• **RESEARCH PAPER** •

Joint task scheduling and multi-UAV deployment for aerial computing in emergency communication networks

Tiankui ZHANG[1], Chaobin CHEN[1], Yu XU[1], Jonathan LOO[2] & Wenjun XU[3]*

[1]*School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing* 100876*, China;*
[2]*School of Computing and Engineering, University of West London, London* W5 5RF*, UK;*
[3]*School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing* 100876*, China*

Received 1 August 2022/Revised 30 October 2022/Accepted 26 December 2022/Published online 28 August 2023

Abstract This article studies mobile edge computing technologies enabled by unmanned aerial vehicles (UAVs) in disasters. First, considering that the ground servers may be damaged in emergency scenarios, we proposed an air-ground cooperation architecture based on ad-hoc UAV networks. We defined the system cost as the weighted sum of task delay and energy consumption because of different delay sensitivity and energy sensitivity tasks in emergency communication networks. Then, we formulated the system cost-minimization problem of task scheduling and multi-UAV deployments. To solve the proposed mixed integer nonlinear programming problem, we decomposed it to two sub-problems that were solved by proposing a swap matching-based task scheduling sub-algorithm and a successive convex approximation-based multi-UAV deployment sub-algorithm. Accordingly, we propose a joint optimization algorithm by iterating the two sub-algorithms to obtain a low complexity sub-optimal solution. Finally, the simulation results show that (i) the proposed algorithm converges in several iterations, and (ii) compared with the benchmark algorithms, the proposed algorithm has better performance of reducing task delay and energy consumption and achieves a good trade-off between them for diverse tasks.

Keywords emergency communication, mobile edge computing, swap matching algorithm, unmanned aerial vehicle

Citation Zhang T K, Chen C B, Xu Y, et al. Joint task scheduling and multi-UAV deployment for aerial computing in emergency communication networks. Sci China Inf Sci, 2023, 66(9): 192303, https://doi.org/10.1007/s11432-022-3667-3

1 Introduction

Nowadays, various natural disasters, such as tsunamis, earthquakes, hurricanes, wildfires, and floods, have caused great damage. In the past 30 years, material losses caused by disasters worldwide have increased by about 100%–150% [1]. When disasters happen, communication infrastructures, such as the base stations (BSs) in cellular networks, fail to provide communication services in emergency scenarios.

An unmanned aerial vehicle (UAV) is a self-contained device that is controlled by specific control and non-payload communications (CNPC) links [2]. As a popular aircraft, UAVs are currently used in many fields [3–5] and have played an important role in emergency scenarios [6]. Given the flexibility of UAVs, they are widely used in disaster relief in the field such as communication recovery, rescue target identification, and aerial mapping of the affected terrain [7, 8].

Ground terminals including various sensor devices generally have limited computing power, which will take a long task execution delay when the computing task burden is heavy. Mobile edge computing (MEC) is a promising technology to solve this problem [9]. MEC is an emerging technology that provides information technology service environment and cloud computing capability at the edge of mobile networks [10]. Accordingly, terminal devices can offload the computing task to the edge server, thereby reducing service delays for users and alleviating network congestion [11]. This will allow tasks with high real-time requirements to be finished quickly, including some specific tasks such as augmented reality and target recognition [12, 13]. However, in traditional MEC networks, the edge server is usually embedded in the ground BS. Damage in ground communication infrastructure makes the traditional MEC networks invalid in emergency scenarios. Thus, aerial computing has drawn extensive attention. It is a new technology that further integrates air radio access network and edge computing through UAVs and other flight equipment [14]. Aerial computing is a part of the UAV-enabled MEC system. Unlike the UAV-assisted MEC system, it generally does not have ground infrastructure. By using the computing ability of edge servers carried by UAVs, aerial computing uses UAV as effective edge nodes to help ground terminals perform task uninstallation calculations [15]. Authors [16] proposed an air-ground collaborative architecture based on UAV, where ground terminals can offload their computing tasks to UAV nodes according to the input data of different tasks.

In emergency relief, better services are necessary for some specific tasks that need assistance of aerial computing. When disasters occur, sensors and cameras carried by the ground terminal are usually used to perform search and rescue tasks. In this case, ground terminals can upload target-area information to the UAV node through MEC networks. It will make more accurate and quick decisions by the high-speed computational capability of UAVs. However, to provide better services, each UAV in emergency scenarios should hover over with suitable velocity and distance [17]. Moreover, owing to the diversity of applications in emergency scenarios, we should further consider tasks with different requirements, including delay-sensitive and energy-sensitive tasks. Therefore, designing a cooperation mechanism between multiple UAVs considering task diversity in emergency scenarios is an urgent problem.

