

Joint Optimization of Handover Control and Power Allocation Based on Multi-Agent Deep Reinforcement Learning

Delin Guo^{1b}, Lan Tang^{1b}, *Member, IEEE*, Xinggan Zhang^{1b}, and Ying-Chang Liang^{1b}, *Fellow, IEEE*

Abstract—In this paper, we study the handover (HO), and power allocation problem in a two-tier heterogeneous network (HetNet), which consists of a macro base station, and some millimeter-wave (mmWave) small base stations. We establish an HO management, and power allocation scheme to maximize the overall throughput while reducing the HO frequency. In particular, considering the interrelationship among decisions made by different user equipments (UEs), we first model the HO, and power allocation problem as a fully cooperative multi-agent task, in which all agents, i.e., UEs, have the same target. Then, to solve the multi-agent task, and get decentralized policies for each UE, we develop a multi-agent reinforcement learning (MARL) algorithm based on the proximal policy optimization (PPO) method, by introducing the centralized training with decentralized execution framework. That is, we use global information to train policies for each UE, and after the training is finished, each UE obtains a decentralized policy, which can be implemented only based on each UE's local observation. Specially, we introduce the counterfactual baseline to address the credit assignment problem in centralized learning. Due to the centralized training, the decentralized policies learned by multi-agent PPO (MAPPO) can work more cooperatively. Finally, the simulation results demonstrate that our method can achieve better performance comparing with other existing works.

Index Terms—Handover, multi-agent deep reinforcement learning, power allocation, HetNet, mmWave.

I. INTRODUCTION

THE 5th generation (5G) networks are expected to meet the growing wireless communication demands. Due to the available broad spectrum and high expected data rate, the millimeter-wave (mmWave), which is in the range of 30–300 GHz, will play a key role in 5G [1]. On the other hand, due

to the limits imposed on the emitted power, the high-temperature noise and the high oxygen absorption, the transmission distance of mmWave signals is short [2]. Dense deployment of small base stations (SBSs) in traffic hotspots is considered one of the potential solutions to overcome shortcomings of mmWave.

In wireless networks, UEs need to switch the serving base station during the movement to maintain the communication quality. Traditionally, the handover (HO) process is determined by the HO events defined in the 3rd Generation Partnership Project (3GPP) protocol [3]. These HO events are triggered based on the periodical signal measurements of UEs, and some HO parameters defined in protocols, such as time-to-trigger (TTT), hysteresis, and event offset. In the dense mmWave network, the traditional HO policies in 3GPP may lead to the frequent HO problem and increase the HO overhead [4]. Therefore, it is crucial to establish an appropriate handover mechanism for mmWave dense systems.

In existing works, HO parameters were tuned to optimize the system performance [5]–[9]. [5] modified HO parameters, including hysteresis and TTT, to improve the network throughput, delay, and jitter. [6] investigated an adaptive TTT scheme to avoid the data link failure and ping-pong effect. In [7], a scheme based on the AHP-TOPSIS method and Q-learning approach was proposed for handover optimization, in which the AHP-TOPSIS method decided the serving base station, and Q-learning approach got the optimal TTT points. In [8], a double threshold algorithm was proposed to reduce unnecessary HOs in macrocell/femtocell integrated networks. [9] proposed a fuzzy Q-Learning algorithm to optimize the HO scheme in small-cell networks, which adjusted offset to reduce the number of dropped handover.

Besides, reinforcement learning (RL) is increasingly being applied to HO optimizations to improve the HO performance [10]. The HO process is modeled as a Markov decision process (MDP) [11] so that the optimal HO policy can be established by RL algorithms. Some existing works have formulated the HO process as a multi-arm bandit (MAB) problem, which is a variant of the RL problem. In [12], a MAB and satisfaction-based learning approach was proposed to improve the overall throughput and reduce the handover failure and ping-pong probabilities. In [13], the authors applied Upper Confidence Bound (UCB) algorithm to select the optimal base station in the single-user system, and in the multi-user system, the base station selection problem was formulated as a linear programming problem. [14]

Manuscript received March 4, 2020; revised June 12, 2020 and July 21, 2020; accepted August 25, 2020. Date of publication September 2, 2020; date of current version November 12, 2020. This work was supported in part by the National Natural Science Foundation of China under Grants 61976113 and U1936201, in part by the National Key R&D Program of China under Grant 2018YFB1801105, in part by the Key Areas of Research, and Development Program of Guangdong Province, China, under Grant 2018B010114001, in part by the Central Universities under Grant ZYGX2019Z022, and in part by the 111 Programme of Introducing Talents of Discipline to Universities under Grant B20064. The review of this article was coordinated by Prof. A. Jamalipour. (Corresponding author: Lan Tang.)

Delin Guo, Lan Tang, and Xinggan Zhang are with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210093, China (e-mail: guodelin@smail.nju.edu.cn; tanglan@nju.edu.cn; zhxg@nju.edu.cn).

Ying-Chang Liang is with the Center for Intelligent Networking, and Communications(CINC), University of Electronic Science, and Technology of China, Chengdu 610051, China (e-mail: liangyc@ieee.org).

Digital Object Identifier 10.1109/TVT.2020.3020400

considered UCB and Bayesian approaches for a parallel contextual MAB problem and applied them to wireless handover optimization. [15] also showed that the handover mechanism could be posed as a contextual MAB problem, where the context was the measurement report collected from the connected UEs, and the goal was to select a cell to maximize the expected reward. In [16], the UCB algorithm was adopted to derive the optimal HO policy, which minimized the HO rate while ensuring a certain throughput. All aforementioned work [12]–[16] had a similar methodology, which is first to formulate the HO process as a MAB problem, and then apply the algorithm like UCB to learn the optimal HO scheme.

In order to optimize the long-run performance, some works exploited the state transition characteristics of the wireless systems to design the HO scheme. [17] proposed a vertical handover management scheme based on the value iteration algorithm, where UEs made decisions according to their velocity and locations. [18] proposed an algorithm based on Q-learning, which decided whether or not to switch to other base stations according to the position and speed information of the pedestrians, by predicting future channel states. [19] proposed an image-to-decision proactive handover framework, which directly mapped pictures to an HO decision by adopting Deep Q-learning. [20] investigated a user association and resource allocation (UARA) problem in heterogeneous networks (HetNets) based on dueling double deep Q-learning, in which UEs made decisions depend on global state information obtained through message passing.

Similar work to ours was given in [21], where authors solved the HO problem in the multi-agent setting by adopting the asynchronous advantage actor-critic (A3C) [22], which is a policy-gradient (PG) based deep RL method. In [21], UEs made HO decisions based on individual candidates' RSRQs and current connection information by using a shared policy network—an artificial neural network (NN). Also, redundant HOs were reduced by adding an HO penalty to the reward function.

However, some issues in the multi-agent settings have not been considered in the above work. Firstly, although UEs shared the global NN, the A3C algorithm in [21] was equivalent to implementing the actor-critic algorithm on each UE independently, since UEs were treated as independent agents in A3C. However, UEs were interactive and the policies of UEs could continually change during the learning process, so that the environment state appeared non-stationary from the view of the UE, which violated Markov assumptions required for the convergence of RL and the independence assumption of agents in A3C. Thereby, it may cause performance degradation. Secondly, multi-agent settings would exacerbate the high variance property of PG based method [23]. Therefore, applying the actor-critic algorithm on each UE independently may lead to learning difficulties. Thirdly, to maximize the system throughput under limited communication resources, we should consider the cooperation and interaction among UEs. Otherwise, it may cause vicious competition and degrade the overall system performance.

In this paper, we aim to establish an HO management and power allocation scheme to maximize the throughput while reducing the handover frequency. We model the multiple-UE

handover and power allocation problem as a cooperative multi-agent reinforcement learning (MARL) problem. Also, we focus on addressing the potential instability and vicious competition brought by the independent implementation of the actor-critic method in [21]. The main contributions of this paper are as follows:

- Firstly, we consider the handover and power allocation problem in a HetNet system with multiple UEs. We maximize the overall throughput while reducing the handover frequency by determining the optimal policy between each UE's actions (HO decision and power allocation) and local observations, including the signal measurement report, current connection, and some public information. In particular, different from [21], we consider the interrelationship among UEs and model our problem as a fully cooperative multi-agent task. We then use a MARL method to learn the optimal cooperative policy for each UE in the HetNet.
- Secondly, we propose a multi-agent PPO (MAPPO) algorithm based on single-agent proximal policy optimization (PPO) in [24] for our multiple UEs system, by introducing the centralized training with decentralized execution framework. That is, we use global information to train policies for each UE, and after the training is finished, each UE obtains a decentralized policy, which makes decisions only based on each UE's local observation. Due to the centralized training, the decentralized policies learned by MAPPO can work more cooperatively. Thereby, the potential instability and vicious competition brought by the independent implementation of the actor-critic can be addressed.
- Thirdly, in the cooperative MARL problem, UEs have shared reward signals so that the learning process would face the credit assignment problem. That is, each UE's action's contribution to the global reward is not explicit, which may affect the learning performance. Therefore, we introduce the counterfactual baseline [25] to MAPPO to get better performance.
- Finally, we verify the feasibility of MAPPO via numerical simulation. We demonstrate our algorithm can achieve better performance by comparing it with other existing works, including the work in [21] and the multi-agent deep deterministic policy gradient (MADDPG) algorithm in [23].

The rest of this paper is organized as follows. Some relevant reinforcement learning backgrounds are introduced in Section II. The system model is presented in Section III. In Section IV, the HO and power allocation problem is formulated as a cooperative multi-agent task. The derivation and the practical implementation of the proposed MAPPO, which is used for tackling the cooperative MARL problems, are elaborated in Section V. The numerical simulations are provided in Section VI. Finally, Section VII concludes this work.

II. REINFORCEMENT LEARNING BACKGROUNDS

Before we introduce the system model, we first give some necessary reinforcement learning backgrounds.

multi-agent task: A fully cooperative N -agents task [25] can be described as a stochastic game Γ , defined by $\Gamma = \langle \mathcal{S}, \mathcal{A}, P, r, \mathcal{Z}, \mathcal{O}, N, \gamma \rangle$. At each time step t , $s_t \in \mathcal{S}$ describes the global state of the environment, and each agent (UE in our setup) $u \in \mathcal{N} = \{1, 2, \dots, N\}$ chooses an action $a_t^u \in \mathcal{A}$. Actions of all agents form a joint action $\mathbf{a}_t \in \mathcal{A} \equiv \mathcal{A}^N$. This incurs a transition in the environment according to the state transition function $P(s_{t+1}|s_t, \mathbf{a}_t) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$. All agents share the same reward function $r(s_t, \mathbf{a}_t)$ (abbreviated to r_t below): $\mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. $\gamma \in [0, 1]$ is the discount factor.

We consider a partially observable setting, in which agents draw individual observations $z \in \mathcal{Z}$ according to the observation function $\mathcal{O}(s) : \mathcal{S} \rightarrow \mathcal{Z}$. The agent u selects action based on a stochastic policy $\pi^u(a_t^u|z_t^u) : \mathcal{Z} \times \mathcal{A} \rightarrow [0, 1]$, where z_t^u denotes the local observation of the agent u .

The discount return is $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$. We denote the policy ensemble as $\Pi \equiv \{\pi^1, \dots, \pi^N\}$, and the joint policy as π . For the joint policy π , we will use the following standard definitions of the state-action value function Q^π , the state value function V^π , and the advantage function A^π :

$$\begin{aligned} Q^\pi(s_t, \mathbf{a}_t) &= \mathbb{E}_{s_{t+1}, \mathbf{a}_{t+1}, \dots} [R_t | s_t, \mathbf{a}_t], \\ V^\pi(s_t) &= \mathbb{E}_{\mathbf{a}_t, s_{t+1}, \dots} [R_t | s_t], \\ A^\pi(s_t, \mathbf{a}_t) &= Q^\pi(s_t, \mathbf{a}_t) - V^\pi(s_t), \\ a_t^u &\sim \pi^u(a_t^u | z_t^u), s_{t+1} \sim P(s_{t+1} | s_t, \mathbf{a}_t), \forall t \geq 0. \end{aligned} \quad (1)$$

We define the expected discount return as

$$\eta(\pi) = \mathbb{E}_{s_0 \sim \rho_0(s_0)} \{V^\pi(s_0)\}, \quad (2)$$

where $\rho_0(s_0)$ is the distribution of the initial state s_0 .

In such a fully cooperative multi-agent task, all agents cooperatively find the optimal decentralized policy ensemble Π^* forming a joint policy π^* which satisfies $\pi^* = \arg \max_{\pi} \eta(\pi)$.

