UAV-Assisted MEC System Considering UAV Trajectory and Task Offloading Strategy

Kun Xiang and Yejun He*
State Key Laboratory of Radio Frequency Heterogeneous Integration
College of Electronics and Information Engineering, Shenzhen University, 518060, China
Email: 1303484745@qq.com, heyejun@126.com*

***Abstract*—As an emerging technology, mobile edge computing (MEC) can provide users with higher quality of service (Qos) such as reducing tasks computing latency and energy consumption of user equipments. Unmanned Aerial Vehicle (UAV) -assisted MEC can apply this technology to more scenarios. In this paper, we design a joint optimization algorithm to optimize the user's task offloading strategy and the trajectory of the UAV. When the MEC server interacts with multiple users at the same time, we adopt the differential evolution (DE) algorithm to obtain the offloading policy of each user in the current time slot based on the user location and UAV location. Aiming at the trajectory optimization problem of the UAV, we adopt the optimistic actor-critic (OAC) algorithm, which can minimize the weighted sum of energy consumption and delay of the system, and derive the optimal path through training. Simulation results show that the proposed algorithm is superior to other algorithms in terms of energy consumption and convergence performance.**

***Index Terms*—Mobile edge computing (MEC), UAV-assited, differential evolution (DE), optimistic actor-critic (OAC)**

I. INTRODUCTION

With the user terminals of 5G networks grows, more emerging smart devices and internet applications such as augmented reality (AR) /virtual reality (VR), cloud VR/AR, internet of vehicles, ultra-high definition video transmission and so on are being chosen by users [1]. But this will make the user terminals generate a large amount of task data. If the data is transferred to the cloud for processing, this will greatly increase the load of the cloud servers and the latency of users. Mobile edge computing (MEC), an emerging technology in 5G networks, is receiving attentions from more and more people, where servers are placed at the edge of the network and users can offload tasks to MEC servers for processing, thereby significantly reducing energy consumption and delay. For traditional terrestrial networks, placing MEC servers at the ground increases signal attenuation due to multipath effects and blocking caused by non-line-of-sight (NLoS) paths, which severely affects communication quality. Due to the high flexibility and easy deployment of the Unmanned Aerial Vehicle (UAV), UAV-assisted communication system has been widely noticed and studied. The MEC server is integrated with the UAV of the UAV-assisted MEC system, which can greatly reduce the energy consumption and latency of the system since the communication link established between the UAV and the ground user can be considered as a line-of-sight (LoS) path.

There has been a lot of relevant research in the field of UAV-assisted MEC system. Wang $et\ al.$ proposed a joint area division and trajectory optimization method to reduce the energy consumption of the UAV and achieve load balancing [2]. Liu $et\ al.$ devised a resource allocation and trajectory design framework and proposed a three-stage iterative algorithm to optimize the beamforming vector, resource allocation, and trajectory of the UAV to achieve system energy minimization [3]. In order to ensure the quality of experience for downlink users, Hu $et\ al.$ proposed a multi-stage alternative optimization algorithm to maximize the computational efficiency [4]. But as the complexity of the environment increases, the computation time of traditional algorithms may increase exponentially.

To overcome the weaknesses of traditional algorithms, the DRL algorithm has been applied to the field of UAV-assisted MEC system. Chen $et\ al.$ applied the DRL algorithm to the MEC domain and proposed an improved reinforcement learning algorithm based on the traditional algorithm for solving the computational offloading and resource allocation problems [5]. For the UAV trajectory optimization problem, Zhu $et\ al.$ proposed a sequence-to-sequence pointer network model to input the clustering of UAV locations and ground devices into the model, and used the actor-critic network to train the model to obtain the optimal trajectory of the UAV [6]. However, using traditional reinforcement learning algorithms cannot train better models in a short time due to the limitations of their exploration efforts. As a result, traditional reinforcement learning algorithms are unable to meet the training needs in complex environments.

