Deep Reinforcement Learning-Based MEC Offloading and Resource Allocation in Uplink NOMA Heterogeneous Network

Wei Liu, Yejun He*, Jinfeng Zhang, Jian Qiao
Guangdong Engineering Research Center of Base Station Antennas and Propagation
Shenzhen Key Laboratory of Antennas and Propagation
College of Electronics and Information Engineering, Shenzhen University, 518060, China
Email:942415342@qq.com, heyejun@126.com*, zhangjf@szu.edu.cn, 446941582@qq.com

2021 Computing, Communications and IoT Applications (ComComAp) | 978-1-6654-2798-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/COMCOMAP53641.2021.9653064

***Abstract*—With the advancement of fifth generation(5G) technology, mobile edge computing (MEC) has been considered an effective solution to 5G technical problems. The applications of non-orthogonal multiple access (NOMA) in heterogeneous networks is gradually being considered as an method to increase network throughput and improve spectrum utilization. By assigning non-orthogonal communication resources to different users at the transmitting end, the utilization rate of the spectrum can be maximized. Based on these advantages, we analyze thoroughly the MEC based on NOMA in this paper. In the NOMA system, we focus on optimizing channel resources, user offloading pattern and transmit power. These all characteristics have major role in obtaining the optimized user energy consumption. In recent years, deep Q network (DQN) is considered to be an effective method to solve the model-free problems. Different from traditional heuristic algorithms, we design multi-agent DQN to solve resource allocation in NOMA system. Due to the strong coupling between multiple decisions and the large solution space in dynamic optimization, there are found great challenges to the optimization of resources allocations. According to the simulation results, we can see that the DQN method for multi-agents can allow each agent to find approximately the optimal solution.**

***Index Terms*—Deep reinforcement learning (DRL), heterogeneous networks (HetNets), mobile edge computing (MEC), non-orthogonal multiple access (NOMA), deep Q network (DQN), resource allocation**

I. Introduction

With the development of fifth generation(5G) technology, the network efficiency and time latency has been improved, but also brought some challenges such as a mobile phone does not have super computing power and the battery capacity is also limited. How to reduce the user energy consumption has become a research hot spot. Mobile edge computing (MEC) has been considered as a promising technique to integrate computation and communication complexity in the future fifth generation systems and it can support various new services [1]. However, with the explosive growth of mobile devices, limited spectrum resources can no longer support large-scale device access.

Heterogeneous networks (HetNets) and non-orthogonal multiple access (NOMA) are considered as two emerging technologies to improve spectral efficiency and system throughput in the future wireless network [2]. The basic principles of NOMA techniques rely on the employment of superposition coding (SC) at the transmitter and successive interference cancelation (SIC) techniques at the receiver [3]. In extreme cases, each user is allocated with all computing resources. The basic idea of HetNets is to deploy numerous small cells underlaidon the microcells, where the small cells are allowed to reuse the subchannel resources of the microcells to improve spectrum efficiency [4]. Since the same channel resources are reused among the cells, it also brings serious communication interference while improving the spectrum utilization. Therefore, how to allocate channel resources reasonably is critical to the NOMA technology.

References [5-6] use traditional heuristic algorithms. Although they pay attention to the resource allocation of the uplink NOMA network, the number of users participating in the numerical simulation is still small and are not capable of handling the reflecting property of the multi-user situation. The influence of serial interference on communication is an ideal simulation result. Reference [7-8] uses the DRL method, and also allocates channel resources, but the resource allocation method is considered as an ideal and does not take into account the actual application scenarios. Reference [9] also divides the channel resource hypothesis into multiple equally and allocates resources under the ideal conditions, also under the premise of a single base station. Reference [10] uses method of user pairing theory. Users in different time slots shares channel and time resources and users in different time slots are combined according to the task amount between different users to find the best energy consumption. However, most methods fails to comprehensively consider the resource allocation problem in NOMA.

For this reason, we propose DQN method based on multi-agent which is highly suitable for multi-base stations and multi-user situations[11]. Each user is regarded as an independent agent to choose the best offloading decision and resource allocation plan according to different situations. The main contributions of this paper are as follows.

1) We design NOMA model of heterogeneous network with multiple base stations and multiple users to dynamically optimize offloading decisions, base station selection, and

channel resource allocation. The purpose of this is to obtain the minimum energy consumption.

