

Joint Resource Allocation and Task Offloading for Hybrid NOMA-assisted MEC Network with Network Slicing

Yong Zhang*, Hao Zhang[†], Yan Li[†], Yong Zhang^{‡§} and Siyu Yuan^{*‡§}

^{*}State Grid Shandong Electric Power Company, Shandong, China

Email: zy5715@wo.cn

[†]Economic & Technical Research Institute

State Grid Shandong Electric Power Company, Shandong China

Email: faszah@yeah.net, liyan@jyy.sd.sgcc.com

[‡]School of Electronic Engineering

Beijing University of Posts and Telecommunications, Beijing, China

Email: yongzhang@bupt.edu.cn, yuanisyu@bupt.edu.cn

[§]Beijing Key Laboratory of Work Safety Intelligent Monitoring

Beijing University of Posts and Telecommunications, Beijing, China

Abstract—Mobile edge computing (MEC) and non-orthogonal multiple access (NOMA) are regarded as promising technologies in fifth-generation (5G) networks. This paper proposes a hybrid NOMA-assisted MEC network slicing scenario. In one network slice, users are divided into some groups and use hybrid NOMA to occupy channel for communication. A two-step resource allocation and task offloading algorithm is proposed to minimize the total network energy consumption under the premise of maximum delay tolerance. The optimization problem is divided into two subproblems. One subproblem is the optimal resource allocation problem for users within a group, and the other is the user grouping problem. A two-layer deep reinforcement learning (DRL) algorithm based on double deep Q network (DDQN) is used to solve the resource allocation within a group, and a swap matching algorithm is used to solve the user grouping. The simulation results show that the proposed algorithm can effectively reduce the total energy consumption. Compared with DQN and Q-learning, the proposed DRL algorithm has faster convergence speed and better convergence value. Compared with exhaustive search, matching search has lower complexity and similar performance.

Index Terms—MEC, hybrid NOMA, network slicing, DRL, swap matching

I. INTRODUCTION

In recent years, as a technology that connects billions of resource-constrained wireless devices (such as mobile devices, sensors, wearables, etc.) to the Internet, the Internet of Things (IoT) has developed explosively and rapidly [1]–[3]. At the same time, the business scenarios of 5G networks put forward higher requirements for indicators such as delay, energy efficiency, number of device connections and reliability [4]. As the key technology of 5G networks, network slicing is an on-demand networking method [5], which allows operators to separate multiple virtual networks on a unified infrastructure to serve various types of applications.

MEC technology provides computing services for edge user devices [6], [7]. MEC deploys computing and storage resources to the edge of radio access network (RAN). At present, researchers focus on the joint communication and computing resources allocation of MEC network to solve the tradeoff between computing delay and energy consumption. Wang F et al. propose an IoT device offloading and scheduling algorithm in the wireless power transmission environment. The IoT device can compute tasks locally or offload tasks to the MEC server [8]. Sun H et al. mainly study the problem of maximizing the energy efficiency of MEC system with joint task offloading and computing resource allocation, and solve this problem through iteration and gradient descent algorithm [9].

NOMA allows multiple users to occupy one channel for transmission at the same time, effectively improving system capacity and spectral efficiency [10]. Now, some researches have combined NOMA technology with MEC network. Pan Y et al. propose an energy saving task offloading algorithm for NOMA-based MEC environment. This algorithm determines the uplink power, solving the problem of task offloading division and resource allocation [11]. Fang F et al. propose a resource allocation optimization algorithm based on imperfect channel state information (CSI) in NOMA-MEC network, minimizing the total network energy consumption through task allocation, power allocation and user association [12].

Reinforcement learning (RL) has a good performance in decision-making in complex scenarios. Limited by the storage form of table, early RL algorithm (Q-Learning) has poor effect on processing high-dimensional data [13]. DRL combines deep learning (DL) and RL, and processes high-dimensional data through deep neural network (DNN), which significantly improves the effect of RL. The DeepMind team proposes DQN [14] in 2013, which is used to play Atari video games and

obtain high scores. In the follow-up research, they propose some DQN variant algorithms [15]–[17].

