Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A UAV-enabled mobile edge computing paradigm for dependent tasks based on a computing power pool*

Xuebin LAI, Yan GUO†‡, Ming HE, Hao YUAN, Wei LI, Xiaonan CUI

Army Engineering University of PLA, Nanjing 210007, China

†E-mail: guoyan_1029@sina.com

Received May 31, 2024; Revision accepted Dec. 1, 2024; Crosschecked

Abstract: With the evolution of 5G and 6G communication technologies, various Internet of Things (IoT) devices and artificial intelligence applications are proliferating, putting enormous pressure on existing computing power networks. Unmanned Aerial Vehicle (UAV)-enabled Mobile Edge Computing (U-MEC) shows potential to alleviate this pressure and has been recognized as a new paradigm for responding to the data explosion. Nevertheless, the conflict between computing demands and resource-constrained UAVs poses a great challenge. Recently, researchers have proposed resource management solutions in U-MEC for computing tasks with dependency. However, the repeatability among the tasks was ignored. In this paper, considering repeatability and dependency, we propose a U-MEC paradigm based on a computing power pool for processing computationally intensive tasks, in which UAVs can share information and computing resources. To ensure the effectiveness of computing power pool construction, the problem of balancing the energy consumption of UAVs is formulated through joint optimization of an offloading strategy, task scheduling, and resource allocation. To address this NP-hard problem, we adopt a two-stage alternate optimization algorithm based on a Successive Convex Approximation (SCA) and an improved Genetic Algorithm (GA). The simulation results show that the proposed scheme reduces time consumption by 18.41% and energy consumption by 21.68% on average, which can improve the working efficiency of UAVs.

Key words: U-MEC; Computing power pool; Dependency; Repeatability
https://doi.org/10.1631/FITEE.2400465 **CLC number:**

1 Introduction

As research on 5G and 6G continues to make strides forward, various Internet of Things (IoT) devices have become ubiquitous. According to Machina Research, the number of global IoT devices is expected to increase to 27 billion by 2025 (Alliance of Industrial Internet, 2023). The applications of artificial intelligence (AI) technologies are also increasing, such as Large Language Models (LLMs) and Augmented Reality (AR). The computing tasks of these applications are mostly delay sensitive and computationally intensive, making them beyond the capacity of mobile devices (Zhang et al., 2021; Michailidis et al., 2022; Ning et al., 2023). The traditional method is to adopt cloud computing technology. However, cloud servers cannot meet the low latency requirements of a massive influx of tasks (Lin et al., 2020). Mobile Edge Computing (MEC) technology can address this shortcomings to a certain extent, and is applied to the network edge closer to users (Kato et al., 2019). Nevertheless, traditional mobile edge servers have limitations in deployment and cannot be rapidly deployed in some emergency situations, deserts, and wilderness places (Kato et al., 2019; Radio Management Institute of the SEDIC Intelligence Centre, 2020). Fortunately, the development of unmanned aerial vehicles (UAVs) is gradually maturing.

‡ Corresponding author

* Project supported by the Natural Science Foundation of Jiangsu Province through Grant (No. BK20211227), and the National Natural Science Foundation of China through Grant (No. 62273356)

UAVs are flexible and easy to deploy, so a new paradigm of UAV-enabled Mobile Edge Computing (U-MEC) has emerged. The U-MEC system offers the prospect of a wide range of applications that can perform data collection, extend the communication range, share the load of a terrestrial Base Station (BS) or Access Point (AP), and perform computational tasks for IoT devices that do not have direct access to communication infrastructure (Ning, et al., 2023).

However, the constraints of weight, power, energy and other factors impose limitations on the communication and computation capabilities of UAVs during data processing. Consequently, the objective of research is to propose efficient resource management methods for the limited resources in U-MEC systems (Abrar et al., 2021). The user tasks provide essential insight into the optimal organization and utilization of resources, thereby facilitating the development of a more targeted approach to resource management that is grounded in the characteristics of user tasks.

The interdependence of subtasks represents a fundamental characteristic of computational tasks. Most computing tasks can be classified into a series of subtasks, each with a defined sequence of completion. These subtasks are collectively referred to as dependent tasks (Liu et al., 2019; Huang et al., 2023). The rationale behind the dependency of computing tasks is that developers typically embrace the concept of modularity when constructing programs and applications. They then proceed to link together multiple modules, each with its own distinct functionality, in accordance with specific logical relationships. (Nguyen et al., 2023). A multitude of applications comprise a set of interdependent tasks. To illustrate, the navigation service in AR consists of four subtasks, which can be performed on disparate modules (Pranoto et al., 2023). The control module initially identifies the geographical location of the current starting point upon receipt of the user-inputted destination. Subsequently, the data are conveyed through the map module and the traffic module, whereby alternative routes and traffic conditions are obtained, respectively. Ultimately, the route module superimposes the virtual route guide within the environment, according to the output results of the two parallel modules (Ning et al., 2023). The data dependency between subtasks represents a significant challenge for the allocation, optimization and utilization of resources, which in turn presents a substantial obstacle to the effective management of U-MEC resources.

In recent research, the interdependence of subtasks has been incorporated into the U-MEC system. With the aim of minimizing the energy and time costs of IoT devices, Yan et al. (2019) investigated the joint optimization problem of the offloading strategy and resource allocation when dependent tasks involve two IoT devices. For the identical optimization objective and problem, a DRL framework was used by Yan et al. (2020), which is appropriate for the application scenarios of time-varying wireless fading channels and random MEC capabilities. Sahni et al. (2020) adopted a heuristic algorithm to solve the problems of task offloading and stream scheduling to minimize the average completion time of tasks. With the goal of minimizing response delay and energy consumption of UAV clusters, Li et al. (2023) studied task scheduling and UAV deployment optimization problems for data analysis service applications with dependency and adopted the DRL algorithm to solve them. Previous research has investigated the issues of task scheduling, resource allocation and offloading strategies in resource management for dependent tasks, with the objective of minimizing the time to task completion or minimizing the energy cost. Nevertheless, the UAVs in these studies were in a hovering state, and the flexibility of the UAVs was not sufficiently utilized. Furthermore, aspects of trajectory optimization and task completion efficiency require improvement. Xu et al. (2022) considered the trajectory optimization problem of UAVs, considering the case of a single UAV assisting the user's task computation. However, due to the sequential execution order requirement between the subtasks of the dependent tasks, the UAV in this study was idle and waiting, and there was scope for improvement in the efficiency of task completion. Nguyen et al. (2023) proposed multi-UAV collaboration with the objective of completing the computation of dependent tasks. While this method does indeed reduce the idle waiting time of the UAV, it fails to account for the correlation between the tasks, resulting in a redundant allocation of resources. Consequently, there is a clear opportunity to enhance the efficiency of resource utilization.

We conclude that most research studies have treated each dependent task in isolation, without considering the repeatability of its subtasks. This approach has the potential to result in the inefficient use of scarce resources. In particular, the completion of each independent task can be divided into multiple sequential subtasks. However, there are repeating parts among them, particularly when the users are situated in the same area and time period. To illustrate, the completion of a personalized recommendation service necessitates use of the user's behavioural data and a personalized algorithmic recommendation model. Additionally, it requires the incorporation of the actual local situation to facilitate the delivery of the most appropriate recommendations to the user (Yuan et al., 2023). In situations where users are situated in the same area and time, the life service application will update recommendations pertaining to nearby food, hotels, and attractions. Additionally, the smart bracelet will update recommendations related to exercise modes and health monitoring. The completion of these tasks necessitates the calling of the subtask pertaining to map information within the area in which the users are located. This component is identical in the tasks of each user. In the context of applications and data collection scenarios of the Internet of Things (IoT), there are numerous computationally intensive tasks that are analogous to those involved in the generation of personalized recommendations. If the recurrent parts are not processed in an appropriate manner, the subtasks will be computed tautologically, which results in a waste of resources.

Accordingly, in this paper we propose a U-MEC computing paradigm based on a computing power pool to address the issues associated with the repeatability and dependency of subtasks. The computing power pool consists of UAVs which are scheduled and managed in a uniform manner by a Data Center (DC). In addition to sharing the results of subtasks, the UAVs in the computing power pool can share their own computing resources. This approach not only addresses the issue of repetitive subtask execution but also prevents the inefficient use of UAV resources and enhances overall task completion efficiency. To guarantee the stability and fairness of the construction of the computing power pool, it is essential to ensure that each UAV participating in the construction of the pool has a relatively balanced consumption of resources. The energy consumption index of the UAV is a key concern for researchers, and therefore, we propose a joint optimization problem to balance the UAV energy consumption. The following section outlines the specific work presented in this paper.