2) We design a multi-agent DQN methodmethod, where each user is regarded as an agent and makes a set of action choices independently according to the environment, which reduces the time for the agent to explore the solution space. The convergence speed of the network is accelerated.

3) We perform numerical simulations to compare the proposed method with the benchmark method, including all-local offloading and random offloading. Numerical results show that our method is close to the optimal solution.

The rest of this paper is organized as follows. In Section II, the system model and the required parameter variables are introduced. In Section III, the optimization model is formulated while the multi-agent based algorithms are proposed. Simulation results are publicized in Section IV. Finally, we conclude this paper in Section V.

II. SYSTEM MODELS AND PROBLEM DESCRIPTION

A. System Model

We design MEC heterogeneous network based on NOMA, and each base station is equipped with MEC as illustrated in Fig.1, and it has macro base station (MBS) and multiple small BSs. We use $U_{MBS} = \{1, 2, ...M_{MBS}\}$ to represent the MBS users, the users of BS service use $U_{BS} = \{1, 2, ...M_{BS}\}$. $B_s = \{1, ...b, ...B\}$ to indicate the number of connectable base stations, we apply $b = 0$ to express connection with MBS, $\forall i \in b > 0$ connect with BS. Specifically, we also use $X_m \in \{0, 1\}$ to indicate the user offloading instruction. when $X_m = 0$, represents the user local processing, Otherwise, the user uploads to MEC for processing. we use P to denote the users transmit power.

Assuming that in each time slot, each user has an intensive task to be processed, $T_{b,c} = \{J_{b,c}, V_{b,c}, D_{max}\}$ represents a collection of tasks. $J_{b,c}$ indicates the amount of tasks that need to be uploaded when the task is ofloading to the MEC, $V_{b,c}$(in cycles per bit) denotes the number of CPU cycles required to complete one bit task and D_{max} denote indicates the maximum endurance time of the user.

B. Communication Model

In this section, we introduce the communication interference when users connect to different base stations to reuse MBS channel resources.

1) Communication Model for MBS: For all users who are offloading to MBS in the cell, the NOMA protocol is reused in the cell and they suffer interference from users who are offloading to the BS and intend to reuse the same channel. The signal-to-interference-plus-noise-ratio (SINR) is defined as follows

$$SINR_{0,c} = \frac{P_{0,c}h_{0,c}}{I_{cr}^{0,c} + N_0} \tag{1}$$

where $h_{0,c} = |h_0|^2 d_{0,c}^{-\alpha}$ indicates the channel gain between the user and the connected channel, h_0 is the Rayleigh fading channel coefficient,which obeys the complex Gaussian distribution $h_0 \sim \mathcal{CN}(0, 1)$ which indicates the distance between the user and the MBS. α indicates the path loss coefficient. N_0 represents Gaussian noise during transmission of power. Interference in macro base station can be expressed as $I_{cr}^{0,c} = \sum_{i \in U_{MBS}} X_i \sum_{i \in U_{BS}} b_i P_{b,c} h_{b,c}$

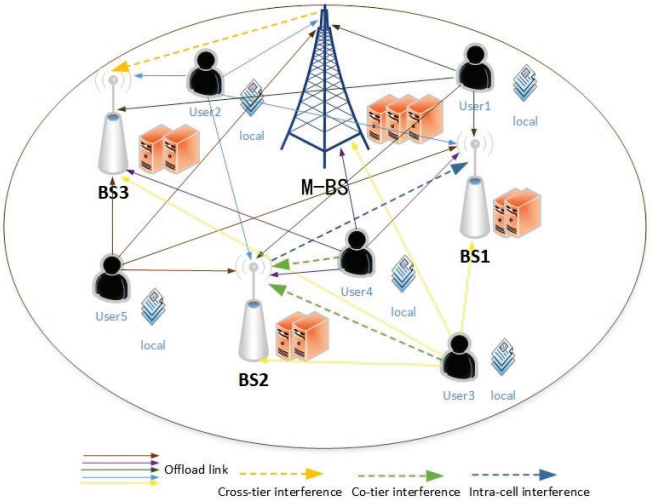

Fig. 1. System model for MEC in uplink NOMA heterogeneous networks.