1.1 Motivations and related work

At present, different contributions have been carried out on the task-completion strategies of UAV networks from single UAV to multiple UAV collaboration. First, in emergency scenarios, most studies focus on how to use the UAV as a BS to achieve fast recovery of communications [18, 19]. These studies do not consider the task computing requirements of ground terminals in emergency scenarios. Second, existing contributions mainly solve the UAV edge computing-task scheduling and offloading under cloud edge collaboration architectures [20–24]. Gu et al. [20] presented an innovative task-completion strategy for multiple UAVs by combining animal group perception with edge resource scheduling and task assignment. The authors in [22] proposed an UAV-enabled MEC system involving interactions among Internet of Things (IoT) devices, UAVs, and ground macro BSs. In [23], the UAV plays the role of an MEC server in the IoT, so that computing tasks can be performed on local devices, UAV, or ground BSs. To minimize the weighted system cost of delay and energy consumption, the authors in [24] proposed a scheme based on game theory to find the optimal solution under the constraints of offloading decision and resource competition. In the above studies, the aerial computing architecture is based on the cloud edge collaboration architecture. Moreover, it relies on the high-speed computing capacity of the ground cloud server. Therefore, a good communication link between ground cloud servers and UAV nodes is necessary. However, in the post-earthquake relief or other emergency scenarios, the communication infrastructure maybe damaged, resulting in the inability to transmit task information from the UAVs to the ground cloud servers.

In addition, some studies examined trajectory optimization in air-ground cooperation architecture [25–30]. Under task constraints and information causality constraints, Hu et al. [25] proposed an alternating optimization algorithm to minimize the weighted total energy consumption of UAVs and terminals. A multi-agent deep reinforcement learning-based trajectory control algorithm was also proposed to manage task scheduling and the trajectory of each UAV [26]. Ning et al. [27] considered trajectory and task-scheduling optimization for 5G-enabled UAV-to-community offloading system. They developed an average throughput maximization-based auction algorithm and a dynamic task admission algorithm to maximize the throughput. Sun et al. [28] examined the joint design of the 3D aerial trajectory and wireless resource allocation in solar-powered MC-UAV communication systems. Cai et al. [29] investigated the trajectory and resource-allocation design for downlink energy-efficient secure UAV communication systems. By jointly optimizing the trajectory of UAV, transmit beamforming, and phase shift of intelligent reflecting surface (IRS), Pang et al. [30] investigated a secure transmission design for an IRS-assisted UAV network in the presence of an eavesdropper. In these contributions of aerial computing, the trajectory optimization of UAVs is generally determined in advance by task information and will not be adjusted with the change in user mobility. Besides, considering some energy-sensitive applications in emergency communications, the energy is quite valuable for UAV nodes and ground terminals. However, most optimization problems in existing task scheduling strategies rarely consider the current energy state of UAV nodes and ignore the network lifetime of UAV node formation.

1.2 Contributions and organization

Gaps in previous studies reflect the need for a distributed air-ground cooperation mechanism for diverse task requirements in emergency communications. In this study, we propose an air-ground cooperation architecture based on the ad-hoc network to jointly optimize task scheduling and UAV node deployment, which can minimize the weighted sum of task delay and energy consumption for mixed time-sensitive and energy-sensitive tasks. The main contributions are as follows.

- We proposed an air-ground cooperation architecture based on the ad-hoc network. In the proposed architecture, we defined the weighted sum of task delay and energy consumption as system cost. Specifically, the total task delay and energy consumption consist of three parts: the task execution consumption in ground terminals and UAV nodes and flight consumption in UAV nodes. On this basis, the system cost-minimization problem of task scheduling and multi-UAV deployment was proposed, in which the multi-UAV deployment included the optimization of the position and velocity of UAV nodes.
- Owing to the intractability of the formulated problem, we transformed the optimization problem into two subproblems: the task scheduling strategy sub-problem and the multi-UAV deployment sub-problem. Since the essence of the two sub-problems was the many-to-many matching problem and continuous nonconvex problem, we used the swap-matching algorithm and the successive convex approximation (SCA) algorithm to solve them. Accordingly, we proposed a joint optimization algorithm by iterating the two sub-algorithms.
- We demonstrated the convergence and effectiveness of the algorithm by numerical simulations. The simulation results showed that in the air-ground cooperation architecture based on the ad-hoc network, the proposed algorithm can greatly reduce task delay and energy consumption compared with benchmark algorithms. Furthermore, it prolonged the survival time of the network and achieved a good trade-off between task delay and energy consumption in diverse task scenarios.