This paper investigates the trajectory optimization of the UAV and tasks offloading of UDs in a UAV-assisted MEC system, where the UAV can integrate a server to handle the tasks of UDs offloading. The system can support communication conditions for UDs in extreme cases, reducing energy consumption and communication delay for UDs. To overcome the shortcomings of the traditional algorithms, we use the OAC reinforcement learning algorithm, where the approximate confidence bounds are explored by maximizing the state-action value function to train the optimal model. The main contributions of this paper are summarized as follows.

1) We propose a UAV-assisted MEC system that combines the tasks offloading strategy of UDs and the flight trajectory of the UAV with the aim of minimizing the weighted sum of energy consumption and delay of the system. The offloading strategy of UDs is optimized based on the current position of the UAV and the weighted sum of energy consumption and

ICC 2023 - IEEE International Conference on Communications | 978-1-5386-7462-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICC45041.2023.10279045

delay of the system, and the strategy includes total offloading, partial offloading, and local processing. The trajectory of the UAV is determined by the weighted sum of the optimized offload policy and the energy consumption and delay of the system.

2) To solve the proposed problem, we propose a joint optimization algorithm that uses the DE algorithm to find the current optimal offloading policy, and then determines the next action of the UAV based on the offloading policy and the weighted sum of energy consumption and delay of the system. Our proposed joint optimization algorithm outperforms other joint optimization algorithms in the weighted sum of energy consumption and delay and has a significant advantage in convergence.

The rest of the paper is organized as follows. Section II presents the proposed system model of the UAV-assisted MEC network and the proposed optimization problem. Section III presents the proposed joint Optimistic Actor-Critic and Differential evolution algorithm. Section IV shows the simulation results. Finally, the full text is concluded in Section V.

II. SYSTEM MODEL

A simple model of UAV-assisted MEC system is shown in Fig. 1. The system consists of an UAV equipped with a MEC server, K users and a base station (BS). Using $\mathcal{K} = \{1, 2, 3, ..., K\}$ denotes the set of UDs, all of which are at arbitrary locations within the region Ω. The location information of the k-th UD is denoted by $U_k = \{X_k, Y_k\}$. BS is used to provide communication services to UEs, and it is assumed that all UEs are outside the BS communication range, so the UAV can be used as a relay to provide services to users. Assuming that each UD has tasks to be processed at each time slot, the set $\mathcal{L} = \{L_1, L_2, ...L_K\}$ denotes the input data size of all UDs. Due to the limited computational resources of UDs, tasks need to be offloaded to the MEC server carried by the UAV for processing. The position of the UAV at time slot t is denoted as $M(t) = \{X(t), Y(t), H\}$, where H denotes the flight altitude of the UAV. The UAV communicates with UDs within the communication range at each time slot, which can greatly reduce the energy consumption and computational delay of UDs. Considering the load pressure on the MEC server, we divide the offloading method of UDs into two schemes: total offloading and partial offloading, which can guarantee the computing requirements of UDs and avoid the overload of the MEC server.

A. Communication Model

Since the height of the UAV is high enough in the simulation of this paper, we can assume the communication channel between the UAV and UDs as LoS. And the energy consumption of transmitting the result back to UDs through downlink after data are processed by MEC server is negligible, so only the uplink is considered in this paper. Assuming that the users' location is known, therefore, based on the location information

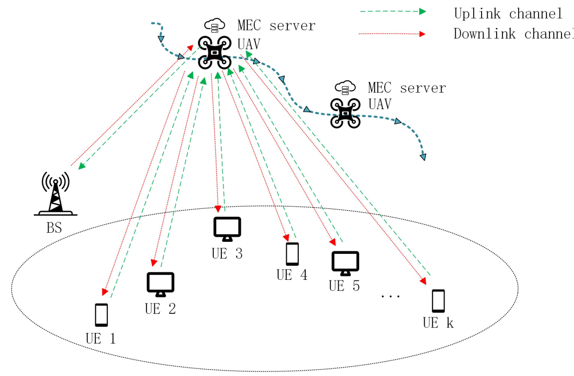


Fig. 1. The system model of UAV-assited MEC.