According to the formula, we can get the uplink transmission rate connected to MBS as

$$r_{0,c} = W log_2(1 + SINR_{0,c}) \tag{2}$$

where W indicates the bandwidth of the channel.

2) Communication Model for BS: Similarly, users can offloads to the BS for task processing. When different users are offloading to the same BS and reuses the same channel, intra-cell interference will occur. When different users are offloading to different BSs and reuses the same channel, inter-cell interference will occur. NOMA protocol applys SIC receivers at the receiving terminal to realize multi-user detection. It judges the users one by one in the received signal, first decodes the signal with the largest channel gain, and subtracts the multiple access interference generated by the user signal from the received signal, and the remaining users are judged again, and the operation is repeated until all the multiple access interference is eliminated. The signal-to-interference-plus-noise-ratio (SINR) as

$$SINR_{b,c} = \frac{P_{b,c}h_{b,c}}{I_{in}^{b,c} + I_{on}^{b,c} + I_{cr}^{b,c} + N_0} \tag{3}$$

where $h_{b,c} = |h_0|^2 d_{b,c}^{-\alpha}$ indicates the channel gain between the user and the connected channel. $d_{b,c}$ indicates the distance between the user and the BS. $I_{in}^{b,c} = \sum_{i \in U_{BS}} b_i P_{b,c} h_{b,c}$ is intra-cell interference that is offloading to the same BS and reuses the same channel. $I_{on}^{b,c} = \sum_{i \notin b} b_i \sum_{i \in U_{BS}} P_{b,c} h_{b,c}$ is inter-cell interference that is offloading to different BS and reuse the same channel. $I_{co}^{b,c} = P_{0,c} h_{b,c}$ is offloading to the

BS, but the cross-layer interference generated by the MBS channel is reused.

According to the formula, we can get the uplink transmission rate connected to BS is

$$r_{b,c} = Wlog_2(1 + SINR_{b,c}). \tag{4}$$

C. Computation Model

In this section, we will introduce the calculation formulas of energy consumption and time when users are offloading to MEC.

1) Local Computing Model: The user can process intensive tasks at the local device, and the energy consumption and time calculations for user i are given by the following formula

$$T_i^{loc} = \frac{\rho_{i,loc}}{f_{i,loc}} \tag{5}$$

and

$$E_i^{loc} = \eta(f_{i,loc})^2 \rho_{i,loc} \tag{6}$$

where $\rho_{i,loc} = J_i V_i$ indicates the number of CPU cycles required to process a task, $f_{i,loc}$ indicates the user local computing power and η indicates the user local computing power.

2) Edge Computing Model: When the user chooses to offload the task to the MEC, the task is first uploaded to the base station through the communication model. The MEC of the base station processes the uploaded task and returns the calculation result. Since the returned calculation result small, so we have ignored it in this paper.

The energy consumption and delay of offloading to MBS are shown

$$T_{0,c}^{tran} = \frac{J_{0,c}}{r_{0,c}} \tag{7}$$

and

$$E_{0,c}^{tran} = P_{0,c} T_{0,c}^{tran}. \tag{8}$$

The energy consumption and delay of offloading to BS are given by

$$T_{b,c}^{tran} = \frac{J_{b,c}}{r_{b,c}} \tag{9}$$

and

$$E_{b,c}^{tran} = P_{b,c} T_{b,c}^{tran}. \tag{10}$$

When users are offloading to MBS or BS's, the computing power of MEC is different and the computing power allocated to users is also different. In order to obtain the best energy consumption and time delay, we assume that MEC allocates all computing power to each user and does not consider the energy consumption of MEC. MEC processing time is defined as follows

$$T_{b,c}^{deal} = \frac{\rho_{i,mec}}{F_{i,mec}} \tag{11}$$

where $F_{i,mec}$ indicates the capacity assigned by MEC to compute upload tasks. The total time and energy consumption for offloading to MBS are shown as

$$T_{0,c}^{off} = T_{0,c}^{tran} + T_{0,c}^{deal} \tag{12}$$

and

$$E_{0,c}^{off} = E_{0,c}^{tran}. \tag{13}$$

The total time and energy consumption for offloading to BS are given by

$$T_{b,c}^{off} = T_{b,c}^{tran} + T_{0,c}^{deal} \tag{14}$$

and

$$E_{b,c}^{off} = E_{b,c}^{tran}. \tag{15}$$

III. PROBLEM FORMULATION AND PROPOSED APPROACHES

In this subsection, we optimize energy consumption based on the offloading decisions, base station selection and channel resource allocation made by users. The design purpose of multi-agent DQN is to minimize the execution energy consumption.

