Hindawi
Complexity
Volume 2022, Article ID 4269102, 16 pages
https://doi.org/10.1155/2022/4269102

WILEY | Hindawi

Review Article

Energy-Efficient Task Migration and Path Planning in UAV-Enabled Mobile Edge Computing System

Can Gong,[1] Li Wei ,[2,3] Deliang Gong,[2,3] Tiantian Li,[2] and Fang Feng[2]

[1]*School of Innovation and Entrepreneurship, Xiangnan University, Chenzhou 423001, China*
[2]*School of Computer and Artificial Intelligence, Xiangnan University, Chenzhou 423001, China*
[3]*Hunan Engineering Research Center of Advanced Embedded Computing and Intelligent Medical Systems, Xiangnan University, Chenzhou 423001, China*

Correspondence should be addressed to Li Wei; weili_xnu@163.com

Received 15 June 2021; Revised 4 February 2022; Accepted 22 February 2022; Published 11 April 2022

Academic Editor: Qingling Wang

With the rapid development of unmanned aerial vehicles (UAVs) technology and the advent of the 5G era, the role of UAV-enabled mobile edge computing (MEC) system has attracted much attention, especially in the event of some emergencies. However, considering the limited battery life and computing capabilities of UAVs, it is challenging to provide energy-efficient services for mobile devices. To solve this challenge, we propose an energy-efficient dynamic task migration algorithm (EDTM) that minimizes the total energy consumption of the system while ensuring UAVs system load balance. Based on the improved ant colony algorithm and path elimination strategy, the proposed algorithm comprehensively considers task migration distance between UAVs, the load situation of UAVs, and environmental factors (e.g., wind speed and air density) and finally plans a reasonable task migration path. The simulation results show that the performance of the proposed EDTM is superior to the benchmark schemes.

1. Introduction

In recent years, 5G communication technology and the Internet of Things (IoTs) have developed rapidly. Mobile edge computing (MEC) attracts much attention, which can leverage the proximate computing resources to provide latency-critical and computation-intensive tasks for mobile users [1, 2]. The computation tasks generated by mobile devices (MDs) can be offloaded to the nearby edge server, such as base stations. However, these edge servers are static. It is difficult to provide powerful and stable computing services for MDs when encountering emergency calls, disaster response, and rural environments. In these cases, UAVs can act as mobile edge servers to provide more flexible and dynamic services [3]. Compared with cellular-based MEC, UAVs have the advantages of fast, high mobility, and controllable mobile management [4].

Despite the advantages of UAV-enabled MEC, there are several challenges. First, the energy capacity of UAVs is constrained. To guarantee the quality of service (QoS) of MDs, it is crucial to consider the energy consumption of task migration. Second, the computation capabilities of UAVs is limited, a single UAV cannot support high computation loads, it is necessary to perform task migration with other UAVs, while ensure load balance among UAVs system. Third, an energy-efficient path planning strategy is also a challenge in UAV-enabled MEC. With an optimal task migration path, UAVs can pay less migration energy cost under load balancing. In the literature, there have been some studies focused on task migration [5–9] in MEC and optimization UAV-assisted network [10–14]. However, on the one hand, these research works mainly focus on energy consumption optimization, while neglecting considering load balance in MEC system. Therefore, the works related to dynamic task migration in MEC system cannot be efficiently

applied in the UAV-enabled MEC systems. On the other hand, the existing works targeted on UAV-assisted network did not comprehensively considers task migration distance between UAVs, the load situation of UAVs, and environmental factors for the path planning of task migration with load balance within the UAVs systems, which is significant for task migration in UAV-assisted network.

In this paper, we consider a joint optimization problem that simultaneously considers the above challenges have not been sufficiently investigated and considering actual environmental factors, such as wind speed and air density. To address these challenges, we propose an energy-efficient dynamic task migration algorithm (EDTM) in a UAV-enabled MEC system, which minimizes the total energy consumption of the system under load balance. The main contributions of the paper are summarized as follows:

(i) We formulate a system model in the UAV-enabled MEC system, which consists of communication model, task computation model, and energy consumption model. We fully consider the energy costs of the MDs and UAVs, including local computation energy consumption on MDs, edge computing energy consumption on UAVs, task migration energy consumption.

(ii) We improve the ant colony algorithm and propose an EDTM algorithm to migrate computation tasks. Considering the load balance of UAVs, we design a path planning elimination strategy, which comprehensively considers task migration distance between UAVs, the load situation of UAVs, and environmental factors.

(iii) We conduct extensive experiments and the simulation results demonstrate that the proposed EDTM algorithm can achieve superior performance than the benchmark schemes, which can effectively reduce the total energy consumption of the system while ensuring the load balance of the UAVs system.

The remainder of the paper is organized as follows. Section 2 introduces related works. In Section 3, the system model and problem are described. Section 4 presents the EDTM algorithm to solve the joint optimization problem. In Section 5, we evaluate the performance of the proposed EDTM with extensive simulation results. Finally, we conclude the paper in Section 6.

2. Related Works

We review the related works by two categories, i.e., dynamic task migration and UAV-assisted networks in MEC.

2.1. Dynamic Task Migration. There are some research efforts on dynamic task migration [5–9]. Wang et al. [5] described the task migration problem as a Markov decision process and designed an optimal service migration strategy in MEC. Kuang et al. [6] studied the partial offloading scheduling and power distribution problem for single-user MEC system. They proposed an iterative algorithm to reduce energy consumption and achieve optimal delay performance. Moreover, the authors of [7] proposed a task migration energy optimization strategy by considering resource caching. Meanwhile, Anajemba et al. [8] studied cooperative offloading scheme in energy-efficient multi-access edge computing and proposed a Lagrangian suboptimal convergent computation offloading algorithm. Zhou et al. [9] studied dynamic task offloading in MIMO MEC system with energy harvesting. The authors proposed an algorithm to minimize the time average of a weighted sum of energy consumption and execution delay, meanwhile stabilizing the battery energy queue. However, the above research works focus on energy consumption optimization, while neglecting considering load balance in the MEC system. Therefore, they cannot be efficiently applied in the UAV-enabled MEC systems.

2.2. UAV-Assisted Networks. UAVs have been deployed to enhance the network capacity and provide services to mobile users with or without infrastructure coverage which attracts much attention in the literature [10–14]. Particularly, the paper [10] studied energy-efficient resource management in UAV-assisted MEC. The goal is to jointly minimize the energy consumption at the IoT devices and the UAVs during task execution. Liu et al. [11] proposed a machine learning framework for trajectory design and power control for multi-UAV assisted networks. In addition, Li et al. [12] studied UAV-assisted MEC with the objective to optimize computation offloading with minimum UAV energy consumption. They adopted the Dinkelbach algorithm and the successive convex approximation technique to solve it. Wu et al. [13] considered a multi-UAV enabled wireless communication system, where multiple UAV-mounted aerial base stations are employed to serve a group of users on the ground. Existing UAV communication and trajectory schemes are inefficient as they assume limited drone mobility and static transmission power. Garg [14] considered a multi-UAV system where UAV-mounted mobile base stations serve users on the ground and proposed an iterative approach using block gradient descent. To summarize, although some research problems related to UAV-assisted networks have been investigated. However, they did not comprehensively consider task migration distance between UAVs, the load situation of UAVs, and environmental factors for the path planning of task migration with load balance within the UAVs systems, which is significant for task migration in UAV-assisted network and motivates our research in this paper.

3. System Model

3.1. System Architecture. We consider a MEC system with multiuser and multiedge servers with dynamic task migration. The system consists of M MDs and N UAVs that integrate MEC servers and communication circuits. We define $M = \{1, 2, \ldots, M\}, m \in M$, and $U = \{1, 2, \ldots, t, \ldots, U\}$, $\forall u \in U$ to represent the set of MDs and UAVs, respectively. As for the total time T for task migration

completion in the system, it is divided into F time slots on average. We use $T\cdot = \{0, 1, \ldots, F\}, t \in T$ to represent the set of time slots, where the length of all time slots t is τ. Moreover, the length of the time slot is small enough to ensure that the position of the UAVs during each time slot is approximately unchanged. The definitions of the main symbols involved in this paper are shown in Table 1.

In this paper, due to the strong mobility of MDs, UAVs will have dynamic task migration in the process of providing services (task processing) to MDs. In order to facilitate the understanding of dynamic task migration, we describe task migration in the UAV-enabled MEC system architecture with multi-user and multiedge server, as shown in Figure 1. Among them, UAVs are distributed around MDs to provide computing services for MDs. We select MD m for the paper detailed description. Initially, MD m was within the service coverage of UAV u. When MD m moved from the initial position 1 to position 2, MD m had already left the service range of UAV u. During the mobile process of MD m, its service quality will be seriously affected and even service termination will occur. In order to ensure the quality of the service, we need to migrate the tasks that were initially processed on UAV u. In addition, task migration also needs to consider whether location 2 is the endpoint of the MD m move. If position 2 is the endpoint of MD m during the movement, MD m will stop moving after reaching position 2; otherwise, MD m will continue to move after reaching position 2. This greatly increases the difficulty of dynamic task migration. The above constitutes the most basic dynamic task migration process of a mobile device. In Figure 1, $d^t_{u,m}$, $d^t_{4,m}$, and $d^t_{U,m}$ represent the distance between UAV u, UAV 4, UAV U and MD m in time slot t, respectively. While $\widetilde{d}^t_{\mathrm{u,1}}$, $\widetilde{d}^t_{\mathrm{u,4}}$, and $\widetilde{d}^t_{\mathrm{u,U}}$ represent the distance between UAV u and UAV 1, UAV 4, and UAV U in time slot t, respectively.