of the UAV, we can calculate the channel gain between the UAV and UDs for each time slot, that is

$$h_{up}(k,t) = \beta_0 d^{-2}(k,t) = \frac{\beta_0}{H^2 + \|M(t) - U_k\|^2} \tag{1}$$

where β_0 denotes the channel gain at a distance of 1 m. $d(k,t)$ denotes the Euclidean distance between the UD-k at the t-th time slot and the UAV in the 3D coordinate system. Assuming the UAV flies within Ω and the communication range between UAV and UDs is defined as

$$\|M(t) - U_k\| \le d_{max} \tag{2}$$

where d_{max} is the maximum communication distance. We set the communication strategy as

$$c(k,t) = \begin{cases} 1, & \text{offload} \\ 0, & \text{local} \end{cases} \tag{3}$$

After obtaining the channel gain, we can therefore calculate the data transmission rate as

$$r_{up}(k,t) = B_0 log_2(1 + \frac{P h_{up}(k,t)}{\delta_0^2}) \tag{4}$$

where B_0 represents the transmission bandwidth, P denotes the signal transmit power of UDs, and δ_0^2 is the power of additive white noise.

We classify the offloading policy of UDs into partial offloading and total offloading, expressed in the form of 0 and 1. The offloading policy is expressed as

$$a(k,t) = \begin{cases} 1, & \text{total offloading} \\ 0, & \text{partial offloading} \end{cases} \tag{5}$$

We set the offload ratio for partial offloading to η. Therefore, the amount of data left for local processing when the policy is partial offloading is $(1-\eta)L_k$, and the ηL_k data amount is offloaded to the MEC server for processing.

B. Computation Model

Depending on the offloading method, we discuss the computation model of the system separately:

- **Total offloading:** UDs with an offloading policy of total offloading offload all data to the UAV and the delay required for offloading is given by

$$T_{to,up}(k,t) = \frac{L_k}{r_{up}(k,t)} \tag{6}$$

Thus, the data offloading energy consumption is expressed as

$$E_{to,up}(k,t) = PT_{to,up}(k,t) \tag{7}$$

After the data is offloaded to the UAV, the elapsed time for the data to be processed in the server is defined as

$$T_{to,m}(k,t) = \frac{L_k C_m}{f_m} \tag{8}$$

where C_m is the number of CPU cycles required by the MEC server to process 1 bit of data and f_m denotes the computational resources of the MEC server. Therefore, the computational energy consumption of UAV when the policy is total offloading is given by

$$E_{to,m}(k,t) = K_m(f_m)^3 T_{to,m}(k,t) \tag{9}$$

where K_m denotes the effective capacitance factor of the server CPU. The total latency and energy consumption required for data processing in UD-k are given by

$$T_{to}(k,t) = T_{to,up}(k,t) + T_{to,m}(k,t) \tag{10}$$

$$E_{to}(k,t) = E_{to,up}(k,t) + E_{to,m}(k,t) \tag{11}$$

• **Partial offloading:** when the offloading policy is partial offloading, $(1-\eta)L_k$ tasks are processed locally. The delay and energy consumed by local processing are defined as

$$T_{pa,ud}(k,t) = \frac{(1-\eta)L_k C_{ud}}{f_{ud}} \tag{12}$$

$$E_{pa,ud}(k,t) = K_{ud}(f_{ud})^3 T_{pa,ud}(k,t) \tag{13}$$

where C_{ud} is the number of CPU cycles required by UDs to process 1 bit of data, f_{ud} and K_{ud} are the computational resources, and the effective capacitance factor of CPU of UDs, respectively. The other part of the data is offloaded to the UAV and then computed, and as with the total offloading, the time and energy consumption of the task offload can be expressed as

$$T_{pa,up}(k,t) = \frac{\eta L_k}{r_{up}(k,t)} \tag{14}$$

$$E_{pa,up}(k,t) = PT_{pa,up}(k,t) \tag{15}$$

The latency and energy consumption of the server to process the task are given by

$$T_{pa,m}(k,t) = \frac{\eta L_k C_m}{f_m} \tag{16}$$

$$E_{pa,m}(k,t) = K_m(f_m)^3 T_{pa,m}(k,t) \tag{17}$$

Therefore, the total energy consumption and delay at time slot t for the UD-k with the offloading strategy being partial offloading are given by