A. Problem Formulation

The energy consumption optimization formula of user i is defined as follows

$$E_i = X_i E_i^{off} + (1 - X_i) E_i^{loc}. \tag{16}$$

The energy consumption of the system is given by

$$\begin{aligned} &\min_{X,P,F,f} \sum_{i \in MBS} \sum_{i \in BS} E_i \\ &s.t. C1 : 0 \le f_i^{loc} \le f_{max}^{loc}, \forall i \in U \\ &\quad C2 : X_i T_i^{off} + (1 - X_i) T_i^{loc} \le D_{max} \\ &\quad C3 : 0 \le P_{0,c} \le P_{max}, 0 \le P_{b,c} \le P_{max} \\ &\quad C4 : X_i \in \{0, 1\}, \forall i \in U_{MBS}, \forall i \in U_{BS} \end{aligned} \tag{17}$$

where constraint $C1$ indicates that the computing power allocated by the task in the local processing is less than the maximum computing power. $C2$ means that regardless of upload to MBS or BS, the upload power is less than the user maximum transmission power. Constraint $C3$ indicates that the delay is less than the user maximum tolerable time regardless of whether it is following local processing or offloading to MEC processing. Constraint $C4$ indicates that the user link instruction is a binary variable.

B. Optimization of Resource Allocation

In this section, we optimize the allocation of resources according to the user maximum endurance time and minimize energy consumption within the user maximum endurance time.

1) Resource Allocation for Local Computing Users: We find the minimum local energy consumption based on the user endurance time.

$$\begin{aligned} &\min_{f_i^{loc}} \eta(f_{i,loc})^2 \rho_{i,loc} \\ &f_{min}^{loc} = \frac{\rho_{i,loc}}{D_{max}}. \end{aligned} \tag{18}$$

According to the obtained minimum local computing power, we can get the minimum energy consumption as

$$E_{min}^{loc} = \eta (f_{min}^{loc})^2 \rho_{i,loc}. \tag{19}$$

2) Resource Allocation for Task Offloading Users: For the offloading users, we get the best energy consumption calculation formula. According to the user maximum endurance time and maximum processing power allocated to the offloading user, we can get the minimum transmission speed as

$$T_{min}^{deal} = \frac{\rho_{i,mec}}{F_{i,max}} \tag{20}$$
$$R_{min}^{tran} = \frac{J_i}{D_{max} - T_{min}^{deal}}.$$

C. Deep Q Network Approach Based on Multi-Agent

The traditional DQN algorithm involves an agent repeatedly interacting with the environment, obtaining actions, and then returning to the environment to continue the interaction. The purpose of DQN is to provide the maximum Q value in the current state $Q_{(s,a)} = r_{(s,a)} + \gamma \max Q_{(s',a')}$ and select the action corresponding to the maximum Q value, which is the best action under the current reward.

We design a multi-agent DQN model. We treat each user as an agent. Each agent may not know each other, interacts independently in the environment, and chooses the best action based on the immediate reward and future reward. Three key elements of DQN are defined as follows.

State: In the multi-agent DQN model, each user is regarded as an agent, and each agent independently interacts with the environment to obtain state, because each agent may not know the situation of other agents, so we use all user uploaded data and channel gains of each base station as the state input of the network. The purpose is to enable each neural network to share the same input value. The status is defined by $state_{i \in U} = \{J_i, \rho_i, h_{b,c}\}, \forall b \in B, \forall c \in C$.