Policy-based methods: Policy-based methods are commonly used for a variety of traditional RL tasks, i.e., single-agent tasks. In policy-based methods, the policy, i.e., the actor, is typically parameterized with parameters θ , and such methods directly apply the policy gradient on the parameters θ by gradient ascent to maximize the expected discount return $\eta(\pi_\theta)$, where the most commonly used policy gradient form is in [26]

$$\Delta\theta = \mathbb{E}_{s_0, a_0, s_1, \dots} \left[\sum_{t=0}^{\infty} \nabla_{\theta} \log \pi_{\theta}(a_t | s_t) A^{\pi}(s_t, a_t) \right]. \quad (3)$$

In RL, we estimate the policy gradient based on the following formula

$$\Delta\theta = \hat{\mathbb{E}}_t [\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) A(s_t, a_t)], \quad (4)$$

where $\hat{\mathbb{E}}_t[\cdot]$ is the sample average and $A(s_t, a_t)$ is the estimation of the true $A^\pi(s_t, a_t)$. How to estimate the advantage function $A^\pi(s_t, a_t)$ gives rise to several practical algorithms. Actor-critic approaches in [11] and [26] estimate A^π based on a critic, which is usually the estimation of Q^π or V^π . For example, considering a parameterized critic $V_{\omega}(s_t)$ which estimates V^π , and then $A^\pi(s_t, a_t)$ can be estimated by the temporal-deference (TD) method: $A(s_t, a_t) = r_t + \gamma V_{\omega}(s_{t+1}) - V_{\omega}(s_t)$. The critic needs to be updated to approach to the true value function. For

the critic $V_{\omega}(s_t)$, we update the parameters ω by minimizing the loss

$$J(\omega) = \hat{\mathbb{E}}_t [(V_{\omega}(s_t) - y_t)^2], \quad (5)$$

where $y_t = r_t + \gamma V_{\omega}(s_{t+1})$, and V_{ω} is the target state-value function [27] and the parameters ω are periodically updated with the most recent ω to stabilize learning process. To solve the above optimization problem, we can adopt stochastic optimization algorithms such as stochastic gradient descent (SGD) or Adam [28]. Actor-critic iteratively updates θ and ω using sampled data by interacting with the environment.

It is worth mentioning that the aforementioned approach is just a typical example of the policy-based framework, and the detailed form may vary in different practical algorithms. Policy gradient methods are known to exhibit high variance gradient estimates, and it is exacerbated in multi-agent settings. In some cases, the probability of taking a gradient step in the correct direction decreases exponentially with the number of agents [23].

III. SYSTEM MODEL AND PROPAGATION MODEL

A. System Model

We consider a two-tier heterogeneous network (HetNet), which consists of M mmWave small base stations (SBSs) and N moving UEs, as well as a macro base station (MBS) using microwave band. We denote the base stations set as $\mathcal{M} = \{0, 1, 2, \dots, M\}$, where 0 is the index of the MBS, and denote the UE set as $\mathcal{N} = \{1, 2, \dots, N\}$. We assume the total number of UEs in the HetNet will not change [17]. Besides, we consider an exemplary model: each UE is served by one SBS or MBS. Coordinated multiple point transmission (COMP), which allows a UE to be served by multiple base stations (BSs) are not considered herein.

The microwave MBS adopts an omnidirectional antenna and eliminates intra-interference by assigning non-overlapping sub-channels to served UEs equally. However, UEs served by the MBS would suffer from inter-cell interference from neighbor MBSs. We assume that the mmWave SBSs are homogeneous and equipped with massive antennas. Because of the excellent directivity of the mmWave beams, the intra-interference of SBSs can be eliminated by appropriate spatial precoding when the number of UEs is relatively small. Thereby, we assume all SBSs employ the same frequency band, and UEs can access to the full bandwidth of each SBS. We denote the available bandwidth of SBSs and MBS as W_S and W_M , respectively.

As shown in Fig. 1, during the move, UEs may switch to another SBS or MBS and require resources (bandwidth or power) from the connected base station. We set the number of UEs served by each BS as the public information. Each UE makes decisions, including BS selection and power requirement, based on its local observations, which consist of signal measurement report, current serving BS, and the public information. The BS allocates transmission power to served UEs proportionally based on power requirements from all UEs under the constraint of total power. Details about power allocation are given in the next section. We assume the system operates in a synchronous

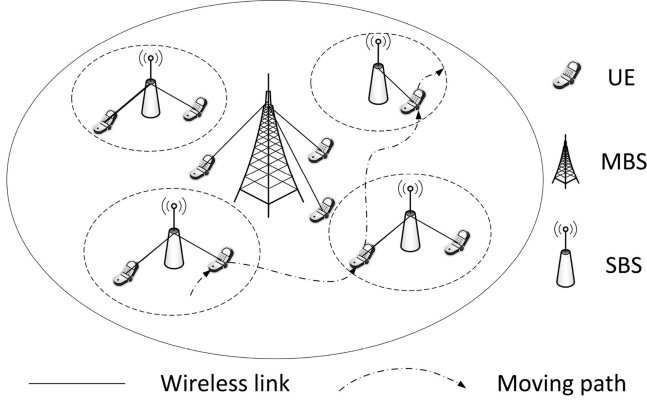


Fig. 1. An illustration of a HetNet system.

equal-length time-step fashion [4], where the time is slotted as $t = 0, 1, \dots$. The length of the time-step is denoted as ΔT . In each time-step, the UEs can switch connection from the current serving BS to another BS, or maintain the current connection, and then the BSs allocate transmission power to served UEs according to all requirements.

We assume the MBS also plays a role of a central processor (CP), which can obtain global information in the centralized training framework and provide cloud computing for our learning missions. We will propose the MAPPO by introducing centralized training with decentralized execution framework. The CP of the MBS learns a centralized critic and a decentralized policy for each UE using our learning algorithm, which would be presented in section V. In the learning process, each UE updates its policy from the recent learning results of the CP periodically.

B. Propagation Model

In this subsection, we give the signal propagation models between SBS (MBS) and UEs.

1) propagation model of the microwave MBS: We assume that the MBS is equipped with an omnidirectional antenna to guarantee coverage area so that there is no extra antenna gain. We adopt the path loss (in dB) model for microwave band in [17], which is expressed by

$$PL(d)[dB] = \alpha_M + 10\kappa_M \log_{10}(d) + \psi + \xi, \quad (6)$$

where d is the distance in meters, κ_M is the path loss exponent which represents the slope of the best linear fit to the propagation measurement in the microwave band, α_M is the path loss factor, ψ is random small-scale fading, and ξ is the random lognormal shadowing.

For UE u , let d_u^0 denote the distance between the UE u and MBS, and p_u^0 denotes the transmission power of MBS allocated to the UE u , which satisfies $\sum_u p_u^0 = P_M$. For the microwave band, we need to consider cochannel interference due to the shared bandwidth. When the UE u is connected to MBS, the received SINR can be expressed as

$$\text{SINR}_u = \frac{PL^{-1}(d_u^0)p_u^0}{I_u + N_M w_u^0}, \quad (7)$$

where I_u is the cochannel interference, N_M is the noise power spectral density of the MBS, and w_u^0 denotes the bandwidth allocated to the UE u by the MBS. The interference is the sum power received on the UE side, which is from MBSs of adjacent macrocells.

2) propagation model of mmWave SBSs: For SBSs, because of the unique propagation characteristic of mmWave, we consider a probabilistic LOS-NLOS model in 3GPP standard [13]. That is, the channel state between UEs and SBSs can vary between the two states: Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS). Let ϑ_u^m denote the LOS probability of the link between the SBS $m \in \mathcal{M}, m \neq 0$ and the UE u . ϑ_u^m can be obtained by using the high-resolution map of surroundings [17]. [29] proposed an estimation method for LOS probability by utilizing the building density of the area. According to [29], the LOS probability between the SBS m and the UE u is given by¹

$$\vartheta_u^m = \exp\left(-\frac{2D_B X_B d_u^m}{\pi}\right), m \neq 0 \quad (8)$$

where D_B is the building density, X_B is the expectation length of buildings, and d_u^m is the distance between the UE u and the BS m .

Similar to [13], the path loss model for SBSs is given by

$$pl(d)[dB] = \alpha_S + 10\kappa_S \log_{10}(d), \quad (9)$$

where d is the distance, α_S and κ_S are the path loss factor and exponential decay factor, respectively. The values of α_S and κ_S are different for LOS and NLOS states. The small-scale fading ψ in (6) can be neglected in mmWave band. Besides, the shadowing ξ in (6) is also ignored for SBSs since the probabilistic LOS-NLOS model emulates a similar effect to shadowing.

We assume that all SBSs are equipped with directional antennas that are necessary to support beamforming and beam tracking for mmWave system, and UEs are equipped with omnidirectional antennas so that the antenna gains are only accounted at SBSs side. As in [13] and [30], antenna gain is given by

$$g(\phi) = \begin{cases} g_{\max}, & \text{if } |\phi| < \frac{\phi_s}{2} \\ g_{\min}, & \text{otherwise} \end{cases} \quad (10)$$

where ϕ is the angle between UE and SBS, and ϕ_s is the width of the antenna main lobe. We assume the perfect beam tracking is performed. Therefore, the UE is always in the main lobe, so that UEs obtain a high antenna gain $g_{\max}$ when connecting to a SBS.

The interference can be ignored for mmWave-SBS [13], [30]. For a UE u , the Signal to Noise Ratio (SNR) when associated with the SBS m can be expressed as

$$\text{SNR}_u^m = \frac{g_{\max} pl^{-1}(d_u^m) p_u^m}{N_S W_S}, m \in \mathcal{M} \& m \neq 0, \quad (11)$$

where p_u^m is the transmission power allocated to the UE u by the SBS m , which satisfies the total power constraint $\sum_u p_u^m = P_S$, and N_S is the noise power spectral density of SBSs.

¹Here, π is Pi, not the policy.

We assume that the length of the time step ΔT is smaller than the coherent time of the channel so that SNRs and SINRs of all UEs keep constant in each time step.

IV. PROBLEM FORMULATION

We aim to optimize the handover and power allocation to maximize the overall throughput and reduce the handover frequency in the HetNet system. Considering the decisions of UEs are interactive, we model the problem in the multiple UEs system as a fully cooperative multi-agent task. As mentioned in section II, this problem can be represented by $\Gamma = \langle \mathcal{S}, \mathcal{A}, P, r, \mathcal{Z}, \mathcal{O}, N, \gamma \rangle$.

A. Action

The action of each UE includes BS selection and power requirement. In order to ensure that the power is sufficient to satisfy UEs' basic requirements, the total power P_S is divided into P_S^e for adaptive allocation and P_S^a for equal allocation. Similarly, the total power P_M of MBS is separated into $P_M = P_M^a + P_M^e$.

At the time step t , the action of UE u can be expressed as a tuple

$$a_t^u = (b_t^u, q_t^u), \quad (12)$$

where $b_t^u \in \mathcal{M} \equiv \{0, 1, \dots, M\}$ is the selected BS, and $q_t^u \in \{0, 1, \dots, q_{\max}\}$ is the discrete power requirement factor which suggests how much power should be allocated to UE u by the selected BS b_t^u . At the beginning of each time slot, each UE u sends the power requirement factor q_t^u to the target BS b_t^u , and then the target BS allocates power to served UEs according to the proportion of power requirement factors. Suppose $b_t^u = m \in \mathcal{M}$. When $m = 0$, the MBS allocates the transmission power $p_u^0(t)$ to UE u based on the following criterion

$$p_u^0(t) = \frac{q_t^u P_M^a}{\sum_{i \in \mathcal{N}} \mathbb{1}\{b_t^i = 0\} q_t^i} + \frac{P_M^e}{\sum_{i \in \mathcal{N}} \mathbb{1}\{b_t^i = 0\}}, \quad (13)$$

where $\mathbb{1}\{\cdot\}$ is the indicator function and $\sum_i \mathbb{1}\{b_t^i = 0\}$ indicates the total number of UEs connected to MBS. When $m \neq 0$, the transmission power allocated by SBS m to UE u is

$$p_u^m(t) = \frac{q_t^u P_S^a}{\sum_{i \in \mathcal{N}} \mathbb{1}\{b_t^i = m\} q_t^i} + \frac{P_S^e}{\sum_{i \in \mathcal{N}} \mathbb{1}\{b_t^i = m\}}, \quad (14)$$

where $\sum_i \mathbb{1}\{b_t^i = m\}$ indicates the total number of UEs connected to SBS m .