(1) We investigate a scenario where multi-UAVs assist multi-users in offloading dependent tasks comprising subtasks that are partly repetitive. We mathematically formulate a problem of balancing the energy consumption of UAVs through jointly optimizing the UAV-user association, communication resources, computing resources, UAV-subtask association, and the trajectories of UAVs, constrained by the demands of tasks and the capacity of the UAVs.

(2) We propose a scheme based on constructing a computing power pool for task linkage. Accordingly, the original problem is disassembled into two subproblems, namely, the optimization of UAV-user association, communication resource allocation and UAV trajectory, and UAV-subtask association and computational resource allocation. We adopt a two-stage alternate optimization algorithm based on SCA technology and an improved genetic algorithm (GA) through operator redesign to solve this problem.

(3) Through a series of experiments, we explore and analyze the effectiveness of the proposed scheme and the impact of the subtask repetition rate. The experimental results show that the proposed scheme can effectively reduce the energy consumption and execution time of UAVs, and the proposed algorithm performs well.

2 Related studies

The computational operations of a user device can typically be classified into two main categories: those with subordinate tasks and those without such tasks. This section presents an analysis of U-MEC work based on the categorization of application tasks according to the presence or absence of subtasks.

2.1 U-MEC for Application tasks without subtasks

For application tasks without subtasks, current research focuses on the design of networks for different scenarios, integration with MEC, and improvement of optimization methods, most of which characterize tasks with two dimensions, namely computational resource requirements and latency requirements. Based on a priority calculation method referring to user delay requirements and residual energy, Tian et al. (2023) used a GA-based method to solve the problem of maximizing user satisfaction by jointly optimizing the user offloading strategy and the UAV trajectory. Wang et al. (2022a), in a study of a scenario of a UAV as a relay and providing server assistance for the BS, considered the problem of jointly optimizing the user offloading strategy, resource allocation and trajectory of the UAV to maximize the computational load and minimize the energy consumption, and decomposed the original problem by using the Block Coordinate Descending (BCD) method. The internal point method was used to solve bandwidth allocation, computing resource allocation and computing offloading problems, and Successive Convex Approximation (SCA) technology was used to solve the UAV trajectory optimization problem. For video offloading, Zhao et al. (2023) proposed a system energy optimization problem that jointly optimizes frame rate, offloading decision, communication resource allocation, and UAV position deployment, and adopted a Deep Reinforcement Learning (DRL) algorithm to solve it. Mei et al. (2019) adopted a clone sharing technique at the cloud and UAV edge to collaborate on user tasks by sharing data and services. Qin et al. (2023) and Zhai et al. (2022) both investigated U-MEC networks assisted by Reflective Intelligence Surface (RIS). Qin et al. (2023) focused on single-dimensional computational resource demand optimization, but their task model considers only the efficient allocation and utilization of computational resources to maximize the network's performance. In contrast, Zhai et al. (2022) considered the time-sensitive nature of the task, balancing the time constraints of task execution with the efficiency of resource allocation. This multi-dimensional optimization strategy demonstrates the flexibility and potential of RIS-assisted networks in handling tasks. Khalid et al. (2023) proposed a U-MEC network with energy harvesting applied to disaster sites to enhance computational efficiency.

Application tasks that do not include subtasks are relatively straightforward and typically stand alone. Consequently, the corresponding resource management program prioritizes the aspects of computational resources and latency requirements.

2.2 U-MEC for Application tasks with subtasks

As a consequence of the growing convergence of AI and the IoT, the nature of application tasks is becoming increasingly complex and diverse. To facilitate their management, it is possible to decompose these tasks into a series of subtasks, which are then executed in a predetermined sequence.

Some studies have sought to enhance task completion efficiency by disregarding the interdependencies between subtasks. This approach involves offloading subtasks to different carriers for parallel computation (Hu et al., 2018; Hu et al., 2020; Wu et al., 2020). However, in practice, most subtasks within applied tasks are interdependent. Consequently, research on task dependence, as exemplified by the U-MEC approach, has increasingly captured the attention of researchers.

The processing of dependent tasks by a single UAV is subject to limitations that may result in periods of idle time. It is therefore becoming increasingly common for multiple UAVs to collaborate to complete the necessary computations (Xu et al., 2022). Huang et al. (2023) adopted the UAV cluster method to calculate dependent tasks, divide UAVs into leading UAVs and computing UAVs, minimize task completion and balance energy consumption between computing UAVs. Nguyen et al. (2023) proposed a method of multi-UAV assisted dependent tasks in computing, in which the BS dominates the flight, communication and computing actions of the UAVs. To minimize the average delay of service, they studied the problem of jointly controlling dependent task offloading decisions and allocating the communication resources of UAVs. Guo et al. (2023) proposed a solution whereby users can forward their dependent task relays to users with good communication links, and then offload tasks to associated UAVs for computation, so as to minimize system delay. Luan et al. (2021) proposed a hierarchical hybrid subtask scheduling algorithm H-HSS, which jointly optimizes the topology and minimizes the average completion time of subtasks in task scheduling. Li et al. (2023) examined the U-MEC cost minimization problem with task dependency and proposed a Directed Acyclic Graph (DAG) optimization algorithm for task scheduling and UAV deployment. Jia et al. (2023) considered the offloading of dependent tasks based on a service function chain in U-MEC systems, and combined UAV location, service function deployment and DAG task scheduling to minimize task completion time. Thus, the consideration of task dependency has received substantial interest. Although the form of multi-UAVs is used to compute the dependent tasks, most studies treat each dependent task as an isolated entity, without considering the repeatability of its subtasks. This approach ultimately results in the inefficient utilization of scarce resources.

In addition to the fundamental requirements of computational resource demand and latency demand, the subtasks of such tasks are typically constrained by the necessity of a sequential completion order. The data dependency between subtasks represents a significant constraint on the allocation, optimization and utilization of resources, thereby posing considerable challenges to the management of U-MEC resources.

3 System model and problem formulation

3.1 System model

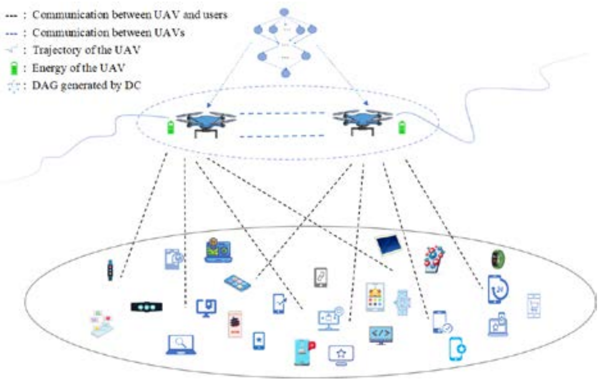


Fig. 1. System model

We studied a scenario in which multi-UAVs are used to assist multi-users in offloading computationally intensive tasks. Following the collection of task specifications of M users by N UAVs, a provisional computing power pool is established to facilitate the execution of the requisite computations (Fig. 1). The DC then assumes responsibility for task scheduling. We adopt a quadruple, (w_m, d_m, D_m, t_m), to describe the user m's task requirements, where $w_m = [x_m, y_m]$ represents m's horizontal coordinate, the vertical coordinate is 0, d_m is the amount of data that m offloads, D_m is a series of dependent subtasks, and t_m is the expected time to receive service. The moving cycle of all the UAVs is recorded as T, which can be divided into K time slots, that is $T = K\Delta T$. The UAV n flies from the starting point $l_{n,init} = [x_{n,init}, y_{n,init}, H]$, where H is the flight height, which is fixed. The flight speed of UAV n in the kth time slot, is denoted as $v_n[k]$, which does not exceed $V_{\max}$, and the minimum distance between UAVs to avoid collision is $d_{\min}$. The trajectory of UAV n on the horizontal coordinate in k's time slot is marked as $q_n[k] = [x[k], y[k]]$. Similar to the work of Liu et al. (2022), the flight constraint of the UAVs can be summarized by Eq. (1), which includes the speed constraint, starting point constraint, and collision avoidance constraint. The key notations are presented in supplementary materials.

$$\begin{cases} v_n[k] = \left\| q_n[k] - q_n[k+1] \right\| / \Delta T \le V_{\max}, \\ \qquad\qquad \forall k \in [1, K-1], \forall n \in N; \\ q_n[1] = q_n[K] = l_{n,init}, \ \forall n \in N; \\ \left\| q_n[k] - q_w[k] \right\| \ge d_{\min}, \ \ \forall n, w \in N, n \ne w \end{cases} \tag{1}$$

3.2 Communication model

3.2.1 UAV-User

Although UAVs can provide convenient line-of-sight (LoS) transmission conditions, due to the complexity and variability of the real channel, it may also receive signals through refraction and reflection. Based on current research, in our study, we adopted the Rician Fading Channel with a channel obedience factor of S to simulate the communication channel between users and UAVs. The channel coefficient considers mainly small-scale fading and channel loss (Khuwaja et al., 2018; Nasir, 2021; Wang et al., 2022a).