In this paper, we propose a MEC network user task offloading scenario based on the hybrid NOMA communication mode. We transform the resource allocation and task offloading problem in the proposed scenario into two subproblems. The first subproblem is to solve the problem of resource allocation and task offloading in user groups on the premise that the user information in the group is known. The second subproblem is to solve the optimal grouping of all users. For the first subproblem, we propose a joint resource allocation and task offloading algorithm based on DDQN. By controlling users' transmission power and task offloading, we minimize the total energy consumption of users in the group within the maximum delay tolerance of the network slice. For the second subproblem, we propose a user grouping algorithm based on swap matching algorithm to minimize the total network energy consumption.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

This paper considers the task offloading scenario in hybrid NOMA-assisted MEC network with network slicing, as shown in Fig. 1. The scenario includes a base station (BS) equipped with MEC server and some users. The user set is denoted as $M = \{1, 2, \dots, |M|\}$, which $|M|$ is the total number of users. The set of network slices is denoted as $S = \{1, 2, \dots, |S|\}$, which $|S|$ is the total number of network slices. Users associate different network slices according to the delay requirements of computing tasks. The maximum delay tolerance of network slice s is $D_s^{\max}$. The delay requirement of the user m is denoted as τ_m . Therefore, the user $m \in M$ associated network slice $s \in S$ needs to satisfy $t_m \leq D_s^{\max} \leq \tau_m$, where t_m is the total delay of the user m to complete the computing task.

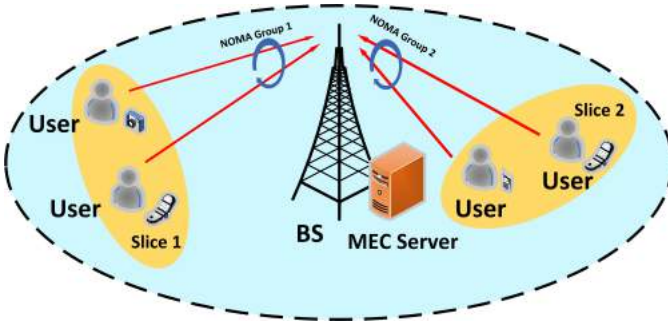


Fig. 1. MEC System Model.

The user's computing task can be divided into several subtasks, and there is a dependency relationship between the subtasks. Users only have limited computing and storage resources, which is difficult to satisfy the delay requirements. To reduce the delay, users can offload part of computing tasks to the MEC server. Assume that the subtasks of users are computationally intensive, timely, and indivisible. For each subtask, users can complete locally or offload to MEC server.

We divide $|M|$ users into n groups, and each group uses hybrid NOMA to communicate on the same channel. Only users associated with the same network slice can be assigned to one group. To reduce the complexity of NOMA decoding, we assume that the number of users in each group is 2, that is $|M| = 2n$. Since each channel is only used by users in the same group, the user's communication is only related to users in the same group, not users in other groups.

In each user group, we sort users according to their delay requirements, and denote two users as user a and user b , so that $\tau_a \leq \tau_b$. Users a and b use the hybrid NOMA to communicate on the same channel. The hybrid NOMA mode is shown in Fig. 2, where t_{NOMA} and t_{OMA} respectively represent the NOMA and OMA transmission times. Since user a 's delay requirement is shorter, in order to enable user a to complete the computing task as soon as possible, user a and user b first use NOMA to share the channel. After user a completes the transmission task, user b uses OMA to occupy the channel alone.

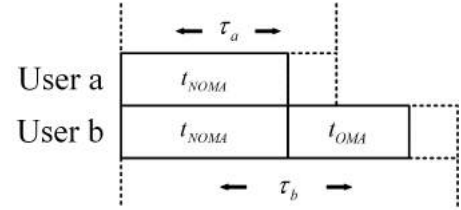


Fig. 2. Hybrid NOMA Transmission Mode.

B. Problem Formulation

In the NOMA transmission process, in order to enable user a to obtain a higher transmission rate, user b is first decoded, and user a 's transmission will not be interfered by user b . According to Shannon channel formula, the transmission rate of user a is shown in (1), where B is the channel bandwidth, p_a is the transmission power of user a , g_a is the channel gain between user a and BS and $B\sigma^2$ is the AWGN noise power.

$$R_a = B \log_2 \left(1 + \frac{p_a g_a}{B\sigma^2} \right) \quad (1)$$

According to the successive interference cancellation (SIC), to enable user a to transmit at the rate in (1), user b 's transmission rate needs to meet the requirements in (2), where p_b^{NOMA} is the transmission power of user b during NOMA transmission, and g_b is the channel gain between user b and BS.

$$R_b^{NOMA} \leq B \log_2 \left(1 + \frac{p_b^{NOMA} g_b}{p_a g_a + B\sigma^2} \right) \quad (2)$$

After the transmission task of user a is completed, user b occupies the channel in OMA transmission mode, and the transmission rate is shown in (3), where p_b^{OMA} is the transmission power of user b during OMA transmission.

$$R_b^{OMA} = B \log_2 \left(1 + \frac{p_b^{OMA} g_b}{B\sigma^2} \right) \quad (3)$$

The user's computing tasks can be divided into I subtasks. These subtasks are denoted as $ST = [L_i, C_i]$, the total bits of the subtask i is denoted as L_i and the number of CPU cycles required is denoted as C_i . In addition, there is a dependency relationship between the subtasks. The computing of some subtasks is based on the results of other subtasks. The dependency relationship between subtasks is represented by a directed acyclic graph (DAG). Taking Fig. 3 as an example, subtask 3 is computing based on the result of subtask 1. If subtask 1 does not be completed, subtask 3 cannot start to compute.