To facilitate the description of the dynamic task migration path, we also define the positions of UAVs and MDs in all time slots. We define the initial position of MDs at the origin (0,0,0) of the spatial rectangular coordinate system, and the direction in which the MDs are connected from position 1 to position 2 is the x-axis of the spatial rectangular coordinate system. Therefore, a line perpendicular to the x-axis on the horizontal plane is the y-axis. The vertical plane can get the z-axis perpendicular to the x-axis and y-axis. Specifically for a certain UAV, we use x^t_u and y^t_u to represent the coordinates of the UAV u on the horizontal x-axis and y-axis in time slot t, that is, the horizontal position of the UAV u in time slot t. Based on the particularity of UAVs, we add a height variable z^t_u to represent the position of UAV u in the vertical direction in time slot t. Therefore, the position of UAV u in time slot t is finally defined as the triple $l^t_u = (x^t_u, y^t_u, z^t_u)$. Similarly, the position of a particular MD, such as the position of MD m in time slot t is defined as a binary groups $\widetilde{l}^t_m = (\widetilde{x}^t_m, \widetilde{y}^t_m)$, where $\widetilde{x}^t_m$ and $\widetilde{y}^t_m$ represent the coordinate positions of MD m in time slot t on the horizontal x-axis and y-axis, respectively.

Because UAVs are widely distributed around MDs, and each MD can access a groups of UAVs within the broadcast range, MDs can offload computing tasks to UAVs within the accessible range [15]. We use $N^t_u \in M$ to represent the set of MDs in the service range of UAV u in time slot t, and the UAVs set in its communication range are defined as $V^t_u \in U$. And because only the digital description of the distance between UAVs and MDs and UAVs can guide the task migration process. Next, we define the distance set d^t_u between UAV u and the MDs included in the set N^t_u in time slot t and the distance set $\widetilde{d}^t_u$ between UAV u and the UAVs included in the set V^t_u. Among them, $d^t_{\mathrm{u,m}} = \sqrt{(x^t_u - \widetilde{x}^t_m)^2 + (y^t_u - \widetilde{y}^t_m)^2 + (z^t_u)^2}$ and $\widetilde{d}^t_{\mathrm{u,1}} = \sqrt{(x^t_u - x^t_1)^2 + (y^t_u - y^t_1)^2 + (z^t_u - z^t_1)^2}$ represent the distances between UAV u and MD m and UAV u and UAV 1 in time slot t, respectively. And and satisfy and respectively. Furthermore, we can get the relative position of UAV u in time slot t, that is, $L^t_u = (d^t_u, \widetilde{d}^t_u)$.

3.2. Communication Model. In this section, we introduce the communication model and give the uplink transmission rate when MDs offload computation tasks to UAVs. We define H_m as the channel gain between MD m and UAVs. At the same time, because the research focus of this paper is to offload the computing tasks that have been offloaded by MDs from the original UAVs to UAVs with relatively low load and close to MDs, then we assume H_m less variable. Therefore, we add a constant error value ς for H_m, and satisfying the following equation:

$$H_m = H_0 + \varsigma, \tag{1}$$

where H_0 indicates the channel gain in the initial state.

In [16], the uplink transmission power of MDs is mainly determined by the channel gain and channel bandwidth. As UAVs are used in this paper to provide edge service functions, the uplink transmission power of MDs is also greatly affected by the dynamic change of the distance between MDs and UAVs. When the distance between MDs and UAVs increases, in order to ensure that the task offload is not affected, the uplink transmission power of MDs will increase accordingly. Otherwise, the uplink transmission power of MDs will decrease accordingly. Specifically, the uplink transmission power of MD m to offload tasks to UAV u in time slot t is defined as

$$\begin{aligned} p^t_{m,u} &= \left(2^{\left(1/B_{\mathrm{actually}}\cdot\tau\right)} - 1\right)\frac{\sigma^2 \cdot d^t_{u,m}}{H_m} \\ &= \left(2^{\left(1/B_{\mathrm{actually}}\cdot\tau\right)} - 1\right)\frac{\sigma^2 \cdot d^t_{u,m}}{H_0 + \varsigma}, \end{aligned} \tag{2}$$

where σ^2 and B_{actually} represent the noise power and the actual network bandwidth in the uplink transmission process, respectively. And $p^t_{m,u}$ satisfies the following equality, i.e.,

$$0 \le p^t_{m,u} \le p^{\max}_{UAV}, \quad m \in M, u \in U, t \in T, \tag{3}$$

where $p^{\max}_{MD}$ is the maximum transmission power of MDs.

Table 1: Definition of main symbols.

Symbols	Definition
M	The set of MDs
U	The set of UAVs
$T\cdot$	The set of time slots
l_u^t	The position of UAV u in time slot t
$\tilde{\mathbf{l}}_m^t$	The position of MD m in time slot t
N_u^t	The set of MDs in UAV u service range at time slot t
V_u^t	The set of MDs in UAV u communication range at time slot t
d_u^t	The set of distances between UAV u and MDs contained in set in time slot t
$\tilde{d}_u^t$	The set of distances between UAV u and UAVs contained in set in time slot t
H_m	The channel gain between MD m and UAVs
$p_{m,u}^t$	The uplink transmission power of MD m to offload tasks to UAV u in time slot t
f_m^t	The CPU cycle frequency of MD m in time slot t
$f_{u,m}^t$	The computing resources of UAV u allocate to MD m in time slot t
γ_m^t	The total number of tasks processed by MD m in time slot t
r_m^t	The total number of tasks that MD m offloaded to UAVs in time slot t
$Q_m^{MD}(t)$	The dynamic task backlog queue of UAV u in time slot t
$E_{\text{com},m}^t$	The task processing energy consumption of MD m in time slot t
$E_{\text{com},u}^t$	The energy consumption for processing offloaded tasks of UAV u in time slot t
$E_{tra,m}^t$	The transmission energy consumption of MD m offloads tasks to UAV u in time slot t
$E_{tra,u}^t$	The task migration energy of UAV u migrates tasks to UAV 1 in time slot t
$E_{\text{fly},u}^t$	The flight energy consumption of UAV u in time slot t
E_{UAV}^t	The total energy consumption of all UAVs in time slot t
E_{total}^t	The total energy consumption of the entire system in time slot t

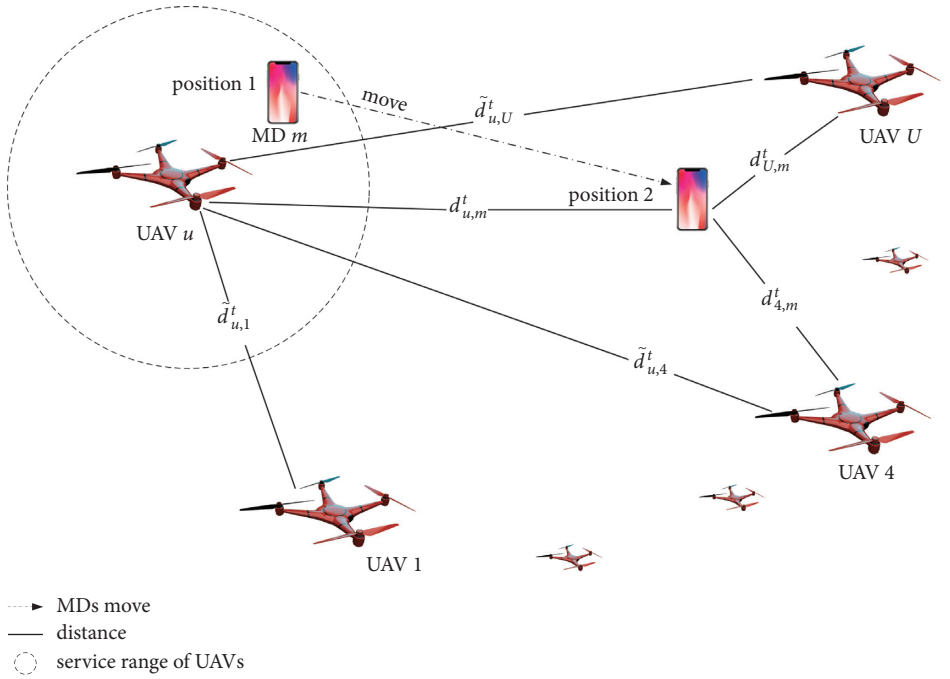


Figure 1: The UAV-enabled MEC system architecture.

At the same time, because this paper uses UAVs to perform edge computing functions, the information transmission between UAVs and MDs and between UAVs is wireless communication [17]. In the process of wireless communication, the theoretical bandwidth and actual bandwidth will be different, because network loss and line attenuation will occur in real scenarios. Table 2 is a set of actual measured data, which shows the relationship between the channel bandwidth and the theoretical download speed as well as the actual download speed.

According to the data in Table 2, we can get the relationship between the actual bandwidth and the theoretical bandwidth of the network, that is,

$$B_{\text{actually}} = 88\% \cdot B_{\text{theory}}. \tag{4}$$

Table 2: Relationship between network bandwidth, theoretical download speed, and actual download speed.

Network bandwidth (M)	Theoretical download speed	Actual download speed
2	256 kB/s	225 kB/s
4	512 kB/s	450 kB/s
6	768 kB/s	675 kB/s
8	1.024 MB/s	901 kB/s
10	1.28 kB/s	1.126 kB/s

In this paper, we do not consider downlink transmission delay and packager loss, because the data size of the result after the task is processed is usually smaller than the data size before the task processed [18], and the downlink transmission rate from UAVs to MDs will be higher than the uplink transmission rate from MDs to UAVs.

3.3. Task Computation Model. In this section, we introduce the computational model of the system. The main research direction of this paper is the task migration strategy in mobile edge computing systems.

In this paper, we define f_m^t as the CPU cycle frequency of MD m in time slot t, and $f_{u,m}^t$ is defined as the computing resource that UAV u allocates to MD m in time slot t for processing the computing tasks offloaded by MD m. And f_m^t and $f_{u,m}^t$ satisfy the following equality, respectively, that is,

$$\begin{aligned} &0 < f_m^t \le f_{MD}^{\max}, \quad m \in M,\ t \in T, \\ &0 < \sum_{m \in M} f_{u,m}^t \le f_{UAV}^{\max}, \quad u \in U,\ t \in T, \end{aligned} \tag{5}$$

where $f_{MD}^{\max}$ and $f_{UAV}^{\max}$ represent the maximum cycle frequencies of MDs and UAVs, respectively.

