Towards Energy Efficient Resource Allocation: When Green Mobile Edge Computing Meets Multi-Agent Deep Reinforcement Learning

Yang Xiao, Yuqian Song, and Jun Liu
School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China
Emails: {zackxy, songyuqian, liujun}@bupt.edu.cn

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838659

***Abstract*—Mobile edge computing (MEC) extends the computing power to the edge of communication networks, which has been considered as a promising technology to further improve the quality of communication services in the near future. Nevertheless, the issue of MEC-empowered energy efficient resource allocation has not been well studied. To maximize the long-term energy efficiency for green MEC-enabled heterogeneous networks (HetNets), we proposed a decentralized multi-agent deep reinforcement learning (MADRL) resource allocation algorithm. Based on the proximal policy optimization (PPO) framework, our proposed algorithm enables observation exchange to coordinate the policies of multiple agents. Simulation results show that our proposed algorithm significantly outperforms three baseline methods in terms of effectiveness, robustness, and scalability.**

***Index Terms*—Green mobile edge computing, multi-agent deep reinforcement learning, energy efficiency, resource allocation.**

I. Introduction

Owing to the exponentially increasing number of mobile users and computation-intensive applications, modern communication systems confront more dynamic network environments and more stringent user quality-of-service requirements. Therefore, resource allocation, as a fundamental networking issue that aims to improve network performance by allocating network resources optimally, has become more significant than ever. Till now, a major obstacle that impedes some of the 5G key technologies is the massively increased energy consumption. Energy efficiency, which focuses on minimizing the consumed energy while performing the same tasks, has received more attention since the past decade [1]. By reducing energy waste and lowering energy costs without sacrificing network performance, energy efficient communication systems are more ecologically and economically sound than existing ones.

Deploying the computing and storage devices close to the user side, mobile edge computing (MEC) extends cloud computing and allows various computing tasks to be processed on the MEC servers associating with the base stations (BSs). As a result, the computation workload is distributed to decentralized MEC servers, which effectively alleviates network delay and congestion. MEC is among the most promising technologies to enhance network management capability and improve users' quality-of-experience [2]. Whereas the majority of existing research works on MEC are concerning task offloading [3], the characteristics of MEC (decentralization, etc.) also make it possible for resource allocation tasks to be performed in a more elaborate and customized manner.

In recent years, deep reinforcement learning (DRL) has made remarkable progress in controlling complex networking systems [4]. More importantly, multi-agent deep reinforcement learning (MADRL), which has better algorithm scalability and strategy complexity, is becoming one of the most promising fields of research in the communications industry. For instance, Nasir *et al.* [5] proposed a MADRL algorithm for power allocation in wireless networks. Li *et al.* [6] designed a MADRL-based spectrum allocation framework for D2D underlay communications. Nie *et al.* [7] introduced a federated MADRL approach for resource management in UAV-aided MEC systems. However, to the best of our knowledge, few works are focusing on leveraging MADRL for energy efficient resource allocation in green MEC scenarios.

In this work, we are motivated to propose a MADRL resource allocation algorithm for energy efficiency in green MEC. The main contributions of this work are summarized as follows:

- We presented the system model for green MEC scenarios and formulated the long-term average energy efficiency maximization problem. The formulated problem can be decoupled into two sub-problems, i.e., subcarrier assignment and power control, which form a mixed-integer nonlinear programming (MINLP) problem.
- We proposed a decentralized MADRL resource allocation algorithm for energy efficiency in green MEC. The design of our algorithm considered the characteristics of MEC, e.g., the training of deep neural networks is executed by the MEC servers, the observation exchange of agents considers communication convenience.
- We conducted extensive experiments to evaluate the effectiveness and robustness of our proposed algorithm. Compared to the three baseline methods (i.e., DQN, FP, and Random), our proposed algorithm achieved the best performance under different environment settings.

The remainder of this paper is organized as follows. In Section II, we present our system model and the energy efficiency problem formulation. In Section III, we introduce our algorithm design. Section IV gives the numerical results of the experiments. Finally, Section V concludes this paper.

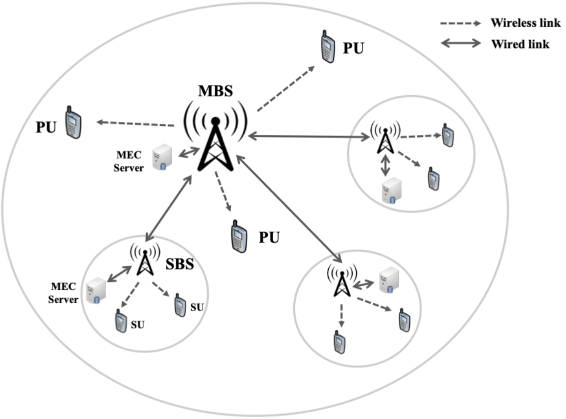


Fig. 1. Illustration of a downlink MEC-enabled HetNet.