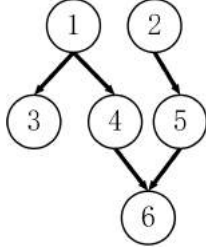


Fig. 3. Subtask Dependency Relationship DAG.

Users can offload some subtasks to the MEC server. ϕ_i is the computing location parameter of the subtask i . $\phi_i = 0$ indicates that the subtask i is computed locally, $\phi_i = 1$ means that the subtask i is offloading to MEC server.

Generally speaking, the length of the result is shorter than the content of the task, and the task offloading is uplink communication. The result return is downlink communication. The transmission power of BS is usually higher than the transmission power of the users. Therefore, the communication delay caused by result return can be ignored.

The delay to complete the subtask i locally is $t_i^{loc} = \frac{C_i}{f^{loc}}$, where f^{loc} is the user's CPU frequency, and the energy consumption is $E_i^{loc} = \kappa C_i (f^{loc})^2$, where κ is the effective capacitance coefficient of each CPU cycle of the user.

The delay to transmit the subtask i 's content to the MEC server is $t_i^{tran} = \frac{L_i}{R}$, where R is the user's transmission rate, and the energy consumption is $E_i^{tran} = \frac{p L_i}{R}$, where p is the user's transmission power. The delay for the MEC server to complete the subtask is $t_i^s = \frac{C_i}{f^s}$, where f^s is the CPU frequency of the MEC server, and the energy consumption is $E_i^s = \kappa_s C_i (f^s)^2$, where κ_s is the effective capacitance coefficient of each CPU cycle of the MEC server. Therefore, the total delay to offload the subtask to the MEC server is $t_i^{off} = t_i^{tran} + t_i^s$, and the total energy consumption is $E_i^{off} = E_i^{tran} + E_i^s$.

For each subtask, we define the prepare time $t_i^{prepare}$ and finish time t_i^{finish} . The prepare time indicates the time when the subtask i can be computed, denoted as $t_i^{prepare} = \max_{l \in dep\{i\}} \{t_l^{finish}\}$, where $dep\{i\}$ indicates the set of antecedent dependent subtasks of subtask i . Finish time refers to the time to complete subtask i , denoted as $t_i^{finish} = t_i^{prepare} + (1 - \phi_i) \cdot t_i^{loc} + \phi_i \cdot t_i^{off}$.

User a always adopts the NOMA transmission, with transmission power p_a and transmission rate R_a . The total time for user a to complete all subtasks is $t_a = \max_{i \in ST_a} \{t_{a,i}^{finish}\}$, and the total energy consumption is $E_a = \sum_{i \in ST_a} [(1 - \phi_{a,i}) \cdot E_{a,i}^{loc} + \phi_{a,i} \cdot E_{a,i}^{off}]$.

User b firstly adopts the NOMA transmission. The duration of NOMA is $t_{noma} = \frac{\sum_{i \in ST_a} (\phi_{a,i} \cdot L_{a,i})}{R_a}$, with transmission power p_b^{noma} and transmission rate R_b^{noma} . After user a completes the transmission, user b uses the OMA transmission. The duration of OMA is $t_{oma} = \frac{\sum_{i \in ST_b} (\phi_{b,i} \cdot L_{b,i}) - t_{noma} \cdot R_b^{noma}}{R_b^{oma}}$, with transmission power p_b^{oma} and the transmission rate R_b^{oma} . The total time for user b to complete all subtasks is $t_b = \max_{i \in ST_b} \{t_{b,i}^{finish}\}$, and the total energy consumption is $E_b = \sum_{i \in ST_b} [(1 - \phi_{b,i}) \cdot E_{b,i}^{loc} + \phi_{b,i} \cdot (E_{b,i}^s + E_{b,i}^{noma} + E_{b,i}^{oma})]$, where $E_{b,i}^{noma} = p_b^{noma} t_b^{noma}$ and $E_{b,i}^{oma} = p_b^{oma} t_b^{oma}$.