We denote $\mathcal{A}$ as the action space of UEs, i.e., $a_t^u \in \mathcal{A}, \forall u \in \mathcal{N}$. The joint action of all UEs is defined as

$$\mathbf{a}_t = (a_t^1, a_t^2, \dots, a_t^N). \quad (15)$$

It can be seen from (13) and (14) that the power allocation of each UE is decided by all $\{q_t^u\}$ s of UEs connected to the same BS. Obviously, the greedy power allocation policy $q_t^u = q_{\max}$ which maximizes the individual reward is not optimal in terms of maximizing the common reward in (21). Therefore, we will optimize the power allocation by centralized training, as stated at the end of this section.

B. State and Observation

The observation of each UE includes its BS connection, signal measurement, and the the number of UEs served by each BS.

The current serving BS of the UE u is chosen at the last time step, i.e., b_{t-1}^u . The public information is sent to UEs by the MBS at the start of each time step. For the BS $m \in \mathcal{M}$, the number of served UEs $U_t^m = \sum_{u \in \mathcal{N}} \mathbb{1}\{b_{t-1}^u = m\}$. Therefore, the public information at time step t is $\mathbf{U}_t = (U_t^0, U_t^1, \dots, U_t^M)$. The signal measurements from MBS and SBSs to UE u are denoted by $S_0^u(t)$ and $S_m^u(t), m \neq 0$, respectively. At the beginning of each time step t , each BS broadcasts reference signals to all UEs, and then each UE u can measure the SINR and SNRs of reference signals from MBS or SBSs. Denoting $\mathbf{S}_t^u = (S_0^u(t), S_1^u(t), \dots, S_M^u(t))$ as the signal measurements for UE u at the time step t , the observation of the UE u at the time step t can be represented by

$$z_t^u = (\mathbf{S}_t^u, \mathbf{U}_t, b_{t-1}^u). \quad (16)$$

We define the global state as the ensemble of observations of all UEs:

$$\mathbf{s}_t = (z_t^1, z_t^2, \dots, z_t^N) \in \mathcal{S}, \quad (17)$$

where $\mathcal{S}$ is the state space.

C. Reward

Firstly, for a UE u with $b_t^u = m$, the throughput can be expressed as

$$R^u(z_t^u, \mathbf{a}_t) = \begin{cases} W_S \log(1 + \text{SNR}_u^m), & m \neq 0 \\ w_u^0 \log(1 + \text{SINR}_u), & m = 0 \end{cases} \quad (18)$$

where $w_u^0 = \frac{W_M}{\sum_{u \in \mathcal{N}} \mathbb{1}\{b_t^u = 0\}}$.

Secondly, we define a HO penalty: if the target BS b_t^u is different from the current serving BS b_{t-1}^u , there will be a HO penalty. The HO penalty of UE u is defined as

$$C_{HO}^u(z_t^u, \mathbf{a}_t) = \mathbb{1}\{b_t^u \neq b_{t-1}^u\}. \quad (19)$$

Besides, there is a interruption when HO occurs. We denote $T_q \leq \Delta T$ as the interruption time. Therefore, the local reward of UE u in the time step t is defined as

$$r^u(z_t^u, \mathbf{a}_t) = (\Delta T - T_q C_{HO}^u(z_t^u, \mathbf{a}_t)) R^u(z_t^u, \mathbf{a}_t) - \beta C_{HO}^u(z_t^u, \mathbf{a}_t), \quad (20)$$

where $\beta \geq 0$ is the weighting factor.

To maximize the long-term overall throughput and reduce the HO frequency, we model this problem as a fully cooperative multi-agent task, where UEs have the following common reward

$$r(\mathbf{s}_t, \mathbf{a}_t) = \sum_{u=1}^N r^u(z_t^u, \mathbf{a}_t). \quad (21)$$

This common reward can guide RL algorithms to achieve the tradeoff between overall throughput and HO frequency by adjusting the weighting factor β . For convenience, in the rest of the paper, we abbreviate $r(\mathbf{s}_t, \mathbf{a}_t)$ as r_t and $r^u(z_t^u, \mathbf{a}_t)$ as r_t^u .

D. Problem Formulation

For the multi-agent task Γ , our objective is to maximize $\eta(\pi)$ defined in (2) by determining the optimal policy π . One idea is to consider a fully centralized learning framework, where all UEs are treated as a single super-agent. In the fully centralized learning framework, we assume CP can know the global state s_t and joint action $\mathbf{a}_t$, and learn a centralized policy π using traditional RL algorithms such as PPO [24]. In this case, the problem can be formulated as

$$\max_{\pi} \eta(\pi), \quad (22)$$

where the centralized policy π makes decisions for all UEs based on the global state, i.e., $\mathbf{a}_t \sim \pi(\mathbf{a}_t|s_t)$. Thereby, the global state s_t is always needed in both the learning and execution phase. However, making decisions according to all UEs' observations is usually impractical, since it may cause the privacy problem [31] and heavy signaling overhead. Besides, the centralized policy π learned by a single-agent RL method may be the best in theory. However, in practical implementations, the action space of π grows exponentially with the number of UEs N , which makes the learning algorithm difficult to obtain the optimal solution with increase of N .

Therefore, we aim to determine decentralized policies for each UE, rather than a centralized policy, by centralized learning. In particular, at the beginning of the time step t , UE $u, \forall u \in \mathcal{N}$ selects an action a_t^u based on the current individual observation z_t^u using the decentralized policy π^u , i.e., $a_t^u \sim \pi^u(a_t^u|z_t^u)$. We assume that during the learning phase, each BS m can know (z_t^u, a_t^u, r_t^u) of all UEs with $b_t^u = m$, and CP in MBS can obtain $(s_t, \mathbf{a}_t, r_t)$ by communication with SBSs. The centralized learning is implemented in CP based on sampled trajectories $\tau = \{s_t, \mathbf{a}_t, r_t\}_{t=1}^T$, where T is the sample length. Once the optimal decentralized policies are determined, UEs can make decision based on local observation. Therefore, the problem can be formulated as

$$\max_{\Pi=\{\pi^1, \dots, \pi^N\}} \eta(\pi). \quad (23)$$

Since each UE makes decisions independently, the joint policy π is to the product of individual policies,

$$\pi(\mathbf{a}_t|s_t) = \prod_{u=1}^N \pi^u(a_t^u|z_t^u). \quad (24)$$

V. LEARNING ALGORITHM FOR MULTI-AGENT TASK

In this section, we investigate the optimization method for solving (23). The optimal joint policy π^* can possibly be found by applying local iterative ascent. That is, starting with an initial guess denoted by π_{old} , update to π such that the target function η increases at each iteration of the algorithm, i.e., $\eta(\pi) > \eta(\pi_{old})$.

Lemma 1 [32]: Let π_{old} denote the current policy. For an arbitrary updated policy π , the expected discount return $\eta(\pi)$

can be expressed as

$$\begin{aligned} \eta(\pi) &= \eta(\pi_{old}) + \mathbb{E}_{s_0, \mathbf{a}_0, \dots \sim \pi} \left\{ \sum_{t=0}^{\infty} \gamma^t A^{\pi_{old}}(s, \mathbf{a}) \right\} \\ &= \eta(\pi_{old}) + \sum_s \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi) \sum_{\mathbf{a} \in \mathcal{A}} \pi(\mathbf{a}|s) A^{\pi_{old}}(s, \mathbf{a}) \\ &= \eta(\pi_{old}) + \sum_s \rho_{\pi}(s) \sum_{\mathbf{a} \in \mathcal{A}} \pi(\mathbf{a}|s) A^{\pi_{old}}(s, \mathbf{a}) \\ &= \eta(\pi_{old}) + \frac{1}{1-\gamma} \mathbb{E}_{\pi} \{ A^{\pi_{old}}(s, \mathbf{a}) \}, \end{aligned} \quad (25)$$

where $\rho_{\pi}(s)$ is the (unnormalized) discounted visitation frequency

$$\rho_{\pi}(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi), \quad \forall s \in \mathcal{S}, \quad (26)$$

and the initial state $s_0 \sim \rho_0(s_0)$. In (25), $\mathbb{E}_{\pi} \{ \dots \} = \sum_s \bar{\rho}_{\pi}(s) \sum_{\mathbf{a} \in \mathcal{A}} \pi(\mathbf{a}|s) [\dots]$, in which $\bar{\rho}_{\pi}(s) = (1-\gamma)\rho_{\pi}(s)$ is the normalized discounted visitation frequency. In order to avoid confusion, we stress that $\rho_0(s_0)$ and $\rho_{\pi}(\cdot)$ are different notations.

It can be seen from (25) that to guarantee $\eta(\pi) \geq \eta(\pi_{old})$, there should be $\mathbb{E}_{\pi} \{ A^{\pi_{old}}(s, \mathbf{a}) \} \geq 0$. However, it is difficult to optimize the Equation (25) directly due to the complex dependency of $\rho_{\pi}(s)$ on π . Therefore, we make a following approximation to $\eta(\pi)$

$$\begin{aligned} F_{\pi_{old}}(\Pi) &= \eta(\pi_{old}) \\ &+ \frac{1}{1-\gamma} \mathbb{E}_{\pi_{old}} \left\{ \sum_{u=1}^N \frac{\pi^u(a^u|z^u)}{\pi_{old}^u(a^u|z^u)} A^{\pi_{old}}(s, \mathbf{a}) \right\}. \end{aligned} \quad (27)$$

For each UE u , we assume the current policy π_{old}^u is a differentiable function of the parameter vector θ_{old}^u , and π^u is a differentiable function of the parameter vector θ^u . We denote $\theta_{old} = \{\theta_{old}^1, \dots, \theta_{old}^N\}$ and $\theta = \{\theta^1, \dots, \theta^N\}$. When $\theta = \theta_{old}$, we have the following Lemma

Lemma 2:

$$\begin{aligned} F_{\pi_{old}}(\Pi)|_{\theta=\theta_{old}} &= \eta(\pi)|_{\theta=\theta_{old}} \\ \nabla_{\theta} F_{\pi_{old}}(\Pi)|_{\theta=\theta_{old}} &= \nabla_{\theta} \eta(\pi)|_{\theta=\theta_{old}}. \end{aligned} \quad (28)$$

Proof: According to the definition of $F_{\pi_{old}}(\Pi)$ and $\eta(\pi)$, we have

$$F_{\pi_{old}}(\Pi)|_{\theta=\theta_{old}} = \eta(\pi)|_{\theta=\theta_{old}} = \eta(\pi_{old}), \quad (29)$$

since

$$\mathbb{E}_{\pi_{old}} \{ A^{\pi_{old}}(s, \mathbf{a}) \} = 0. \quad (30)$$

The gradient of $\eta(\pi)$ is given by

$$\nabla_{\theta} \eta(\pi) = \sum_s \rho_{\pi}(s) \sum_{\mathbf{a}} \nabla_{\theta} \pi(\mathbf{a}|s) A^{\pi_{old}}(s, \mathbf{a})$$

$$\begin{aligned}
&= \sum_s \rho_\pi(s) \sum_a \pi(a|s) \nabla_\theta \log \pi(a|s) A^{\pi_{old}}(s, \mathbf{a}) \\
&= \sum_s \rho_\pi(s) \sum_a \pi(a|s) \sum_u \frac{\nabla_\theta \pi^u(a^u|z^u)}{\pi^u(a^u|z^u)} A^{\pi_{old}}(s, \mathbf{a}).
\end{aligned} \tag{31}$$

Therefore, when $\theta = \theta_{old}$, the following equation holds

$$\begin{aligned}
&\nabla_\theta \eta(\pi)|_{\theta=\theta_{old}} \\
&= \frac{1}{1-\gamma} \mathbb{E}_{\pi_{old}} \left\{ \sum_{u=1}^N \frac{\nabla_\theta \pi^u(a^u|z^u)}{\pi_{old}^u(a^u|z^u)} A^{\pi_{old}}(s, \mathbf{a}) \right\} |_{\theta=\theta_{old}} \\
&= \nabla_\theta F_{\pi_{old}}(\Pi)|_{\theta=\theta_{old}}.
\end{aligned} \tag{32}$$

Lemma 2 implies that for a given π_{old} , when the update step from θ_{old} to θ is sufficiently small, improving the objective $F_{\pi_{old}}(\Pi)$ will also guarantee the target $\eta(\pi)$ is non-decreasing. However, it is not explicit that how big of an update step to take. Therefore, based on Lemma 2, we derive the following Theorem,

Theorem 1: Let $D_{KL}(\cdot||\cdot)$ denote the Kullback-Leibler (KL) divergence which can measure the similarity between two probability distributions, and $D_{KL}^{\max}(\pi_{old}^u||\pi^u) = \max_{z^u} D_{KL}(\pi_{old}^u(\cdot|z^u)||\pi^u(\cdot|z^u))$, $\forall u$. Let

$$M_{\pi_{old}}(\Pi) = F_{\pi_{old}}(\Pi) - C \max_u \sqrt{D_{KL}^{\max}(\pi_{old}^u, \pi^u)}, \tag{33}$$

where $C = \frac{4N\epsilon}{(1-\gamma)}$ and $\epsilon = \max_{s, \mathbf{a}} |A^{\pi_{old}}(s, \mathbf{a})|$. Then we have the following results

$$\begin{aligned}
\eta(\pi) &\geq M_{\pi_{old}}(\Pi), \\
\eta(\pi_{old}) &= M_{\pi_{old}}(\Pi_{old}),
\end{aligned} \tag{34}$$

where $\Pi_{old} = \{\pi_{old}^u, \dots, \pi_{old}^N\}$.