II. System Model and Problem Formulation

A. System Model

Consider a downlink MEC-enabled heterogeneous network (HetNet) scenario with one macro base station (MBS) and M small base stations (SBSs), where each BS is equipped with a specified MEC server (shown in Fig. 1). The BSs and MEC servers are connected by wired networks, through which the BSs can offload computation-intensive tasks to the MEC servers. There are L orthogonal frequency-division multiplexing (OFDM) subcarriers and two types of users, i.e., primary users (PUs) and secondary users (SUs). All the PUs are directly served by the MBS, whereas the SUs are served by their associated SBSs, respectively. N_m denotes the number of all users of the m-th BS. Among the entire $M+1$ BSs, the MBS is indexed as 0 and the SBSs are indexed as 1 to M. Similarly, each user is indexed as (m,n), where m is the index of its associated BS, and n is the index within the set of the users grouped by the same BS. Therefore, the n-th SU of the m-th SBS is indexed as (m,n), where $m=1,...,M$. Specifically, the n-th PU is indexed as $(0,n)$, and the number of all PUs is N_0. For each assigned user (m,n), the BS m determines the subcarrier assignment and the downlink transmitting power $p^t_{m,(m,n)}$ during each timestep t, where $p^t_{m,(m,n)}$ is a non-negative value smaller than the maximum transmitting power P_{max}, i.e., $0 \leq p^t_{m,(m,n)} \leq P_{max}$. Moreover, we denote the subcarrier assignment for user (m,n) at timestep t by $\boldsymbol{A}^t_{(m,n)}$, which is expressed as

$$\boldsymbol{A}^t_{(m,n)} = \{\alpha^t_{1,(m,n)}, ..., \alpha^t_{l,(m,n)}, ..., \alpha^t_{L,(m,n)}\}, \tag{1}$$

where $\alpha^t_{l,(m,n)} \in [0,1]$ denotes the indicator representing whether the user (m,n) is assigned to the l-th subcarrier. Each user can only be assigned to one subcarrier during each timestep, which inevitably causes complex and time-varying co-channel interference due to limited channel resource.

Let $g^t_{m,(m,n)}$ denotes the channel gain between the BS m and the user n at timestep t. Each $g^t_{m,(m,n)}$ consists of the small-scale fading factor and the large-scale fading factor, which is expressed as

$$g^t_{m,(m,n)} = |h^t_{m,(m,n)}|^2 \beta_{m,(m,n)}, \tag{2}$$

where $h^t_{m,(m,n)}$ denotes the small-scale Rayleigh fading factor, and $\beta_{m,(m,n)}$ denotes the large-scale fading factor including path loss and shadowing. At each timestep, a user receives interference from both the communications between the MBS and PUs, and the communications between SBSs and SUs that share the same subcarrier with it. For any two users (m,n) and (m',n'), we denote by $\omega^t_{(m,n),(m',n')} \in [0,1]$ the coefficient indicating whether they share the same subcarrier at t, which is calculated by

$$\omega^t_{(m,n),(m',n')} = \sum_{l=1}^{L} \alpha^t_{l,(m,n)} \alpha^t_{l,(m',n')}. \tag{3}$$

Then, the signal-to-interference-plus-noise ratio (SINR) of an arbitrary PU $(0,n)$ is given by

$$\gamma^t_{(0,n)} = \frac{g^t_{0,(0,n)} p^t_{0,(0,n)}}{I^t_{co,PU} + I^t_{cr,SU} + \sigma^2}, \; n = 1, ..., N_0, \tag{4}$$

$$I^t_{co,PU} = \sum_{\substack{n'=1,\\ n'\neq n}}^{N_0} \omega^t_{(0,n),(0,n')} g^t_{0,(0,n)} p^t_{0,(0,n')}, \tag{5}$$

$$I^t_{cr,SU} = \sum_{m=1}^{M} g^t_{m,(0,n)} \sum_{n'=1}^{N_m} \omega^t_{(0,n),(m,n')} p^t_{m,(m,n')}, \tag{6}$$