Small-scale fading is composed of the LoS and the non-line-of-sight (NLoS), denoted as $h_m^{n,LoS}[k]$ and $h_m^{n,NLoS}[k]$, respectively, where $\left|h_m^{n,LoS}[k]\right| = 1$, $h_m^{n,NLoS}[k] \sim N(0,1)$. The channel loss coefficient is denoted as $g_m^n[k] = g_0 (l_m^n[k])^{-\alpha}$, where $l_m^n[k] = \sqrt{\left\| q_n[k] - w_m \right\|^2 + H^2}$ indicates the distance between the user m and the UAV n. Therefore, its channel gain can be expressed as Eq. (2):

$$h_m^n[k] = \left(\sqrt{\frac{S}{S+1}} h_m^{n,LoS}[k] + \sqrt{\frac{1}{1+S}} h_m^{n,NLoS}[k] \right) g_m^n[k] \tag{2}$$

Assuming that the communication mode between the UAV and the users is OFDM, the UAVs adopt different frequency bands to avoid interference and the user's transmitting power is constant, denoted as P_m (Hu et al., 2021b; Dai et al., 2023). $B_{\max}$ is the allocatable bandwidth and $b_{n,m}[k]$ refers to the bandwidth ratio of UAV n allocated by user m. The noise power is σ^2, so the signal-to-noise ratio (SNR) between the user m and the UAV n in the kth time slot can be expressed as Eq. (3).

$$snr_m^n[k] = P_m \left| h_m^n[k] \right|^2 / (b_{n,m}[k] \sigma^2) \tag{3}$$

The data transmission rate is expressed as Eq. (4), where CFT_n is the time slot when UAV n completes information collection.

$$R_m^n[k]=\begin{cases} b_m^n[k]B_{\max}\log_2(1+snr_m^n[k]), & t_m^I \le k \le CFT_n \\ 0 & , else \end{cases} \tag{4}$$

We use $\alpha_{m,n}(k)$ to indicate whether the user m 's task is captured by UAV n in the k th time slot. If $\alpha_{m,n}[k]=1$, it means yes, otherwise $\alpha_{m,n}[k]=0$. The sets of $\alpha_{m,n}[k]$ are denoted as A, which is one of the variables to be optimized. When the access of m associated users, i.e., $\alpha_{m,n}[k]=1$, $b_{n,m}[k]$ should meet the constraints of Eq. (5).

To ensure that user data are fully received, Eq. (6) needs to be satisfied.

$$\begin{cases} 0 \le b_{n,m}[k] \le 1, & \forall m \in M, \forall n \in N, k \in K \\ \sum_{m\in M} b_{n,m}[k] \le 1, & \forall k \in K, \forall n \in N \end{cases} \tag{5}$$

$$d_m \le \alpha_{m,n} \sum_{k=t_m^I}^{CFT_n} R_m^n[k]\Delta T, \forall m \in M, n \in N \tag{6}$$

The time slot UAV n completes information collection $CFT_n=\max(t_m^I+\sum_{k=t_m^I}^{SSC}\alpha_{m,n}[k]\cdot t_{n,m}^{off})$. SSC means the time slot in which all the dispatched UAVs have completed the information collection task, that is, the time slot of the computing linkage. We ignore the delay commanded by DC. So $SSC=\max(CFT_n), \forall n \in N$. $t_{n,m}^{off}$ represents the amount of time that the UAV n assigns to the user m to offload the task, which can be calculated by Eq. (7).

$$t_{n,m}^{off}=d_m / \sum_{k=t_m^I}^{CFT_n} R_m^n[k] \tag{7}$$

3.2.2. UAV-UAV

The LoS link between UAVs is assumed here. The dispatch of UAVs including communication, is directed by DC. Like in most studies, we ignore the influence of mutual interference between UAVs (Guo et al., 2023; Huang et al., 2023; Nguyen et al., 2023). We assume that the UAV sends data at constant power P_u. Therefore, the average channel gain between UAV n and UAV w is $h_n=h_0(l_{n,w})^{-\alpha}$, the SNR $snr_n^w=P_u h_n/\sigma^2$, and the data transmission rate between UAVs is expressed by Eq. (8).

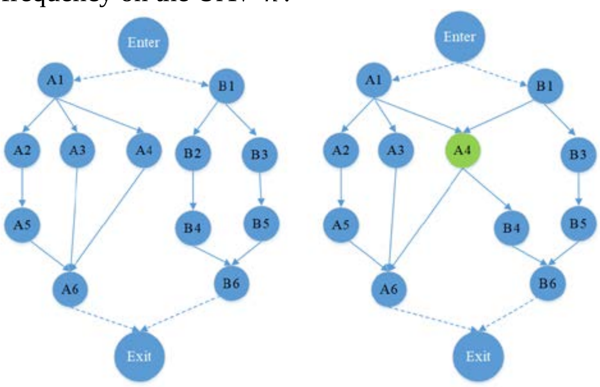
$$R_m^n=B_{\max}\log_2(1+snr_n^w) \tag{8}$$

Assuming that the amount of data transmitted between UAVs is d_u, the communication time taken to transmit the data can be calculated by Eq. (9).

$$t_{u,comm}=d_u / R_m^n \tag{9}$$

3.3. Dependent tasks model

Without loss of generality, we consider the case of one user corresponding to one task, and the user m 's dependent tasks are represented by DAG, $D_m=(T_m,E_m,C_m)$, where $T_m=\{T_{m,1},...T_{m,s}\}$ represents the set of subtasks required to complete user m 's task (Luan et al., 2021; Huang et al., 2023; Jia et al., 2023; Li et al., 2023). m and s correspond one to one, so if m is certain, then s is also certain. E_m is the dependency between subtasks, represented by an $s\times s$ dimensional matrix. For example, $e_m(i,j)$ indicates the completion sequence of the user m ' s subtask i and subtask j. A value greater than 0indicates that T_i must be completed before T_j starts computing. Otherwise, the value is negative. C_m represents the computing time of the subtask on UAVs, expressed in a matrix of $s\times N$ dimensions, $C_m(i,n)$, representing the time taken by the user's subtask i to compute at the maximum computing frequency on the UAV n.



(a) Before merging subtasks (b) After merging subtasks

Fig. 2. The DAG of the dependent task

Each user's computing task is a dependent task and can be represented by a DAG. To assist UAVs to perform joint task computations, we combine the DAG for all the user tasks into one DAG, which is completed by the DC and realized through task migration and reuse technology (Yang et al., 2021; Liu et al., 2023b). First, two virtual tasks are introduced (the enter task and exit task with weight 0). Furthermore, we simplify the DAG graph to consider the repeatability between subtasks. There are two user subtasks A and B (Fig. 2). In Fig. 2(a), if subtasks A4 and B2 are the same, there is no need to compute repeatedly once the temporary computing power pool is built, where the computing results and computing resources are shared, so the DAG can be further simplified to Fig. 2(b).

Whether a subtask $T_{m,i}$ can start computation is constrained by the dependency relationship between subtasks, i.e. it must wait for its predecessor subtask (only the first level precursor subtask is considered) to complete before computation. For ease of designation, we refer to the predecessor subtask of $T_{m,i}$ as $T_{pre(m,i)}$. For example, the A5 subtask cannot be executed until the A2 subtask is completed, so we call $T_{A,2}$ the precursor subtask of the subtask $T_{A,5}$, i.e., $T_{pre(A,5)} = T_{A,2}$. There are dependent data among interdependent subtasks, i.e. the output of the previous subtask is the input of the current subtask, and its data quantity is denoted as $M_{pre(m,i)}$. Therefore, the output of $T_{A,2}$ equals the input of $T_{A,5}$, which can also be denoted as $M_{pre(A,5)}$.