The optimization objective of this paper is to minimize the total energy consumption of all users under the maximum delay tolerance of the network slices. The optimization problem of joint resource allocation and task offloading in hybrid NOMA-assisted MEC network is shown in (4) - (9).

$$\min_{\{\phi_{a,i}\}, \{\phi_{b,i}\}, p_a, p_b^{noma}, p_b^{oma}} \sum_{j=0}^n (E_{a,j} + E_{b,j}) \quad (4)$$

$$s.t. \quad p_a, p_b^{noma}, p_b^{oma} \in [0, p_{max}] \quad (5)$$

$$\phi_{a,i} \in [0, 1], \forall i \in ST_a \quad (6)$$

$$\phi_{b,i} \in [0, 1], \forall i \in ST_b \quad (7)$$

$$t_a \leq D_s^{max} \leq \tau_a \quad (8)$$

$$t_b \leq D_s^{max} \leq \tau_b \quad (9)$$

Due to the nonlinear constraints of continuous variables (such as p_a) and binary variables (such as $\phi_{a,i}$), the optimization problem is a non-convex optimization problem. Deep reinforcement learning is a feasible way for this kind of non-convex optimization problem. Therefore, we propose a resource allocation algorithm based on deep reinforcement learning to solve this optimization problem.

III. TWO-LAYER DRL-BASED RESOURCE ALLOCATION ALGORITHM WITHIN A GROUP

A. DDQN

Reinforcement learning is a common method to solve decision-making problem. The RL agent needs to obtain a good strategy in the continuous exploration learning. RL is usually described by Markov Decision Process (MDP).

The most common RL algorithm is Q-Learning algorithm, which uses Q-Table to store the actions' values. The update

formula of Q-Learning is (10). Through constant updating, we can get an excellent Q-Table to carry out the decision-making process.

$$Q(s, a) = Q(s, a) + \alpha [R(s, a) + \gamma \max_{a'} Q(s', a') - Q(s, a)] \quad (10)$$

DRL combines RL with DL, and uses DNN to replace the original table to calculate the actions' values. DQN is a representative algorithm of DRL. DNN is used to calculate the Q value (action value function) in DQN. In the decision-making process, DQN takes the state as the input of the DNN, and calculates the Q value of each action through DNN. The original Q value $Q(s, a)$ is replaced by a new form $Q(s, a; \theta)$, where θ represents the parameters of DNN. In order to reduce the problems caused by the correlation between data, DQN uses two key technologies: experience pool replay and fixed target network.

The loss function of DQN is defined as the square error form of target Q and estimated Q. (11) is the form of DQN loss function, where B is the set of mini batch, and γ is the learning rate.

$$L^{DQN} = \sum_{b \in B} [r_b + \gamma \max_{a'} Q(s_b', a_b'; \theta^-) - Q(s_b, a_b; \theta)]^2 \quad (11)$$

DDQN solves the problem of over-estimation of DQN. DDQN uses different value functions to realize action selection and action evaluation. The loss function form of DDQN is shown in (12).

$$L^{DDQN} = \sum_{b \in B} [r_b + \gamma Q(s_b', \arg\max_{a'} Q(s_b', a_b'; \theta); \theta^-) - Q(s_b, a_b; \theta)]^2 \quad (12)$$

B. Two-Layer DRL-Based Resource Allocation Algorithm (MDDQN)

In order to solve the problem of resource allocation in MEC network, this paper proposes a two-layer resource allocation algorithm based on DDQN algorithm. The lower layer uses task controller to allocate computing resources. The upper layer uses the power controller to allocate communication resources according to the task controller's allocation results, as shown in Fig. 4.

The basic elements of task controller and power controller are set as follows:

- **Agent:** The task controller is configured at the users to manage the user's subtask offloading. The power controller is configured at BS to manage the transmission power of users.
- **State:** The state of the task controller is expressed as $S_{tc} = \{a_{i,j}\}_{L \times L}$, where $\{a_{i,j}\}_{L \times L}$ is the adjacency matrix of the subtask dependency DAG. The state of the power controller is expressed as $S_{pc} =$

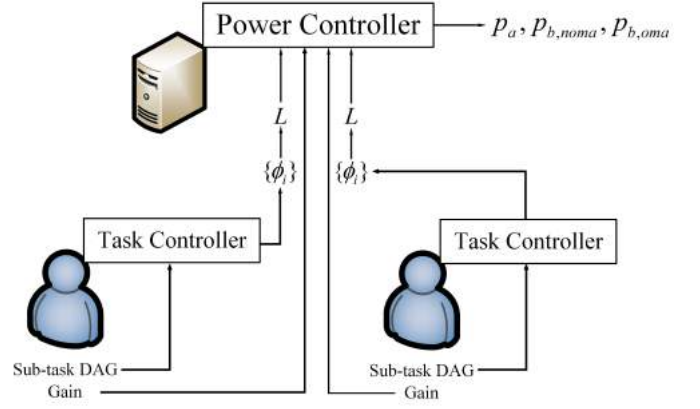


Fig. 4. Two-Layer DRL-Based Resource Allocation Algorithm.

$[g_a, g_b, L_a^{tran}, L_b^{tran}]$, where L_a^{tran} is the total number of bits transmitted by the user a .