where $I^t_{co,PU}$ represents the co-tier interference caused by other PUs, and $I^t_{cr,SU}$ represents the cross-tier interference caused by SUs. σ^2 is the power of addictive white Gaussian noise (AWGN). Analogously, the SINR of an arbitrary SU (m,n) is given by

$$\gamma^t_{(m,n)} = \frac{g^t_{m,(m,n)} p^t_{m,(m,n)}}{I^t_{cr,PU} + I^t_{co,SU} + \sigma^2}, m = 1, ..., M, n = 1, ...N_m, \tag{7}$$

$$I^t_{cr,PU} = \sum_{i=1}^{N_0} \omega^t_{(m,n),(0,i)} g^t_{0,(m,n)} p^t_{0,(0,i)}, \tag{8}$$

$$\begin{aligned} I^t_{co,SU} = & \sum_{\substack{n'=1,\\ n'\neq n}}^{N_m} \omega^t_{(m,n),(m,n')} g^t_{m,(m,n)} p^t_{m,(m,n')} + \\ & \sum_{\substack{m'\geq 1,\\ m'\neq m}}^{M} g^t_{m',(m,n)} \sum_{n'=1}^{N_m} \omega^t_{(m,n),(m',n')} p^t_{m',(m',n')}, \end{aligned} \tag{9}$$

where $I^t_{cr,PU}$ is the cross-tier interference caused by PUs. $I^t_{co,SU}$ is the co-tier interference caused by SUs, whose first term and second term in (9) represent the intra-cell interference component and the inter-cell interference component, respectively. Therefore, the maximum achievable downlink transmission rate for user (m,n) with normalized bandwidth is expressed as

$$c^t_{(m,n)} = \log_2\left(1 + \gamma^t_{(m,n)}\right), \; m = 0, ..., M, n = 1, ..., N_m. \tag{10}$$

The energy efficiency η_{EE} of a communication system is defined as the ratio of total achievable transmission rate to the total consumed power, which in our considered scenario is expressed as

$$\eta_{EE}^{t} = \frac{\sum_{m=0}^{M} \sum_{n=1}^{N_m} c_{(m,n)}^{t}}{\sum_{m=0}^{M} \sum_{n=1}^{N_m} p_{m,(m,n)}^{t} + \sum_{m=0}^{M} p_{m,static}^{t}}, \tag{11}$$

where $p_{m,static}^{t}$ denotes the static power of the m-th BS.

B. Problem Formulation

In this work, we aim to maximize the energy efficiency of the MEC-enabled HetNet by optimizing the resource allocation strategy. To better realize long-term energy efficient network operations, we further formulate our considered problem as maximizing the average energy efficiency over a period, which is defined as

$$\max_{\boldsymbol{A}^t, \boldsymbol{P}^t} \frac{1}{T} \sum_{t=t_0}^{t_0+T} \eta_{EE}^{t}, \tag{12}$$

$$s.t. \quad \alpha_{l,(m,n)}^{t} \in [0,1], \forall \alpha_{l,(m,n)}^{t} \in \boldsymbol{A}_{(m,n)}^{t}, \forall \boldsymbol{A}_{(m,n)}^{t} \in \boldsymbol{A}^{t}, \tag{12.c1}$$

$$\sum_{l=1}^{L} \alpha_{l,(m,n)}^{t} = 1, \forall \alpha_{l,(m,n)}^{t} \in \boldsymbol{A}_{(m,n)}^{t}, \forall \boldsymbol{A}_{(m,n)}^{t} \in \boldsymbol{A}^{t}, \tag{12.c2}$$

$$0 \leq p_{m,(m,n)}^{t} \leq P_{max}, \ \forall p_{m,(m,n)}^{t} \in \boldsymbol{P}^{t}, \tag{12.c3}$$

where t_0 is an arbitrary timestep and T is a long period of time. As we discussed in the previous section, the resource allocation problem can be decoupled into two sub-problems, i.e., subcarrier assignment and power control. Therefore, we in (12) denote by $\boldsymbol{A}^t$ and $\boldsymbol{P}^t$ the forms of solutions for the two sub-problems, respectively. Specifically, $\boldsymbol{A}^t = \{\boldsymbol{A}_{(0,0)}^{t}, ..., \boldsymbol{A}_{(m,n)}^{t}, ..., \boldsymbol{A}_{(M,N_M)}^{t}\}$ is the joint subcarrier assignment for all users, and $\boldsymbol{P}^t = \{p_{0,(0,0)}^{t}, ..., p_{m,(m,n)}^{t}, ..., p_{M,(M,N_M)}^{t}\}$ is the joint power control for all users. Constraints (12.c1) and (12.c2) represent that each user can only be assigned to one subcarrier at each timestep, and (12.c3) represents that the transmitting power for each user cannot exceed the maximum value P_{max}.