$$T_{pa}(k,t) = max\left\{T_{pa,ud}(k,t), T_{pa,up}(k,t) + T_{pa,m}(k,t)\right\} \tag{18}$$

$$E_{pa}(k,t) = E_{pa,ud}(k,t) + E_{pa,up}(k,t) + E_{pa,m}(k,t) \tag{19}$$

In order to minimize the delay of partial offloading, we can adjust the offloading ratio to obtain $min(T_{pa}(k,t))$. $T_{pa,ud}$ and $T_{pa,up} + T_{pa,m}$ are inversely proportional to each other since the amount of tasks for each UD is constant. According to the specific of mathematical increase and decrease function, the minimum value can be obtained when $T_{pa,ue} = T_{pa,up} + T_{pa,m}$. Therefore, the offload ratio can be obtained by the following relationship

$$\begin{aligned} \frac{(1-\eta)L_k C_{ud}}{f_{ud}} &= \frac{\eta L_k}{r_{up}} + \frac{\eta L_k C_m}{f_m} \\ \Rightarrow \eta &= \frac{C_{ud} f_m r_{up}}{(C_m f_{ud} + C_{ud} f_m) r_{up} + f_{ud} f_m} \end{aligned} \tag{20}$$

• **Local computation:** the UDs located out of the UAV communication range can only process the tasks locally. Similarly, the latency and energy consumption of the computation for UDs computed locally are given by

$$T_{lo}(k,t) = \frac{L_k C_{ud}}{f_{ud}} \tag{21}$$

$$E_{lo}(k,t) = K_{ud}(f_{ud})^3 T_{lo} \tag{22}$$

C. Problem Formulation

We express the delay and energy consumption of UD-k at time slot t as

$$\begin{aligned} T(k,t) = c(k,t)\left[a(k,t)T_{to}(k,t) + (1-a(k,t))T_{pa}(k,t)\right] \\ +(1-c(k,t))T_{lo}(k,t) \end{aligned} \tag{23}$$

$$\begin{aligned} E(k,t) = c(k,t)(a(k,t)E_{to}(k,t) + (1-a(k,t))E_{pa}(k,t)) \\ +(1-c(k,t))E_{lo}(k,t) \end{aligned} \tag{24}$$

In this subsection, we discuss the question of minimizing the weighted sum of energy consumption and latency in our UAV-assisted MEC system. Since the UAV advances at a uniform speed in this system and communicates with UDs after advancing the same distance after each acquired flight maneuver. Therefore, we only need to consider the energy consumption and time delay for communication and computation between the UAV and UDs. It is assumed that the UAV acquires one action at the end of each time slot and that all UDs regenerate their tasks. We denote the design problem as

$$\text{P1: } \min_{A,M} \sum_{t=1}^{T}\left[\omega\sum_{k=1}^{K}E(k,t) + (1-\omega)\sum_{k=1}^{K}T(k,t)\right] \tag{25a}$$

$$\text{s.t.} \quad M(t), U_k \in \varOmega \quad \forall k \in \mathcal{K} \tag{25b}$$

$$c(k,t) \in \{0,1\} \tag{25c}$$

$$a(k,t) \in \{0,1\} \tag{25d}$$

$$M^{start} \rightarrow M^{end} \tag{25e}$$

where ω denotes the weight parameter to indicate the importance of energy consumption and delay. The constraint (25b) indicates that both UDs and the UAV must be within the range of $\varOmega$. The constraint (25c) indicates that only the UDs within the communication range of the UAV can communicate. The constraint (25e) indicates that UAVs have fixed starting and ending points.

III. THE PROPOSED JOINT OPTIMIZATION ALGORITHM

Since our proposed optimization problem is non-convex, we cannot obtain the optimal solution directly. This section presents a joint optimization algorithm for solving the tasks offloading problem of UDs and the trajectory optimization problem of the UAV, respectively.