- **Action:** The action of the task controller is expressed as $A_{tc} = \{\phi_i\}_{1 \times L}$. The action of the power controller is expressed as $A_{pc} = [p_a, p_{b,noma}, p_{b,oma}]$.
- **Reward:** The reward functions of task controller and power controller need to consider whether the users meet the network slice maximum delay tolerance. If the network slice maximum delay tolerance is not met, the reward value will be 0. The step function meets our expectations for the reward function. However, the step function is discontinuous at the point $x=0$, which makes it difficult to update the gradient of DNN. We use Sigmoid function instead of step function, and the expression of Sigmoid function is $S(x) = \frac{1}{1+e^{-x}}$. The network slice delay constraint is shown in (13), where t_a and t_b respectively represent the total delay of user a and b . Appendix I introduces the calculation methods of t_a and t_b .

$$f(t_a, t_b) = S(D_s^{\max} - t_a) \cdot S(D_s^{\max} - t_b) \quad (13)$$

The reward function is shown in (14).

$$R = \frac{f(t_a, t_b)}{E_a + E_b} \quad (14)$$

IV. MATCHING THEORY-BASED USER GROUPING ALGORITHM

In Section III, we propose a resource allocation algorithm within a group. In this section, we will extend the intra-group resource allocation scheme to multi-user scenarios by proposing a user grouping algorithm. Usually, the exhaustive search method can be used to solve this problem. However, when the number of users is large, due to the high complexity of the exhaustive search method, it takes a long time to obtain the optimal solution. To overcome this problem, we propose a low complexity user grouping algorithm based on swap matching theory, to minimize the total energy consumption of MEC network system.

A. Swap Matching-Based User Grouping Algorithm

The user grouping algorithm can realize bilateral matching between user set M and user group set N . In order to combine the resource allocation algorithm in Section III, it is assumed that each user group contains only two users, which means $|M| = 2|N|$. Therefore, we propose a two-to-one user grouping algorithm based on swap matching algorithm, which is defined as follows:

[Definition 1] The two-to-one mapping relationship Ψ from M to N need to meet the following requirements:

1. $\Psi(m) \subseteq N, \forall m \in M$
2. $\Psi^{-1}(n) \subseteq M, \forall n \in N$
3. $|\Psi(m)| = 1$
4. $|\Psi^{-1}(n)| = 2$

In Definition 1, condition 1 indicates that each user has a corresponding group. Condition 2 indicates that each group corresponds to several users. Condition 3 indicates that each user corresponds to only one group. Condition 4 indicates that there are two users in each group.

$E_m(\Psi)$ is the energy consumption of user m in the case of user group matching Ψ . $E_n(\Psi)$ is the total energy consumption of group n in the case of user group matching Ψ . n_m denotes the group of user m .

To minimize the total energy consumption, two users from different user groups are allowed to exchange. The two users form a swap pair, denoted as (m, m') . $\Psi_m^{m'}$ is the matching relationship after exchanging swap pair (m, m') . After exchanging swap pair (m, m') , if the energy consumptions of each user and group don't increase and the energy consumption of at least one user or group decreases, the swap pair is called an effective swap pair. Effective swap pairs are defined as follows:

[Definition 2] An effective swap pair (m, m') need to meet the following requirements:

1. $\forall i \in \{m, m', n_m, n_{m'}\}, E_i(\Psi_m^{m'}) \leq E_i(\Psi)$
2. $\exists i \in \{m, m', n_m, n_{m'}\}, E_i(\Psi_m^{m'}) < E_i(\Psi)$

The matching process will repeatedly search for effective swap pairs until a stable matching state is reached. The stable matching state is defined as follows:

[Definition 3] In a matching state, if there is no effective swap pair, the matching is in a stable matching state.

Algorithm 1 is the proposed two-to-one user grouping algorithm based on the swap matching algorithm.

B. Complexity Analysis

In this section, we analyze the time complexity of the user grouping algorithm based on matching search, and compare it with the time complexity of exhaustive search. Suppose there are n users in total, each group has two users. The time complexity of exhaustive search is $O(\frac{n!}{2^{n/2}})$. According to the natural logarithm, the time complexity is rewritten to $O(\ln(n!))$. According to Stirling's formula $\ln(x!) = x \ln x - x + O(\ln(x))$, the time complexity is rewritten to $O(n \ln(n))$. The time complexity of the proposed matching search algorithm in one cycle is $O(n^2)$. According to the natural logarithm, the time complexity is rewritten to $O(\ln(n))$. Obviously, $O(\ln(n)) <$