Proof: We provide the proof in Appendix A.

Theorem 1 implies that for a given π_{old} , maximizing $M_{\pi_{old}}(\Pi)$ guarantees the non-decreasing of the original objective $\eta(\pi)$. Therefore, maximizing $M_{\pi_{old}}(\Pi)$ iteratively will finally solve the original problem. In order to take larger update steps in a robust way, similar to [32], we maximize $F_{\pi_{old}}(\Pi)$ with constraints $D_{KL}^{\max}(\pi_{old}^u, \pi^u) \leq \delta$, $\forall u$. By expanding $F_{\pi_{old}}(\Pi)$ with (27), the problem of maximizing $M_{\pi_{old}}(\Pi)$ can be reformulated as

$$\max_{\theta} \mathbb{E}_{\pi_{old}} \left\{ \sum_{u=1}^N \frac{\pi^u(a^u|z^u)}{\pi_{old}^u(a^u|z^u)} A^{\pi_{old}}(s, \mathbf{a}) \right\} \tag{35a}$$

$$\text{subject to : } D_{KL}^{\max}(\pi_{old}^u||\pi^u) \leq \delta, \forall u. \tag{35b}$$

We call (35b) as *the trust region constraints*, which restrain the update size of the decentralized policies.

However, the trust region constraints are imposed on every observation point, which makes the problem in (35) impractical to solve. In order to make the optimization process more simple, similar to the proximal policy optimization (PPO) in [24], we remove the trust region constraints by applying the clip function on each individual policy, and transform the problem in (35) to

the following unconstrained problem

$$\max_{\{\theta^1, \dots, \theta^N\}} \mathbb{E}_{\pi_{old}} \left\{ \sum_{u=1}^N f(r(\theta^u), A^{\pi_{old}}(s, \mathbf{a})) \right\} \tag{36}$$

where

$$\begin{aligned}
&f(r(\theta^u), A^{\pi_{old}}(s, \mathbf{a})) \\
&= \min\{r(\theta^u) A^{\pi_{old}}(s, \mathbf{a}), c^\epsilon(r(\theta^u)) A^{\pi_{old}}(s, \mathbf{a})\}.
\end{aligned} \tag{37}$$

In (37), the clip function $c^\epsilon(x)$ restricts x into the interval $[1 - \epsilon, 1 + \epsilon]$, and $r(\theta^u) = \frac{\pi^u(a^u|z^u)}{\pi_{old}^u(a^u|z^u)}$ is the probability ratio for UE u .

As the equation (36) implies, the incentive moving each $r(\theta^u)$ outside of the interval $[1 - \epsilon, 1 + \epsilon]$ is removed. Thereby, the policy update can be limited to a small range. Empirically, the clipping mechanism can effectively restrict the $r(\theta^u)$ to the clip range [24].

Lemma 3: For a certain ϵ , if $1 - \epsilon \leq r(\theta^u) \leq 1 + \epsilon$ holds for all z^u and a^u , the trust region constraints in (35b) could also be guaranteed.

Proof: Suppose a discrete action space.² If $\log \frac{1}{1-\epsilon} \leq \delta$, we have

$$\begin{aligned}
&D_{KL}(\pi_{old}^u(\cdot|z^u)||\pi^u(\cdot|z^u)) \\
&= \sum_{a^u \in A} \pi_{old}^u(a^u|z^u) \log \frac{\pi_{old}^u(a^u|z^u)}{\pi^u(a^u|z^u)} \\
&\leq \sum_{a^u \in A} \pi_{old}^u(a^u|z^u) \log \left(\frac{1}{1-\epsilon} \right) = \log \frac{1}{1-\epsilon} \leq \delta.
\end{aligned} \tag{38}$$

The problem in (36) can be solved by the gradient ascent method. The gradient of the objective function is

$$\Delta \theta = \mathbb{E}_{\pi_{old}} \left\{ \sum_{u=1}^N \nabla_\theta f(r(\theta^u), A^{\pi_{old}}(s, \mathbf{a})) \right\}. \tag{39}$$

Therefore, for each individual policy π^u , the policy gradient is

$$\Delta \theta^u = \mathbb{E}_{\pi_{old}} \{ \nabla_{\theta^u} f(r(\theta^u), A^{\pi_{old}}(s, \mathbf{a})) \}, \forall u. \tag{40}$$

Based on the policy gradient (40), we will investigate two practical algorithms to obtain the sub-optimal decentralized policies $\{\pi^1, \dots, \pi^N\}$: IPPO, which is a straightforward decentralized algorithm, and MAPPO based on centralized training framework.

A. Independent PPO

Inspired by the independent Q-learning in [33], we propose an independent proximal policy optimization (IPPO). IPPO is to implement the PPO algorithm on each UE independently, where each UE learns with its own actor and critic. In IPPO, each UE u learns according to the individual trajectory $\tau^u = \{z_t^u, a_t^u, r_t\}_{t=1}^T$, where r_t is the common reward broadcast by MBS to all UEs.

²It is straightforward to extend the result to a continuous action space by replacing the sums with integrals.

It can be seen from the policy gradients in (40), the policy gradient $\Delta\theta^u$ depends on decentralized policy π^u and joint advantage function $A^{\pi_{old}}(s, \mathbf{a})$. In the independent learning, due to the lack of global state s_t and joint action $\mathbf{a}_t$, we can only estimate the advantage function in (40) for each UE u based on the shared reward r_t and each UE's local information.

For each UE u , the critic is the parameterized function $V_{\omega^u}(z^u)$ with parameters ω^u , and the advantage function $A^{\pi_{old}}(s_t, \mathbf{a}_t)$ is estimated by the truncated version of generalized advantage estimation (GAE) in [26],

$$A_t^u = \delta_t + (\gamma\lambda)\delta_{t+1} + \dots + (\gamma\lambda)^T \delta_T, \quad (41)$$

where $\delta_t = r_t + \gamma V_{\bar{\omega}^u}(z_{t+1}^u) - V_{\bar{\omega}^u}(z_t^u)$, and $V_{\bar{\omega}^u}$ is the target local critic of UE u .

In the practical implementation, the policy gradient for each UE u is calculated as

$$\Delta\theta^u = \nabla_{\theta^u} \hat{\mathbb{E}}_t \{f(r_t(\theta^u), A_t^u)\}, \quad (42)$$

where $\hat{\mathbb{E}}_t\{\cdot\}$ is the sample average, and $r_t(\theta^u) = \frac{\pi^u(a_t^u|z_t^u)}{\pi_{old}^u(a_t^u|z_t^u)}$.

As suggested in Sec. II, the critic also need to be updated to approach to the accurate value function. According to [24], [34], we update each critic by minimizing the loss function

$$J(\omega^u) = \hat{\mathbb{E}}_t [(y_t^u - V_{\omega^u}(z_t^u))^2], \quad (43)$$

where $y_t^u = A_t^u + V_{\bar{\omega}^u}(z_t^u)$.

In summary, in IPPO, for the given policy π_{old}^u with parameters θ_{old}^u and corresponding trajectory τ^u , to get the sub-optimal solution to (36), we iteratively update θ^u with the gradient in (42) and ω^u by minimizing (43). Then, θ_{old}^u and $\bar{\omega}^u$ are updated iteratively until the learning process terminals.

The learning process for each UE u is presented in Algorithm 1, where L is the maximum iteration rounds, K is the number of training epoch, and B is the minibatch size.

However, the absence of the global state causes that the local critic cannot well approximate the true global state-value function $V^{\pi_{old}}(s_t)$. Thereby, the advantage function $A^{\pi_{old}}(s_t, \mathbf{a}_t)$ can not be estimated accurately. Besides, due to the lack of joint action information, it is challenging to learn cooperative policies, and for each UE, it is difficult to estimate the influences of its actions on the common reward.

B. Multi-Agent PPO

In this subsection, we develop a MAPPO algorithm in the centralized training with decentralized execution framework. That is, we use global information to train decentralized policies for each UE. In the practical system, we assume the global information is collected by MBS, and the learning task is finished in CP of MBS.

Note that the policy gradients in (40) are related to the joint advantage function $A^{\pi_{old}}(s, \mathbf{a})$. Therefore, we consider a decentralized actors and centralized critics framework. We assume that the global information, including s_t and $\mathbf{a}_t$, is available for training. The centralized critic estimates the joint value function (Q or V) based on global information. The decentralized actors make decisions only based on UEs' local observations. After the training process, global information is not needed, and UEs can

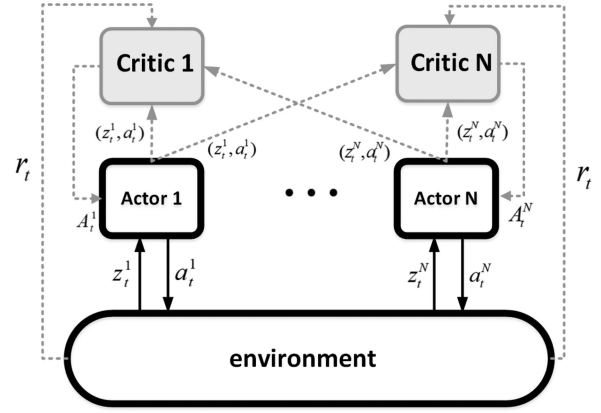


Fig. 2. Overview of our centralized critic and decentralized actor framework.

Algorithm 1: PPO for Each UE u .

Initiate V_{ω^u} , π^u . Initiate π_{old} with $\theta_{old}^u \leftarrow \theta^u$, and $V_{\bar{\omega}^u}$ with $\bar{\omega}^u \leftarrow \omega^u$
for iteration = 1, 2, ..., L
for an episode $t=1, 2, \dots, T$
UE executes action according to $\pi_{old}(a_t^u|z_t^u)$
Get the reward r_t , and the next observation z_{t+1}^u
end for
Get a trajectory: $\tau = \{z_t^u, a_t^u, r_t\}_{t=1}^T$
Compute advantages $\{A_t^u\}_{t=1}^T$ using (42)
Compute $y_t^u = V_{\bar{\omega}^u}(z_t^u) + A_t^u$
Get data $\{z_t^u, a_t^u, y_t^u, A_t^u\}_{t=1}^T$
for $k=1, 2, \dots, K$
Shuffle the data's order and renumber
for $j=0, 1, \dots, \frac{T}{B} - 1$
Select B group of data $\{z_i^u, a_i^u, y_i^u, A_i^u\}_{i=1+Bj}^{B(j+1)}$
Compute gradient:
 $\Delta\theta^u = \frac{1}{B} \sum_{i=1}^B \{\nabla_{\theta^u} f(r_i(\theta^u), A_i^u)\}$
 $\Delta\omega^u = \frac{1}{B} \sum_{i=1}^B \{\nabla_{\omega^u} (y_i^u - V_{\omega^u}(z_i^u))^2\}$
Apply gradient ascent on θ^u using $\Delta\theta^u$ by Adam
Apply gradient descent on ω^u using $\Delta\omega^u$ by Adam
end for
end for
update $\theta_{old}^u \leftarrow \theta^u$ and $\bar{\omega}^u \leftarrow \omega^u$
end for

choose actions in a decentralized manner. In Fig. 2, we depict the overview of the centralized training with decentralized executing framework, where the gray and dotted line parts only are needed during the training process.

An explicit centralized critic is the estimated state-value function $V_{\omega^u}(s_t)$, where ω^u is the critic parameters of the UE u . In RL, the expectation is replaced by sample averages. Hence, we update each policy using the following gradient,

$$\Delta\theta^u = \nabla_{\theta^u} \hat{\mathbb{E}}_t \{f(r_t(\theta^u), A^u(s_t, \mathbf{a}_t))\}, \quad (44)$$

where $A^u(s_t, \mathbf{a}_t)$ is the estimation of $A^{\pi_{old}}(s_t, \mathbf{a}_t)$, and it is calculated by GAE with the target value function $V_{\bar{\omega}^u}(s_t)$. We

must stress that, although UEs have their own critics, these critics estimate the same state-value function and are trained based on the same data. Thereby, all value functions V_{ω^u} with different u should be the same in theory, so that such critics provide the same advantage function estimation for each UE. Therefore, it is not explicit that how a particular agent's actions contribute to the common reward, which causes the credit assignment problem [25].