A. Tasks offloading policy optimization

The DE algorithm is a multi-objective optimization algorithm that can be used to solve the overall optimal solution in a multi-dimensional space. We use the DE to solve the optimal offloading strategy for UDs. The process of the DE includes population initialization, variation, crossover and selection. The details of the algorithm are shown in Algorithm 1.

• **Population initialization:** N individuals are randomly and uniformly generated in the solution space, and each individual consists of a K-dimensional vector, i.e., the number of UDs, and the initial population target vector is defined as

$$\begin{aligned} &\boldsymbol{X}_n(0) = \{\boldsymbol{x}_{n,1}(0), \boldsymbol{x}_{n,2}(0), ..., \boldsymbol{x}_{n,K}(0)\} \\ &n \in \{1, 2, ..., N\}, k \in \mathcal{K} \end{aligned} \tag{26}$$

• **Variation:** There are various variation strategies, such as DE/rand/1, DE/best/1, DE/rand/2, etc. In this paper, we adopt the strategy of DE/rand/1. Firstly, three different vectors $\boldsymbol{x}_{r1}^g$, $\boldsymbol{x}_{r2}^g$, $\boldsymbol{x}_{r3}^g$ are selected in each generation of the population, where g denotes the number of iterative generation. Thus the variance vector can be expressed as

$$\boldsymbol{v}_n^g = \boldsymbol{x}_{r1}^g + F \cdot (\boldsymbol{x}_{r2}^g - \boldsymbol{x}_{r3}^g) \quad (r1 \neq r2 \neq r3) \tag{27}$$

where F is the scaling factor. According to the number of iterations, F can be obtained by

$$F = F_0 \cdot 2^{e^{(1-\frac{G}{G+1-g})}} \tag{28}$$

where F_0 is the initial scaling factor and G is the largest number of iterations.

• **Crossover:** The experiment vector generated by the variational vector using the binomial crossover operation is given by

$$\boldsymbol{u}_{n,k}^g = \begin{cases} \boldsymbol{v}_{n,k}^g, & rand(0,1) \leq CR \ or \ k = k_{rand} \\ \boldsymbol{x}_{n,k}^g, & otherwise \end{cases} \tag{29}$$

where CR is the crossover rate and $CR \in (0,1)$, and k_{rand} is a randomly chosen integer in the range $[1 : N]$.

• **Selection:** The target vector or experiment vector of the current generation is selected as the next generation target vector which is given by

$$\boldsymbol{x}_n^{g+1} = \begin{cases} \boldsymbol{u}_n^g, & f(\boldsymbol{u}_n^g) < f(\boldsymbol{x}_n^g) \\ \boldsymbol{x}_n^g, & otherwise \end{cases} \tag{30}$$

where the $f(\cdot)$ in (26) is our proposed objective function, and we adopt the greedy algorithm to select the optimal vector as the next generation target vector.

Algorithm 1: Tasks Offloading Policy Optimization

```
Initialize F_0, x_n^0, n ∈ {1, 2, ..., N};
for episodes g = 0 : G do
    for n = 1 : N do
        Obtain the variance vector v_n^g based on Eq.(27);
        Obtain the experiment vector u_n^g based on Eq.(29);
        Obtain the next generation target vector x_n^{g+1} based on Eq.(30);
    end
    Update the scaling factor F based on Eq.(28);
end
```

B. UAV trajectory optimization

We use the OAC reinforcement learning algorithm to obtain continuous actions of the UAV as a way to optimize the UAV flight trajectory. The OAC algorithm is explored by maximizing the approximate confidence bound of the state-action value function. The algorithm enables efficient training in continuous control tasks. We use the location of the UDs, the task volume size and the location of the UAV as status information which is given by

$$State = \{U_k, L_k, M(t)\}, k \in \mathcal{K} \tag{31}$$

We use the deflection angle of the UAV as the action of the algorithm, that is

$$Action \in [0, \pi/2] \tag{32}$$

We obtain the action by sampling from the exploration strategy π_E, which takes the form $\pi_E = N(\mu_E, \Sigma_E)$ and is denoted as