Action: Each independent agent needs to choose the offloading decision, the base station linking decision, and the channel decision. The amount of action are given by the following formula;

$$action_i = \begin{cases} X_i \in \{0,1\} \\ Y_i \in \{0,1...b\} \\ Z_i \in \{0,1...c\} \end{cases}$$

Reward: We set the energy consumption of offloading or local consumption as the reward of each agent. Our goal is to maximize the reward of each agent. If the task is not completed within the maximum tolerable time, the task is regarded as a failure and a poor reward is given as

$$reward_i = \begin{cases} -E_i, & X_i T_i^{off} + (1 - X_i) T_i^{loc} \le D_{max} \\ -\psi, & otherwise \end{cases}$$

The traditional DQN can only output a single discrete action and cannot handle multi-action tasks. The construction of multiple agents can execute multiple discrete actions, where each user outputs its own actions and each network trains its own state without interfering with each other, but uses the overall reward as a bridge to connect all agents. The goal of the agent is to perform best action to ensure the minimum energy consumption of all users. The pseudo code of the algorithm is shown below.

Algorithm 1: Multi-agent based resource allocation and task offloading algorithm

for $i \in U$ **do**
  Initialize replay memory as rpm;
  Initialize action-value network Q and target network $Q^{'}$;
  for $episode = 1, 2, 3, ...max$ **do**
    Initialize the state s_t of each agent;
    for $step = 1, 2, 3, ...max$ **do**
      According to the random number ψ select action a_t based on ε

$$action_i = \begin{cases} a_i = \psi, & \psi \le \epsilon \\ a_i = agrmaxQ(s,a), & otherwise \end{cases}$$

      Perform actions and interact with the environment and get reward r_t , the next state s_{t+1};
      Weighted sum of the reward of each agent as r_t^{sum};
      Store $(s_t, a_t, r_t^{sum}, s_{t+1}, done)$ in rpm;
      Sample a random mini-batch from rpm;
      Send the data to the network to get the output value of the prediction network;

$$y_i = \begin{cases} y_i = r_t^{sum}, & t = max \\ y_i = r_t^{sum} + \gamma maxQ^{'}(s^{'}, a^{'}), & otherwise \end{cases}$$

      Calculated for MSE using gradient descent $loss = (y_i - Q_{s_t,a_t})^2$;
      Exchange the parameters of the prediction network and the target network;
      $\theta = \theta^{'}$
    end
  end
end

IV. SIMULATION RESULTS

In this section, we designed four baselines to compare with multi-agent DQN model. They are "All local processing", "Random offload processing", "Approximate iterative optimization", and "Particle swarm algorithm" for comparison. The simulation results show that under the complex system model, the DQN model of the multi-agent can approximately find the optimal solution. The parameters of the simulation are given in the Table I.

In the proposed cell system, we use eight users for simulation. In order to compare the performance of all network convergence, we also added two baselines. While exploring the best action, each user made a comparison between all local and all offloading. Each agent interacts with the environment,

TABLE I
SIMULATION PARAMETERS

Parameters	Value
Cell coverage radius	500m
All of user	8
Pathloss exponent α	3.7
Bandwidth	1MHz
User maximum transmission power	23dBm
Noise power	-70dBm
MBS server computing capacity	25GHz
BS server computing capacity	2.5GHz
Maximum local computing capacity	1GHz
Iuput data size	[100,200]Kbit
CPU cycles of per bit	150 cycles/bit
Maximum user delay	0.5s
Coefficient η	10^{-27}
Penalty coefficient ψ	-0.02

learns from previous experience, and gradually moves towards the direction of maximum reward, and finally converges.

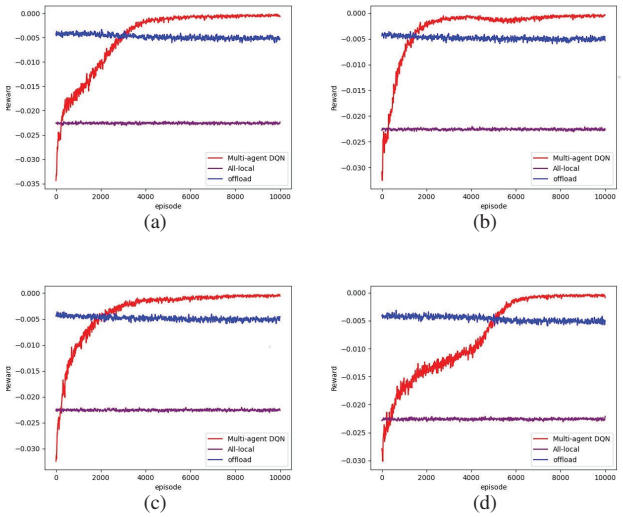

Fig. 2. Convergence diagram of different agents.