In order to address the credit assignment problem, we adopt the counterfactual baseline proposed by [25]. We utilize a centralized critic $Q_{\omega^u}(s_t, \mathbf{a}_t)$ to estimate action-value function $Q^{\pi_{old}}(s_t, \mathbf{a}_t)$. We denote joint quantities over UEs other than a given UE u with the superscript $-u$, e.g., a joint action other than the UE u is $\mathbf{a}_t^{-u}$. Then the advantage function for each UE u is calculated by comparing the Q-value estimated by the critic for the executed action a_t^u to a counterfactual baseline that marginalizes out a_t^u , while keeping actions of other UEs fixed:

$$A^u(s_t, \mathbf{a}_t) = \hat{Q}^u(s_t, \mathbf{a}_t) - b(s_t, \mathbf{a}_t^{-u}), \quad (45)$$

where $b(s_t, \mathbf{a}_t^{-u}) = \sum_{a^u} \pi_{old}^u(a^u | z_t^u) Q_{\omega^u}(s_t, (\mathbf{a}_t^{-u}, a^u))$ is the counterfactual baseline, and $\hat{Q}^u(s_t, \mathbf{a}_t)$ is the estimation of $Q^{\pi_{old}}(s_t, \mathbf{a}_t)$. $\hat{Q}^u(s_t, \mathbf{a}_t)$ is computed by the truncated version of TD(λ) [11]

$$\hat{Q}^u(s_t, \mathbf{a}_t) = Q_{\bar{\omega}^u}(s_t, \mathbf{a}_t) + \delta_t + (\gamma\lambda)\delta_{t+1} + \dots + (\gamma\lambda)^T \delta_T, \quad (46)$$

where the TD error $\delta_t = r_t + \gamma Q_{\bar{\omega}^u}(s_{t+1}, \mathbf{a}_{t+1}) - Q_{\bar{\omega}^u}(s_t, \mathbf{a}_t)$. Therefore, (45) computes a separate advantage function for each UE that uses the centralized critic to infer counterfactuals in which only the UE u 's action changes. Although each $\hat{Q}^u(s_t, \mathbf{a}_t)$ is calculated by using different critics, they estimate the same joint action-value function $Q^{\pi_{old}}(s_t, \mathbf{a}_t)$.

In this paper, we consider a discrete action space $\mathcal{A}$. The output node of the deep neural network (DNN) of a typical critic is 1. When the state-action pair $(s_t, \mathbf{a}_t)$ is input to the critic, we get a scalar $Q_{\omega^u}(s_t, \mathbf{a}_t)$. Thereby, computing the counterfactual baseline for one UE needs $|\mathcal{A}|$ times evaluation. It would be time-consuming when the action space size is large. In order to improve the evaluation efficiency, we adopt the critic structure in [25]. As presented in Fig. 3, the input of the NN of the critic for agent u is $(s_t, \mathbf{a}_t^{-u})$, and then the network outputs all the state-action values of the agent u . The critic structure in Fig. 3 requires that each UE has a critic. Otherwise, the critic can not distinguish which UE's action is marginalized.

Algorithm: The whole pseudo-code of MAPPO is presented in Algorithm 2. We construct the critic for each UE with the structure in Fig. 3. The critic Q_{ω^u} and actor π^u are parameterized by ω^u and θ^u , respectively. We assign the policy parameters θ^u and ω^u to the current policy π_{old}^u and the target critic $Q_{\bar{\omega}^u}$, respectively. L is the maximum iteration round.

In the period of data collecting, each UE interacts with the environment using policy π_{old}^u . Because the time scale can be regarded as infinity in wireless communication systems, we do not need to consider the terminal state. In each round of iteration, each UE u collects a T -timestep trajectory τ^u . Then in the CP

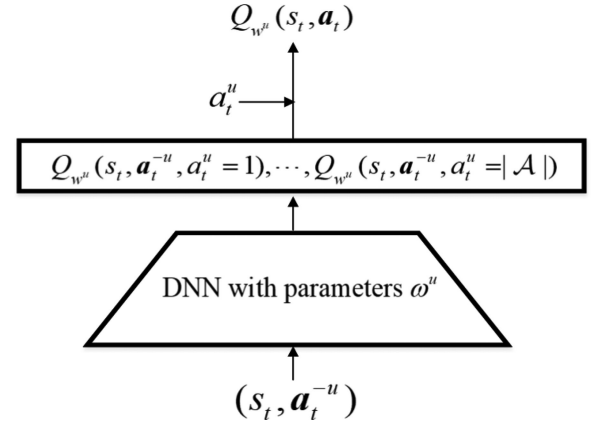


Fig. 3. The diagram of the critic's structure.

of MBS, $\{A^u(s_t, \mathbf{a}_t)\}_{t=1}^T$ and $\{\hat{Q}^u(s_t, \mathbf{a}_t)\}_{t=1}^T$ are calculated according to (45) and (46), respectively.

In the period of training, we update the parameters iteratively in K epochs with collected data to improve the sample efficiency. In each epoch, we randomly shuffle the order of data and reorder them, by which the correlations between data are broken and therefore the training process is stabilized [27]. With these sampled data, θ^u for all u are updated using the gradient in (44) with minibatch Adam [28]. Also, for each u , ω^u is updated by minimizing the loss function $J(\omega^u) = (\hat{Q}^u(s_t, \mathbf{a}_t) - Q_{\omega^u}(s_t, \mathbf{a}_t))^2$. After K epochs, we update θ_{old}^u and $\bar{\omega}^u$ to new θ^u and ω^u for all UEs.

VI. NUMERICAL SIMULATION

In this section, we first give the simulations setups. Then we present the simulation results with discussion and analysis.

A. Simulations Setups

HetNet settings: We consider a two-tier heterogeneous network deployed in urban areas, which consists of a microwave MBS and several mmWave SBSs. Let M_S denote the number of SBSs. We default $M_S = 4$, and $N = 10$. We describe the location by Cartesian coordinates in meters. We assume a circular area with a radius of 500 meters, where the MBS is located at the origin (0,0), and four SBSs are located at (400,0), (0,400), (-400,0), and (0,-400), respectively. The default total transmission power of SBSs and MBS are $P_S = 3W$, and $P_M = 120W$, respectively. Half of P_S and P_M are allocated adaptively, i.e., $P_S^a = \frac{P_S}{2}$ and $P_M^a = \frac{P_M}{2}$. The log-normal shadow fading of MBS ξ has zero mean and standard deviation 3 dB, and the small-scale fading in non-dB form $10^{\frac{\xi}{10}}$ obeys an exponential distribution with unit mean. We summarize other simulation parameters in the table I. We assume UEs are randomly distributed in the area at the beginning of each learning episode. We adopt the Random Direction model in [35] to simulate the UEs' mobility patterns. The velocity is uniformly distributed in the range [3, 20] m/s.

We demonstrate the simulation results by comparing our work with following benchmark algorithms:

TABLE I
SIMULATION PARAMETERS OF THE HETNET

Notations	Definition
Total bandwidth of SBSs W_S	100MHz[17]
Total bandwidth of an MBS W_M	20MHz[17]
Pathloss parameters α_S, κ_S for LOS	$\alpha_S = 70, \kappa_S = 2$ [17]
Pathloss parameters α_S, κ_S for NLOS	$\alpha_S = 82, \kappa_S = 3$ [13]
Pathloss parameter α_M, κ_M	$\alpha_M = 38, \kappa_M = 3$ [17]
mmWave noise power density N_S	-164dBm/Hz[13]
Microwave noise power density N_M	-174dBm/[13]
The high antenna again g_{max}	10dB
Interval of each timestep ΔT	100ms
Interruption time T_q	100ms[17]
Building density D_B	$1 \times 10^{-4}/m^2$ [30]
Expectation of buildings' length X_B	25m[30]
Maximum power allocation factor q_{max}	4

Algorithm 2: Multi-Agent PPO.

Initiate critic Q_{ω^u} and actor π^u with $\theta^u, \forall u$.
Initiate the current policies π_{old}^u with $\theta_{old}^u \leftarrow \theta^u$, and the target critic $Q_{\bar{\omega}^u}$ with $\bar{\omega}^u \leftarrow \omega^u$.
Initiate a memory buffer D
for iteration = 1, 2, ..., L
 s_1 =initiate state
 for an episode $t=1, 2, \dots, T$
 Each agent u executes action according to $\pi_{old}^u(a_t^u | z_t^u)$
 Get the reward r_t , and the next state s_{t+1}
 end for
 Get a trajectory for each UE u : $\tau^u = \{z_t^u, a_t^u, r_t\}_{t=1}^T$
 Compute $\{\hat{Q}^u(s_t, \mathbf{a}_t)\}_{t=1}^T$ according to (46)
 Compute advantages $\{A^u(s_t^u, \mathbf{a}_t)\}_{t=1}^T$ according to (45)
 Store data $\{[z_t^u, a_t^u, \hat{Q}^u(s_t, \mathbf{a}_t), A^u(s_t, \mathbf{a}_t)]_{u=1}^N\}_{t=1}^T$ into D
 for $k=1, 2, \dots, K$
 Shuffle and renumber the data's order
 for $j=0, 1, \dots, \frac{T}{B} - 1$
 Select B group of data D_j :
 $D_j = \{[z_i^u, a_i^u, \hat{Q}^u(s_i, \mathbf{a}_i), A^u(s_i, \mathbf{a}_i)]_{u=1}^N\}_{i=1+Bj}^{B(j+1)}$
 for $u=1, 2, \dots, N$
 $\Delta\theta^u = \frac{1}{B} \sum_{i=1}^B \{\nabla_{\theta^u} f(r_i(\theta^u), A^u(s_i, \mathbf{a}_i))\}$
 $\Delta\omega^u = \frac{1}{B} \sum_{i=1}^B \{\nabla_{\omega^u} (\hat{Q}^u(s_i, \mathbf{a}_i) - Q_{\omega^u}(s_i, \mathbf{a}_i))^2\}$
 Apply gradient ascent on θ^u using $\Delta\theta^u$ by Adam
 Apply gradient descent on ω^u using $\Delta\omega^u$ by Adam
 end for
 end for
 end for
 Update $\theta_{old}^u \leftarrow \theta^u$ and $\bar{\omega}^u \leftarrow \omega^u$ for each UE u
 Empty D
end for

- 1) A3C: [21] adopted A3C in [22] to address the HO problem. In [21], UEs are treated as independent agents, and the interactions among UEs are neglected. Each UE calculates the policy gradient using the individual trajectory with local reward, i.e., $\{z_t^u, a_t^u, r_t^u\}_{t=1}^T$, according to the equation (4). Then, the calculated policy gradients of each UE are asynchronously pushed to the global server for updating policy parameters. The policy of each UE will be updated from the global server, periodically. For comparison fairness, when comparing with [21], we default that the reward in the learning process is the shared reward r_t rather than the local reward r_t^u .
- 2) MADDPG [23]: This method extends the deep deterministic policy gradient (DDPG) to multi-agent settings by introducing a centralized training with decentralized execution framework. In MADDPG, each UE trains a private policy and a centralized critic according to the global information $\tau = \{s_t, \mathbf{a}_t, r_t\}$, and the policy gradient is calculated by

$$\Delta\theta^u = \hat{\mathbb{E}}_t \{ \nabla_{\theta^u} \pi^u(z_t^u) \nabla_{\mathbf{a}^u} Q_{\omega^u}(s_t, \mathbf{a}_t^u, \mathbf{a}^u) |_{\mathbf{a}^u = \pi^u(z_t^u)} \}, \quad (47)$$

where π^u is the deterministic policy, and Q_{ω^u} is the centralized critic. Besides, the derivation in equation (47) holds only for continuous actions. To support discrete action space in this paper, the Gumbel-Softmax estimator in [36] is adopted.

- 3) Central-V: Abbreviation for the MAPPO with central V, which is a variant of MAPPO in Algorithm 2. In this algorithm, we construct a centralized critic to estimate the global state-value function, i.e., $V_{\omega^u}(s_t)$, and then the advantage function $A^u(s_t, \mathbf{a}_t)$ is calculated by GAE using the target value function $V_{\bar{\omega}^u}(s_t)$. The other processes are the same as those in Algorithm 2.
- 4) CC-PPO: Abbreviation for the central controller obtained by solving the problem (22) with the single-agent PPO in [24]. In particular, we apply PPO to obtain the centralized policy π by treating all UEs in the system as a super single-agent.
- 5) σ_{sig} : A simple Signal-based HO algorithm. UEs perform HOs only based on the received signal measurements. Specifically, UEs always switch to the BS with the highest SNR or SINR when their SNR or SINR is lower than 0 dB, and the power is allocated to connected UEs equally.