$$\mu_E = \mu_T + \frac{\sqrt{2\delta}}{\left\| \left[\bigtriangledown_a \hat{Q}_{UB}(s,a) \right]_{a=\mu_T} \right\|_{\Sigma}} \Sigma_T \left[\bigtriangledown_a \hat{Q}_{UB}(s,a) \right]_{a=\mu_T} \tag{33}$$

$$\Sigma_E = \Sigma_T \tag{34}$$

To obtain the solution of P2, we use the opposite of the objective function as the reward of the OAC, so the reward is expressed as

$$reward = -\sum_{t=1}^{T} \left[\omega \sum_{k=1}^{K} E(k,t) + (1-\omega) \sum_{k=1}^{K} T(k,t) \right] \tag{35}$$

C. Joint optimization algorithm

The specific details of our proposed joint OAC_DE optimization algorithm are shown in Algorithm 2.

Algorithm 2: OAC_DE algrithm

Initial parameters ω_1, ω_2 of the critic and θ of the target policy π_T;
Initialize target network weights and replay pool: $\breve{\omega}_1 \leftarrow \omega_1$,$\breve{\omega}_2 \leftarrow \omega_2$,$D \leftarrow \phi$;
for $episode = 1 : ep_{max}$ **do**
  Randomly initialize the state s_1;
  for $stept = 1 : T$ **do**
    Sample action from exploration policy as in (34)(35):$a_t \sim \pi_E(a_t \mid s_t)$;
    Inputting a_t and s_t into the environment;
    Obtain the current offloading policy according to Algorithm 1;
    Obtain reward r_t and new state s_{t+1};
    Store $(s_t, a_t, r_t, s_{t+1}, done)$ in rpm;
    Sample a batch size of $(s_t, a_t, r_t, s_{t+1}, done)$ from rpm;
    Update two bootstraps of the critic:
    for $i \in 1, 2$ **do**
      Update ω_i with $\hat{\bigtriangledown}_{\omega_i} \|\hat{Q}^i_{LB}(s_t, a_t) - R(s_t, a_t) - \gamma \min(\breve{Q}^1_{LB}(s_{t+1}, a), \breve{Q}^2_{LB}(s_{t+1}, a))\|^2_2$;
    end
    Update policy gradient θ with $\bigtriangledown_\theta \hat{J}^\alpha_{\hat{Q}'_{LB}}$;
    Update target networks:
    $\breve{\omega}_1 \leftarrow \tau\omega_1 + (1 - \tau)\breve{\omega}_1, \breve{\omega}_2 \leftarrow \tau\omega_2 + (1 - \tau)\breve{\omega}_2$;
  end
  $reward = \sum_{t=1}^{T} r_t$
end

IV. Simulation Results

In this section, we show the simulation results of our proposed algorithm. To better visualize the advantages of our proposed algorithm, we compare it with other joint algorithms. In addition, the simulation parameters of the model are shown in Table I.

This paper compares the proposed algorithm with three other methods as follows.

• **DDPG_PDPSO**: It optimizes the trajectory of the UAV by DDPG reinforcement learning algorithm and the offloading strategy of UDs by PDPSO algorithm [7].

• **DQN_DE**: It optimizes the trajectory of the UAV by DQN reinforcement learning algorithm and the offloading strategy of UDs by DE algorithm.

• **SP_DE**: The UAV flies in a fixed direction, we do not optimize the trajectory of the UAV, but only the offloading strategy of the UDs.

In the parameters of OAC algorithm, the learning rate is 0.0001, the decay factor γ is set to 0.99, and the soft update factor is 0.001. In the parameters of DE algorithm, the initial scaling factor F_0 is set to 2 and the crossover rate CR is set to 0.3. We evaluate the performance of the algorithm by using python. The simulation diagram of the optimized trajectory of the UAV with different UDs distribution is shown in Fig. 2.