Considering that $\boldsymbol{A}^t$ is a set of one-hot encoded vectors while $\boldsymbol{P}^t$ is a set of continuous values, our formulation (12) is a time-variant mixed-integer nonlinear programming (MINLP) problem. Therefore, variables such as the Rayleigh fading factor $h_{m,(m,n)}^{t}$ will change continuously with the timestep t shifting. Existing approaches for solving MINLP problems (e.g., branch and bound, cutting-plane methods) will treat the formula (12) as a consecutive one-shot optimizations problem. As a result, these approaches have to recalculate optimal solutions during each timestep. Owing to the recalculation process requiring the nearly real-time acquisition of global network status, such schemes will put tremendous pressure on a centralized computing unit, leaving the MEC servers underused. Moreover, since the problem is non-convex and NP-hard, the computational complexity of these approaches is further increased.

III. Algorithm Design

Deep reinforcement learning (DRL) has been proved effective in dealing with communications and networking problems [8]. In a typical DRL scheme, an agent (or multiple agents) interacts with the environment and optimizes its policy (or their policies) in a trial-and-error manner. By leveraging the deep neural network (DNN) module, the DRL agent can approximate any continuous-valued input states and respond with actions as the near-optimal solutions. Therefore, DRL is suitable for handling continuous decision-making tasks. Here in this work, we propose a MADRL resource allocation algorithm for energy efficiency maximization in green MEC scenarios. In this section, we first introduce our design of essential DRL components. Then, we provide the details of the algorithm process.

A. Deep Reinforcement Learning Components

1) Agent: A DRL agent is essentially an instance of the DNN-empowered reinforcement learning algorithms. In green MEC scenarios, the MEC servers are located at the edge BS side, enabling services with high-density computing and low-latency requirements to be deployed. To fully exploit the computing capacity of the MEC servers, we apply the MADRL scheme and leverage each agent to allocate resources for each BS-user pair. Supposing the m-th BS has N_m authorized users, there will be N_m agents that are assigned to N_m BS-user pairs. In addition, these N_m agents reside on the same MEC server specified to the m-th BS.

2) State: The state is the necessary information for each agent to perceive the environment. For better practicality, we model the problem as a partially observable Markov decision process (POMDP) in which each agent can only observe a small portion of the environment, i.e., each agent residing on any one of the MEC servers cannot acquire global observation of the HetNet in real time. We first define the local observation of user (m,n) as

$$o_{(m,n)}^{t} = \{H_{m,(m,n)}^{t}, p_{m,(m,n)}^{t}, c_{(m,n)}^{t}\}, \tag{13}$$

where $H_{m,(m,n)}^{t}$ is the CSI of the channel between the m-th BS and the user (m,n) measured at timestep t, $p_{m,(m,n)}^{t}$ and $c_{(m,n)}^{t}$ are the corresponding transmitting power and maximum transmission rate, respectively. Noted that the agents on the same MEC server are easy to communicate with each other, we design the state as the combination of the local observation of each agent and the exchanged observations from other agents. The introduction of exchanged observations will bring additional information that help facilitate coordination between agents [9]. The PUs, which usually have higher service priority, are served by the same MBS in our considered scenario. Formally, we define the state of the agent associated with PU $(0,n)$ as

$$s_{(0,n)}^{t} = \{o_{(0,1)}^{t}, ..., o_{(0,N_0)}^{t}\}. \tag{14}$$

As mentioned in the previous section, the BSs and MEC servers are connected by wired networks, so the communication between them is relatively low-cost. Besides, agents on the same SBS, as analogous to the case for an arbitrary PU, are also easy to communicate. Therefore, we define the state of the agent associated with SU (m, n) as

$$s^t_{(m,n)} = \{\underbrace{o^t_{(m,1)}, ..., o^t_{(m,N_m)}}_{\textit{co-tier information}}, \underbrace{o^t_{(0,1)}, ..., o^t_{(0,N_0)}}_{\textit{cross-tier information}}\}, \tag{15}$$

where the terms $o^t_{(m,1)}, ..., o^t_{(m,N_m)}$ compose the co-tier information, and the terms $o^t_{(0,1)}, ..., o^t_{(0,N_0)}$ compose the cross-tier information.