Hyperparameters: For MAPPO, IPPO, Central-V, and CC-PPO, we use the Adam optimizer with a learning rate $\alpha = 3.5 \times 10^{-4}$ which is linearly annealed to 0 in the learning progress. We parameterize the policies and critics by adopting the two-layer Rectified Linear Unit (ReLU) Multi-Layer Perceptron (MLP) with 64 units per layer. Although many existing works have adopted recurrent neural networks (RNN), we use MLP for simplicity. We set the maximum length of each episode $T = 1024$, the number of epoch $K = 4$, the number of minibatch $B = 4$, $\lambda = 0.5$, and the discount factor $\gamma = 0.9$. The clip range $\varepsilon = 0.2$ is the best empirical value [24]. For A3C and MADDPG, we also

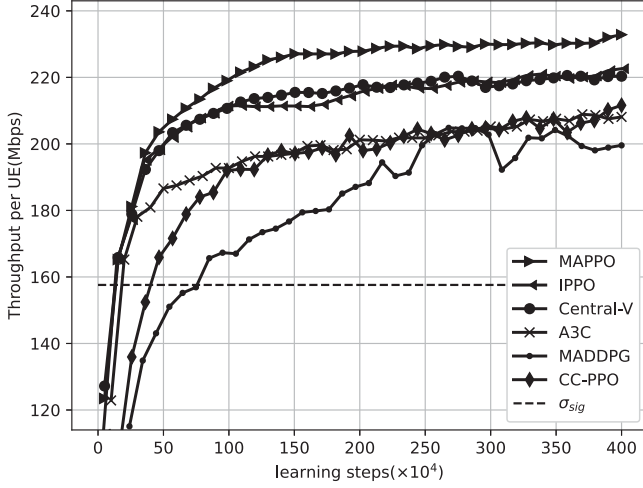


Fig. 4. Comparison of throughput performance between different algorithms.

adopt MLP as approximator and use the default hyperparameters in [34] and [23] respectively, except for $\gamma = 0.9$.

Simulation platform: The simulation of the HetNet system and algorithm programs are constructed in python 3.6, and the neural networks are built and trained by using Tensorflow [37] with version 1.7.0.

B. Simulation Results

In this subsection, unless otherwise specified, all learning algorithms are guided by the shared reward r_t with the HO penalty weight factor $\beta = 0$.

Fig. 4 compares the throughput performance of different approaches in the same HetNet system. It is shown that our MAPPO significantly outperforms other methods, and σ_{sig} has the worst performance. Although CC-PPO might be the best in theory, it performs badly in the practical implementations. A plausible cause is that learning a centralized policy with a large action space is more difficult than learning multiple decentralized policies with small action space. Surprisingly, IPPO achieves similar throughput performance to the Central-V. A possible reason for the relatively poor performance of Central-V is the lack of explicit guidance for updating decentralized policies because of the same advantage estimation. This result verifies the benefit brought by applying the counterfactual baseline. By comparing A3C and IPPO, we can find that although both A3C and IPPO have not considered the interrelationship among UEs, IPPO achieves more reasonable policies than A3C. This result suggests that maximizing the clipped objective in IPPO can achieve higher throughput than maximizing the traditional objective in A3C. Among all learning algorithms, the rate of convergence of MADDPG is the slowest. In MADDPG, 2.5 million steps are required to make the throughput surpass 200 Mbps. Although MADDPG also adopted a centralized training with decentralized execution framework, the deterministic policy has a poor exploration capability, and is easy to cause overestimation of the Q function and thus bring severe fluctuation of the learning curves. As a result, it performs relatively poorly in the considered HetNet system.

TABLE II
THE COMPARISON OF AVERAGE THROUGHPUT (MBPS) BETWEEN MAPPO AND CC-PPO OVER DIFFERENT NUMBER OF UES

Method \ N	2	4	6	8	10
MAPPO	302	276	265	249	238
CC-PPO	313	277	261	235	222

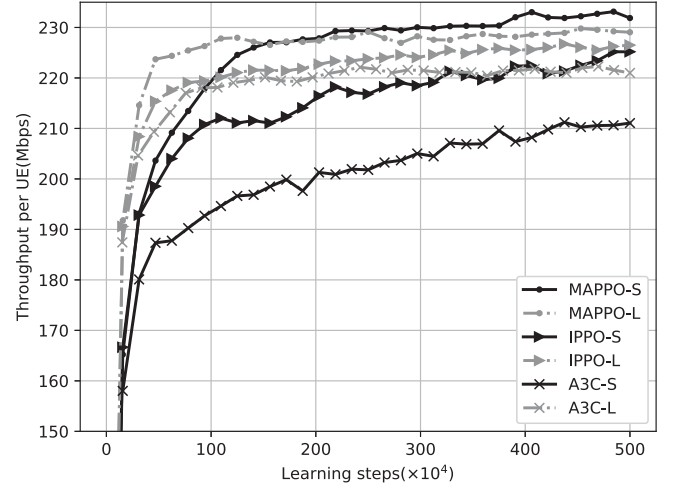


Fig. 5. Throughput performance between different algorithms with different reward conditions.

In Table II, we compare the average throughputs of MAPPO and CC-PPO with different numbers of UEs. These results are obtained by running 5 million steps of both two methods. When the number of UEs is relatively small ($N = 2$ and 4), the fully centralized learning method, CC-PPO, is superior to the proposed MAPPO. However, when $N = 6$, the performance of MAPPO overtakes CC-PPO, and the performance gap increases with the increase of N . These comparisons further verify that learning a centralized policy with a large action space is more difficult than learning multiple decentralized policies with small action space. Therefore, MAPPO is a more suitable learning algorithm than CC-PPO in the HetNet system with large number of UEs.

In Fig. 5, we compare the throughput performance of different algorithms (MAPPO, IPPO, A3C) with shared and local rewards. We use the suffix ‘-S’ to indicate methods with shared reward and ‘-L’ to indicate the methods with the local reward. In the method with shared reward, the learning process is guided by the shared reward r_t , while in the method with local reward, the actor and critic of each UE u are learned using the local reward r_t^u . For example, in MAPPO-L, the shared reward r_t in Fig. 2 is replaced by the local reward r_t^u . The simulation results show that MAPPO-S achieves the maximum throughput since parameters in critics are optimized based on shared rewards. Although MAPPO-S and IPPO-S achieve higher steady-state value than MAPPO-L and IPPO-L, respectively, MAPPO-L and IPPO-L have faster rate of convergence than their counterparts. Because the global state and joint action information are used during learning, MAPPO-L performs better than other methods except MAPPO-S. Simulation results suggest that in some cases

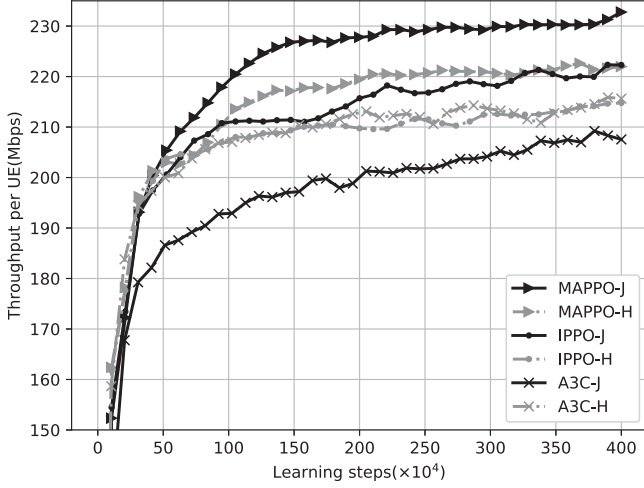


Fig. 6. The throughput performance comparisons between HO-only and joint optimization scheme.

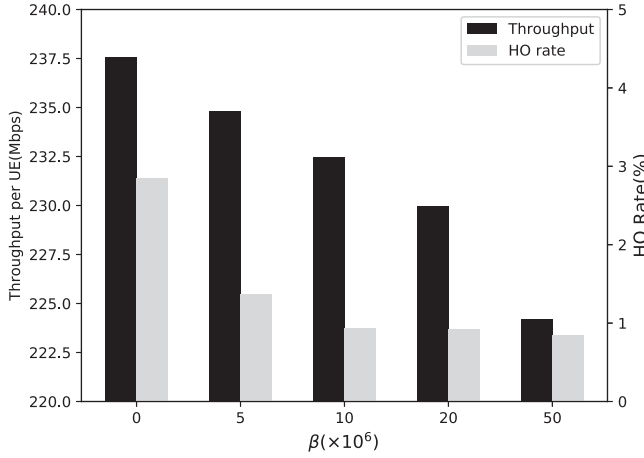


Fig. 7. The tradeoff between HO rate and throughput with different β .

with low-latency requirements, MAPPO-L can be applied to reduce the learning time.

In Fig. 6, we investigate the performance gain brought by power allocation by comparing joint optimization with HO-only optimization. In the HO-only scheme, we only optimize HO decisions, and the transmit power is equally allocated to UEs served by the same BS. The suffix ‘-H’ and ‘-J’ are used to indicate the HO-only scheme and joint optimization scheme, respectively. It is shown that, for MAPPO and IPPO, joint optimization can achieve a notable performance gain compared with HO-only schemes. In A3C, joint optimization achieves worse performance than HO-only partly because the power allocation leads to a larger action space and thus enhances the complexity of the learning process. Therefore, in comparison with A3C, MAPPO and IPPO can still perform excellently in a larger action space.

In Fig. 7, we vary the the weight factor of HO penalty β in the reward to balance the throughput and HO frequency. We run MAPPO for 5×10^6 steps and test 10^4 independent Monte-Carlo episodes. There are 25 time steps in each episode. Obviously, increasing β encourages the algorithm to lower both HO frequency

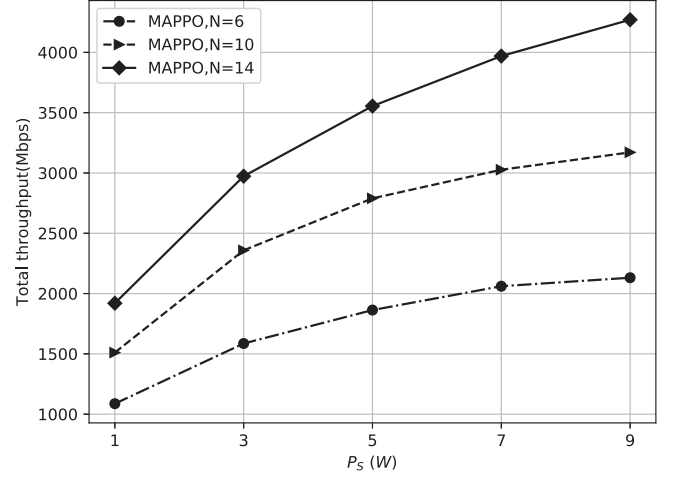


Fig. 8. Overall throughput versus different transmission power using MAPPO with different N .

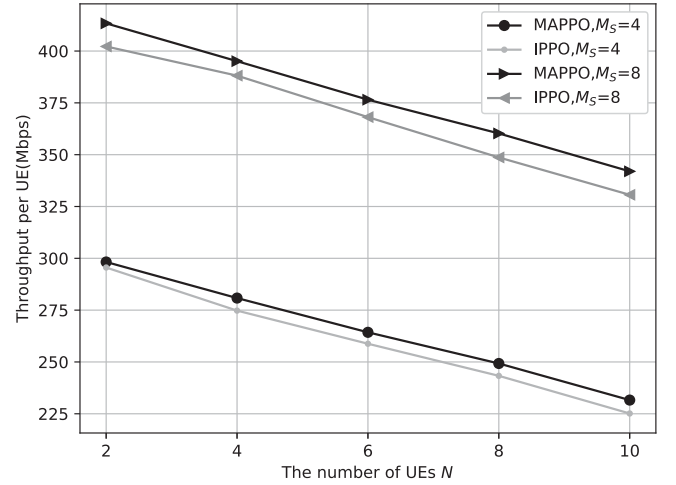


Fig. 9. Average throughput per UE versus different number of UEs.

and throughput. We can find that the HO frequency decreases significantly from 2.85% to 1.36% as β increases from 0 to 5×10^6 . However, with the increasing of β , the effect of β on the HO rate gradually decreases. Specifically, when β increases from 10×10^6 to 50×10^6 , the HO frequency remains approximately unchanged (slightly decreases from 0.94% to 0.85%), while the throughput has an apparent degradation. Results in Fig. 7 provide meaningful reference for choosing appropriate β to achieve the best tradeoff between throughput and HO rate.