If the subtask $T_{m,i}$ is assigned to the UAV n, it is denoted that $\chi_{m,i,n} = 1$, otherwise it is 0, and the sets of $\chi_{m,i,n}$ are called as X. If $T_{m,i}$ and $T_{pre(m,i)}$ are assigned to the same UAV for processing, then the dependent data between the two tasks do not need to be transmitted, otherwise, the transmission time of intermediate data needs to be considered. Because the UAV is in a hovering state at this time, the energy consumed by hovering should be considered. The transfer time of the dependent data can be obtained according to Eq. (9). The data size of dependent data, $M_{pre(m,j)}$, is extremely small compared to that of the input of users' tasks, so the energy consumption of communication between UAVs is negligible (Guo et al., 2023; Nguyen et al., 2023).

3.4. Energy consumption model

3.4.1. Flight energy consumption

Kim et al. (2023) and Liu et al. (2023a), in the kth time slot, denoted the flight energy consumption of the UAV n is as $E_n^f[k]$, and $E_n^f[k] = 0.5 M_n v^2[k]$. M_n is the weight of the UAV n. Therefore, the total flight energy consumption of the UAV n can be calculated by Eq. (10), where CFT_n is the time slot in which the UAV n finished collection tasks.

$$E_n^f = \sum_{k=1}^{CFT_n} 0.5 M_n v^2[k] \times \Delta T \tag{10}$$

3.4.2. Hovering energy consumption

The hovering energy consumption of UAV n can be calculated by $p_n^{hov} = \dfrac{\eta\sqrt{\eta}}{\varphi_n \sqrt{2\pi q r^2 \vartheta}}$ (He et al., 2023; Nguyen et al., 2023), where η is a value proportional to the mass of the UAV, φ_n is the power efficiency of the UAV, q is the number of rotors belonging to a single UAV, r is the diameter of each rotor, and ϑ is the density of the air.

Hovering energy consumption consists of two main parts. One part is the generated when the UAV is waiting for other UAVs after collecting the task information, which is recorded as Eq. (11).

$$E_n^{h1} = p_n^{hov} \sum_{k=CFT_n}^{SSC} \Delta T \tag{11}$$

The second part occurs in the stage of computing, which is recorded as Eq. (12).

$$E_n^{h2} = p_n^{hov} \sum_{k=SSC}^{SFC} \Delta T \tag{12}$$

If the total hover energy consumption of the UAV n is noted as E_n^h, then

$$E_n^h = E_n^{h1} + E_n^{h2} \tag{13}$$

The moving energy consumption of the UAV n, E_n^{move}, is the sum of the flight and hovering energy consumption, so

$$E_n^{move} = E_n^f + E_n^h \tag{14}$$

3.5. Computing consumption model

Similar to the computing models of other studies, assuming that the maximum computing frequency of the UAV n is $f_n^{\max}$, the computing frequency in the kth time slot is represented as $f_n[k]$, and the sets of $f_n[k]$ are represented as F, C_n is the number of CPU cycles required by the UAV n to process 1 bit of data, so the time required to process 1 bit of data is $C_n / f_n[k]$ (Bai et al., 2022; Jang et al., 2023; Kim et al., 2023). As in most studies, we assumed that the computing energy consumed by the UAV per unit time is $\kappa_n f_n^3[k]$. κ_n is the effective switching capacitance of the CPU, so the energy consumed by computing 1 bit of data is $\kappa_n C_n f_n^2[k]$. Therefore, the energy consumed by the UAV n for computing the subtasks assigned to it can be expressed as in Eq. (15).

$$E_n^c = \sum_{k=SSC}^{SFC_n}\sum_{m=1}^{M}\sum_{i=1}^{s} \kappa_n f_n^{\,2}[k]\times f_n^{\max}\times C_m(i,n)\times\chi_{m,i,n}[k]\times\Delta T \tag{15}$$

where SSC represents the time slot in which the UAVs start computing, when all the UAVs have collected the task information and started the task linkage. SFC_n indicates the time slot in which the UAV n has completed computing all the assigned subtasks.

3.6. Problem formulation

Based on the idea of building a computing power pool, we investigated the problem P of minimizing the maximum weighted energy consumption by joint optimization of the UAV trajectory Q, communication resource allocation B, computing resource allocation F, UAV-user association A, and UAV-subtask association X.

$$\mathrm{P}: \min_{\mathrm{A},\mathrm{X},Q,B,F} \max(\gamma E_n^{move}+E_n^c), \forall n\in N$$

s.t. (1), (4), (5), (6)

$$\alpha_{m,n}\cdot\max\{t_m^I\}\le CFT_n\le SSC\le SFC_n\le T,\quad \forall m\in M, n\in N, \alpha_{m,n}=1 \tag{16}$$

$$\sum_{n=1}^{N}\alpha_{m,n}[k]=1, \forall m\in M, \forall k\in K \tag{17}$$

$$\sum_{n=1}^{N}\chi_{m,i,n}=1, \forall m\in M \tag{18}$$

$$E_n^{move}[k]+E_n^c[k]\le E_n^0, \forall n\in N, \forall k\in K \tag{19}$$

where γ is the weight coefficient of the moving energy consumption of UAV (Zhang et al., 2020). Eq. (1) is the flight constraint; Eq. (4) ensures that the UAV will not waste time on meaningless time; Eq. (5) limits the bandwidth resource constraints of UAVs; Eq. (6) ensures the integrity of the collected data; Eq. (16) is the causal constraint of time completion, i.e. the time that the UAV needs to ensure the associated user tasks are collected before the task joint computation begins; Eq. (17) ensures that each user's task information will be collected by only one UAV; Eq. (18) guarantees that each subtask will be computed on only one UAV; Eq. (19) is the energy consumption limit of the UAV.

The scheme also requires that the time for each UAV to collect task information is as consistent as possible, so is the computing process, to ensure the fairness of the temporary computing power pool. If the difference of collecting time is too large, it will cause the UAVs to wait too long for each other, consuming meaningless energy. Likewise, if the time gap of computing is too large, the UAV will exit the computing power pool early, and the available resources in the computing power pool will be reduced, which is not in line with the original intention of this scheme.

4. Proposed solution

The problem is NP-hard because it is a mixed-integer nonlinear programming (MINLP) problem, which includes both integer and continuous variables (Qin et al., 2021; Zhang and Chakareski, 2022). Considering that the moving of UAVs can be divided into two main stages, namely information collection and task linkage, we also divide the problem into two subproblems according to this clue. However, the final positions of the UAVs obtained from the collection stage are the input of carrying out the task linkage, which creates some problems for the solution process. Thereby, we propose a two-stage alternate optimization algorithm to tackle this problem, which combines the advantages of SCA and GA. Specifically, we split problem P into two subproblems, P1 and P2. Firstly, the UAV-user association A, the communication resource allocation B and the UAV trajectory Q are obtained by solving problem P1. Then the UAV-subtask association X and computational resource allocation F are obtained by solving P2. The results of P1 and P2 iterate over each other, and finally we obtain the optimal solution of the problem P. The detailed analysis is as follows.

4.1. Optimizing UAV-user association, communication resource allocation and UAV trajectory problem solving

In this section, we discuss the problem of jointly optimizing the UAV-user association A, the communication resource allocation B and the UAV trajectory Q to minimize the maximum energy consumed by the UAV when the UAV-subtask association X and the allocation of computational resources F are given. At this point, problem P can be converted to problem P1, which is described as follows:

$$\mathrm{P1}: \min_{\mathrm{A},Q,B} \max(\gamma E_n^{move}+E_n^c), \forall n\in N$$

s.t. (1), (4), (5), (6), (16), (17), (19)

There are two main difficulties in solving this problem. The first is the UAV-user association, which is an integer variable, and the second is that the problem is non-convex. Considering that the objective function is the energy consumption of one of the UAVs, and P1 can be converted to a single UAV energy minimization problem if the UAV-user association is given, we adopt an approach combining clustering and SCA to solve it.

Firstly, we use a clustering method with weights to pre-assign the UAV-user. Considering that the energy consumption and time spent by the UAV in collecting information are affected by the user's location and the size of the data, the clustering rule considers mainly these factors. We adopt the reciprocal distance between the user and the starting point as the weight of each user, and the K-Means ++ algorithm to initiate the center of mass (Bahmani et al., 2012; Hämäläinen et al., 2020). Then, when updating the center of mass, the weight is considered, so that the farther away from the starting point, the less user data are clustered (Ahmed et al., 2020).