Whether it is an UAV or a MD, the tasks in the task queue are processed according to the "first come, first served" rule. In particular, the unit of the task amount is unified as bits, and the CPU cycle required to process one bit of any type of computing task is c. We assume that task arrival MDs are a random model and that task arrival is independent. The total number of tasks processed by MD m in time slot t is γ_m^t, that is,

$$\gamma_m^t = \frac{f_m^t \cdot \tau}{c}. \tag{6}$$

While the total number of tasks that MD m offloads to UAVs is r_m^t.

When MDs offload some tasks to UAVs, UAVs will process these tasks. Although the computing power of UAVs is much stronger than that of MDs, the computing power of UAVs is extremely limited compared to cloud servers. UAV u handles the number of tasks from MD m offload at time slot t is $b_{u,m}^t$. Therefore, we can get the dynamic task backlog queue of UAV u, i.e.,

$$Q_u^{UAV}(t+1) = \left(Q_u^{UAV}(t) - \sum_{m \in M} b_{u,m}^t \right)^+ + \sum_{m \in M} r_m^t, \tag{7}$$

where $b_{u,m}^t = (f_{u,m}^t \cdot \tau / c)$. And $Q_u^{UAV}(t)$ satisfies equality (8).

$$\sum_{m \in M} b_{u,m}^t \le Q_u^{UAV}(t), \quad u \in U,\ t \in T, \tag{8}$$

In a real application scenario, some MDs will offload tasks to UAVs, while some MDs may not [19]. So, $\sum_{m \in M} r_m^t \ge 0$. In addition, because our optimization goal is to ensure that each UAV achieves load balancing in each time slot as much as possible, that is, the number of tasks that each UAV needs to process in each time slot is approximately equal. Therefore, we define the limiting conditions as shown in equality (9), so that the number of tasks of each UAV in each time slot cannot exceed 10\% of the average number of tasks in the queue of all UAVs. Furthermore, MDs can make full use of the computing resources of UAVs without causing a waste of resources.

$$Q_u^{UAV}(t) \le \frac{1}{U} \sum_{u=1}^{U} Q_u^{UAV}(t) \cdot (1 + 10\%), \quad u \in U,\ t \in T. \tag{9}$$

Next, we introduce the system energy consumption model in detail.

4. Energy Consumption Model

First, the task processing energy consumption includes the energy consumption of MDs and UAVs when processing tasks. According to the capacitance theory, whether it is a MD or an UAV, the processing energy consumption is mainly determined by the CPU performance of the electronic device, that is, chip architecture [20]. In the paper [21], we can know that the CPU cycle frequency varies with the CPU voltage and is linearly related. Therefore, the task processing energy consumption of MD m and the energy consumption of UAV u processing offloaded tasks are respectively expressed as (9) and (10), that is,

$$E_{\text{com},m}^t = \kappa \cdot \left(f_m^t \right)^3 \cdot \tau, \tag{10}$$

$$E_{\text{com},u}^t = \sum_{m \in M} \kappa \cdot \left(f_{u,m}^t \right)^3 \cdot \tau. \tag{11}$$

Among them, κ represents the effective capacitance switch of the CPU, and the value of κ is determined by the performance of the CPU.

Secondly, the communication energy consumption consists of the energy consumption of the MDs offload task and the energy consumption of task migration between UAVs. As mentioned earlier, if MD m chooses to offload computing tasks to UAV u through the uplink, then the transmission energy consumption of MD m offloads tasks to UAV u in time slot t can finally get and show as follows:

$$E_{tra,m}^t = p_{m,u}^t \cdot \tau. \tag{12}$$

In addition, the dynamic task migration from UAV u to UAV 1 is shown in Figure 2. In Figure 2, MD 3 is initially in the service range of UAV u, so MD 3 offloads tasks to UAV u. However, with the movement of MD 3, MD 3 eventually moves away from the service range of UAV u. In order to ensure the normal service of MD 3, UAV u needs to reselect

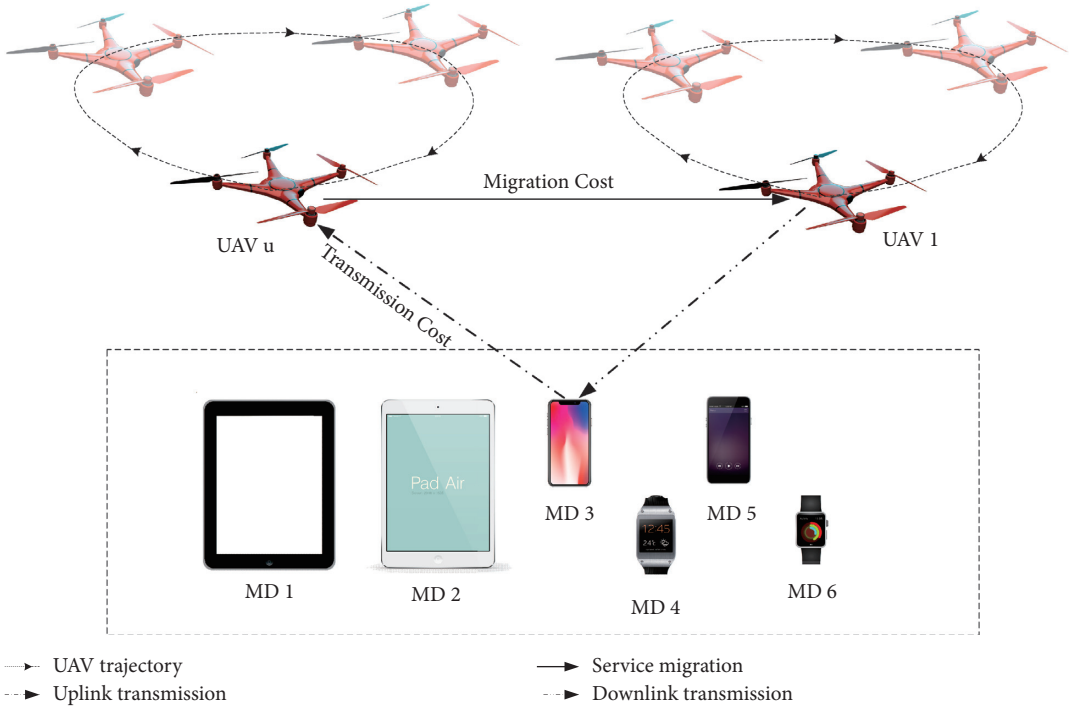


Figure 2: The dynamic process of task migration from UAV u to UAV 1.

a new UAV and migrate the tasks to the new UAV according to the load of each UAV and the distance from each UAV within its communication range V_u^t. As shown in Figure 2, the new UAV is UAV 1. At this time, we can get the task migration energy consumption of UAV u migrates the tasks to UAV 1, that is,

$$E_{tra,u}^t = p_{u,1}^t \cdot \tau, \tag{13}$$

where $p_{u,1}^t$ is the transmission power between UAV 1 and UAV u, and its specific computation method is similar to (2) in Section 3.2. And $p_{u,1}^t$ cannot exceed the maximum transmission power of UAV, that is,

$$0 \le p_{u,1}^t \le p_{UAV}^{\max}, \quad u \in U, t \in T. \tag{14}$$

Finally, in addition to the energy consumption of task computation and task migration for UAVs, the energy consumed by UAVs for a flight cannot be ignored. Most of the papers only consider the flying speed of UAVs as the independent variable when studying the flight energy consumption of UAVs. However, it shows that the flight energy consumption of UAVs depends not only on the speed of UAVs but also on the acceleration of UAVs [21–23]. Next, we define the velocity and acceleration of UAV u in time slot t, which are expressed as follows:

$$\begin{aligned} v_u^t &= \frac{\left\| \mathbf{l}_u^{t+1} - \mathbf{l}_u^t \right\|}{\tau}, \\ a_u^t &= \frac{v_u^t - v_u^{t-1}}{\tau}. \end{aligned} \tag{15}$$

Therefore, we can get the flight energy consumption of UAV u in time slot t, that is,

$$E_{\text{fly},u}^t = k_{u,1} \left\| v_u^t \right\|^3 + \frac{k_{u,2}}{\left\| v_u^t \right\|} \left(1 + \frac{\left\| a_u^t \right\|^2}{g^2} \right). \tag{16}$$

Among them, $k_{u,1} = 0.5 \cdot \rho \cdot G_u \cdot \tau$, $k_{u,2} = (2 \cdot G_u^2 \cdot \tau / (\pi \rho A_R S_r))$. G_u represents the weight of UAV u, and g represents the acceleration of gravity. ρ stands for air density (kg/m^3). A_u^t and S_u^t represent the number of revolutions per second of the propeller of UAV u and the speed of the wind around the UAV u (m/s) in time slot t, respectively.

Regarding the flight of UAVs, there are three special restrictions that need to be specified. First, the flying speed of every UAV at each moment should be less than the maximum flying speed $v^{\max}$, that is,

$$\left\| v_u^t \right\| \le v^{\max}, \quad u \in U, t \in T. \tag{17}$$

Equality (18) restrict the position of UAVs. The meaning of equality (18) means that each UAV's own trajectory history cannot be changed. The meaning of equality (18) means that during the total time T, each UAV has an initial position and a final position. The specific restrictions are as follows:

$$\begin{aligned} \sum_{t=0}^{T-1} \left\| l_u^{t+1} - l_u^t \right\| &\le \eta, \quad u \in U, t \in T, \\ l_u^0 &= l_{\text{u,start}}, \\ l_u^{F-1} &= \mathbf{l}_{\text{u,end}}, \quad u \in U, \end{aligned} \tag{18}$$

where η is a small nonnegative value. $l_{\text{u,start}}$ and $l_{\text{u,end}}$ represent the initial and final positions of UAV u, respectively.