The rest of this paper is organized as follows. Section 2 presents the system model and formulates the optimization problem. In Section 3, we decompose the problem into two subproblems and propose a joint optimization algorithm. In Section 4, the simulation results are provided. Finally, Section 5 presents the summary.

2 System model and problem formulation

To further reduce task delay and energy consumption, studies should focus on task scheduling and offloading technology of air-ground collaborative computing in emergency scenarios. Compared with the literature, we have made the following improvements to the system model: (1) we proposed an air-ground cooperation architecture based on the ad-hoc network to solve the problem of ground communication infrastructure damage and (2) the remaining energy weight was set in the system cost to solve the problem of uneven energy consumption of UAV nodes. The main notations in this paper are summarized in Table 1.

2.1 Design of air-ground cooperation architecture

As shown in Figure 1, in the air-ground coordination scenario with UAV nodes for emergency communications, multiple tasks are required from the ground terminals. To speed up task execution, task offloading will be provided by air computing. The UAV node network should further consider flight position, communication interference, and other factors in the air-ground cooperation architecture. Considering the damaged ground traditional communication link in the emergency scenario, UAV nodes must communicate with each other according to their computing power and cache resources. The mobile edge cloud system based on the ad-hoc network is a suitable MEC architecture for distributed collaborative computing. The virtual cloud based on the ad-hoc network does not need to rely on remote servers but can calculate and be completed by a group of mobile devices. A recent study showed that the MEC architecture based on the ad-hoc network can use the resources of adjacent mobile devices to achieve

Table 1 Notation descriptions

Description	Notation
Period time (s) of task collaboration strategy update	T_s
Number of UAV nodes	M
Number of ground terminal	K
Number of tasks	N
Amount of the input data (bits) of task n	I_n
Ground terminal generating task n	H_n
Cycles of computation required for task	ξ_n
Whether task n is offloaded to the UAV node m	x_{mn}
Whether task n is computed locally	$x_{M+1,n}$
Whether the ground terminal H_n is associated with the UAV node m	$\Theta(m,k)$
Computation capability (GHz) of the ground terminal k	f_k
Computation delay (s) for the local execution of task n	t_n^l
Energy consumption (J) of local execution of task n	e_n^l
Local task execution cost of task n	Z_n^l
Weight of task delay	α
Weight of energy consumption	β
Position vector of the UAV node m	$\boldsymbol{u}_m$
Position vector of the ground terminal k	$\boldsymbol{u}_k$
Whether ground terminal k' has communication interaction with other UAVs	$\Gamma(m,k')$
Transmission rate (Mbps) between the UAV node m and the ground terminal k	$r_{m,k}$
Transmission delay (s) of task n for offloading	$t_n^{\mathrm{c,tr}}$
Computing task capability (GHz) of the UAV node m	f_m
Computing delay (s) of task n executed at the UAV node m	$t_n^{\mathrm{c,ex}}$
Task delay (s) of task n for offloading	t_n^{c}
Energy consumption (J) of offloading task n	e_n^{c}
Cost of task n executed in the UAV node m	Z_n^c
Flight velocity (m/s) of the UAV node m	v_m
Flight time (s) of the UAV node m	t_m^F
Flight power (W) of the UAV node m	$p^F(v_m)$
Flight cost of the UAV node m	Z_m^F
Computing resources (G CPU cycles) of the UAV node m	c_m

better overall system performance [31]. It not only can complete the tasks with lower energy consumption but can also significantly reduce the computation delay. Therefore, we chose to establish the ad-hoc self-organizing network by the UAV nodes and built an air-ground cooperation architecture.

Based on the above design of air-ground cooperation architecture, the self-organizing network was assumed to be composed of multiple middle-high-altitude and long-duration rotary-wing UAV nodes. In this case, the UAV nodes will not be affected by the non-line-of-sight (NLoS) link and can provide communication and computing services for ground terminals. We adopted mobile ad-hoc network routing protocol, such as the zone routing protocol (ZRP), to deal with the problem of controlling information overhead and obstacles in communication [32]. Moreover, the UAV node was assumed to have two communication modules: a module for communication with ground terminals and another for communication with other UAVs. The spectral resources of these two communication modules were assumed to be orthogonal with each other and do not have inter-frequency interference. Therefore, communications between UAV nodes will not affect that between UAV nodes and ground terminals.