The non-convexity of the problem is caused mainly by Eq. (4), where the variable $\log_2(1+snr_m^n[k])$ is non-convex with respect to $l_m^n[k]$. For ease of subsequent derivation, we denote $\log_2(1+snr_m^n[k])$ as $\eta_m^n[k]$, and use the SCA method to solve (Scutari et al., 2016). The specific derivations are as follows:

Since $\eta_m^n[k]$ is convex with respect to $\left|l_m^n[k]\right|^2$, its lower bound is obtained at the r th iteration:

$$\begin{aligned}\eta_m^n[k] &\geq \eta_m^{n,r}[k]-\Phi_m^n[k](\left\|W_m-q_n^r[k]\right\|^2-\left\|W_m-q_n[k]\right\|^2)\\ &\triangleq \xi_m^{n,r}[q[k]]\end{aligned} \tag{20}$$

where

$$\begin{aligned}\Phi_m^n[k]=\ln\frac{\sqrt{2}}{2}\cdot\alpha P g_0&\left|\sqrt{\frac{S}{S+1}}h_m^{LoS}[k]+\sqrt{\frac{1}{1+S}}h_m^{NLoS}[k]\right|^2\\ &/[(H^2+\left\|q^r[k]-W_m\right\|^2)Pg_0\\ &\cdot\left|\sqrt{\frac{S}{S+1}}h_m^{LoS}[k]+\sqrt{\frac{1}{1+S}}h_m^{NLoS}[k]\right|^2\\ &+(b_{n,m}^r[k]\sigma^2)(H^2+\left\|q^r[k]-W_m\right\|^2)^{1+\alpha/2}]\end{aligned} \tag{21}$$

$$\eta_m^{n,r}[k]=\log_2(1+\frac{P_m\left|\sqrt{\frac{S}{S+1}}h_m^{LoS}[k]+\sqrt{\frac{1}{1+S}}h_m^{NLoS}[k]\right|^2 g_0}{(b_{n,m}^r[k]\sigma^2)(H^2+\left\|w_m-q^r[k]\right\|^2)^{\alpha/2}}) \tag{22}$$

Accordingly, $R_m^{n,r}[k]=b_{n,m}^r[k]B\eta_m[k]$, which denotes the communication rate between the user and the UAV during the iteration process. $q_n^r[k]$ denotes the horizontal position of the UAV during the iteration process. $\Phi_m^n[k]$ can be obtained by the derivation of $\eta_m^n[k]$ on the variable $\left|l_m^n[k]\right|^2$.

We introduce slack variables $Z=\{z_m^n[k]\geq 0,\forall m\in M,n\in N,k\in K\}$ and $U=\{u_m^n[k]\geq 0,\forall m\in M,n\in N,k\in K\}$ here, then it can be inferred that

$$u_m^n[k]\leq \xi_m^{n,r}[q_n[k]] \tag{23}$$

$$b_{m,n}[k]u_m^n[k]\geq z_m^n[k] \tag{24}$$

Eq. (23) can be replaced by the equivalent convex difference function, $\frac{\left(b_{m,n}[k]+u_m^n[k]\right)^2-\left(b_{m,n}[k]-u_m^n[k]\right)^2}{4}$. At the r th iteration, $\frac{\left(b_{m,n}[k]+u_m^n[k]\right)^2}{4}$ is convex, so Eq.（24）can be transformed as:

$$\begin{aligned}&z_m^n[n]+\frac{\left(b_{m,m}[k]-u_m^n[k]\right)^2}{4}-\\ &\frac{\left(b_{m,n}^r[k]-u_m^{n,r}[k]\right)\left(b_{m,n}[k]-b_{m,n}^r[k]+u_m^n[k]-u_m^{n,r}[k]\right)}{2}\leq 0\end{aligned} \tag{25}$$

After the above analysis, we can transform P1 into a convex optimization problem. Therefore, we adopt Algorithm 1 to solve P1 (the algorithm flow is provided in the supplementary materials).

4.2. Optimizing UAV-subtask association, computational resource allocation problem solving

In this section, we discuss the problem of jointly optimizing the UAV-subtask association X and computational resource allocation F to minimize the maximum energy consumption of the UAV given the UAV-user association A, communication resource allocation B and UAV trajectory Q. At this point, the problem P can be described as P2, which is formulated as follows:

$$\begin{aligned}\text{P2}: &\min_{X,F}\ \max(\gamma E_n^{move}+E_n^c),\forall n\in N\\ &\text{s.t. (18), (19)}\end{aligned}$$

The size of subtasks in this scenario is large. A meta-heuristic algorithm has obvious advantages in solving large-scale solutions. Therefore, we adopt an improved GA to solve this problem. GA is a kind of meta-heuristic algorithm inspired by natural selection and genetic evolution. The basic idea is to simulate the process of natural evolution, natural selection and survival of the fittest. In the process of evolution, solutions with low fitness will be eliminated, and those with better fitness will further reproduce, and more alternative solutions closer to the optimal solution will be derived. However, the existing GA was not suitable for solving our problem. Therefore, we redesigned the relevant operators, including coding, population initialization, and fitness function, which are included in Algorithm 2 (provided in the supplementary materials).

Through the above analysis, we divided the original problem P into two sub-problems, P1 and P2, and adopted Algorithm 1 and Algorithm 2 respectively to iteratively obtain the optimal solution. The algorithm is described in Algorithm 3. The algorithm description and complexity analysis are provided in the supplementary materials.

5. Simulation results

We conducted a number of experimental simulations to analyze the performance and effectiveness of the proposed method. Since the energy consumption of the UAV can reflect the rationality of the trajectory of the UAV and other resource allocation to a certain extent, and the objective function is energy consumption, we focused on the analysis of energy consumption and execution time. Firstly, to validate the superior performance of the proposed algorithm, we compared it with four other algorithms and analyzed the corresponding changes in energy consumption and time, as well as the effect of computing pool construction as the number of users increases (the simulation results and analysis are provided in supplementary materials). Then, to measure the improvement of the proposed scheme in dealing with the trouble brought about by the subtasks with repeatability and dependency, we compared it with the method of repeating subtasks without merging and analyzed changes in time and energy consumption indices as the number of users and repetitions increased.

5.1. Parameter setting

Similar to previous studies (Samir et al., 2019; Fu et al., 2022; Wang et al., 2022b; Yang et al., 2022; Nguyen et al., 2023), we set an area of 1×1 km, with two UAVs with a weight of ,20 kg starting from horizontal coordinates (0,0) and (1000,1000) respectively. Referring to Nguyen et al. (2023), the topology of user tasks is generated layer by layer. The number of subtasks in each layer is normally distributed. The data size of user subtasks is randomly generated, with a value between 1 and 8 Mbit and the value of dependent data is between 150 and 300 kbit. The simulation parameters are provided in the supplementary materials.

To test the performance of the proposed algorithm comprehensively, we increased the number of users from 10 to 50, and compared it with another four algorithms:

Queue-based algorithm: This algorithm is derived from the tasks dealing method of Yang et al. (2022), which proposes a task queue model and adopts a Lyapunov-based online algorithm to balance task queue stack length and energy consumption by the UAVs.

Regional-based algorithm: This algorithm is derived from Nguyen et al. (2023). The user area is divided geometrically and each UAV is responsible for the computing tasks in its respective area.

Distance-based algorithm: This algorithm is derived from Fu et al. (2022). The UAV-user association is obtained by referring to the distance-based clustering method. On this basis, resource allocation and task scheduling are further optimized to balance the energy consumption of the UAV.

Data volume-based algorithm: This algorithm refers to the method of Wang et al. (2022b) for balancing the data volume of the UAV, and we further optimize the offloading strategy.

5.2. Simulation results and analysis

5.2.1. The effect of the number of users on energy consumption, time and computing pool construction

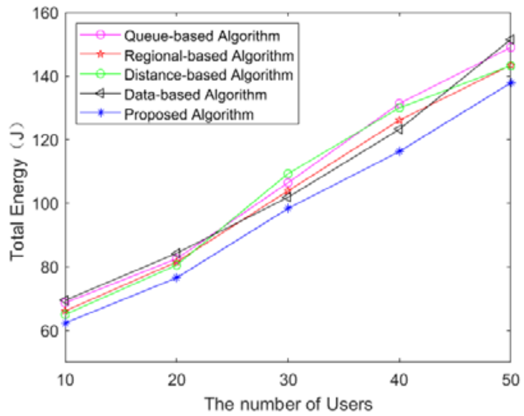


Fig. 3. Total energy consumption

We analyzed the energy consumption index of the UAVs. The overall result of all algorithms was that the total energy consumption increases with the increase in the number of users (Fig. 3). No matter how many users there were, compared with other algorithms, the energy consumption of our proposed algorithm was the lowest, highlighting its advantage in terms of energy consumption. As the number of users increased, the performance of other algorithms was unstable.