3) Action: The action of each agent is defined in accordance with the resource allocation task, which is composed of the subcarrier assignment and the power control schemes for user (m, n). Formally, the action of the agent associated with user (m, n) is expressed as

$$a^t_{(m,n)} = \{\boldsymbol{A}^t_{(m,n)}, p^t_{m,(m,n)}\}, \tag{16}$$

where $\boldsymbol{A}^t_{(m,n)}$ is a one-hot encoded vector for subcarrier assignment, and $p^t_{m,(m,n)}$ is a non-negative real number smaller than P_{max}. Note that the joint action of all agents (denoted by a^t) is what finally led to the state transitions during the POMDP. Mathematically, the a^t is formulated as

$$\begin{aligned} a^t &= \{a^t_{(0,0)}, ..., a^t_{(M,N_M)}\} \\ &= \{\boldsymbol{A}^t_{(0,0)}, ..., \boldsymbol{A}^t_{(M,N_M)}, p^t_{0,(0,0)}, ..., p^t_{M,(M,N_M)}\} \\ &= \{\boldsymbol{A}^t, \boldsymbol{P}^t\}, \end{aligned} \tag{17}$$

which reflects that each of the joint action exactly gives a complete solution for our proposed resource allocation problem (12) at each timestep.

4) Reward: The reward is the quantified feedback from the environment to evaluate the effectiveness of the executed action based on the observed state. To reflect the optimization objective of the resource allocation task, we design the forms of the reward functions similar to the energy efficiency η_{EE}. Following the range of observation exchange of the state definition, we also define the reward function of an arbitrary PU or SU separately. The rewards of the agents associated with PU $(0, n)$ and SU (m, n) are expressed as

$$r^t_{(0,n)} = \frac{\sum_{n=1}^{N_0} c^t_{(0,n)}}{\theta_1 \cdot \sum_{n=1}^{N_0} p^t_{0,(0,n)} + \theta_2 \cdot p^t_{0,static}}, \tag{18}$$

$$r^t_{(m,n)} = \frac{\sum_{n=1}^{N_m} c^t_{(m,n)} + \sum_{n=1}^{N_0} c^t_{(0,n)}}{\theta_3 \cdot \big(\sum_{n=1}^{N_m} p^t_{m,(m,n)} + \sum_{n=1}^{N_0} p^t_{0,(0,n)}\big) + \theta_4 \cdot \big(p^t_{m,static} + p^t_{0,static}\big)}, \tag{19}$$

where θ_1 and θ_3 are the coefficients of dynamic power terms, and θ_2 and θ_4 are the coefficients of static power terms. The four coefficients are applied for two purposes: adjusting the relative scale between the dynamic power and static power, and controlling the overall reward scaling. Consequently, each agent (m, n) can optimize the energy efficiency of a subset of the entire HetNet, where the subset consists of all the agents exchanging information with agent (m, n).

Algorithm 1 MADRL Resource Allocation Algorithm for Energy Efficiency in Green MEC

```
Initialize the actor and critic networks for all agents.
for episode = 1 to $E$ do
    for timestep = 1 to $S$ do
        for agent = $(0, 0)$ to $(M, N_M)$ do
            Agent exchanges observations.
            Agent interacts with the environment.
            if step % $U$ == 0 then
                Calculate mean squared advantage value.
                Update the critic network.
                Calculate clipped surrogate objective.
                Update the actor network.
            end if
        end for
    end for
end for
```

