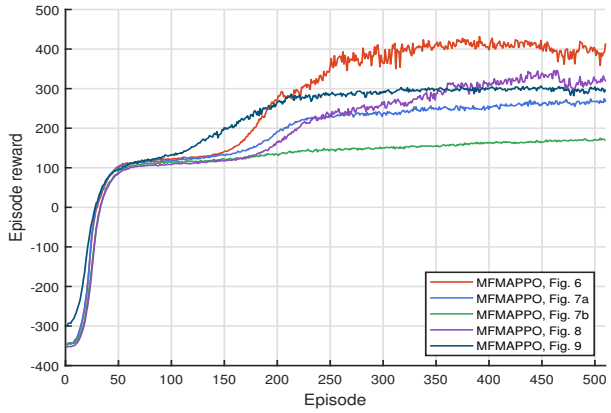


Fig. 11. The convergence of MFMAPPO under different scenarios.

E. Test Traffic Different from Training Traffic

In previous simulations, the training traffic is the same as testing traffic, i.e., the packet arrival rate in training equals that in the test. To see the generalization capacity of the proposed MFMAPPO algorithm, we evaluate the performance of the MFMAPPO in the condition that test traffic is different from training traffic.

First, we consider the scenario that there are $N = 100$ agents who have the identical traffic accessing the network through only one RU. The training traffic is set to be saturated, i.e., $\lambda = 1$. Table V presents the throughput in test, where the target denotes the upper bound of throughput. As Table V illustrates, the performance of MFMAPPO with each test traffic can maintain nearly optimal even when the packet arrival rate is different from that in training.

Then we consider the scenario of dynamic and diverse traffic with $N = 60$ and $M = 10$. By dividing all N STAs into three groups as $\boldsymbol{N_1} = \{1, \ldots, 20\}$, $\boldsymbol{N_2} = \{21, \ldots, 40\}$ and $\boldsymbol{N_3} = \{41, \ldots, 60\}$, the packet arrival rate of each group for training and testing is shown in Table VI. In 1st and 2nd row, the packet arrival rate averaged over time for training equals

TABLE V
PERFORMANCE WITH VARIOUS TEST TRAFFIC

Test Traffic	Throughput	Target	JFI
$\lambda = 0.05$	0.9679	1	0.9943
$\lambda = 0.005$	0.4838	0.5	0.9930
$\lambda \in \{0.005, 0.001, 0.003\}$	0.2877	0.3	0.9901

that in the test for each group. In this case, the throughput and fairness performance will not be greatly affected. In the 3rd and 4th row, the packet arrival rate averaged over time for training does not equal that in the test for each group. As a result, the throughput and the fairness will be greatly affected.

TABLE VI
TRAINING TRAFFIC AND CORRESPONDING TEST TRAFFIC

Training Traffic	Test Traffic	Throughput	Target	JFI
$\lambda_i \in \{0.05, 0.07, 0.05\}, \forall i \in \boldsymbol{N_1}$ $\lambda_i \in \{0.1, 0.12\}, \forall i \in \boldsymbol{N_2}$ $\lambda_i \in \{0.15, 0.17\}, \forall i \in \boldsymbol{N_3}$	$\lambda_i \in \{0.03, 0.07, 0.07\}, \forall i \in \boldsymbol{N_1}$ $\lambda_1 \in \{0.09, 0.11, 0.13\}, \forall i \in \boldsymbol{N_2}$ $\lambda_i \in \{0.12, 0.2\}, \forall i \in \boldsymbol{N_3}$	0.63	0.65	0.998
$\lambda_1 \in \{0.05\}, \forall i \in \boldsymbol{N_1}$ $\lambda_i \in \{0.1\}, \forall i \in \boldsymbol{N_2}$ $\lambda_i \in \{0.15\}, \forall i \in \boldsymbol{N_3}$	$\lambda_i \in \{0.03, 0.07\}, \forall i \in \boldsymbol{N_1}$ $\lambda_i \in \{0.08, 0.12\}, \forall i \in \boldsymbol{N_2}$ $\lambda_i \in \{0.12, 0.18\}, \forall i \in \boldsymbol{N_3}$	0.527	0.6	0.99
$\lambda_i \in \{0.05\}, \forall i \in \boldsymbol{N_1}$ $\lambda_i \in \{0.1\}, \forall i \in \boldsymbol{N_2}$ $\lambda_i \in \{0.15\}, \forall i \in \boldsymbol{N_3}$	$\lambda_i \in \{0.08, 0.12\}, \forall i \in \boldsymbol{N_1}$ $\lambda_i \in \{0.12, 0.18\}, \forall i \in \boldsymbol{N_2}$ $\lambda_i \in \{0.03, 0.07\}, \forall i \in \boldsymbol{N_3}$	0.481	0.6	0.977
$\lambda_i \in \{0.05, 0.07, 0.05\}, \forall i \in \boldsymbol{N_1}$ $\lambda_i \in \{0.1, 0.12\}, \forall i \in \boldsymbol{N_2}$ $\lambda_i \in \{0.15, 0.17\}, \forall i \in \boldsymbol{N_3}$	$\lambda_i \in \{0.18, 0.12\}, \forall i \in \boldsymbol{N_1}$ $\lambda_i \in \{0.1, 0.06\}, \forall i \in \boldsymbol{N_2}$ $\lambda_i \in \{0.13, 0.07\}, \forall i \in \boldsymbol{N_3}$	0.545	0.66	0.969
Multiple Traffic Combination	Same as 3rd row	0.552	0.6	0.994
Multiple Traffic Combination	Same as 4th row	0.576	0.66	0.982