The distance-based algorithm expended slightly less energy than the other algorithms when the number of users was small, but did not maintain this trend when the number of users increased to 30. This was because the distance-based algorithm largely depends on the geographical distribution of users. Although the flight energy consumption of UAVs can be optimized, it ignores the factor of data volume. Therefore, with an increasing number of users, the amount of user task data associated with the corresponding UAV gradually becomes unbalanced, which leads to an increase in the cost of the mutual waiting stage and the computing stage. This is also reflected in the UAV's energy consumption. The total energy consumption of queue-based algorithms was the highest overall. This algorithm aims to provide the fastest response service on demand and is driven by the time users propose tasks. However, it does not consider the geographical location and data volume between users, which will lead to the phenomenon of overlapping of the trajectory of a single UAV. The trajectory of the UAV is not optimized and the flight energy loss of re-entry is large. Especially as the number of users grows, the performance will only deteriorate. The region-based algorithm is greatly affected by the distribution density of users. If the regional distribution is not balanced, the UAV energy consumption and time distribution will also be unbalanced. A UAV with fewer tasks will have a longer waiting time, and a UAV with more tasks needs to accelerate the completion of tasks, resulting in increased energy consumption. This is not conducive to the construction of the computing power pool in terms of time and energy consumption. Due to the average user-related data we used, the algorithm's performance in terms of energy consumption was mediocre. The moving energy consumption of the data-based algorithm was high, mainly because the distance factor was not considered, which prevents further optimization of the flight trajectory, and increases cost in the flight stage. The performance of indicators was similar to that of the queue-based algorithm.

We analyzed the performance of the algorithm from the aspect of the time index. With an increase of the number of users, our proposed algorithm was the shortest in terms of total time, collection time, computing time and mutual waiting time, and performed well in terms of time index (Fig. 4). After analysis, we concluded that the biggest difference between the various algorithms is the time the UAV waits. The tasks undertaken by the UAVs of the comparison algorithm are not balanced, resulting in some spending too much time waiting. The difference in time spent by each algorithm in the data collection and computing stages is not large, but our proposed algorithm has obvious advantages in the mutual waiting time of the UAVs (Fig. 5).

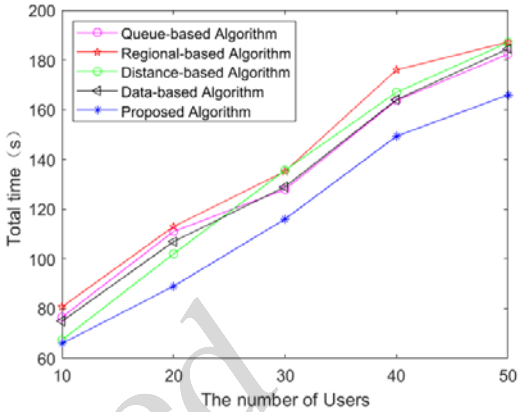


Fig. 4. Total time consumption

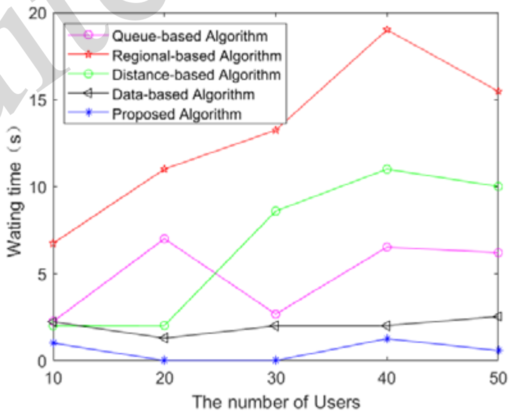


Fig. 5. UAV waiting time for different users

Taking the region-based algorithm as an example, in the case of different users, the data collection time and computing time of the UAV are less than those of other algorithms, but the corresponding waiting time is much higher, resulting in an increase in the total time. This is because the distribution of tasks among multiple UAVs is unbalanced, resulting in one of the UAVs finishing data collection prematurely, spending more time and energy consumption in waiting, resulting in the lowest workload of the UAVs but consuming the most energy. Moreover, the region-based algorithm depends largely on the density of users and the size of the data. Based on the original long waiting time, if the newly added users happen to belong to the jurisdiction of the UAV, the UAV can use its idle time to collect these new data, so the waiting time will be reduced, which is why the waiting time of the algorithm is reduced when the number of users is 50.

The collection time of queue-based algorithms is relatively long due to the sacrifice of energy consumption. But its waiting time is relatively random and is affected mainly by the order of user tasks. The collection time of the distance-based algorithm is shorter than that of the region-based algorithm, but the waiting time is longer because the data volume is not considered. The waiting time of the data-based algorithm is relatively short and close to that of our proposed algorithm, but its collection time is longer. Due to the lack of consideration of distance, the flight trajectory of the UAV is not further optimized, and the time cost eventually increases.

In summary, our proposed algorithm can ensure the shortest average waiting time while balancing energy consumption, effectively ensure the fairness of the construction of the UAV computing power pool, and lay the foundation for task linkage.

5.2.2. Effect of repetition rate

The main aim of experiments described in this section was to verify the advantages and the factors that affect the computing power pool. Considering a sample of user tasks with 12 subtasks being repeated, we designed two experiments to analyze the changes in energy consumption and time as the numbers of users and repetitions increased. The number of repetitions is defined as the average number of times a subtask is reused (only subtasks that are adopted two or more times were considered). The specific analysis is as follows.

(1) Changes in energy consumption and time as the number of users increases

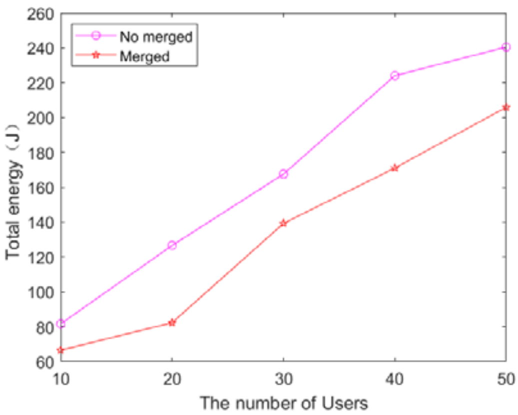


Fig. 6. Changes in total energy consumption of different users

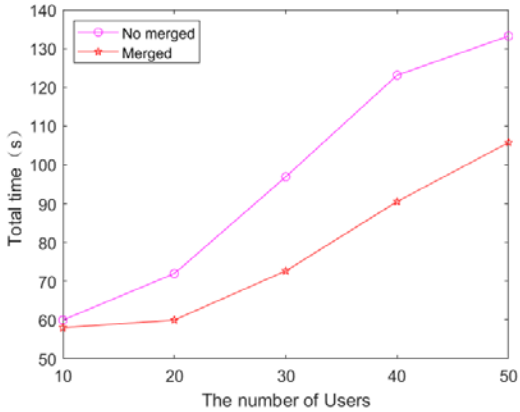


Fig. 7 Changes in total time consumption of different users

As the number of users increases, the method of merging the same subtasks reduces total computing time by 18.41% and energy consumption by 21.68% on average (Figs 6 and 7, respectively). In terms of total time, with the increase of the number of users, the time that can be saved increases, because with the increase of the number of users, the repetition rate of the corresponding subtasks will increase, and the time for processing this part is reduced.

(2) Changes in energy consumption and time with an increasing average number of repetitions

The time and energy consumption of both the merged and unmerged subtasks methods tend to increase as the average number of repetitions increases, because increasing the average number of repetitions means that more users need to participate (Figs 8 and 9). The time and energy consumption after merging the same subtasks is significantly reduced, and the change in computing time is smoother. After repeated experiments, we obtained the results shown in Table 1. With the increase of the average number of repetitions, the average time growth rate was 15.33%, which was 11.57 percentage points lower than that without merging. The average growth rate of energy consumption was slightly higher, and the change of energy consumption was not as smooth as that of time. The main reason is that with an increase of the number of users, the subtasks with no repetition will also increase, and the computing energy consumption of these subtasks cannot be ignored. However, the energy consumption consumed by the merging method is less than that without merging, with an average reduction of 57.63 J.

In conclusion, the scheme of constructing a provisional computing power pool can effectively solve the dependency and repeatability problems of subtasks, reduce the computing energy and time consumption, and efficiently manage the limited resources of UAVs.