In summary, the total energy consumption of all UAVs in the system in time slot t is defined as follows:

$$E_{UAV}^{t} = \sum_{u \in U} \left(E_{\text{com},u}^{t} + E_{tra,u}^{t} + \varepsilon \cdot E_{\text{fly},u}^{t} \right), \tag{19}$$

where ε is a nonnegative penalty coefficient, which represents the ratio of the flying energy consumption $E_{\text{fly},u}^{t}$ to the total energy consumption E_{UAV}^{t} of UAVs. The larger the value of ε, the greater the proportion of $E_{\text{fly},u}^{t}$ in E_{UAV}^{t}. Otherwise, the smaller the proportion of $E_{\text{fly},u}^{t}$ in E_{UAV}^{t}. In the MEC system, we adjust the ε value to achieve a balance between UAVs flight energy consumption and UAVs computation migration energy consumption. Similarly, we can get the total energy consumption of the entire system in time slot t, that is,

$$E_{\text{total}}^{t} = \varpi_1 \cdot \sum_{m \in M} \left(E_{\text{com},m}^{t} + E_{tra,m}^{t} \right) + \varpi_2 \cdot E_{UAV}^{t}, \tag{20}$$

where ϖ_1 and ϖ_2 represent energy consumption weighting factors of MDs and UAVs, respectively, and satisfy $0 \le \varpi_1 \le 1$, $0 \le \varpi_2 \le 1$, and $\varpi_1 + \varpi_2 = 1$. When ϖ_1 increases, ϖ_2 will decrease accordingly. At this time, the energy consumption ratio of MDs becomes larger and the energy consumption ratio of UAVs becomes smaller. Conversely, the energy consumption ratio of MDs becomes smaller, and the energy consumption ratio of UAVs becomes larger. In this model, we can dynamically adjust ϖ_1 and ϖ_2 according to the actual situation to meet the priority energy demand, and finally, achieve the energy balance between MDs and UAVs.

4.1. Problem Formulation. Next, we mainly optimize from two aspects of energy consumption, UAVs load balancing (that is, to ensure that the task queue of UAVs in each time slot in the system has a minimum backlog), and comprehensively consider performance indicators such as task offloading, CPU cycle frequency allocated by UAVs to MDs offloaded tasks and flight energy consumption of UAVs. Finally, we worked out the problem of minimizing the energy consumption of the system under the premise of UAVs load balancing, that is,

$$P: \min_{X} \frac{1}{T} \sum_{t=0}^{T-1} E_{\text{total}}^{t},$$
$$\text{s.t. } (3),\ (5),\ (8),\ (10),\ (11),\ (15),\ (18),\ (19),\ (20), \tag{21}$$

where X represents a migration decision, which is a set of nodes on a migration path. For example, $X = \{1, 3, 4\}$ means that tasks are migrated from UAV 1 to UAV 3 first, and then they are migrated from UAV 3 to UAV 4. The research focus of this paper is to find the optimal edge servers (i.e., UAVs) migration path that is most suitable for tasks to complete the migration while ensuring that each edge server is load balanced, so that edge servers are not overloaded. Especially, the optimal migration path with the lowest energy consumption.

5. Joint Optimization Strategy of Task Migration and Load Balancing

5.1. Description of the Dynamic Migration Algorithm. In this section, we propose a new and improved iterative algorithm, called Energy-efficient Dynamic Task Migration (EDTM). The proposed EDTM algorithm is modified on the basis of the ant colony algorithm. The traditional ant colony algorithm is a heuristic optimization algorithm. Although the initial optimization results of the problem can be obtained by the traditional ant colony algorithm, it has the disadvantages of being easily trapped into a local optimum and slower convergence speed [24]. In this paper, we combine the specific research content of this paper to improve the traditional ant colony algorithm. According to the inspiration of [25], the proposed EDTM algorithm has adaptive consistency and is more suitable for MEC. The main improvements are as below.

First, we modified the pheromone update strategy in the ant colony algorithm. The idea of the ant colony algorithm originally originated from the process of ants finding the best path. An ant leaves an odor when walking through a certain path. This odor is called a pheromone. The ants behind it would be based on the pheromone concentration to choose a path, the greater the pheromone concentration, the higher the probability that an ant will choose this path, and vice versa. In this model, the proposed EDTM algorithm will also select the next UAV node according to the number of different UAV nodes and the location of each UAV node in each time slot. It can prevent the proposed EDTM algorithm from falling into a local optimum too quickly, and at the same time increase the global search capability of the algorithm.

Secondly, based on the goal of this paper is to jointly optimize UAV load balancing and reduce the total energy consumption of the system, so we add the task queue of UAVs as a variable to the formula transition probability. The meaning of transition probability is the probability that the ant chooses the next UAV node. This helps us find a reasonable migration path.

Thirdly, in the initial iterative process of the ant colony algorithm, the ant has to try all possible paths in order to find the optimal path, which will cause the algorithm to be redundant and increase the complexity. In the proposed EDTM algorithm, we propose a method of path elimination. By setting the "number of pathfinding nodes" to control the number of iterations of the algorithm. On the basis of ensuring the accuracy of the algorithm results, the complexity of the algorithm is further reduced, and the algorithm execution efficiency is improved.

The proposed EDTM algorithm comprehensively considers the migration distance, the UAVs load situation, and the migration cost to finally plan a reasonable task migration path. The algorithm has a very strong ability to find solutions to combined optimization problems. In addition, it has the advantages of distributed computing and the advantages of easy integration with other algorithms. In particular, it can show a high degree of flexibility and robustness in dynamic environments [26, 27]. Because the scenario in this paper uses UAVs as edge servers to provide services to users, we attach great importance to the advantages of mutual cooperation between UAVs.Finally, these UAVs process the sub-tasks and return corresponding results to MDs.

5.2. Solution to Problem P. In this section, we specifically explain how the proposed EDTM algorithm solves problem *P*. In the previous model, we have defined the UAVs set $U = \{1, 2, \ldots, t, \ldots, U\}$, $\forall u \in U$, and $\widetilde{d}_u^t$ represents the distance between UAVs nodes in time slot *t*. Specifically, our description of the proposed EDTM algorithm is divided into the following seven steps.

Step 1. Initialize. In this paper, UAVs nodes can communicate with each other, so we initialize the pheromone on each edge to a small constant value, and then randomly assign *A* ants to *U* UAVs nodes. At the same time, we set the starting UAV node to the tabu list. One thing that needs to be explained here is that the reason why we did not initialize the pheromone directly to zero is that the ants choose the next hop UAV node will receive the influence of the pheromone concentration.

Step 2. Each ant will select the next UAV node according to formulas (22) and (23), and update the tabu list.

$$j = \begin{cases} \arg\max_{j \notin Y_a} \left\{ \left[\phi_{ij}(t)\right] \cdot \left[\varphi_{ij}(t)\right]^{\beta} \cdot \left[\chi_{ij}(t)\right]^{\lambda} \right\}, & \text{if } q \le q_0, \\ p_{ij}^a, & \text{otherwise,} \end{cases} \tag{22}$$

where $i \in U$, $j \in U$, and Y_a is a set, that is, tabu table. Y_a is used to record the UAV nodes passed by ant *a*. The tabu table Y_a will be updated as the algorithm iterates. $\phi_{ij}(t)$ indicates the value of the pheromone transferred from UAV node *i* to UAV node *j* in time slot *t*. The greater the value of the pheromone, the greater the probability that the ant chooses the UAV node as the migration node, and the two show a proportional relationship. $\phi_{ij}(t)$ is the heuristic information of the ant transferred from UAV node *i* to UAV node *j* in time slot *t*, and the heuristic information will affect the probability that the ant chooses the next node. In this paper, we take the distance between UAV node *i* and UAV node *j* as the independent variable of the heuristic information, so we define $\phi_{ij}(t) = 1/\widetilde{d}_{\mathrm{i,j}}^t$. The longer the distance between UAV node *i* and UAV node *j*, the smaller the value of heuristic information; otherwise, the larger the value of heuristic information. Next, we named the $\chi_{ij}(t)$ in formula (22) as a statin, which represents the value of the statin of the ants from UAV node *i* to UAV node *j* in time slot *t*. The effect of statin is opposite to that of pheromone. The greater the value of statin, the lower the probability that the ant will choose the UAV node as the migration node, and the two show an inverse relationship. The optimization goals of this paper are two aspects, namely, to ensure the load balance of UAV groups and reduce the total energy consumption of the system. Therefore, we specifically represent the statin $\chi_{ij}(t)$ in formula (22) as $\chi_{ij}(t) = k_3 \cdot Q_j^{UAV}(t)$. It can be seen that the larger the load of the UAV node (that is, the longer the task backlog queue length of the UAV node in time slot *t*), the larger the statin of this path. Similarly, the smaller the load of the UAV nodes in this path is, the smaller the statin of this path is. Therefore, $\chi_{ij}(t)$ and $Q_j^{UAV}(t)$ have a function proportional relationship, where k_3 is used as the coefficient of the function expression. It should be noted that $q_0 \in [0, 1]$ is a parameter initially set by the system, and *q* is a random number and satisfies $q \in [0, 1]$; finally, p_{ij}^a represents the probability that the ant *a* moves from UAV node *i* to UAV node *j*. The specific expression is as follows,

$$p_{ij}^a = \begin{cases} \dfrac{\left[\phi_{ij}(t)\right]^{\alpha} \cdot \left[\varphi_{ij}(t)\right]^{\beta} \cdot \left[\chi_{ij}(t)\right]^{\lambda}}{\sum_{a \notin Y_a} \left\{ \left[\phi_{ia}(t)\right]^{\alpha} \cdot \left[\varphi_{ia}(t)\right]^{\beta} \cdot \left[\chi_{ia}(t)\right]^{\lambda} \right\}}, & j \notin Y_a, \\ 0, & j \in Y_a. \end{cases} \tag{23}$$

In particular, formula (22) and formula (23) contain the weighting factors of pheromone, heuristic information, and statin, which represent the importance of pheromone on paths *i* and *j*, represent the importance of heuristic information and represent the importance of statin on paths *i* and *j*.