From the Fig. 2, we can see that all agents with different curves are following convergence. We can see that as the training progresses, the energy consumption curve of each agent gradually rises. In our design, the training data input to the neural network is the channel gain and task volume of all users. Each neural network is trained independently, and each agent is in a selfish environment, and there is no parameter transfer between each other. As each agent optimizes the search, it also compares local and random offloading. Besides, we can see that at the beginning, the agent has not learned enough experience, so it is difficult to make the best action and cannot be optimized. As the training progresses, the agent learns the experience, and the curve of energy consumption gradually rises, which is better than local and offloading, and finally converges. Each agent approaches the approximate optimal solution. When each agent converges, we can conclude that the reward of the total network also converges.

Energy consumption comparison chart of other methods as shown in Fig.3. We set different hyperparameters for the multi-agent algorithm.

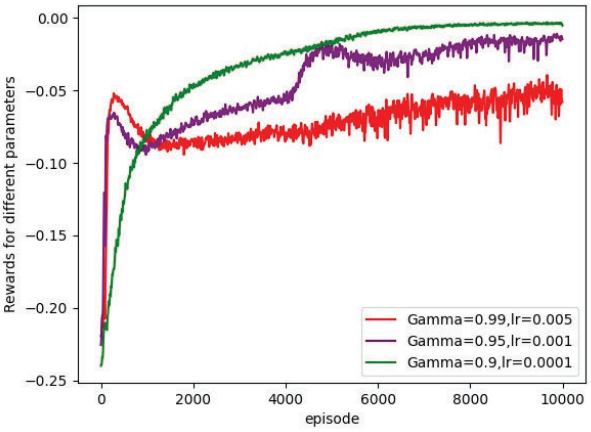

Fig. 3. Multi-agent rewards with different parameters.

From the Fig. 3, we can see that the learning rate and the gamma parameter of the future reward will affect the convergence speed and degree of the curve. The size of the learning rate affects the magnitude of the gradient drop of the loss function. In the multi-agent algorithm, the large learning rate causes the agent to update too quickly, but the agents are selfish and cannot share information quickly, which further causes in low convergence of the curve.

When the DQN algorithm updates the reward, the role of the gamma parameter is similar to a pair of myopia glasses. When the value of the parameter is too large, it will cause the agent to care too much about future rewards and not pay attention to current benefits. Otherwise, it will be overly concerned about immediate benefits. When the $\gamma = 0.9$, the agent can weigh the future rewards and the instant rewards well. The agent can avoid falling into a state of local convergence due to excessive concern about the future or instant rewards, and finally reach a state of stable convergence.

The total energy consumption under different delay is shown in Fig.4. As can be seen from the Fig.4, the total energy consumption of users increases as the maximum delay decreases. As the user maximum endurance time decreases, the system allocates more resources to the task. For the offloading task, it is fond of selecting a server with larger capacity and a better channel. However, it will lead to the increase of interference in the system. In order to satisfy the maximum latency of users, more computing resources are consumed when processing tasks locally and offloading to MEC, resulting in increased energy consumption.

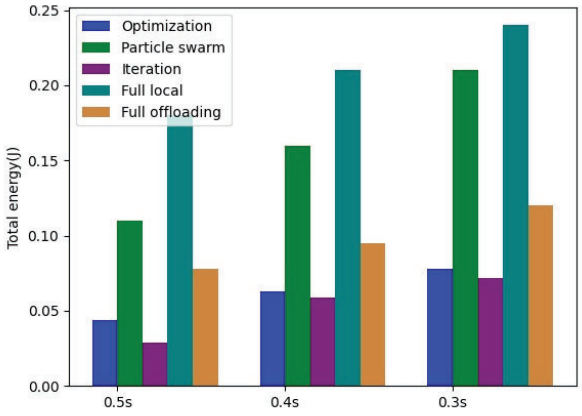

Fig. 4. Total power consumption with different delays.