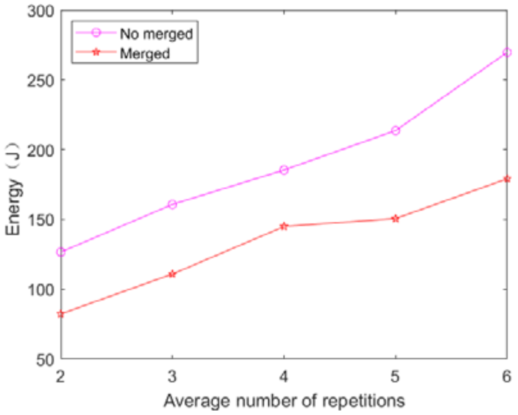


Fig. 8 The change of energy consumption under different average repetitions

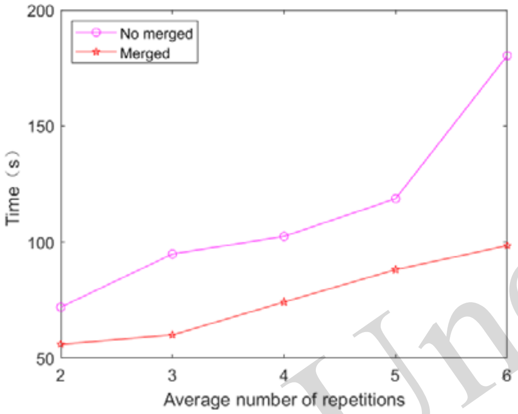


Fig. 9 The change of time consumption under different average repetitions

Table 1 The average growth rate of time and energy consumption with and without subtask merging

Average growth rate	No Merged	Merged
Energy consumption	20.93%	22.08%
Time consumption	26.90%	15.33%

6. Conclusions

We studied the redundancy in resource allocation by UAVs caused by repetitive subtasks of dependent tasks and propose a U-MEC paradigm based on a computing power pool. Furthermore, considering the fairness of the construction of the computing power pool, the energy consumption of the UAV participating in the construction is balanced by jointly optimizing the UAV-user association, communication resources, computing resources, UAV-subtask association, and UAV trajectory, and an alternate optimization algorithm based on a SCA and improved GA is proposed to solve the problem. The sharing of computing power and task information by UAVs in the computing power pool can effectively circumvent the repetition of identical subtasks, enhance the flexibility of task scheduling, and optimize the efficiency of resource allocation. Results from experimental simulations showed that the proposed method can effectively enhance the efficiency of UAV task processing. On average, the execution time was reduced by 18.41% and the energy consumption by 21.68%. Furthermore, the proposed method shows remarkable efficacy in reducing task processing time and energy consumption as the number of subtask repeats increases. Compared to the scenario in which no repetitive tasks are processed, the execution time increases at a gradual rate, with a reduction in growth of 11.57%. Energy consumption is also reduced by 57.63 on average.

In future research, we intend to facilitate the participation of UAVs in a computing power pool. This would enable the UAVs to autonomously determine whether to contribute their own computing power and, if so, to assess the quantity of computing power to provide based on the accumulated task information within the computing power pool. During the implementation phase, we will focus on addressing the issues of rule-making and task heterogeneity in the construction of the computing power pool.

Contributors

Xuebin LAI designed the research and drafted the manuscript. Yan GUO and Ming HE revised and finalized the paper. Hao YUAN and Wei LI helped debug code. Xiaonan CUI helped organize the manuscript.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

Abrar M, Ajmal U, Almohaimeed ZM, et al., 2021. Energy efficient UAV-enabled mobile edge computing for IoT devices: a review. *IEEE Access*, 9:127779-127798. https://doi.org/10.1109/ACCESS.2021.3112104

Ahmed M, Seraj R, Islam SMS, 2020. The *k*-means algorithm: a comprehensive survey and performance evaluation. *Electronics*, 9(8):1295. https://doi.org/10.3390/electronics9081295

Alliance of Industrial Internet, 2023. Citing electronic sources of information. Available from https://www.aii-alliance.org/uploads/1/20230703/7194a30af14f25d771980cffe93e525f.pdf [Accessed on May. 21,

2024].
Bahmani B, Moseley B, Vattani A, et al., 2012. Scalable K-means++. *Proceedings VLDB Endowment*, 5(7):622-633.
Bai ZY, Lin YF, Cao Y, et al., 2022. Delay-aware cooperative task offloading for multi-UAV enabled edge-cloud computing. *IEEE Trans Mob Comput*, 23(2):1034-1049. https://doi.org/10.1109/TMC.2022.3232375
Dai MH, Huang N, Wu Y, et al., 2023. Latency minimization oriented hybrid offshore and aerial-based multi-access computation offloading for marine communication networks. *IEEE Trans Commun*, 71(11): 6482-6498. https://doi.org/10.1109/TCOMM.2023.3306581
Fu S, Guo XH, Fang F, et al., 2023. Towards energy-efficient data collection by unmanned aerial vehicle base station with NOMA for emergency communications in IoT. *IEEE Trans Veh Technol*, 72(1):1211-1223. https://doi.org/10.1109/TVT.2022.3206213
Guo HZ, Wang YT, Liu JJ, et al., 2024. Multi-UAV cooperative task offloading and resource allocation in 5G advanced and beyond. *IEEE Trans Wirel Commun*, 23(1):347-359. https://doi.org/10.1109/TWC.2023.3277801
Hämäläinen J, Kärkkäinen T, Rossi T, 2020. Improving scalable K-means++. *Algorithms*, 14(1):6. https://doi.org/10.3390/a14010006
He YJ, Gan YH, Cui HX, et al., 2023. Fairness-based 3-D multi-UAV trajectory optimization in multi-UAV-assisted MEC system. *IEEE Internet Things J*, 10(13):11383-11395. https://doi.org/10.1109/JIOT.2023.3241087
Hu QY, Cai YL, Yu GD, et al., 2019. Joint offloading and trajectory design for UAV-enabled mobile edge computing systems. *IEEE Internet Things J*, 6(2):1879-1892. https://doi.org/10.1109/JIOT.2018.2878876
Hu XY, Wong KK, Zhang YY, 2020. Wireless-powered edge computing with cooperative UAV: task, time scheduling and trajectory design. *IEEE Trans Wirel Commun*, 19(12):8083-8098. https://doi.org/10.1109/TWC.2020.3019097
Hu ZZ, Zeng FZ, Xiao Z, et al., 2021. Computation efficiency maximization and QoE-provisioning in UAV-enabled MEC communication systems. *IEEE Trans Netw Sci Eng*, 8(2):1630-1645. https://doi.org/10.1109/TNSE.2021.3068123
Huang XM, Peng CD, Wu Y, et al., 2023. Joint interdependent task scheduling and energy balancing for multi-UAV-enabled aerial edge computing: a multiobjective optimization approach. *IEEE Internet Things J*, 10(23): 20368-20382. https://doi.org/10.1109/JIOT.2023.3288379
Jang Y, Jeong S, Kang J, 2024. Energy-efficient vehicular edge computing with one-by-one access scheme. *IEEE Wirel Commun Lett*, 13(1): 39-43. https://doi.org/10.1109/LWC.2023.3318632
Jia RN, Zhao K, Wei XL, et al., 2023. Joint trajectory planning, service function deploying, and DAG task scheduling in UAV-empowered edge computing. *Drones*, 7(7):443. https://doi.org/10.3390/drones7070443
Kato N, Fadlullah ZM, Tang FX, et al., 2019. Optimizing space-air-ground integrated networks by artificial intelligence. *IEEE Wirel Commun*, 26(4):140-147. https://doi.org/10.1109/MWC.2018.1800365
Khalid R, Shah Z, Naeem M, et al., 2024. Computational efficiency maximization for UAV-assisted MEC networks with energy harvesting in disaster scenarios. *IEEE Internet Things J*, 11(5): 9004-9018. https://doi.org/10.1109/JIOT.2023.3322001
Khuwaja AA, Chen YF, Zhao N, et al., 2018. A survey of channel modeling for UAV communications. *IEEE Commun Surv Tutorials*, 20(4):2804-2821. https://doi.org/10.1109/COMST.2018.2856587
Kim D, Jeong S, Kang J, 2024. Energy-efficient secure offloading system designed via UAV-mounted intelligent reflecting surface for resilience enhancement. *IEEE Internet Things J*, 11(3): 3768-3778. https://doi.org/10.1109/JIOT.2023.3336919
Li J, Pan Y, Xia YC, et al., 2024. Optimizing dag scheduling and deployment for IoT data analysis services in the multi-UAV mobile edge computing system. *Wireless Netw*, 30(7):6465-6479. https://doi.org/10.1007/s11276-023-03451-0
Lin H, Zeadally S, Chen ZH, et al., 2020. A survey on computation offloading modeling for edge computing. *J Netw Comput Appl*, 169:102781. https://doi.org/10.1016/j.jnca.2020.102781
Liu CT, Guo Y, Li N, et al., 2022. Aoi-minimal task assignment and trajectory optimization in multi-UAV-assisted IoT networks. *IEEE Internet Things J*, 9(21):21777-21791. https://doi.org/10.1109/JIOT.2022.3182160
Liu CT, Guo Y, Li N, et al., 2023. Satisfaction-driven cooperative trajectory optimisation for multi-UAV-assisted mobile edge computing. *Int J Ad Hoc Ubiquitous Comput*, 43(1):29-40. https://doi.org/10.1504/IJAHUC.2023.130980
Liu SM, Yu Y, Lian X, et al., 2023. Dependent task scheduling and offloading for minimizing deadline violation ratio in mobile edge computing networks. *IEEE J Sel Areas Commun*, 41(2):538-554. https://doi.org/10.1109/JSAC.2022.3233532
Luan QJ, Cui HY, Zhang LF, et al., 2022. A hierarchical hybrid subtask scheduling algorithm in UAV-assisted MEC emergency network. *IEEE Internet Things J*, 9(14):12737-12753. https://doi.org/10.1109/JIOT.2021.3138263
Mei HB, Yang K, Liu Q, et al., 2020. Joint trajectory-resource optimization in UAV-enabled edge-cloud system with virtualized mobile clone. *IEEE Internet Things J*, 7(7):5906-5921. https://doi.org/10.1109/jiot.2019.2952677
Michailidis ET, Maliatsos K, Skoutas DN, et al., 2022. Secure