Step 3. Local update of pheromone. According to the idea of bionics in biology, if an ant passes a certain path, then the pheromone concentration on this path will increase again. The specific formula for the pheromone update is as follows,

$$\phi_{ij}(t+1) = (1 - \mu(t))\phi_{ij}(t) + \mu(t) \cdot \Delta\phi_{ij}^a, \quad i \in U,\ j \in U, \tag{24}$$

where $\mu(t)$ is called the pheromone volatility factor, which indicates the volatility coefficient of the pheromone in time slot *t*. The setting of $\mu(t)$ in the proposed EDTM algorithm is very important, it directly affects the execution efficiency of the algorithm. $\mu_{\min}$ is the minimum value of the pheromone volatilization rate set by the system in advance, and its role is to prevent the convergence speed of the proposed EDTM algorithm from being too low due to $\mu(t)$ too small. At the beginning of the algorithm, $\mu(t)$ should take a larger value. The advantage of this is that it can speed up the convergence speed of the algorithm. With the increase of the number of iterations, in order to prevent the convergence from falling too quickly into a local optimum, the value of $\mu(t)$ should be gradually reduced to improve the global search capability of the proposed EDTM algorithm. Therefore, this model uses the method shown in (25) to adaptively adjust $\mu(t)$.

$$\mu(t) = \begin{cases} 0.95\mu(t-1), & 0.95\mu(t-1) > \mu_{\min}, \\ \mu_{\min}, & 0.95\mu(t-1) \le \mu_{\min}. \end{cases} \tag{25}$$

In addition, $\Delta\phi_{ij}^a$ in (24) represents the amount of pheromone left on the path when ant *a* passes through UAV node *i* and UAV node *j*. The specific expression of $\Delta\phi_{ij}^a$ is shown:

$$\Delta\phi_{ij}^a = \begin{cases} \dfrac{Q}{D_{ij}}, & \text{when ant a walks across the path of UAV node } i \text{ and } j, \\ 0, & \text{otherwise,} \end{cases} \tag{26}$$

where Q represents the total amount of information released by ant a for one week, and it is a constant; D_{ij} is the length of the path that ant a has traveled from the starting UAV node to the current UAV node.

Step 4. Compute the best path. After A ants have walked through all UAV nodes, we need to choose a path suitable for migration among these paths. In the task migration, in addition to the load of the nodes, we also need to consider the energy consumption caused by the migration [28, 29]. In the model part of the paper, equation (13) defines the migration cost, so we use migration cost as a measure of optimal path quality. The specific formula is as follows:

$$E_{\min}^{\text{mig}} = \min\{E_a\}, \tag{27}$$

where E_a is the migration cost caused by the path taken by ant a.

Step 5. Path elimination rules. In this paper, we set the "number of pathfinding nodes" B to control the number of iterations of the algorithm, and we define $B = \lceil 20\% \cdot U \rceil$. It can be seen that the number of pathfinding nodes is related to the total number of nodes in the entire system. When the proposed EDTM algorithm starts to run, A ants do not have to iterate over all UAV nodes, but only needs to filter out the B nodes closest to the initial UAV node in the system for access. After each iteration is completed, the algorithm automatically executes Step 4 to find the optimal migration path from path set, and the remaining paths no longer participate in the operation of subsequent algorithms.

Step 6. Global update of pheromone. After the ants have walked through all UAV nodes, we need to distinguish the best path from the normal path. Therefore, we process the pheromone on the best path according to formula (28) to convert the old pheromone $\phi_{ij}^{\text{old}}(t)$ into the new pheromone $\phi_{ij}^{\text{new}}(t)$.

$$\phi_{ij}^{\text{new}}(t) = (1-\sigma)\phi_{ij}^{\text{old}}(t) + \sigma \cdot \Delta\phi_{ij}, \tag{28}$$

where σ is the global pheromone volatility coefficient, $\Delta\phi_{ij}$ represents the sum of the pheromone amount of all ants passing through the path between i and j, and $\Delta\phi_{ij}$ is defined as follows:

$$\Delta\phi_{ij} = \begin{cases} \dfrac{1}{D_{\text{good}}}, & \text{if the global optimal solution across the path of UAV node } i \text{ and } j, \\ 0, & \text{otherwise,} \end{cases} \tag{29}$$

where D_{good} represents the length of the optimal path.

Step 7. In this paper, as the number of iterations increases, the change of the migration node-set X will stabilize. Therefore, we set a search number in the proposed EDTM algorithm. If the specified search number is not reached, the tabu list is cleared and the above process is repeated.

5.3. Path Elimination Rules. The path elimination rule is a process of determining the number of valid UAV nodes. If the subset of UAV nodes that can be the most likely to become valid nodes can be filtered first, this will greatly reduce the running time and complexity of the proposed EDTM algorithm. Before the proposed EDTM algorithm starts to execute, we can determine the valid nodes from the source nodes. This is a pre-processing process.

In Step 5 of Section 5.2 of this paper, we mentioned "Number of Pathfinding Nodes" B. These UAV nodes are the possible UAV nodes closest to the initial UAV node. At the same time, these UAV nodes are also the most likely to become valid UAV nodes. Therefore, we have reason to believe that the probability that the optimal path passes through these nodes will be relatively large. In addition, B meets $B \le U - 1$, and U is the total number of UAV nodes in the system. The value of B can only be a constant between 1 and U-1. The value of B will directly affect the convergence speed and accuracy of the proposed EDTM algorithm.

To facilitate understanding, we use Figures 3–5 to illustrate the process of obtaining the optimal migration path after MD m offloads tasks to UAV 1. In detail, Figure 3 shows that it is currently in the initial state and no task migration has occurred. There are 6 UAV nodes distributed from UAV node 1 to UAV node 6, respectively. Furthermore, node 1 is the initial node and node 6 is the destination node. All we need to do is plan an optimal path from node 1 to node 6. Before we start using the path elimination method, we first introduce the situation of the traditional ant colony algorithm in the early iteration, as shown in Figure 4. When the algorithm started running, UAV node 1 had to calculate the transition probability with the remaining 5 UAV nodes and then make a judgment, which means that the time complexity of the first iteration is $O(n)$. As the number of iterations increases, the running time of the algorithm will increase at an exponential rate, that is, $O(n^{n^2-2n+1})$.

To sum up, the traditional ant colony algorithm does have certain defects in the initial iteration. We start with the path elimination method, and the (a-e) graph contained in Figure 5 corresponds to the five processes of optimal path acquisition.

There are 6 nodes in Figure 5(a), so it can be computed by the formula $B = \lceil 20\% \cdot 6 \rceil = \lceil 1.2 \rceil = 2$. As shown in Figure 5(a), the two nodes closest to node 1 are node 2 and node 3. The next-hop UAV node can be obtained by computing the transition probability. We assume that the

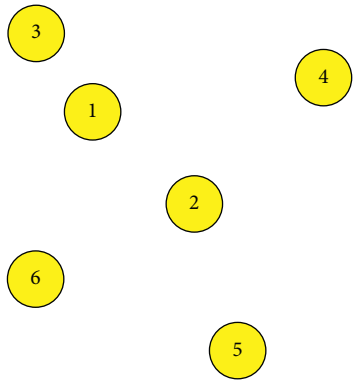


Figure 3: Initial state.

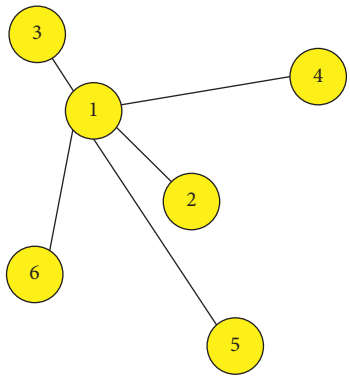
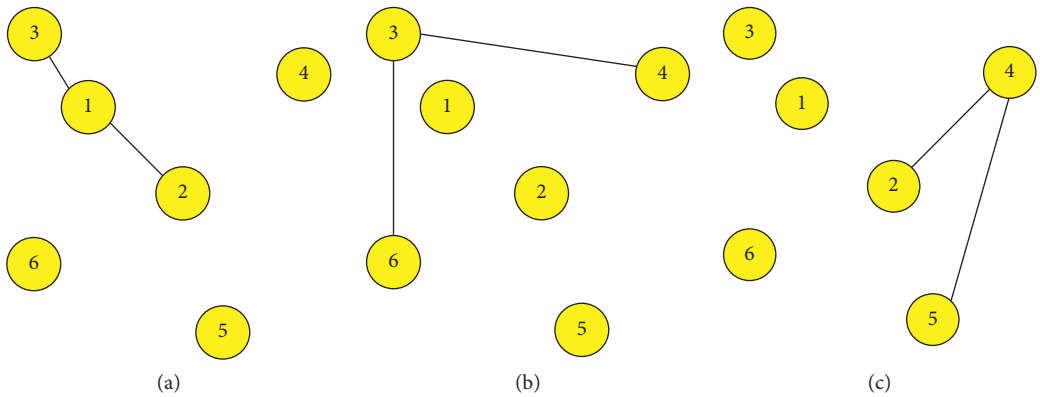
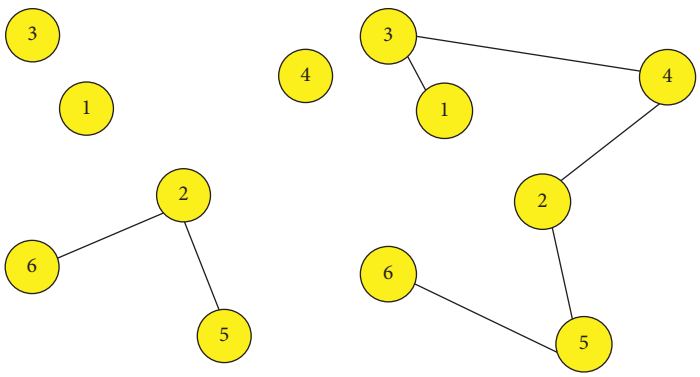


Figure 4: The situation of the traditional ant colony algorithm in the early iteration.



(a) (b) (c)



(d) (e)

Figure 5: The situation of the traditional ant colony algorithm in the early iteration.

computation result is node 3. Figure 5(b) shows two computation possibilities for the next node. For node 3, we only need to compare its transition probability with node 4 and node 6. We assume that the computation result is that node 4 as the next hop UAV node. Since the ant has searched for the destination node in the process of finding a valid node, we first update the migration node-set to $X_u^t = \{1, 3, 6\}$. Next, it is node 4 to start searching for new valid nodes. At this time, the two nodes closest to node 4 are node 1 and node 2. But since node 1 is the initial node, node 1 is excluded from the valid nodes. At this time, we need to compute the transition probability of node 2 and 5. We assume that the computation result is that node 2 as the next-hop node, as shown in Figure 5(c). As shown in Figure 5(d), the two nodes closest to node 2 and not yet visited are node 5 and node 6. Then we need to continuously compare the total migration cost with the migration cost of the previous migrated nodes set $X_u^t = \{1, 3, 6\}$ and select the optimal migration path. Finally, the migration node-set X^t is updated. Now, only the last UAV node is left, it is shown in Figure 5(e).