To improve the performance when the packet arrival rates averaged over time in training is different with that in the test, we consider a multiple training traffic combination in 5th and 6th row at Table VI. In particular, each STA includes traffic combinations with different time-averaged value during training, i.e., one of the following three training traffic combination is randomly selected over episodes: 1) $\lambda_i = 0.05, \forall i \in \boldsymbol{N_1}$, $\lambda_i = 0.1, \forall i \in \boldsymbol{N_2}$, $\lambda_i = 0.15, \forall i \in \boldsymbol{N_3}$; 2) $\lambda_i = 0.1, \forall i \in \boldsymbol{N_1}$, $\lambda_i = 0.15, \forall i \in \boldsymbol{N_2}$, $\lambda_i = 0.05, \forall i \in \boldsymbol{N_3}$; 3) $\lambda_i = 0.15, \forall i \in \boldsymbol{N_1}$, $\lambda_i = 0.05, \forall i \in \boldsymbol{N_2}$, $\lambda_i = 0.1, \forall i \in \boldsymbol{N_3}$. The performance comparison with different test traffics is also shown in Table VI. As shown in Table VI, the MFMAPPO with such training setting can have better generalization capacity in terms that even when the time-averaged value of test traffic is different with that in training traffic combinations as shown in the 5th, 6th row, the MFMAPPO can also achieves a better throughput and fairness performance compared with those in 3rd, 4th row respectively.

We can conclude from the aforementioned simulations that: 1) When all agents have identical packet arrival rate, the MFMAPPO with saturated training traffic can adapt to various test traffic without retraining. 2) When agents have different packet arrival rate, the MFMAPPO can adapt to the packet arrival rates in the test that have the same time-averaged value as those in training. 3) When the packet arrival rate of MFMAPPO in training includes multiple traffic combinations, the MFMAPPO can adapt to those packet arrival rates that have different time-averaged from those in training.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proposed the MFMAPPO algorithm based on MARL for the uplink OFDMA scenario in IEEE 802.11ax networks. To address the massive access problem, a novel global state motivated by the MFE is introduced in MFMAPPO to enhance the convergence performance and greatly reduce the computational complexity during training. Moreover, the CTDE architecture is applied to improve performance in terms of both throughput and fairness and reduce the computational complexity during execution stage. Considering the conflict between throughput and fairness, the MCSP framework is introduced in the proposed MFMAPPO algorithm to address the fairness issue while improving throughput by evaluating those state values through different critic networks. Furthermore, we have introduced the action mechanism that lets agents decide action every certain slots to expand the size of action space, which avoids from converging to the local optimal strategy due to excessive collisions. Simulation results show that MFMAPPO can achieve nearly optimal throughput performance while guaranteeing the fairness in various traffic configurations. Moving forward, it would be interesting to consider other performance metrics such as delay, to fully exploit the power of RL in the multiple access problem for the next generation wireless networks.

REFERENCES

[1] "IEEE approved draft standard for information technology– telecommunications and information exchange between systems local and metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 1: Enhancements for high efficiency wlan," *IEEE P802.11ax/D8.0, October 2020 (approved draft)*, pp. 1–820, 2021.

[2] B. Bellalta, "Ieee 802.11ax: High-efficiency wlans," *IEEE Wireless Communications*, vol. 23, no. 1, pp. 38–46, 2016.

[3] L. Lanante, H. O. T. Uwai, Y. Nagao, M. Kurosaki, and C. Ghosh, "Performance analysis of the 802.11ax ul ofdma random access protocol in dense networks," in *Proc. IEEE ICC*, 2017, pp. 1–6.