TABLE I
Simulation Parameters

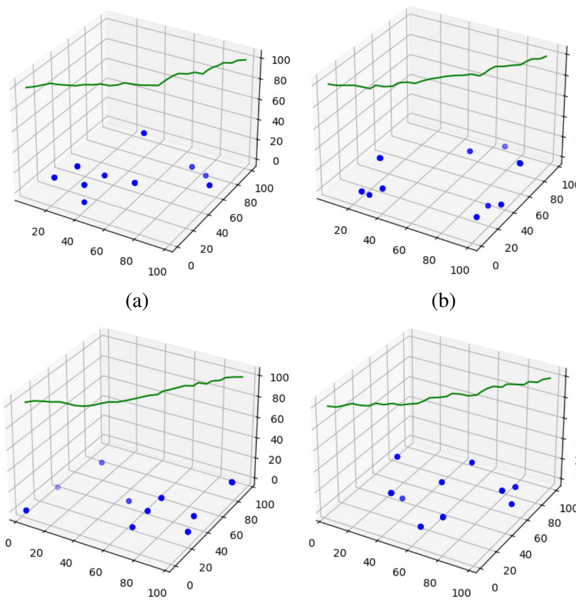
Parameters	Value
The number of UDs $\mathcal{K}$	10
The amount of data generated by each UD $\mathcal{L}$	[1,10] Mbits
The height of UAV H	100 m
Range of UDs and UAV movement Ω	[100 m,100 m]
Channel gain per unit distance β_0	-50 dB
Channel bandwidth for each UD B_0	10 MHz
The transmission power of the UDs P	0.5 W
The noise power at the UAV δ_0^2	-70 dBm/Hz
The CPU capacitance coefficient of UDs K_{ud}	10^{-27}
The CPU capacitance coefficient of UAV K_m	10^{-28}
Required CPU cycles per bit computation at local C_{ud}	800 cycles/bit
Required CPU cycles per bit computation at UAV C_m	1000 cycles/bit
The UDs computing resources f_{ud}	1 GHz
The UAV computing resources f_m	3 GHz
The weight of energy consumption and delay ω	0.75

The maximum communication range of the UAV at this time is set to 100 m, and the number of UDs is 10. From the four subplots, it can be seen that when the UAV is in the area with a high number of UDs, the flight trajectory of the UAV is more curved and biased toward the dense area, and vice versa, it is smoother. This is because the UAV needs to fly longer in the area with dense UDs to achieve the purpose of reducing the offloading energy consumption and time delay.

Comparison of the training results of our proposed joint optimization algorithm with other joint optimization algorithms is shown in Fig. 3 and Fig 4. Fig. 3(a) and (b) show that the maximum communication range of the UAV is 100 m and 50 m, respectively, and the number of UDs is 10. Firstly, it can be seen from the final convergence results that our proposed optimization algorithm has the highest reward value, which means that the weighted sum of energy consumption and time delay of the system is the smallest. Secondly, in terms of convergence speed, our proposed algorithm converges faster than the DDPG_PDPSO algorithm, and the convergence results are smoother. Compared with the DQN_DE algorithm, DQN is difficult to obtain discrete actions and thus optimize trajectory through training, so the convergence results are significantly lower than other algorithms. In the comparison with SP_DE, the optimized trajectory significantly reduce the energy consumption and delay of the system. Fig. 4 shows the effect of the number of UDs on the weighted sum of energy consumption and delay of the system. It can be seen that the increased number of UDs makes the system more burdened, but our proposed algorithm always outperforms the others. The performance of DQN_DE is always the worst, which indicates that continuous actions are more suitable for designing the trajectory of the UAV. The DDPG_PDPSO is suitable for the case of a small number of UDs and inferior to our proposed algorithm in terms of environmental adaptation.