5.4. The Proposed EDTM Algorithm. In this section, we mainly introduce the specific process of the proposed EDTM algorithm to solve the joint optimization problem of load balancing and total energy consumption in the system. The number of target UAV nodes that we screen is $r = \lceil 15\% \cdot U \rceil$. We select r UAV nodes with the shortest task backlog queue length (that is, the lowest load) as the target UAV nodes. After clarifying the target UAV node, we begin to use the proposed EDTM algorithm to compute the optimal migration path. Because the target UAV node is not unique, the final migration path must also be unique. We can divide the task and then migrate to these UAV nodes. These UAV nodes will perform the distributed processing on the task, and finally, return the results to MDs. The detailed process of the proposed EDTM algorithm is shown in Algorithm 1. In Algorithm 1, e represents the number of iterations, $Q^{UAV}(t)$ represents the task backlog queue set of all UAVs nodes in the system at time slot t.

Besides, the special nature of the proposed EDTM algorithm, the task migration efficiency of multiple users will be higher. Because the ants look for UAV nodes and paths are based on the pheromone left by the previous ants, when there are multiple MDs that need task migration, it is no longer necessary to run the proposed EDTM algorithm to replan the migration path.

5.5. Experimental Results. In this experiment, we simulated a $600\,m \times 600\,m$ rectangular area in which a UAV group flies and provides services to MDs. The horizontal movement range of each UAV is 240 square meters. There is no limit to the flight trajectory of each UAV, but one thing we need to emphasize is that the initial position and end position of each UAV is always the same. We assume that the number of UAVs and MDs is 15 and 100, respectively. Besides, the service radius of each UAV is $150\,m$. The main parameters are defined in Table 3.

5.6. Setting of Experimental Parameters. In this experiment, we simulated a $600\,\text{m} \times 600\,\text{m}$ rectangular area in which a UAV group flies and provides services to MDs. The horizontal movement range of each UAV is 240 square meters. The flight trajectory of each UAV can be the same or different, but one thing we need to emphasize is that the initial position and end position of each UAV is always the same. We assume that the number of UAVs and MDs is 15 and 100, respectively. Besides, the service radius of each UAV is 150 m. In summary, each UAV is flying within a fixed range as an edge server and each MD offloads tasks that it cannot process or has a large amount of computation to the UAVs. The main parameters are defined in Table 3.

For some parameters in the proposed EDTM algorithm, we define $\alpha = 3$, $\beta = 2$, $\lambda = 1.5$, $A = 100$, $\mu(0) = 0.5$, $\mu_{\min} = 0.1$, $Q = 20000$, $q_0 = 0.7$, $\sigma = 0.3$. And we set the number of iterations of the entire algorithm during the run to 4500. Besides, we need to explain that if the values of (α/β) and (α/λ) are too large, the ant has a strong dependence on the pheromone when choosing the path, which will easily fall into a local optimum; otherwise, the ant will not rely on the pheromone left by other ants to make judgments.

5.7. The Effect of B and r on the Proposed EDTM Algorithm. In this paper, to accelerate the early convergence speed of the proposed EDTM algorithm, we introduce the variable B. The value of B will directly affect the convergence speed and the accuracy of the results. The B value selected in Figure 6 and Figure 7 are $10\% \cdot u$, $15\% \cdot u$, $20\% \cdot u$, $25\% \cdot u$. Since we have a small number of UAVs, our percentage difference during the experiment is 5%. In Figure 6, the time points at which $25\% \cdot u$, $20\% \cdot u$, $15\% \cdot u$, and $10\% \cdot u$ converge are 3560 ns, 4516 ns, 5231 ns, and 6010 ns, respectively. Especially, $25\% \cdot u$ has the fastest convergence speed and requires the fewest actual iterations. $10\% \cdot u$ has the slowest convergence speed, and it has the most iterations. In Figure 7, the ordinate represents the length of the optimal path. Also, we can know that $10\% \cdot u$ has the slowest convergence speed but it has the shortest optimal path length. And $25\% \cdot u$ has the fastest convergence speed, but its optimal distance is proportional to the other three best paths. In this paper, we need to consider both the convergence speed of the proposed EDTM algorithm and the accuracy of the results, so we choose $B = \lceil 20\% \cdot U \rceil$. Next, we continue to analyze the effect of r value on the proposed EDTM algorithm. To guarantee the users Qos, we split the tasks and then migrate them to multiple UAVs. These UAVs jointly perform the task and then return the final computation result to the MDs. One of our research priorities is how many UAVs we migrate to during this process. As shown in Figures 8 and 9, we consider the impact of r value on the proposed EDTM algorithm from two aspects: migration cost and load balancing. In Figure 8, the migration costs corresponding to $5\% \cdot u$, $10\% \cdot u$, $15\% \cdot u$ and $21\% \cdot u$ increase with time. Among them, the migration cost of $5\% \cdot u$ is the lowest, and the migration cost of $21\% \cdot u$ is the highest. This is because the larger the number of target UAVs, the higher the final migration cost. Also, we can find from Figure 8, the

Input: $\mathbf{Q}^{UAV}(t), q_0, \alpha, \beta, \lambda, Q, \text{num}, A, r$
Output: $\mathbf{Q}^{UAV}(t+1), \mathbf{X}^t$
(1) Initialize the target UAV node Set and initial UAV nodes
(2) for $e = 1$: num do
(3) the ant looks for the next hop UAV node according to formula (23)
(4) get $\phi_{ij}, \Delta\phi, \mu$ according to formula (24)–(26)
(5) Update the tabu table
(6) Update local pheromone ϕ_{ij}
(7) Update global pheromone ϕ_{ij}^{new} by formula (29)
(8) for $u \in U$
(9) $B = \lceil 20\% \cdot U \rceil$
(10) Update migration node set X
(11) $U = U - B$
(12) Compute total migration cost E_{total}^t
(13) Remove invalid UAV nodes and eliminate nonoptimal paths
(14) end for
(15) end for

Algorithm 1: The proposed EDTM algorithm.

Table 3: The definition of the main experimental parameters.

Parameters	Value
The theoretical communication bandwidth	10 M
The channel gain	$H_0 = 170$, $\varsigma = 15.5$
The noise power	2×10^{-9}W
The maximum transmission power of the MDs	1.7 W
The maximum transmitting power of the UAVs	100 W
The maximum CPU cycle frequency of the MDs	1.8 GHz
The maximum CPU cycle frequency of the UAVs	20 GHz
The maximum memory of the UAVs	1 T
The CPU effective capacitance switch of the MDs	1.2×10^{-17}
The CPU effective capacitor switch of the UAVs	3×10^{-16}
The density of processing tasks	10^3 cycles/bit
The mass of each UAV	19.7 kg
The maximum speed of the UAVs	13 m/s
The initial height of the UAVs	50 m
The gravitational acceleration	9.8 m/s^2
The air density	1.293 kg/m^3
The range of the UAV propeller rotations per second	[50, 160]
The wind speed range	[3.5 m/s, 10 m/s]

migration cost of the system rises rapidly between 2263 ns and 3700 ns, which is because the requirements of the migration task conform to Poisson distribution.

In Figure 9, the ordinate indicates the extreme difference of the UAV task queue length as a percentage of its maximum task queue length at the current moment. We use the computation method of extreme difference to judge the load balancing of UAVs in the MEC system. As shown in Figure 9, 0.9 represents 90%, 0.8 represents 80%, and the following values are deduced by analogy. In the previous paper, we explained that our optimization goal is to ensure that the task queue length of each UAV does not exceed 10% of the average number of tasks in the queues, so we set a baseline in the experiment. Also, we can learn that $5\% \cdot u$, $10\% \cdot u$, $15\% \cdot u$, $21\% \cdot u$ will all converge to 10%, which proves that the proposed EDTM algorithm can make the load of the UAV group reach equilibrium, but their convergence speed will be different. Among them, $21\% \cdot u$ has the fastest convergence speed, and $5\% \cdot u$ has the slowest convergence speed.

In summary, we choose $r = \lceil 15\% \cdot U \rceil$, which ensures the convergence speed of the proposed EDTM algorithm, and the system migration cost is low.

5.8. The Analysis of Weight Factor. In formula (20), we use two weighting factors, i.e., ϖ_1 and ϖ_2. Due to $0 \le \varpi_1 \le 1$, $0 \le \varpi_2 \le 1$ and $\varpi_1 + \varpi_2 = 1$, we just need to choose the value of ϖ_2. As shown in Figure 10, we selected $\varpi_2 = 0$, $\varpi_2 = 0.1$, $\varpi_2 = 0.2$ and $\varpi_2 = 0.5$. As ϖ_2 decreases, ϖ_1 will increase accordingly. At this time, the MEC system will tend to minimize the energy consumption of MDs. And the average energy consumption (AEC) of the UAVs will also increase with the acceleration. When the acceleration changes from

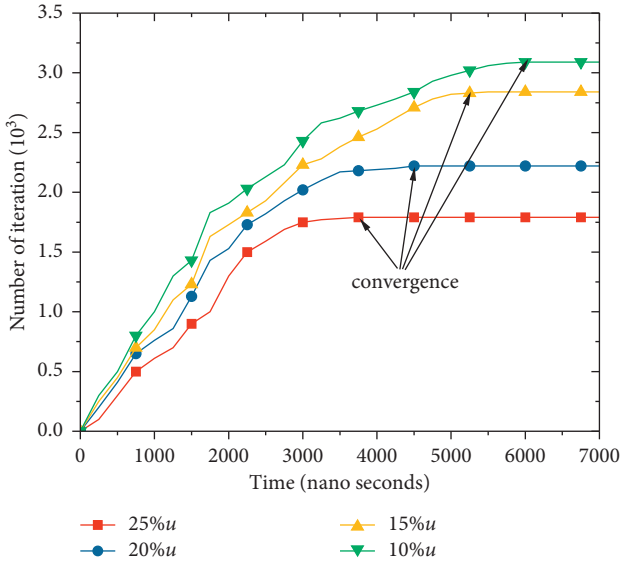


Figure 6: The number of iterations versus time and B.