In Fig. 8, we illustrate the overall throughput of the HetNet system versus the different transmission power P_S when MAPPO is applied. P_M is fixed to 120 W. For the same number of UEs N , the total throughput increases with the increasing of P_S . Meanwhile, we can see that due to multiuser diversity, the power utilization efficiency improves as N increases. In Fig. 9, we present the throughput per UE versus different number of UEs in HetNet systems with $M_S = 4$ and $M_S = 8$. In the system with $M_S = 8$, the four added SBSs are located at $(100\sqrt{2}, 100\sqrt{2})$, $(-100\sqrt{2}, 100\sqrt{2})$, $(-100\sqrt{2}, -100\sqrt{2})$,

and $(100\sqrt{2}, -100\sqrt{2})$, respectively. It shows that the throughput per UE decreases with the increasing of N . When comparing IPPO with MAPPO, we can find that the throughput gap in the system with $M_S = 8$ is bigger than that in the system with $M_S = 4$, which implies that MAPPO performs better than IPPO in a larger action space.

VII. CONCLUSION

This paper has investigated the joint HO and power allocation in a HetNet system with multiple UEs, in which the objective is to maximize the overall throughput while reducing the HO frequency. The problem is formulated as a fully cooperative multi-agent task. To solve the multi-agent task, we derive the theoretical lower bound of the objective function and policy gradients of decentralized policies. Based on derived policy gradients, a MARL algorithm-MAPPO, is developed. In MAPPO, we apply global information to train decentralized policies for each UE. Simulation results demonstrate that MAPPO can achieve better performance than existing methods in the considered HetNet system, and the frequent HO problem can be relieved by adjusting the HO penalty factor in the reward function.

APPENDIX PROOF OF THEOREM 1

Because the proof needs the coupling and total variation divergence theory in [38], we begin the proof with the following definition and proposition.

Definition 1 [32]: For each u , (π_{old}^u, π^u) is an α^u -coupled policy pair if it defines a joint distribution $(a_{old}^u, a^u)|z^u$, such that $P(a_{old}^u \neq a^u|z^u) \leq \alpha^u$ for all z^u . π_{old}^u and π^u denote the marginal distributions of a_{old}^u and a^u , respectively.

Let $D_{TV}(\pi_{old}^u(\cdot|z^u)||\pi^u(\cdot|z^u)) = \frac{1}{2} \sum_{a^u} |\pi_{old}^u(a^u|z^u) - \pi^u(a^u|z^u)|$ denote the total variation divergence between the probability distributions of $\pi_{old}^u(\cdot|z^u)$ and $\pi^u(\cdot|z^u)$, and denote $D_{TV}^{\max}(\pi_{old}^u, \pi^u) = \max_{z^u} D_{TV}(\pi_{old}^u(\cdot|z^u)||\pi^u(\cdot|z^u))$. Based on the proposition 4.7 and remark 4.8 in [38], we have the following proposition.

Proposition 1: For each u , if $D_{TV}^{\max}(\pi_{old}^u, \pi^u) = \alpha^u$, there exists a joint distribution $(a_{old}^u, a^u)|z^u$, whose marginal distributions are π_{old}^u and π^u , for which $P(a_{old}^u \neq a^u|z^u) \leq \alpha^u$ holds for all z^u .

This proposition will be used in the proof of Lemma 1.

To obtain the inequality relationship in (34), we first expand Equations in (25) and (27), respectively,

$$\eta(\pi) = \eta(\pi_{old}) + \sum_{t=0}^{\infty} \gamma^t \sum_{s_t} P(s_t|\pi) \mathbb{E}_{\mathbf{a} \sim \pi} \{A^{\pi_{old}}(s_t, \mathbf{a})\}, \quad (48)$$

where $\mathbb{E}_{\mathbf{a} \sim \pi} \{\cdot\} = \sum_{\mathbf{a}} \pi(\mathbf{a}|s)[\cdot]$, and

$$F_{\pi_{old}}(\Pi) = \eta(\pi_{old}) + \sum_{t=0}^{\infty} \gamma^t \sum_{s_t} P(s_t|\pi_{old}) \sum_{u=1}^N \mathbb{E}_{\mathbf{a} \sim \pi_{old}} \left\{ \frac{\pi^u(a^u|z_t^u)}{\pi_{old}^u(a^u|z_t^u)} A^{\pi_{old}}(s_t, \mathbf{a}) \right\}. \quad (49)$$

For simplicity, we define $\tilde{\pi}_u(\mathbf{a}|s) = \pi_{old}(\mathbf{a}|s) \frac{\pi^u(a^u|z^u)}{\pi_{old}^u(a^u|z^u)}$ so that $\mathbb{E}_{\mathbf{a} \sim \pi_{old}} \left\{ \frac{\pi^u(a^u|z_t^u)}{\pi_{old}^u(a^u|z_t^u)} [\cdot] \right\}$ in (49) can be replaced by $\mathbb{E}_{\mathbf{a} \sim \tilde{\pi}_u} [\cdot]$. Then, Equation (49) can be re-denoted as

$$F_{\pi_{old}}(\Pi) = \eta(\pi_{old}) + \sum_{t=0}^{\infty} \gamma^t \sum_{s_t} P(s_t|\pi_{old}) \sum_{u=1}^N \mathbb{E}_{\mathbf{a} \sim \tilde{\pi}_u} \{A^{\pi_{old}}(s_t, \mathbf{a})\}. \quad (50)$$

Based on Equations (48) and (50), we get

$$\begin{aligned} |\eta(\pi) - F_{\pi_{old}}(\Pi)| &= \sum_{t=0}^{\infty} \gamma^t \left| \sum_{s_t} P(s_t|\pi) \mathbb{E}_{\mathbf{a} \sim \pi} \{A^{\pi_{old}}(s_t, \mathbf{a})\} \right. \\ &\quad \left. - \sum_{s_t} P(s_t|\pi_{old}) \sum_{u=1}^N \mathbb{E}_{\mathbf{a} \sim \tilde{\pi}_u} \{A^{\pi_{old}}(s_t, \mathbf{a})\} \right| \\ &\leq \sum_{t=0}^{\infty} \gamma^t \cdot \left(\sum_{s_t} P(s_t|\pi) |\mathbb{E}_{\mathbf{a} \sim \pi} \{A^{\pi_{old}}(s_t, \mathbf{a})\}| \right. \\ &\quad \left. + \sum_{s_t} P(s_t|\pi_{old}) \sum_{u=1}^N |\mathbb{E}_{\mathbf{a} \sim \tilde{\pi}_u} \{A^{\pi_{old}}(s_t, \mathbf{a})\}| \right) \\ &\stackrel{(e)}{\leq} \sum_{t=0}^{\infty} \gamma^t \cdot (2N\alpha^* \epsilon \sum_{s_t} P(s_t|\pi_{old}) + 2N\alpha^* \epsilon \sum_{s_t} P(s_t|\pi)) \\ &= \sum_{t=0}^{\infty} \gamma^t 4N\alpha^* \epsilon = \frac{4N\alpha^* \epsilon}{1 - \gamma}, \end{aligned} \quad (51)$$

where $\alpha^* = \max_u D_{TV}^{\max}(\pi_{old}^u, \pi^u), \forall u$, $\epsilon = \max_{s, \mathbf{a}} |A^{\pi_{old}}(s, \mathbf{a})|$, and the inequality (e) holds due to the following Lemma.

Lemma 4: Let $\alpha^u = D_{TV}^{\max}(\pi_{old}^u, \pi^u), \forall u$, $\alpha^* = \max_u \alpha^u$, and $\epsilon = \max_{s, \mathbf{a}} |A^{\pi_{old}}(s, \mathbf{a})|$, we have

$$|\mathbb{E}_{\mathbf{a} \sim \pi} \{A^{\pi_{old}}(s, \mathbf{a})\}| \leq 2N\alpha^* \epsilon, \quad (52)$$

and

$$\sum_{u=1}^N |\mathbb{E}_{\mathbf{a} \sim \tilde{\pi}_u} \{A^{\pi_{old}}(s, \mathbf{a})\}| \leq 2N\alpha^* \epsilon. \quad (53)$$

Proof: For any coupled policy pair $(\pi_{old}^u, \pi^u), \forall u$, whose marginal distributions are π_{old}^u and π^u , respectively, we have

$$\begin{aligned} |\mathbb{E}_{\mathbf{a} \sim \pi} \{A^{\pi_{old}}(s, \mathbf{a})\}| &\stackrel{(e)}{=} |\mathbb{E}_{(\mathbf{a}, \mathbf{a}_{old}) \sim (\pi, \pi_{old})} \{A^{\pi_{old}}(s, \mathbf{a}) - A^{\pi_{old}}(s, \mathbf{a}_{old})\}| \\ &= P(\mathbf{a}_{old} \neq \mathbf{a}|s) \cdot \\ &|\mathbb{E}_{(\mathbf{a}, \mathbf{a}_{old}) \sim (\pi, \pi_{old}) | \mathbf{a} \neq \mathbf{a}_{old}} \{A^{\pi_{old}}(s, \mathbf{a}) - A^{\pi_{old}}(s, \mathbf{a}_{old})\}| \\ &\leq P(\mathbf{a}_{old} \neq \mathbf{a}|s) \cdot 2 \max_{s, \mathbf{a}} |A^{\pi_{old}}(s, \mathbf{a})| \\ &= \left(1 - \prod_{u=1}^N P(a_{old}^u = a^u|z^u) \right) \cdot 2\epsilon, \end{aligned} \quad (54)$$

where (e) follows from $\mathbb{E}_{\mathbf{a} \sim \pi_{old}} \{A^{\pi_{old}}(s, \mathbf{a})\} = 0$. According to Proposition 1, there exists an α^u -coupled policy pair (π_{old}^u, π^u) for each u , such that $P(a_{old}^u \neq a^u | z^u) \leq \alpha^u$ for all z^u . Therefore, for each u , when (π_{old}^u, π^u) is α^u -coupled, we have

$$\left(1 - \prod_{u=1}^N P(a_{old}^u = a^u | z^u)\right) \leq \left(1 - \prod_{u=1}^N (1 - \alpha^u)\right). \quad (55)$$

Since (54) holds for any coupled policy pair (π_{old}^u, π^u) with marginals π_{old}^u and π^u , it must hold for α^u -coupled policy pair. Therefore, based on (54) and (55), we get

$$\begin{aligned} |\mathbb{E}_{\mathbf{a} \sim \pi} \{A^{\pi_{old}}(s, \mathbf{a})\}| &\leq 2(1 - \prod_{u=1}^N (1 - \alpha^u))\epsilon \\ &\leq 2(1 - (1 - \alpha^*)^N)\epsilon \leq 2N\alpha^*\epsilon. \end{aligned} \quad (56)$$

Next, we will prove (53) with the similar method. For any coupled policy pair $(\pi_{old}, \tilde{\pi}_u)$ that defines a joint distribution $(\mathbf{a}_{old}, \tilde{\mathbf{a}}) | s$, where π_{old} and $\tilde{\pi}_u$ are the marginals of $\mathbf{a}_{old}$ and $\tilde{\mathbf{a}}$, respectively, we have

$$\begin{aligned} |\mathbb{E}_{\mathbf{a} \sim \tilde{\pi}_u} \{A^{\pi_{old}}(s, \mathbf{a})\}| &= |\mathbb{E}_{(\tilde{\mathbf{a}}, \mathbf{a}_{old}) \sim (\tilde{\pi}_u, \pi_{old})} \{A^{\pi_{old}}(s, \tilde{\mathbf{a}}) - A^{\pi_{old}}(s, \mathbf{a}_{old})\}| \\ &= P(\mathbf{a}_{old} \neq \tilde{\mathbf{a}} | s) \\ &\quad \cdot |\mathbb{E}_{(\tilde{\mathbf{a}}, \mathbf{a}_{old}) \sim (\tilde{\pi}_u, \pi_{old})} \{A^{\pi_{old}}(s, \tilde{\mathbf{a}}) - A^{\pi_{old}}(s, \mathbf{a}_{old})\}| \\ &\leq P(\mathbf{a}_{old} \neq \tilde{\mathbf{a}} | s) \cdot 2 \max_{s, \mathbf{a}} |A^{\pi_{old}}(s, \mathbf{a})|. \end{aligned} \quad (57)$$