Algorithm 1 Two-to-One User Grouping Algorithm Based on Swap Matching

```

1: Initialization: Randomly group users, get  $\Psi_{init}$ ;
2: while the current user group matching does not have an
   effective swap pair do
3:   for each  $m \in M$  do
4:     User  $m$  and user  $m'$  in other group form swap pairs
       in turn;
5:     if  $(m, m')$  is an effective swap pair according to
       Definition 2 then
6:       User  $m$  and  $m'$  exchange group;
7:     else
8:       User  $m$  and  $m'$  don't exchange group;
9:     end if
10:  end for
11: end while

```

TABLE I
HYPERPARAMETERS OF MDDQN

Papameter	Value
Memory pool sizes	20000
Mini-batch size	32
Learning rate δ	0.005
Activation function	ReLU
Optimizer	Adam

$O(n \ln(n))$, which proves that the complexity of the proposed user grouping algorithm based on matching search is lower than that of the user grouping algorithm based on exhaustive search.

V. SIMULATION SETTING AND RESULTS

A. Simulation Settings

In the MEC network, one BS equipped with MEC server and 10 users are deployed. The size of the network is 100m * 100m. The location of the BS is fixed in the center of the scene, and users are randomly distributed in the range of the network model. The user can offload part of computing task to the BS. The maximum transmission power of the user is $P_{\max} = 40\text{dBm}$. Every two users are divided into one user group, and two users of the same user group occupy the same channel. The bandwidth of each channel is 10MHz. The noise power of AWGN is $\sigma^2 = -174\text{dBm/Hz}$. We consider the channel fading model including large-scale fading and small-scale fading. Large-scale fading includes path loss and shadow fading. The expression of path loss is $L(d) = 37 + 30 \log(d)$ [18]–[20], where d is the distance between the BS and the user. The standard deviation of shadow fading is 8dB. Rayleigh channel model with mean value of 0dB and variance of 8dB is used for small-scale fading [21].

The hyperparameter settings for MDDQN algorithm are shown in Table I.

B. Simulation Results

The resource allocation algorithm proposed in this paper includes an intra-group resource allocation algorithm based

on DRL and an inter-group user grouping algorithm based on swap matching algorithm. This section first verifies the simulation performance of the MDDQN used to solve the resource allocation within a group. Then, the rest of the paper verifies the performance of the swap matching algorithm for user grouping.

This paper compares MDDQN algorithm with MDQN and MQ-learning. The performance of each algorithm is shown in Fig. 5. MQ-learning stores Q values in a table, which requires discretization in both action space and state space. MDDQN and MDQN only need to discretize the action space. Therefore, MQ-learning is affected by quantization error caused by state space discretization, and its convergence speed and convergence value are significantly lower than MDDQN and MDQN. MDDQN solves the over-estimation problem of MDQN, and its convergence speed, stability and convergence value are superior to MDQN.

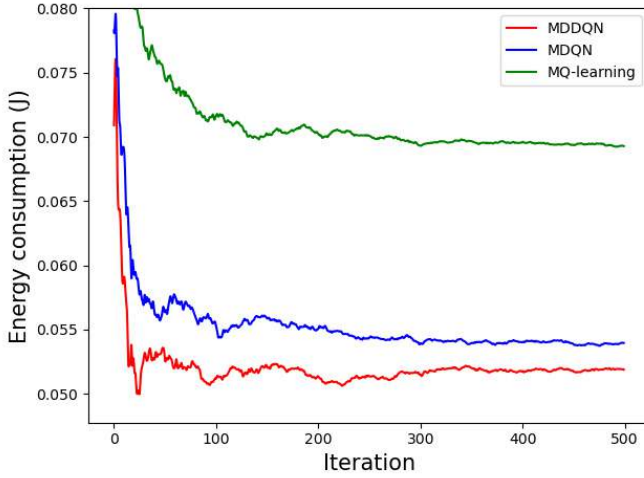


Fig. 5. Performance Comparison of Different Algorithms.

Learning rate is one of the key hyperparameters of DRL. Fig. 6 shows the performance comparison of MDDQN under different learning rates. When the learning rate is small ($LR=0.0005$), the convergence speed of MDDQN is slow. With the increase of learning rate ($LR=0.005$), the convergence speed of MDDQN increases. When the learning rate is too large ($LR=0.05$), the convergence speed of MDDQN further increases. However, the excessive learning rate makes the agent easy to fall into the local optimal solution. As shown in the figure, the convergence energy consumption of the curve $LR=0.05$ is higher than that of the low learning rate.