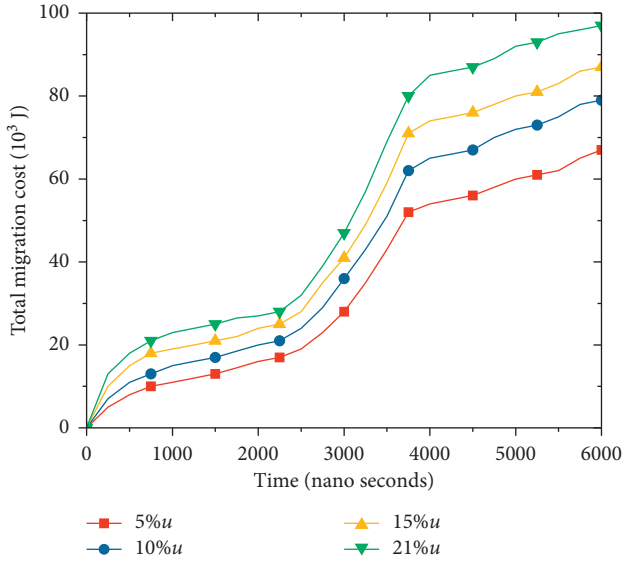


Figure 8: The total migration cost versus time and r.

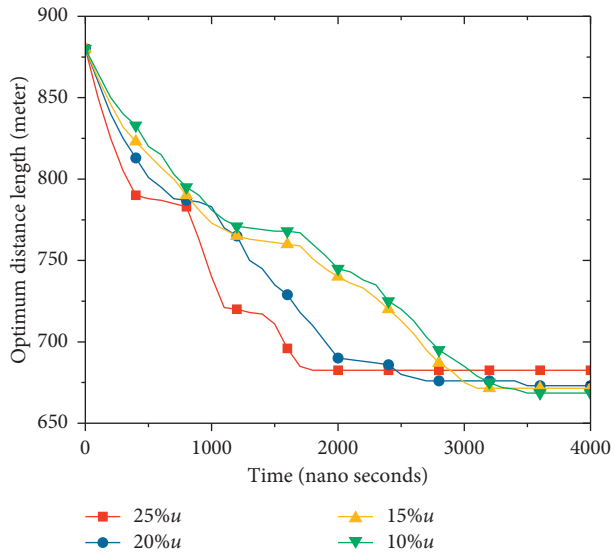


Figure 7: The optimum distance length versus time and B.

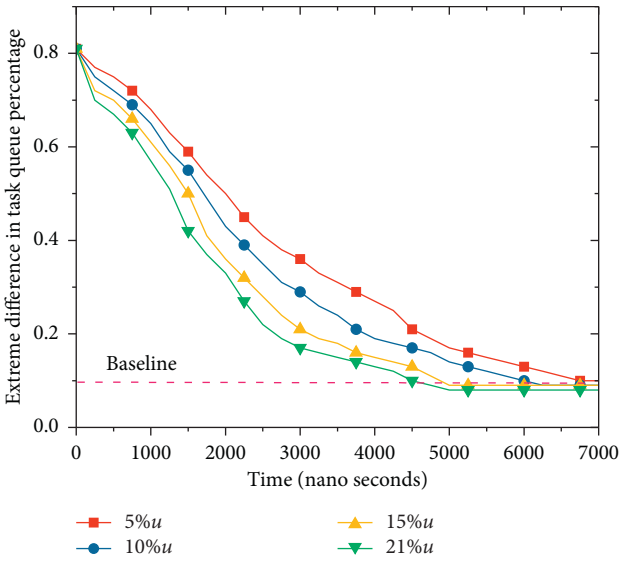


Figure 9: The load status versus time and r.

40 m/s^2 to 47 m/s^2, the AEC of the UAVSs has a growth jump, which mainly comes from two aspects. One is because we set the wind speed in the entire area as a random number in [3.5 m/s, 10 m/s], a sudden increase in wind speed will increase the flight energy consumption of the UAVs, which in turn will increase the AEC of the UAVs.

In summary, we can find that the smaller the ϖ_2, the larger the AEC of the UAVs. When $\varpi_2 = 0$, the AEC of the UAVs is the largest because the system goal is only to focus on minimizing the energy consumption of MDs.

5.9. The Analysis of the Proposed EDTM Algorithm Performance. In this section, we analyze the performance of the proposed EDTM algorithm. As shown in Figure 11, the proposed E DTM algorithm converges after approximately 2650 iterations, while the ACO algorithm enters a convergence state after approximately 3460 iterations. Obviously, under the same conditions, the proposed EDTM algorithm converges faster than the ACO algorithm.

As shown in Figure 12, the abscissa represents time, and the ordinate represents the extreme difference of the UAV task queue length as a percentage of its maximum task queue length at the current moment in the UAV group. In this experiment, we assume that the process of task arrival conforms to the Poisson distribution, so we can know from Figure 12 that the task queue extreme value of the UAV group between 1500 ns and 2700 ns becomes large and rises very quickly. Between 2700 ns and 5000 ns, the proposed EDTM algorithm makes the extreme difference smaller and stays around 0.1 in the end.

According to the above experiment, the proposed EDTM algorithm is superior to the ACO algorithm in terms of

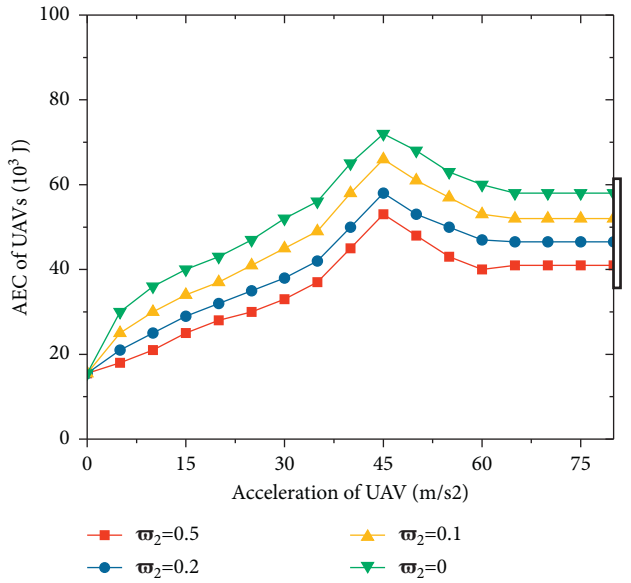


Figure 10: The AEC of UAVs versus acceleration and v2.

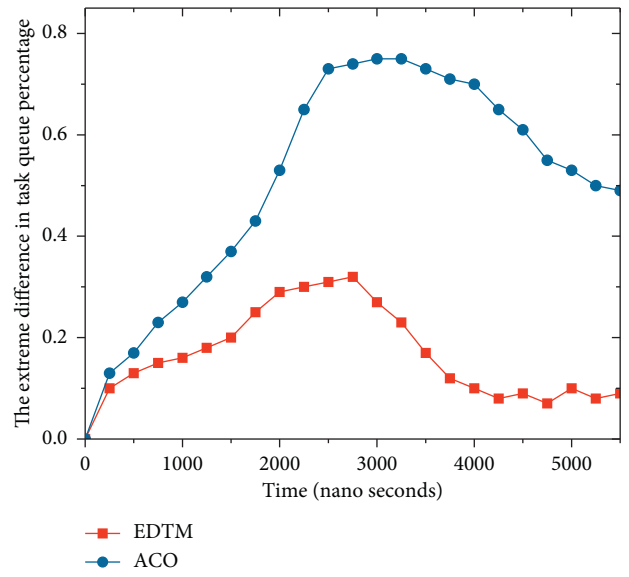


Figure 12: The comparison in optimizing UAVs load between the proposed EDTM algorithm and the algorithm ACO.

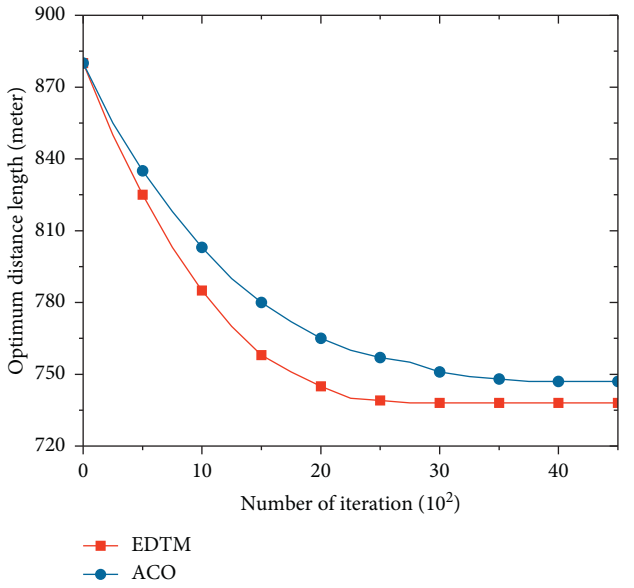


Figure 11: The convergence comparison between the proposed EDTM algorithm and the ACO algorithm.

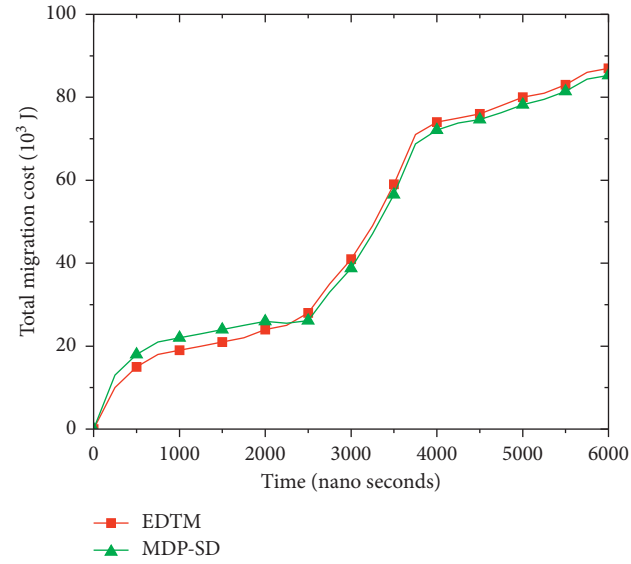


Figure 13: The comparison of total migration cost between the proposed EDTM algorithm and the algorithm MDP-SD algorithm.

convergence speed and optimization of load balancing efficiency of the UAV group. At the same time, we have also proven that the proposed EDTM algorithm can help the UAV group achieve load balancing.