UAV-aided mobile edge computing for IoT: a review. *IEEE Access*, 10:86353-86383. https://doi.org/10.1109/ACCESS.2022.3199408

Nasir AA, 2021. Latency optimization of UAV-enabled MEC system for virtual reality applications under rician fading channels. *IEEE Wirel Commun Lett*, 10(8):1633-1637. https://doi.org/10.1109/LWC.2021.3075762

Nguyen LX, Tun YK, Dang TN, et al., 2023. Dependency tasks offloading and communication resource allocation in collaborative UAV networks: a metaheuristic approach. *IEEE Internet Things J*, 10(10): 9062-9076. https://doi.org/10.1109/JIOT.2022.3233667

Ning ZL, Hu H, Wang XJ, et al., 2023. Mobile edge computing and machine learning in the internet of unmanned aerial vehicles: a survey. *ACM Comput Surv*, 56(1):13. https://doi.org/10.1145/3604933

Pranoto H, Saputra PP, Sadekh M, et al., 2023. Augmented reality navigation application to promote tourism to local state attraction “lawang sewu”. *Procedia Comput Sci*, 216:757-764. https://doi.org/10.1016/j.procs.2022.12.193

Qin XT, Song ZY, Hou TW, et al., 2023. Joint optimization of resource allocation, phase shift, and UAV trajectory for energy-efficient RIS-assisted UAV-enabled MEC systems. *IEEE Trans Green Commun Netw*, 7(4):1778-1792. https://doi.org/10.1109/TGCN.2023.3287604

Qin Z, Wang H, Wei ZH, et al., 2021. Task selection and scheduling in UAV-enabled MEC for reconnaissance with time-varying priorities. *IEEE Internet Things J*, 8(24):17290-17307. https://doi.org/10.1109/JIOT.2021.3078746

Radio Management Institute of the SEDIC Intelligence Centre, 2020. Citing electronic sources of information. Available from https://media.baogao.com/uploads/soft/2020/200327/20032GJP1.pdf [Accessed on May. 23, 2024].

Sahni Y, Cao JN, Yang L, et al., 2021. Multihop offloading of multiple DAG tasks in collaborative edge computing. *IEEE Internet Things J*, 8(6):4893-4905. https://doi.org/10.1109/JIOT.2020.3030926

Samir M, Sharafeddine S, Assi CM, et al., 2020. UAV trajectory planning for data collection from time-constrained IoT devices. *IEEE Trans Wirel Commun*, 19(1):34-46. https://doi.org/10.1109/TWC.2019.2940447

Scutari G, Facchinei F, Lampariello L, 2017. Parallel and distributed methods for constrained nonconvex optimization—part I: theory. *IEEE Trans Signal Process*, 65(8):1929-1944. https://doi.org/10.1109/TSP.2016.2637317

Tian J, Wang D, Zhang HX, et al., 2023. Service satisfaction-oriented task offloading and UAV scheduling in UAV-enabled MEC networks. *IEEE Trans Wirel Commun*, 22(12):8949-8964. https://doi.org/10.1109/TWC.2023.3267330

Wang LJ, Zhou Q, Shen Y, 2023. Computation efficiency maximization for UAV-assisted relaying and MEC networks in urban environment. *IEEE Trans Green Commun Netw*, 7(2): 565-578. https://doi.org/10.1109/TGCN.2022.3222398

Wang ZQ, Rong HG, Jiang HB, et al., 2022. A load-balanced and energy-efficient navigation scheme for UAV-mounted mobile edge computing. *IEEE Trans Netw Sci Eng*, 9(5):3659-3674. https://doi.org/10.1109/TNSE.2022.3188670

Wu GX, Miao YM, Zhang Y, et al., 2020. Energy efficient for UAV-enabled mobile edge computing networks: intelligent task prediction and offloading. *Comput Commun*, 150:556-562. https://doi.org/10.1016/j.comcom.2019.11.037

Xu B, Kuang ZF, Gao J, et al., 2023. Joint offloading decision and trajectory design for UAV-enabled edge computing with task dependency. *IEEE Trans Wirel Commun*, 22(8): 5043-5055. https://doi.org/10.1109/TWC.2022.3231408

Yan J, Bi SZ, Zhang YJ, et al., 2020a. Optimal task offloading and resource allocation in mobile-edge computing with inter-user task dependency. *IEEE Trans Wirel Commun*, 19(1):235-250. https://doi.org/10.1109/TWC.2019.2943563

Yan J, Bi SZ, Zhang YJA, 2020b. Offloading and resource allocation with general task graph in mobile edge computing: a deep reinforcement learning approach. *IEEE Trans Wirel Commun*, 19(8):5404-5419. https://doi.org/10.1109/TWC.2020.2993071

Yang XM, Luo H, Sun Y, et al., 2021. Coalitional game-based cooperative computation offloading in MEC for reusable tasks. *IEEE Internet Things J*, 8(16):12968-12982. https://doi.org/10.1109/JIOT.2021.3064186

Yang ZY, Bi SZ, Zhang YJA, 2022. Online trajectory and resource optimization for stochastic UAV-enabled MEC systems. *IEEE Trans Wirel Commun*, 21(7):5629-5643. https://doi.org/10.1109/TWC.2022.3142365

Yuan WY, Chang DN, Han T, 2023. A context-aware smart product-service system development approach and application case. *Comput Ind Eng*, 183:109468. https://doi.org/10.1016/j.cie.2023.109468

Zhai ZY, Dai XH, Duo B, et al., 2022. Energy-efficient UAV-mounted RIS assisted mobile edge computing. *IEEE Wirel Commun Lett*, 11(12):2507-2511. https://doi.org/10.1109/LWC.2022.3206587

Zhang L, Chakareski J, 2022. UAV-assisted edge computing and streaming for wireless virtual reality: analysis, algorithm design, and performance guarantees. *IEEE Trans Veh Technol*, 71(3):3267-3275. https://doi.org/10.1109/TVT.2022.3142169

Zhang PY, Wang C, Jiang CX, et al., 2021. UAV-assisted multi-access edge computing: technologies and challenges. *IEEE Internet Things Mag*, 4(4):12-17. https://doi.org/10.1109/IOTM.001.2100092

Zhang XC, Zhang J, Xiong J, et al., 2020. Energy-efficient multi-UAV-enabled multiaccess edge computing incorporating NOMA. *IEEE Internet Things J*, 7(6):5613-5627. https://doi.org/10.1109/JIOT.2020.2980035

Zhao TT, Li F, He LJ, 2024. Secure video offloading in multi-UAV-enabled MEC networks: a deep reinforcement learning approach. *IEEE Internet Things J*, 11(2): 2950-2963. https://doi.org/10.1109/JIOT.2023.3296613