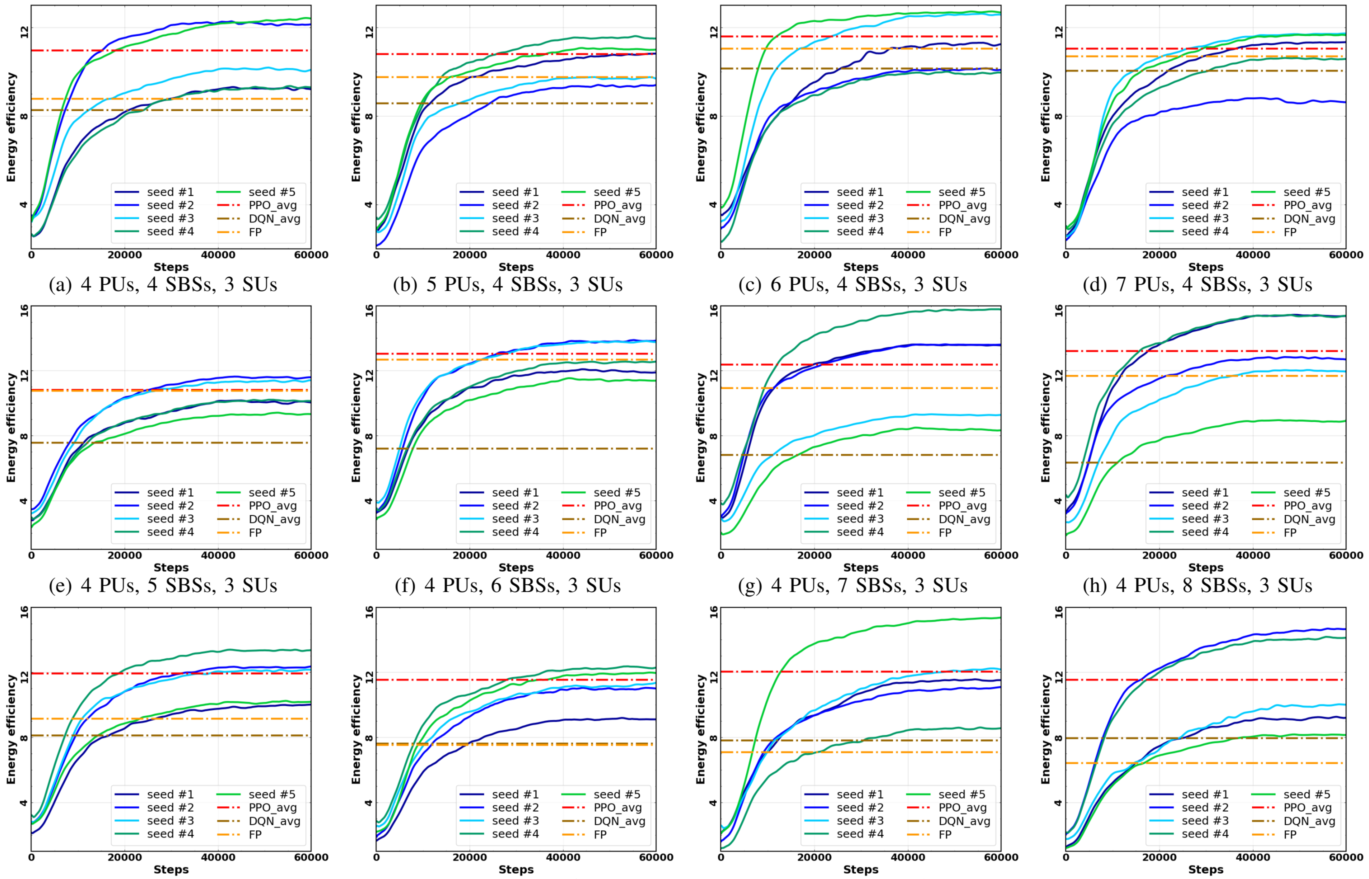
B. Algorithm Process

On the premise that the DRL components are defined, we further introduce the details of our proposed algorithm. Owing to the existence of the real number component of the action space, value-based DRL algorithms such as the DQN [10] are not applicable. We propose to adopt the proximal policy optimization (PPO) [11], one of the widely applied policy gradient methods, as the underlying algorithm for all agents. In addition, we make two major modifications to algorithm structure according to our problem formulation.

- To adapt to the real number component of the action space, the PPO algorithm is implemented in the actor-critic style [12]. Each agent has an actor network mapping from states to actions and a critic network mapping from state-action pairs to the predicted value.
- To conform to the multi-agent settings with information exchange, the actor networks and critic networks of all agents are trained parallelly and independently, whereas a small amount of information is exchanged within the MEC server, or between the BSs through wired networks.

At the beginning of the training stage, the parameters of the actor and critic networks of all agents are initialized independently. During each timestep, the agents first exchange their local observations so that the states of all agents are obtained. Then, the agents interact with the environment by recurrently observing states, executing actions, and receiving rewards. Periodically, the policy $\pi(a|s)$ mapping from states to actions is updated. Firstly, the rewards collected from interactions are used to calculate the mean squared advantage value as the critic loss, whose derivative is subsequently back-propagated to update the parameters of the critic network. Afterward, we calculate the clipped surrogate objective function by

$$L^t(\theta) = \hat{\mathbb{E}}^t \Big[\min \big[r(\theta)\hat{A}^t, clip(r(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}^t \big] \Big], \tag{20}$$



(a) 4 PUs, 4 SBSs, 3 SUs (b) 5 PUs, 4 SBSs, 3 SUs (c) 6 PUs, 4 SBSs, 3 SUs (d) 7 PUs, 4 SBSs, 3 SUs