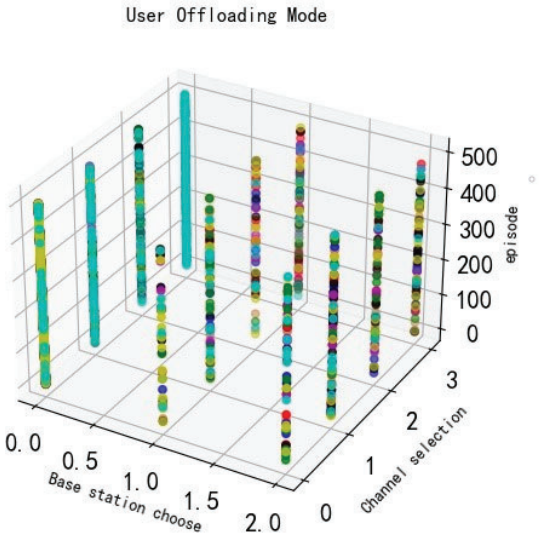

Fig. 5. User Offloading Selection.

We show the base station selection and channel selection methods of all users in Fig.5. We randomly selected 500 user offloading decisions from all episodes as sample points. The intensity of the scatter plot represents the intensity of user selection. From the Fig.5, we can see that the method of (0, 0) has the largest density of scattered points. This way represents local and macro base stations. Because the MEC processing capacity of the macro base station is slightly larger than that of the small cell BS's, in order to obtain the minimum energy consumption, users try to select the macro base station. When too many users select the macro base station, it causes in excessive channel interference and each agent needs to dynamically obtain the minimum energy consumption according to different offloading methods.

V. Conclusion

In this paper, we investigated the energy minimization task offloading and resource allocation for MEC in NOMA-HetNets. To minimize the energy consumption of all users while satisfing the QoS requirement, we jointly optimized task offloading decision, substation resource allocation and subchannel resource allocation. In our model, each agent interacts with the environment, learns from experience, and learns optimal actions. Multi-agents not only reduce the difficulty of exploring the action space, but also increase the convergence speed of the network, and can quickly achieve the optimal solution. The simulation results show that our algorithm is superior to other methods and approximately closer to the optimal solution.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 62071306, and in part by Shenzhen Science and Technology Program under Grants JCYJ20200109113601723 and JSG-G20210420091805014.

References

[1] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322-2358, Fourthquarter 2017.

[2] A. Kiani and N. Ansari, "Edge Computing Aware NOMA for 5G Networks," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1299-1306, April 2018.

[3] Y. Liu, Z. Qin, M. Elkashlan, Z. Ding, A. Nallanathan and L. Hanzo, "Nonorthogonal Multiple Access for 5G and Beyond," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2347-2381, Dec. 2017.

[4] S. Singh and J. G. Andrews, "Joint Resource Partitioning and Offloading in Heterogeneous Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 2, pp. 888-901, February 2014.

[5] C. Xu, G. Zheng and X. Zhao, "Energy-Minimization Task Offloading and Resource Allocation for Mobile Edge Computing in NOMA Heterogeneous Networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16001-16016, Dec. 2020.

[6] L. P. Qian, A. Feng, Y. Huang, Y. Wu, B. Ji and Z. Shi, "Optimal SIC Ordering and Computation Resource Allocation in MEC-Aware NOMA NB-IoT Networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2806-2816, April 2019.

[7] Liu Y , Yu H , Xie S , *et al*. "Deep Reinforcement Learning for Offloading and Resource Allocation in Vehicle Edge Computing and Networks," *IEEE Transactions on Vehicular Technology*, PP.99(2019):1-1.

[8] X. Qiu, W. Zhang, W. Chen and Z. Zheng, "Distributed and Collective Deep Reinforcement Learning for Computation Offloading: A Practical Perspective," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1085-1101, 1 May 2021.

[9] X. Wang, Y. Zhang, R. Shen, Y. Xu and F. -C. Zheng, "DRL-Based Energy-Efficient Resource Allocation Frameworks for Uplink NOMA Systems," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7279-7294, Aug. 2020.

[10] J. Li, F. Wu, K. Zhang and S. Leng, "Joint Dynamic User Pairing, Computation Offloading and Power Control for NOMA-based MEC System," *in 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2019, pp. 1-6.

[11] X. Zhu, Y. Luo, A. Liu, M. Z. A. Bhuiyan and S. Zhang, "Multiagent Deep Reinforcement Learning for Vehicular Computation Offloading in IoT," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9763-9773, 15 June15, 2021.