Fig. 7 shows the performance comparison of different communication modes under different slice maximum delay tolerance. Compared with the pure NOMA (P-NOMA) and OMA, the hybrid NOMA (H-NOMA) can significantly reduce the energy consumption. Compared with offloading computing tasks to the MEC server, the energy consumption for completing locally is lower, but the delay for completing locally is longer. As the maximum slice delay tolerance decreases,

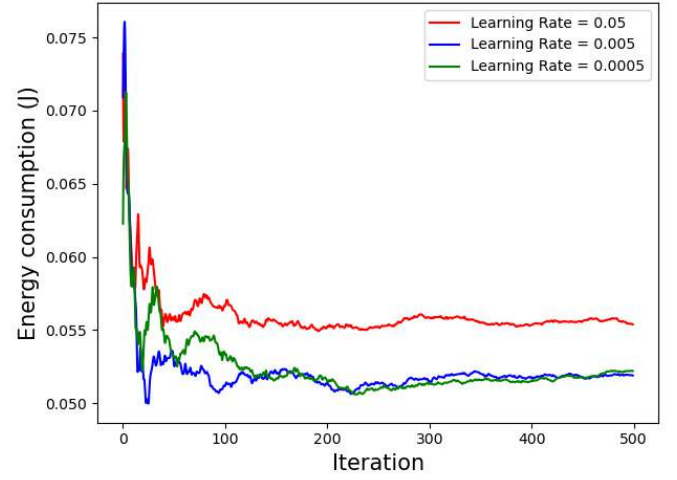


Fig. 6. Performance Comparison of MDDQN with Different Learning Rates.

users offload more computing tasks to the MEC server, which increases energy consumption.

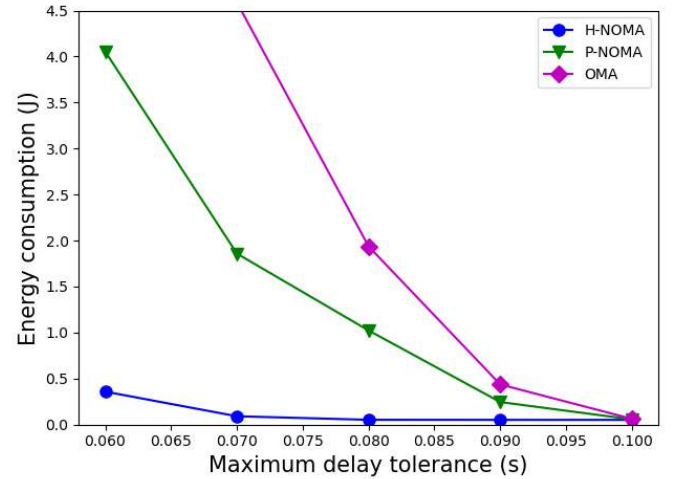


Fig. 7. Performance Comparison of MDDQN with Different Communication Mode under Different Maximum Delay Tolerance.

In order to verify the performance of the user grouping algorithm, this paper combines the trained MDDQN agent with swap matching search and exhaustive search respectively, and the performance comparison is shown in Fig. 8. The total energy consumption of the MDDQN + Matching search algorithm is very close to the optimal solution obtained by the MDDQN + exhaustive search algorithm. As described in Section IV-B, the time complexity of swap matching search algorithm is lower than that of exhaustive search. In general, swap matching search algorithm has lower time complexity and similar performance with exhaustive search algorithm.

VI. CONCLUSION

In this paper, we propose a hybrid NOMA-assisted MEC network scenario. Users can offload part of subtasks to the

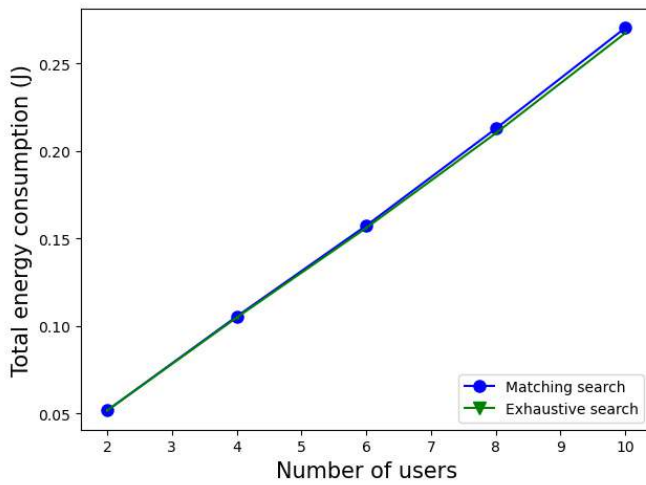


Fig. 8. Performance Comparison of Matching Search and Exhaustive Search under Different Number of Users.