(e) 4 PUs, 5 SBSs, 3 SUs (f) 4 PUs, 6 SBSs, 3 SUs (g) 4 PUs, 7 SBSs, 3 SUs (h) 4 PUs, 8 SBSs, 3 SUs

(i) 4 PUs, 4 SBSs, 4 SUs (j) 4 PUs, 4 SBSs, 5 SUs (k) 4 PUs, 4 SBSs, 6 SUs (l) 4 PUs, 4 SBSs, 7 SUs

Fig. 2. Comparison of energy efficiency between our proposed algorithm and baseline methods under different environment settings.

where $r(\theta) = \frac{\pi_\theta(a^t|s^t)}{\pi_{\theta_{old}}(a^t|s^t)}$ is the ratio of current policy to old policy under parameter θ. $\hat{A}^t$ denotes the estimated advantage value. The operation $\hat{\mathbb{E}}^t$ represents calculating the expectation of the following expression over a sample batch. The clipping operation, whose parameter ϵ is usually a small value close to zero, limits the margin of policy updates, which ensures the policy to be optimized smoothly. The process of our proposed algorithm is summarized in **Algorithm 1**, where E and S denote the maximum episodes and the maximum steps of each episode, respectively. U denotes the period of policy update.

IV. Numerical Results

A. Simulation Setup

We conduct extensive experiments to evaluate the performance of our proposed MADRL resource allocation algorithm. In our experiments, the default setting of the green MEC HetNet consists of one MBS, four PUs, and four SBSs ($N_0 = 4, M = 4$). For simplicity, we uniformly set the number of SUs of all SBSs as three ($N_m = 3, m = 1, ..., M$). The maximum communication ranges of the MBS and SBS are $1km$ and $0.1km$, respectively. All the users are randomly located within the coverage of their assigned BSs. There are three authorized orthogonal subcarriers for users to choose from ($L = 3$). The maximum transmitting power P_{max} and the AWGN power σ^2 are 38dBm and -114 dBm, respectively.

In terms of the channel model, the small-scale fading factor is simulated by the Jakes model [13]. The large-scale fading factor is calculated by $\beta = 114.8 + 36.7\log_{10}(d) + 10\log_{10}(z)$, where d is the distance between the user and its corresponding BS. z is a log-normal random variable whose standard deviation is 8 dB.

The algorithm is implemented based on the PyTorch library. Empirically, we set the actor network and critic network of each agent to both have three hidden layers with 128, 64, and 64 neurons. The learning rates of the actor network and critic network are 3×10^{-4} and 1×10^{-3}, respectively. The *Adam* optimizer is applied for the training processes of all DNNs. The maximum episodes $E = 1200$ and the maximum steps of each episode $S = 50$. The period of policy update U is 400 steps. The coefficients in reward functions are set as $\theta_1 = \theta_3 = 1$, $\theta_2 = \theta_4 = 0.1$, respectively. The parameter ϵ that limits policy updates in PPO is set as 0.2.

B. Performance Evaluation

We evaluate the performance of our proposed algorithm (referred to PPO in figures) against three baseline methods:

- *Deep Q-network (DQN)*: The multi-agent structure and the definitions of state, action, and reward are the same

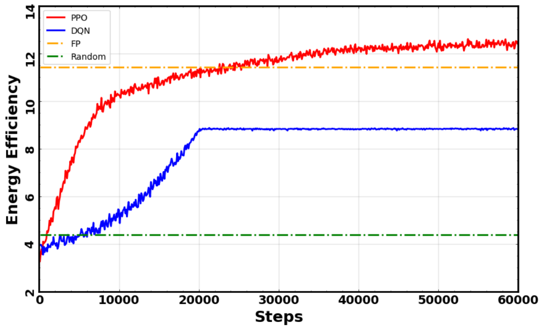


Fig. 3. Comparison of the training process between our proposed algorithm and baseline methods.

as our proposed algorithm, except that the DRL algorithm is replaced by a benchmark implementation of DQN [14].

- *Fractional programming (FP)*: Apply the fractional programming after relaxing the discretization of the subcarrier assignment component [15], then approximate the subcarrier assignment to the discrete space of $\boldsymbol{A}^t$.
- *Random*: Each agent randomly and individually chooses a subcarrier and a transmitting power.