[4] R. Kassab, A. Destounis, D. Tsilimantos, and M. Debbah, "Multi-agent deep stochastic policy gradient for event based dynamic spectrum access," in *Proc. IEEE PIMRC*, 2020, pp. 1–6.

[5] J. Bai, H. Song, Y. Yi, and L. Liu, "Multiagent reinforcement learning meets random access in massive cellular internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17 417–17 428, 2021.

[6] V. Raj, I. Dias, T. Tholeti, and S. Kalyani, "Spectrum access in cognitive radio using a two-stage reinforcement learning approach," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 20–34, 2018.

[7] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 310–323, 2018.

[8] C. Zhong, Z. Lu, M. C. Gursoy, and S. Velipasalar, "A deep actor-critic reinforcement learning framework for dynamic multichannel access," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1125–1139, 2019.

[9] Y. Xu, J. Yu, W. C. Headley, and R. M. Buehrer, "Deep reinforcement learning for dynamic spectrum access in wireless networks," in *Proc. IEEE MILCOM*, 2018, pp. 207–212.

[10] S. Wang, H. Liu, P. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, pp. 257–265, 2018.

[11] Z. Guo, Z. Chen, P. Liu, J. Luo, X. Yang, and X. Sun, "Multi-agent reinforcement learning-based distributed channel access for next generation wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 5, pp. 1587–1599, 2022.

[12] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2282–2292, 2019.

[13] H.-H. Chang, H. Song, Y. Yi, J. Zhang, H. He, and L. Liu, "Distributive dynamic spectrum access through deep reinforcement learning: A reservoir computing-based approach," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1938–1948, 2018.

[14] C. Bowyer, D. Greene, T. Ward, M. Menendez, J. Shea, and T. Wong, "Reinforcement learning for mixed cooperative/competitive dynamic spectrum access," in *Proc. IEEE DySPAN*, 2019, pp. 1–6.

[15] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1277–1290, 2019.

[16] M. Sohaib, J. Jeong, and S.-W. Jeon, "Dynamic multichannel access via multi-agent reinforcement learning: Throughput and fairness guarantees," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 3994–4008, 2022.

[17] S. K. Sharma and X. Wang, "Collaborative distributed q-learning for rach congestion minimization in cellular iot networks," *IEEE Communications Letters*, vol. 23, no. 4, pp. 600–603, 2019.

[18] H. Gao, W. Li, R. A. Banez, Z. Han, and H. V. Poor, "Mean field evolutionary dynamics in dense-user multi-access edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 7825–7835, 2020.

[19] K. Kar, S. Sarkar, and L. Tassiulas, "Achieving proportional fairness using local information in aloha networks," *IEEE Transactions on Automatic Control*, vol. 49, no. 10, pp. 1858–1863, 2004.

[20] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. M. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative, multi-agent games," *Avaliable as ArXiv: 2103.01955*, 2021.

[21] N. D. Nguyen, T. T. Nguyen, P. Vamplew, R. Dazeley, and S. Nahavandi, "A prioritized objective actor-critic method for deep reinforcement learning," *Neural Computing and Applications*, vol. 33, no. 16, pp. 10 335–10 349, 2021.

[22] L. Lanante, C. Ghosh, and S. Roy, "Hybrid ofdma random access with resource unit sensing for next-gen 802.11 ax wlans," *IEEE Transactions on Mobile Computing*, vol. 20, no. 12, pp. 3338–3350, 2020.

[23] S.-W. Jeon and H. Jin, "Online estimation and adaptation for random access with successive interference cancellation," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022.

[24] S.-H. Lee, B. C. Jung, and S.-W. Jeon, "Successive interference cancellation with feedback for random access networks," *IEEE Communications Letters*, vol. 21, no. 4, pp. 825–828, 2017.

[25] C.-H. Ke and L. Astuti, "Applying multi-agent deep reinforcement learning for contention window optimization to enhance wireless network performance," *ICT Express*, 2022.

[26] S. Cho, "Reinforcement learning for rate adaptation in csma/ca wireless networks," in *Advances in Computer Science and Ubiquitous Computing*, J. J. Park, S. J. Fong, Y. Pan, and Y. Sung, Eds. Singapore: Springer Singapore, 2021, pp. 175–181.

[27] R. Ali, B. Kim, S. W. Kim, H. S. Kim, and F. Ishmanov, "(relbt): A reinforcement learning-enabled listen before talk mechanism for lte-laa and wi-fi coexistence in iot," *Computer Communications*, vol. 150, pp. 498–505, 2020.

[28] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proc. ICML*, 2018.

[29] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[30] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Neural Information Processing Systems*, 2017.