MEC server. Users associate corresponding network slices according to their own delay requirements. The users of the same network slice are associated for grouping, and the users of the same group use the hybrid NOMA to occupy the channel. In this paper, the problem of joint resource allocation and task offloading is transformed into two subproblems. The first subproblem is the optimal resource allocation and task offloading problem within a group. This paper proposes an optimization algorithm based on DRL, which minimizes the total energy consumption under the premise of the maximum delay tolerance of slices by controlling users' transmission power and task offloading selection. The second subproblem is the user grouping problem. This paper proposes a low complexity user grouping algorithm based on swap matching search to minimize the total energy consumption of the network. Simulation results show that compared with DQN and Q-learning, the proposed DRL algorithm (MDDQN) has faster convergence speed and better convergence value. Compared with exhaustive search, swap matching search has less complexity and similar performance.

ACKNOWLEDGMENT

This research has been supported by the Science and Technology Project of State Grid Corporation of China: Research on Key Technologies of Flexible Ethernet(FlexE) Technology in Energy Internet (Grant No. 5700-202116378A-0-0-00).

REFERENCES

- [1] Wan S, Hu J, Chen C, et al. Fair-hierarchical scheduling for diversified services in space, air and ground for 6g-dense internet of things[J]. *IEEE Transactions on Network Science and Engineering*, 2020, 8(4): 2837-2848.
- [2] Balasubramanian V, Aloqaily M, Reisslein M. An SDN architecture for time sensitive industrial IoT[J]. *Computer Networks*, 2021, 186: 107739.
- [3] Wan S, Gu R, Umer T, et al. Toward offloading internet of vehicles applications in 5G networks[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2020, 22(7): 4151-4159.
- [4] Spinelli F, Mancuso V. Toward enabled industrial verticals in 5G: A survey on MEC-based approaches to provisioning and flexibility[J]. *IEEE Communications Surveys & Tutorials*, 2020, 23(1): 596-630.
- [5] Katsalis K, Nikaein N, Schiller E, et al. Network slices toward 5G communications: Slicing the LTE network[J]. *IEEE Communications Magazine*, 2017, 55(8): 146-154.
- [6] Chen X, Jiao L, Li W, et al. Efficient multi-user computation offloading for mobile-edge cloud computing[J]. *IEEE/ACM transactions on networking*, 2015, 24(5): 2795-2808.
- [7] Liu L, Chang Z, Guo X, et al. Multiobjective optimization for computation offloading in fog computing[J]. *IEEE Internet of Things Journal*, 2017, 5(1): 283-294.
- [8] Wang F, Xu J, Wang X, et al. Joint offloading and computing optimization in wireless powered mobile-edge computing systems[J]. *IEEE Transactions on Wireless Communications*, 2017, 17(3): 1784-1797.
- [9] Sun H, Zhou F, Hu R Q. Joint offloading and computation energy efficiency maximization in a mobile edge computing system[J]. *IEEE Transactions on Vehicular Technology*, 2019, 68(3): 3052-3056.
- [10] Dai L, Wang B, Yuan Y, et al. Non-orthogonal multiple access for 5G: solutions, challenges, opportunities, and future research trends[J]. *IEEE Communications Magazine*, 2015, 53(9): 74-81.
- [11] Pan Y, Chen M, Yang Z, et al. Energy-efficient NOMA-based mobile edge computing offloading[J]. *IEEE Communications Letters*, 2018, 23(2): 310-313.
- [12] Fang F, Wang K, Ding Z, et al. Energy-efficient resource allocation for NOMA-MEC networks with imperfect CSI[J]. *IEEE Transactions on Communications*, 2021, 69(5): 3436-3449.
- [13] Watkins C J C H, Dayan P. Q-learning[J]. *Machine learning*, 1992, 8(3): 279-292.
- [14] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. *nature*, 2015, 518(7540): 529-533.
- [15] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double q-learning[C]//*Proceedings of the AAAI conference on artificial intelligence*. 2016, 30(1).
- [16] Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning[C]//*International conference on machine learning*. PMLR, 2016: 1995-2003.
- [17] Hessel M, Modayil J, Van Hasselt H, et al. Rainbow: Combining improvements in deep reinforcement learning[C]//*Thirty-second AAAI conference on artificial intelligence*. 2018.
- [18] Yang Z, Xu W, Xu H, et al. User association, resource allocation and power control in load-coupled heterogeneous networks[C]//*2016 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2016: 1-7.
- [19] ETSI TR 125 942 (V17.0.0): "Universal Mobile Telecommunications System (UMTS); RF system scenarios (3GPP TR 25.942 version 17.0.0 Release 17)"
- [20] Yang Z, Xu W, Shi J, et al. Association and load optimization with user priorities in load-coupled heterogeneous networks[J]. *IEEE Transactions on Wireless Communications*, 2017, 17(1): 324-338.
- [21] Dighe P A, Mallik R K, Jamuar S S. Analysis of transmit-receive diversity in Rayleigh fading[J]. *IEEE transactions on communications*, 2003, 51(4): 694-703.