To evaluate the effectiveness and robustness of our proposed algorithm, we compared the energy efficiency of our proposed algorithm and baseline methods under different environment settings. For the two DRL-based algorithms (i.e., PPO and DQN), we choose five independent random seeds (the same numbers of PUs, SBSs, and SUs, but different location distributions) for repeated experiments. As shown in Fig. 2, we conducted 12 experiments with different environment settings. Fig. 2(a) is the result under the default environment setting where $\{N_0, M, N_m\} = \{4, 4, 3\}$. Fig. 2(b-d), Fig. 2(e-h), and Fig. 2(i-l) are grouped by additional PUs, SBSs, and SUs, respectively. For each subfigure, seed #1 to seed #5 are the training curves of our proposed algorithm under five independent random seeds, and the average of their convergence values is demonstrated as *PPO_avg*. The same experiments are also conducted on the DQN algorithm. For a better demonstration, we leave out these training curves and only present the average convergence values *DQN_avg*. The convergence value of the *FP* algorithm is also presented in the figures as the benchmark result of the traditional method.

As illustrated in Fig. 2, whereas the the *DQN_avg* can hardly surpasses *FP*, *PPO_avg* achieves the highest energy efficiency in all experiments. In terms of algorithm scalability, we can observe from Fig. 2(e-h) and Fig. 2(i-l) that the outperformance of *PPO_avg* is sustained and widened with the network scaling up. At the same time, though the location distributions that are initialized differently lead to the performance variation of multiple random seeds, the worst seed can still perform no worse than *DQN_avg* in most cases. Thus, the effectiveness and the robustness of our proposed algorithm are further guaranteed.

In Fig. 3, we look into the training processes of PPO and DQN. Both the two DRL-based approaches can improve the performance smoothly and outperform the random method at the beginning of training, while the performance of PPO grows much faster than that of DQN. At around 20,000 steps, the DQN falls into a local minimum and cannot continue to improve. In contrast, the performance of PPO can improve steadily and surpass FP at around 25,000 steps. The final convergence value of PPO outperforms the three baseline methods by a significant margin.

V. Conclusion

In this paper, we investigated the resource allocation problem in green MEC-enabled HetNets. To maximize the long-term average energy efficiency, we proposed a decentralized MADRL-based resource allocation algorithm. Our proposed algorithm features the observation exchange mechanism that contributes to the policy coordination of multiple agents. Simulation results demonstrate the outperformance of our algorithm in terms of effectiveness, robustness, and scalability.

Acknowledgment

This work was supported in part by the MoE-CMCC Artificial Intelligence Project under Grant MCM20190701, and the BUPT Excellent Ph.D. Students Foundation under Grant CX2021112. (*Corresponding author: Jun Liu.*)

References

[1] X. Huang, T. Han, and N. Ansari, "On green-energy-powered cognitive radio networks," *IEEE Commun. Surv. Tutor.*, vol. 17, no. 2, pp. 827–842, 2015.

[2] Y. Mao, C. You *et al.*, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 4, pp. 2322–2358, 2017.

[3] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 3, pp. 1628–1656, 2017.

[4] Y. Xiao, J. Liu, J. Wu, and N. Ansari, "Leveraging deep reinforcement learning for traffic engineering: A survey," *IEEE Commun. Surv. Tutor.*, pp. 1–1, 2021.

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

[6] Z. Li and C. Guo, "Multi-agent deep reinforcement learning based spectrum allocation for d2d underlay communications," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1828–1840, 2020.

[7] Y. Nie, J. Zhao *et al.*, "Semi-distributed resource management in uav-aided mec systems: A multi-agent federated reinforcement learning approach," *IEEE Trans. Veh. Technol.*, pp. 1–1, 2021.

[8] N. C. Luong, D. T. Hoang *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 4, pp. 3133–3174, 2019.

[9] H. Mao, Z. Gong *et al.*, "Accnet: Actor-coordinator-critic net for "learning-to-communicate" with deep multi-agent reinforcement learning," *arXiv preprint arXiv:1706.03235*, 2017.

[10] V. Mnih, K. Kavukcuoglu *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[11] J. Schulman, F. Wolski *et al.*, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[12] M. Holzleitner, L. Gruber *et al.*, "Convergence proof for actor-critic methods applied to ppo and rudder," in *Transactions on Large-Scale Data-and Knowledge-Centered Systems*. Springer, 2021, pp. 105–130.

[13] P. Dent, G. E. Bottomley, and T. Croft, "Jakes fading model revisited," *Electronics letters*, vol. 29, no. 13, pp. 1162–1163, 1993.

[14] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[15] K. Shen and W. Yu, "Fractional programming for communication systems–part I: Power control and beamforming," *IEEE Trans. Signal Process.*, vol. 66, no. 10, pp. 2616–2630, Apr. 2018.