Besides, part of the current research work solves the task migration problem in MEC by setting up Markov Decision Process (MDP). For example, researchers proposed an MDP algorithm based on static distance (MDP-SD). Next, we compare the performance of the proposed EDTM algorithm and the MDP-SD algorithm in terms of migration cost and average load percentage of the UAV groups. As shown in Figure 13, the reason why the curve changes as shown is because we assume that the computation task request conforms to the Poisson distribution. When $t = 0$, there is no computation task in the system, so the migration cost of UAVs is 0. When $0 < t < 1650$ ns, the computation task request amount over time slowly rising, and the migration cost of the MDP-SD algorithm is slightly higher than that of the proposed EDTM algorithm. This is because the MDP-SD algorithm consumes more computing resources than the proposed EDTM algorithm when it uses the value iterative method to solve the migration strategy in the early stage. When $1650 < t < 3800$ ns, the amount of computation task requests rises rapidly, and the migration cost of the two algorithms also increases rapidly. When $t > 3800$ ns, the amount of computation task request decrease and the rate of increase in migration costs begin to decrease.

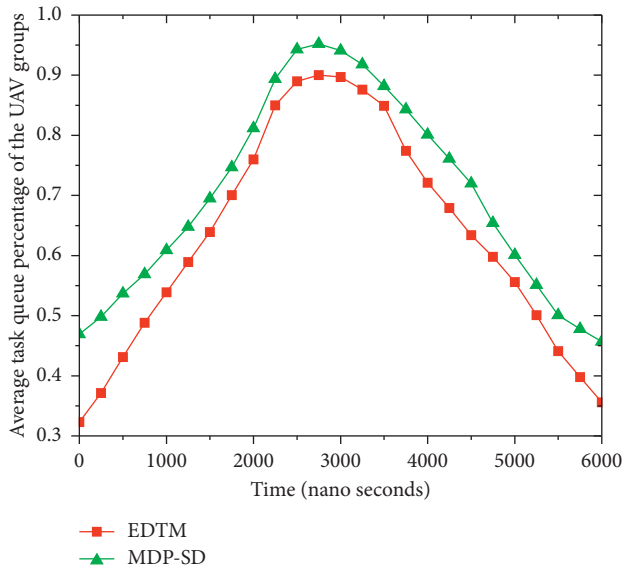


Figure 14: The comparison of average task queue percentage of the UAV groups between the proposed EDTM algorithm and the MDP-SD algorithm.

Table 4: The performance comparison.

	Convergence time (ns)	Total energy consumption (J)	Maximum system load status (%)
ACO	3550	101120	89
MDP	3611	91312	81
EDTM	2000	89991	35

After comparing the migration costs of the proposed EDTM algorithm and the MDP-SD algorithm, next, we compare the performance of the two algorithms from the average load percentage of the UAV groups. As shown in Figure 14, 0.9 means that the task queue of the UAV groups at this time accounts for 90% of the total length of the task queue. Besides, the percentage of the task queue length under the MDP-SD algorithm is higher than the proposed EDTM algorithm. This is because we considered the small storage space and computing power of the UAVs when designing the proposed EDTM algorithm. So that the proposed EDTM algorithm does not occupy too much UAV resources. Also, because the amount of computation task request conforms to the Poisson distribution, when $0 < t < 2800$ ns, the average task queue percentage of the UAV groups will increase. And when $t = 2800$ ns, whether it is the proposed EDTM algorithm or the MDP-SD algorithm, the average task queue percentage of the UAV groups will reach the highest point. Until $t > 2800$ ns, the average task queue percentage of the UAV groups slowly decrease. Obviously, from the perspective of the entire process, the average task queue percentage of the UAV groups under the proposed EDTM algorithm is always lower than the MDP-SD algorithm. Therefore, the proposed EDTM algorithm is more suitable for UAV-enabled MEC systems.

To sum up, we compare the performance of the algorithm from the following three aspects, including convergence time, energy consumption and maximum system load status. The experimental results are shown in Table 4.

6. Conclusion

In this paper, we formulated the joint optimization problem in UAV-enabled MEC system, which considers migration costs, total system energy consumption, and load balance. To address this problem, we then propose the EDTM algorithm and path elimination rules, the weight factors and the pheromone update strategy in the algorithm show the stability and robustness of the system. Finally, simulation experiments demonstrate the efficacy of the proposed EDTM algorithm. For future works, we plan to apply the algorithm in a real environment and test its performance.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the Scientific Research Project of Education Department of Hunan Province in 2019 (Grant no. 19C1725), College Innovation and Entrepreneurship Education Center in 2019, Xiangnan University Students Innovation and Entrepreneurship Education Center (Grant no. 2019-333), Teaching Reform Research Project of Colleges and Universities of Hunan Province in 2020 (Grant no. 232 HNJG-2020-0934), and Innovation and Entrepreneurship Base of Hunan Province in 2020, Network Security Innovation and Entrepreneurship Education Base (Grant no. [2020]301 no. 65).

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, 2017.

[2] F. Vhora and J. Gandhi, "A Comprehensive Survey on Mobile Edge Computing: Challenges, Tools, Applications," in *Proceedings of the 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 49–55, Erode, India, March 2020.

[3] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV mounted cloudlet: optimization of bit allocation and path planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, 2018.

[4] L. Park, C. Lee, W. Na, S. Choi, and S. Cho, "Two-stage computation offloading scheduling algorithm for energy-harvesting mobile edge computing," *Energies*, vol. 12, no. 22, p. 4367, 2019.

[5] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic Service Migration in Mobile Edge Computing Based on Markov Decision Process," *IEEE/ACM Transactions on Networking*, no. 99, , 2019.

[6] Z. Kuang, L. Li, J. Gao, L. Zhao, and A. Liu, "Partial offloading scheduling and power allocation for mobile edge computing systems," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6774–6785, 2019.

[7] R. Liang, G. Wang, and J. Hu, "Resource caching and task migration strategy of small cellular networks under mobile edge computing," *Mobile Information Systems*, vol. 2021, no. 2, pp. 1–16, 2021.

[8] J. H. Anajemba, T. Yue, C. Iwendi, M. Alenezi, and M. Mittal, "Optimal cooperative offloading scheme for energy efficient multi-access edge computation," *IEEE Access*, vol. 8, pp. 53931–53941, 2020.

[9] W. Zhou, L. Xing, J. Xia, L. Fan, and A. Nallanathan, "Dynamic Computation Offloading for MIMO Mobile Edge Computing Systems with Energy Harvesting," *IEEE Transactions on Vehicular Technology*, 2021.

[10] Y. K. Tun, Y. M. Park, N. H. Tran, W. Saad, S. R. Pandey, and C. S. Hong, "Energy-efficient resource management in UAV-assisted mobile edge computing," *IEEE Communications Letters*, vol. 25, no. 1, pp. 249–253, 2021.

[11] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, "Trajectory design and power control for multi-UAV assisted wireless networks: a machine learning approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 99, p. 1, 2019.

[12] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, and X. Shen, "Energy-efficient UAV-assisted mobile edge computing: resource allocation and trajectory optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 99, p. 1, 2020.

[13] Q. Wu, Z. Yong, and Z. Rui, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 99, p. 1, 2017.

[14] A. Garg, "Machine Learning Coupled Trajectory and Communication Design for UAV-Facilitated Wireless Networks," 2020, https://arxiv.org/abs/2101.10454.

[15] K. Sungwook, "One-on-one Contract Game-Based Dynamic Virtual Machine Migration Scheme for Mobile Edge Computing," *European Transactions on Telecommunications*, vol. 29, 2018.

[16] T. M. Cover and J. A. Thomas, "Elements of Information Theory (Cover/Elements of Information Theory, Second Edition) || Index," *Index*, pp. 727–748, 2005.

[17] H. Huang, A. V. Savkin, and W. Ni, "Energy-efficient 3D navigation of a solar-powered UAV for secure communication in the presence of eavesdroppers and No-fly zones," *Energies*, vol. 13, no. 6, p. 1445, 2020.

[18] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[19] P. Mach and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[20] C. Iwendi, J. H. Anajemba, J. A. Ansere, F. Sam, and G. Srivastava, "Optimal soft error mitigation in wireless communication using approximate logic circuits," *Sustainable Computing: Informatics and Systems*, vol. 30, no. 4, Article ID 100521, 2021.

[21] T. D. Burd and R. W. Brodersen, "Processor design for portable systems," *The Journal of VLSI Signal Processing*, vol. 13, no. 2-3, pp. 203–221, 1996.

[22] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, 2017.

[23] M.-A. Messous, H. Sedjelmaci, N. Houari, and S.-M. Senouci, "Computation offloading game for an UAV network in mobile edge computing," in *Proceedings of the 2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, Paris, France, May 2017.

[24] J. Xu, L. Chen, and S. Ren, "Online learning for offloading and autoscaling in energy harvesting mobile edge computing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 3, pp. 361–373, 2017.

[25] Q. Wang and C. Sun, "Adaptive Consensus of Multiagent Systems with Unknown High-Frequency Gain Signs under Directed Graphs," *IEEE Transactions on Systems Man & Cybernetics Systems*, pp. 1–6, 2018.

[26] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927–1941, 2018.

[27] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao, "Joint computation and communication design for UAV-assisted mobile edge computing in IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5505–5516, 2020.

[28] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Automation in Construction*, vol. 81, pp. 25–33, 2017.

[29] L. Ru, L. Ya-fei, and H. Zhong-xi, "A model of mission planning for cooperative UAVs," in *Proceedings of the 27th Chinese Control and Decision Conference (2015 CCDC)*, pp. 67–72, Qingdao, China, May 2015.