According to proposition 1, if $D_{TV}^{\max}(\pi_{old}^u, \tilde{\pi}_u) = \alpha^u$, there exists an α^u -coupled policy pair $(\pi_{old}^u, \tilde{\pi}_u)$ so that $P(\mathbf{a}_{old} \neq \tilde{\mathbf{a}} | s) \leq \alpha^u, \forall s$. We can prove $D_{TV}^{\max}(\pi_{old}^u, \tilde{\pi}_u) = D_{TV}^{\max}(\pi_{old}^u, \pi^u) = \alpha^u$ by the following derivation

$$\begin{aligned} D_{TV}(\pi_{old}(\cdot | s) || \tilde{\pi}_u(\cdot | s)) &= \frac{1}{2} \sum_{\mathbf{a}} |\pi_{old}(\mathbf{a} | s) - \tilde{\pi}_u(\mathbf{a} | s)| \\ &= \frac{1}{2} \sum_{\mathbf{a}} |\prod_{i \neq u} \pi_{old}^i(a^i | z^i) \cdot (\pi_{old}^u(a^u | z^u) - \tilde{\pi}_u^u(a^u | z^u))| \\ &= \sum_{\mathbf{a}^u} \prod_{i \neq u} \pi_{old}^i(a^i | z^i) \times \frac{1}{2} \sum_{a^u} |\pi_{old}^u(a^u | z^u) - \tilde{\pi}_u^u(a^u | z^u)| \\ &= D_{TV}(\pi_{old}^u(\cdot | z^u) || \tilde{\pi}_u^u(\cdot | z^u)). \end{aligned} \quad (58)$$

Since (57) holds for an any coupled policy pair $(\pi_{old}, \tilde{\pi}_u)$, we have

$$|\mathbb{E}_{\mathbf{a} \sim \tilde{\pi}_u} \{A^{\pi_{old}}(s, \mathbf{a})\}| \leq 2\alpha^u\epsilon. \quad (59)$$

Then, we get

$$\sum_{u=1}^N |\mathbb{E}_{\mathbf{a} \sim \tilde{\pi}_u} \{A^{\pi_{old}}(s, \mathbf{a})\}| \leq 2\epsilon \cdot \sum_{u=1}^N \alpha^u \leq 2N\alpha^*\epsilon. \quad (60)$$

Based on (51), we thus obtain that

$$\eta(\pi) \geq F_{\pi_{old}}(\Pi) - \frac{4N\epsilon}{(1-\gamma)} \max_u D_{TV}^{\max}(\pi_{old}^u, \pi^u). \quad (61)$$

Finally, due to the relationship between TV divergence and KL divergence [39], i.e., $D_{TV}(\pi_{old}^u(\cdot | z^u) || \pi^u(\cdot | z^u))^2 \leq D_{KL}(\pi_{old}^u(\cdot | z^u) || \pi^u(\cdot | z^u))$, we replace the TV divergence in (61) by the KL divergence. Therefore,

$$\eta(\pi) \geq F_{\pi_{old}}(\Pi) - \frac{4N\epsilon}{(1-\gamma)} \max_u \sqrt{D_{KL}^{\max}(\pi_{old}^u, \pi^u)}, \quad (62)$$

where the equality holds when $\pi = \pi_{old}$ and $\Pi = \Pi_{old}$.

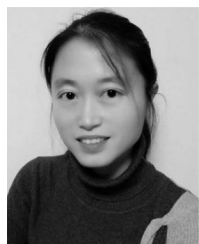
REFERENCES

- [1] J. G. Andrews *et al.*, "What will 5G be?," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.
- [2] B. V. Quang, R. V. Prasad, and I. Niemegeers, "A survey on handoffs – Lessons for 60 GHz based wireless systems," *IEEE Commun. Surv. Tut.*, vol. 14, no. 1, pp. 64–86, Jan.–Mar. 2012.
- [3] "Protocol Specification: Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC)," 3GPP TS 36.331, V10.10.0, Release 10. [Online]. Available: https://www.etsi.org/deliver/etsits/136300/136399/136331/10.10.00/60_2013.
- [4] C. Shen and M. Van Der Schaar, "A learning approach to frequent handover mitigations in 3GPP mobility protocols," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2017, pp. 1–6.
- [5] I. N. M. Isa, M. D. Baba, A. L. Yusof, and R. A. Rahman, "Handover parameter optimization for self-organizing lte networks," in *Proc. IEEE Symp. Comput. Appl. Ind. Electron.*, 2015, pp. 1–6.
- [6] J. Kim, G. Lee, and H. P. In, "Adaptive time-to-trigger scheme for optimizing LTE handover," *Int. J. Control Automat.*, vol. 7, no. 4, pp. 35–44, 2014.
- [7] T. Goyal and S. Kaushal, "Handover optimization scheme for LTE-advance networks based on AHP-TOPSIS and Q-learning," *Comput. Commun.*, vol. 133, pp. 67–76, 2018.
- [8] G. Yang, X. Wang, and X. Chen, "Handover control for lte femtocell networks," in *Proc. Int. Conf. Electron., Commun. Control (ICECC)*, Ningbo, China, Sep. 2011, pp. 1–4.
- [9] J. Wu *et al.*, "Dynamic fuzzy Q-learning for handover parameters optimization in 5G multi-tier networks," in *Proc. Int. Conf. Wireless Commun. Signal Process.*, Oct. 2015.
- [10] N. C. Luong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surv. Tut.*, vol. 21, no. 4, pp. 3133–3174, Oct.–Dec. 2019.
- [11] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.
- [12] M. Simsek, M. Bennis, and S. Gven, "Context-aware mobility management in HetNets: A reinforcement learning approach," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2015 pp. 1536–1541.
- [13] Y. Sun, G. Feng, S. Qin, Y. Liang, and T. P. Yum, "The smart handoff policy for millimeter wave heterogeneous cellular networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 6, pp. 1456–1468, Jun. 2018.
- [14] I. Colin, A. Thomas, and M. Draief, "Parallel contextual bandits in wireless handover optimization," 2019, *arXiv:1902.01931*.
- [15] V. Yajnanarayana *et al.*, "5G handover using reinforcement learning," in *Proc. IEEE 5G World Forum (5GWF)*, 2020, pp. 1–6.
- [16] C. Shen, C. Tekin, and M. Van Der Schaar, "A non-stochastic learning approach to energyefficient mobility management," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3854–3868, Dec. 2016.
- [17] S. Zang, W. Bao, P. L. Yeoh, B. Vucetic, and Y. Li, "Managing vertical handovers in millimeter wave heterogeneous networks," *IEEE Trans. Commun.*, vol. 67, no. 2, p. 1629–1644, Feb. 2019.
- [18] K. Youssef, Y. Koji, N. Takayuki, and M. Morikura, "Reinforcement learning based predictive handover for pedestrian-aware mmwave," in *Proc. IEEE Conf. Comput. Commun. Workshops.*, 2018 pp. 692–697.
- [19] Y. Koda *et al.*, "Handover management for mmwave networks with proactive performance prediction using camera images and deep reinforcement learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 2, pp. 802–816, 2019.
- [20] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5141–5152, Nov. 2019.
- [21] Z. Wang, L. Li, Y. Xu, H. Tian, and S. Cui, "Handover control in wireless systems via asynchronous multiuser deep reinforcement learning," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4296–4307, Dec. 2018.

- [22] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, New York, NY, USA, Jun. 2016, pp. 1928–1937.
- [23] R. Lowe *et al.*, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Advances Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 6379–6390.
- [24] J. Schulman *et al.*, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
- [25] J. Foerster *et al.*, "Counterfactual multi-agent policy gradients," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, Louisiana USA, Feb. 2018, pp. 2974–2983.
- [26] J. Schulman *et al.*, "High-dimensional continuous control using generalized advantage estimation," in *Proc. Int. Conf. Learn. Representations (ICLR)*, San Juan, Puerto Rico, May 2016, pp. 1–14.
- [27] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Comput. Sci.*, 2014.
- [29] T. Bai *et al.*, "Using random shape theory to model blockage in random cellular networks," in *Proc. Int. Conf. Signal Process. Commun.*, Jul. 2012, pp. 1–5.
- [30] H. Elshaer, M. N. Kulkarni, F. Boccardo J. G. Andrews, and M. Dohler, "Downlink and uplink cell association with traditional macrocells and millimeter wave small cells," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6244–6258, Sep. 2016.
- [31] L. Qiang, J. Li, and C. Touati, "A user centered multi-objective handoff scheme for hybrid 5G environments," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 3, pp. 380–390, Jul.–Sep. 2017.
- [32] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust region policy optimization," in *Proc. 32nd Int. Conf. Mach. Learn.*, Lille, France, Jul. 2015, vol. 37, pp. 1889–1897.
- [33] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. 10th Int. Conf. Mach. Learn.*, 1993, Art. no. 330337.
- [34] P. Dhariwal *et al.*, "Openai baselines," 2017. [Online]. Available: <https://github.com/openai/baselines>
- [35] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *Wireless Commun. Mobile Comput.*, vol. 2, no. 5, pp. 483–502, 2002.
- [36] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *Proc. Int. Conf. Learn. Representations (ICLR)*, Palais des Congrès Neptune, Toulon, France, Apr. 2017, pp. 1–13.
- [37] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: <http://tensorflow.org/mixing>
- [38] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. Providence, RI, USA: American Mathematical Society, 2009.
- [39] D. Pollard, "Asymptopia: An exposition of statistical asymptotic theory," 2009. [Online]. Available: <http://www.stat.yale.edu/pollard/Books/Asymptopia>



Delin Guo received the B.S. degree in electronic information science and technology from the Jilin University, Jilin, China, in 2017. He is currently working toward the Ph.D. degree with the School of Electronic Science and Engineering, Nanjing University, Nanjing, China. His current research interests include energy harvesting wireless communication, handover control, physical-layer security, reinforcement learning, and multi-agent reinforcement learning.



Lan Tang (Member, IEEE) received the Ph.D. degree in communication and information science from Southeast University, Nanjing, China, in 2009. Since 2009, she has joined the School of Electronic Science and Engineering, Nanjing University, Nanjing, China, where she is currently an Associate Professor. In 2014, she was a Visiting Researcher with Columbia University, USA. Her current research interests include optimization and machine learning in wireless communications, and statistical signal processing applied for radar and communications.



tion and tracking.

Xinggan Zhang received the B.S. degree in electrical engineering and the S.M. and Ph.D. degrees from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1982, 1988, and 2001, respectively. In 1992, he joined the Department of Electronic Engineering, NUAU, where he was an Associate Professor. In 1999, he joined the School of Electronic Science and Engineering, Nanjing University, Nanjing, China, where he is currently a Professor. His research interests lie in the areas of radar signal processing, including radar imaging, target localization and tracking.



Ying-Chang Liang (Fellow, IEEE) was a Professor with The University of Sydney, Australia, a Principal Scientist and a Technical Advisor with the Institute for Infocomm Research, Singapore, and a Visiting Scholar with Stanford University, USA. He is currently a Professor with the University of Electronic Science and Technology of China, Chengdu, China, where he leads the Center for Intelligent Networking and Communications. He is also a Deputy Director of the Artificial Intelligence Research Institute. His research interests include wireless networking and communications, cognitive radio, symbiotic networks, dynamic spectrum access, the Internet of Things, artificial intelligence, and machine learning techniques. He is a foreign member of Academia Europaea. He has been recognized by the Thomson Reuters (Clarivate Analytics) as a Highly Cited Researcher, since 2014. He received the Prestigious Engineering Achievement Award from The Institution of Engineers, Singapore, in 2007, the Outstanding Contribution Appreciation Award from the IEEE Standards Association, in 2011, and the Recognition Award from the IEEE Communications Society Technical Committee on Cognitive Networks, in 2018. He was the recipient of numerous paper awards, including the IEEE Jack Neubauer Memorial Award, in 2014, and the IEEE Communications Society APB Outstanding Paper Award, in 2012. He was the Chair of the IEEE Communications Society Technical Committee on Cognitive Networks. He served as the TPC Chair and the Executive Co-Chair for the IEEE Globecom 2017. He was an Associate Editor-in-Chief for the *Random Matrices: Theory and Applications* (World Scientific). He serves as an Associate Editor-in-Chief for *China Communications*. He is the Founding Editor-in-Chief for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS: COGNITIVE RADIO SERIES, and the Key Founder. He serves as the Editor-in-Chief for the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He served as a Guest/Associate Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the *IEEE Signal Processing Magazine*, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORK. He was a Distinguished Lecturer from the IEEE Communications Society and the IEEE Vehicular Technology Society.