[31] C. Liu, X. Xu, and D. Hu, "Multiobjective reinforcement learning: A comprehensive overview," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 3, pp. 385–398, 2014.

[32] H. Zhang, Y. Kang, L. Song, Z. Han, and H. V. Poor, "Age of information minimization for grant-free non-orthogonal massive access using mean-field games," *IEEE Transactions on Communications*, vol. 69, no. 11, pp. 7806–7820, 2021.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[34] Y. Yang and J. Wang, "An overview of multi-agent reinforcement learning from game theoretical perspective," *Avaliable as ArXiv: 2011.00583*, 2020.

[35] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[36] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[37] M. Hessel, H. Soyer, L. Espeholt, W. Czarnecki, S. Schmitt, and H. van Hasselt, "Multi-task deep reinforcement learning with popart," in *Proc. AAAI*, vol. 33, no. 01, 2019, pp. 3796–3803.

[38] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[39] L. Oliveira, D. Schneider, J. De Souza, and W. Shen, "Mobile device detection through wifi probe request analysis," *IEEE Access*, vol. 7, pp. 98 579–98 588, 2019.

[40] Y. Li, J. Barthelemy, S. Sun, P. Perez, and B. Moran, "A case study of wifi sniffing performance evaluation," *IEEE Access*, vol. 8, pp. 129 224–129 235, 2020.

Mingqi Han received the B.Eng. degree in Communication Engineering from School of Electronics and Communication Engineering, Shenzhen Campus of Sun Yat-sen University, Shenzhen, China in 2022. He is currently pursing for the M.E. degree in Information and Communication Engineering with School of Electronics and Communication Engineering, Shenzhen Campus of Sun Yat-sen University, Shenzhen, China. His research interests include machine learning, reinforcement learning and multiple access.

Xinghua Sun (M'13) received the B.S. degree from Nanjing University of Posts and Telecommunications (NJUPT), China, in 2008 and the Ph.D. degree from City University of Hong Kong (CityU), China, in 2013. In 2010, he was a visiting student with the National Institute for Research in Digital Science and Technology (INRIA), France. In 2013, he was a postdoctoral fellow at CityU. From 2015 to 2016, he was a postdoctoral fellow at University of British Columbia, Canada. From July to Aug. 2019, he was a visiting scholar at Singapore University of Technology and Design, Singapore. From 2014 to 2018, he was an associate professor with NJUPT. Since 2018, he has been an associate professor with Sun Yat-sen University, Guangdong, China. Dr. Sun was a co-recipient of the Best Paper Award from the EAI IoTaaS in 2023. He served as the Technical Program Committee Member and the Organizing Committee Member for numerous conferences. His research interests are in the area of stochastic modeling of wireless networks and machine learning for networking.

Wen Zhan (Member, IEEE) received the B.S. and M.S. degrees from the University of Electronic Science and Technology of China, China, in 2012 and 2015, respectively, and the Ph.D. degree from the City University of Hong Kong, China, in 2019. He was a Research Assistant and a Postdoctoral Fellow with the City University of Hong Kong. Since 2020, He has been with the School of Electronics and Communication Engineering, Sun Yat-sen University, China, where he is currently an Assistant Professor. His research interests include Internet of Things, modeling, and performance optimization of next-generation mobile communication systems.

Yayu Gao (Member, IEEE) received the B.S. degree in electronic engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2009, and the Ph.D. degree in electronic engineering from the City University of Hong Kong, Kowloon, Hong Kong, in 2014. Since March 2014, she has been with the School of Electronic Information and Communication, Huazhong University of Science and Technology,Wuhan, China, where she is an Associate Professor. Her research interests include decrentralized multiple access, next-generation WiFi networks, heterogeneous network coexistence, and network intelligence.

Yuan Jiang (Member, IEEE) received the B.Eng. degree in satellite communication and the M.Sc. degree in communication and electronic system from the Information Engineering University, Zhengzhou, China, in 1991 and 1998, respectively, and the Ph.D. degree in information and communication engineering from Zhejiang University, Hangzhou, China, in 2004. From 1991 to 2005, he was with the Department of Communication Engineering, Information Engineering University. From 2005 to 2008, he was with the Postdoctoral Workstation of Computer Science and Engineering, South China University of Technology, Guangzhou, China. From 2008 to 2018, he was successively a Vice General Manager, Chief Engineer, and the Vice President in a listed company. Since 2018, he has worked as a professor, doctoral supervisor, and director of a provincial key laboratory at Sun Yat-sen University. He has presided or participated in more than 20 research projects, including National Key Research and Development Program Special Project, etc. His research interests include cognitive communications, satellite communication networks, satellite navigation, and their applications to wireless communication systems.