V. Conclusion

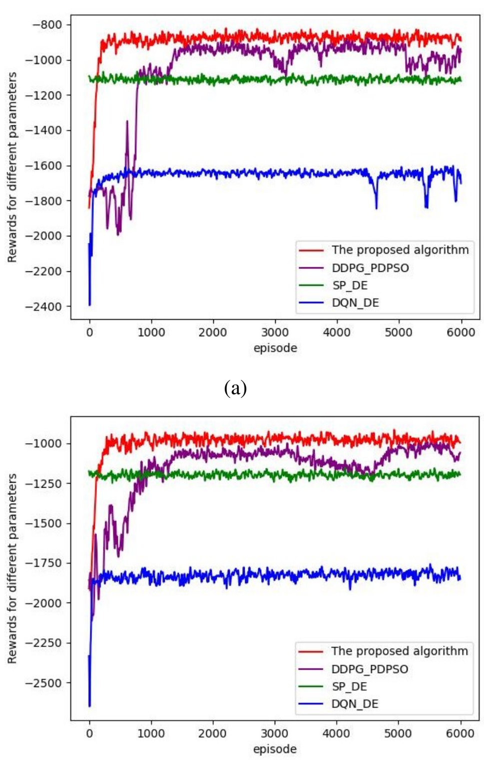
In this paper, we presented a problem of minimizing the weighted sum of energy consumption and delay of the system based on a UAV-assisted MEC system by jointly optimizing



(a) (b)

(c) (d)

Fig. 2. Optimized trajectory of UAV.



(a)

(b)

Fig. 3. Comparison of training results of different joint optimization algorithms.

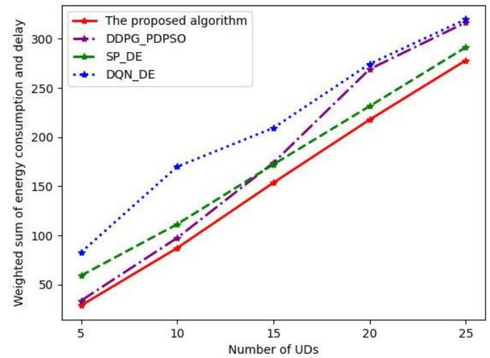


Fig. 4. Impact of the number of UDs on the weighted sum of energy consumption and delay.

the trajectory of the UAV and the offloading strategy of the UDs. Since this problem is nonconvex, we proposed an OAC_DE joint optimization algorithm. We first solved the optimal unloading strategy for the current time slot by the DE algorithm, and then used the OAC reinforcement learning algorithm to obtain the next action and optimize the flight trajectory of the UAV. The final results show that our proposed joint optimization algorithm achieves better performance in terms of reducing energy consumption and delay as well as the convergence of the algorithm.

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, JSGG20210420091805014, JSGG20210802154203011.

REFERENCES

[1] Z. Liu, Y. Cao, P. Gao, X. Hua, D. Zhang and T. Jiang, "Multi-UAV Network Assisted Intelligent Edge Computing: Challenges and Opportunities," *China Communications*, vol. 19, no. 3, pp. 258-278, Mar. 2022.

[2] D. Wang, J. Tian, H. Zhang and D. Wu, "Task Offloading and Trajectory Scheduling for UAV-Enabled MEC Networks: An Optimal Transport Theory Perspective," *IEEE Wireless Communications Letters*, vol. 11, no. 1, pp. 150-154, Jan. 2022.

[3] B. Liu, Y. Wan, F. Zhou, Q. Wu and R. Q. Hu, "Resource Allocation and Trajectory Design for MISO UAV-Assisted MEC Networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4933-4948, May 2022.

[4] Z. Hu, F. Zeng, Z. Xiao, B. Fu, H. Jiang and H. Chen, "Computation Efficiency Maximization and QoE-Provisioning in UAV-Enabled MEC Communication Systems," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1630-1645, Apr. 2021.

[5] J. Chen, H. Xing, Z. Xiao, L. Xu and T. Tao, "A DRL Agent for Jointly Optimizing Computation Offloading and Resource Allocation in MEC," *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17508-17524, Dec. 2021.

[6] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton and J. Henry, "UAV Trajectory Planning in Wireless Sensor Networks for Energy Consumption Minimization by Deep Reinforcement Learning," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9540-9554, Sept. 2021.

[7] Y. Gan and Y. He, "Trajectory Optimization and Computing Offloading Strategy in UAV-Assisted MEC System," in *Proc. Computing, Communications and IoT Applications (ComComAp)*, 2021, pp. 132-137.