2.2 Task execution model

According to the characteristics of the air-ground cooperation architecture at the edge, we assume that the period time of task collaboration strategy update is T_S. There are M UAV nodes, the UAV node set is denoted by $\mathcal{M}=\{1,\ldots,M\}$. There are K ground terminals, and the set of which is denoted by $\mathcal{K}=\{1,\ldots,K\}$. In the system, it has N tasks, the set of which is denoted by $\mathcal{N}=\{1,\ldots,N\}$. The tasks can be executed locally at the ground terminal according to the task information, or offloaded to the UAV nodes. The information of task n includes (I_n,ξ_n,H_n), where I_n represents the amount of input data (in bits) of task n, H_n represents the ground terminal generating task n, i.e., $H_n\in\mathcal{K}$. Assume that

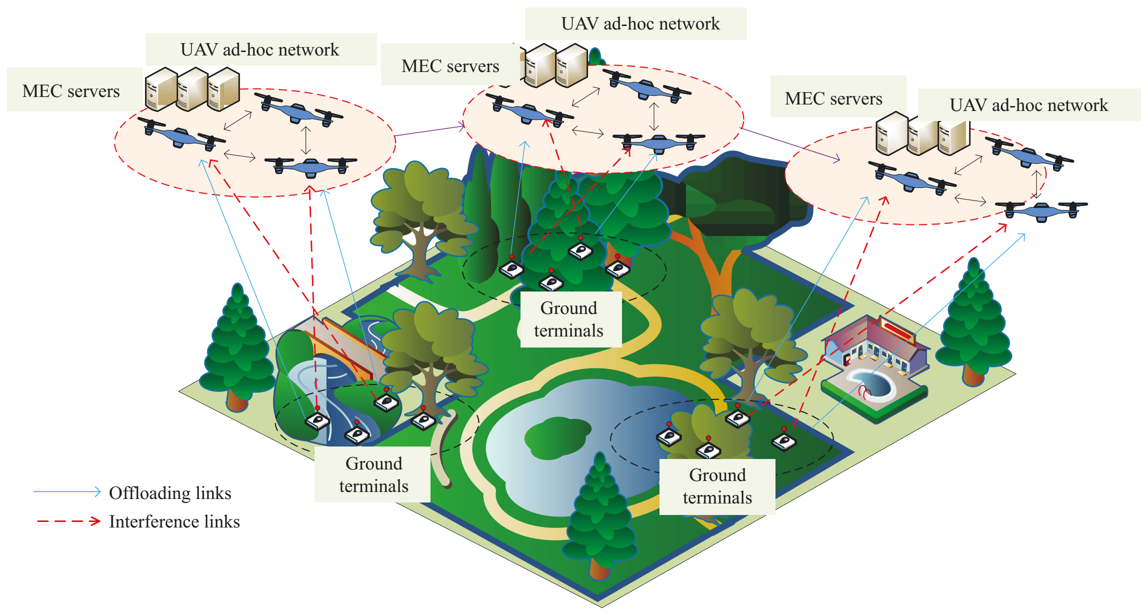

Figure 1 (Color online) Air-ground cooperation architecture based on the ad-hoc network.

the cycles of computation required for task, i.e., ξ_n, is proportional to the input data size. Thus, we have $\xi_n = \kappa I_n$, where the coefficient κ represents required CPU cycles for executing one bit of task data.

Define the task scheduling variable matrix of the UAV nodes as $\boldsymbol{x}$. The matrix element $x_{mn}, m \in \mathcal{M}, n \in \mathcal{N}$, is a binary variable, which indicates whether task n is offloaded to UAV node m. When $x_{mn} = 1$, the ground terminal generating task n is offloaded to UAV node m, otherwise $x_{mn} = 0$. In particular, let X_{M+1} indicates whether the ground terminal performs computing tasks locally. If $X_{M+1,n} = 1$, task n will perform calculation locally, otherwise $X_{M+1,n} = 0$. On this basis, the association function between the ground terminal and UAV node is set as $\Theta(m, k)$,

$$\Theta(m,k) = \begin{cases} 1, \ x_{mn} = 1, \ H_n = k, \\ 0, \ x_{mn} = 0, \ H_n = k. \end{cases} \tag{1}$$

If task n is scheduled to the m-th UAV node, $\Theta(m, k) = 1$ denotes the ground terminal H_n is associated with UAV node m.

According to the scheduling of the tasks, the task execution can be divided into two types: local task execution and UAV task execution. The two types of task execution models are as follows.

(1) Local task execution model. In the air-ground cooperation architecture, tasks can be executed locally at the ground terminals. Suppose that task n is generated in ground terminal k. Define the computation capability (in CPU cycles per second) of ground terminal k as f_k. the computation delay for local execution of task n is

$$t_n^l = \frac{\xi_n}{f_k} = \frac{\kappa I_n}{f_k}. \tag{2}$$

The energy consumption of local execution is described as $e_n^l = \mu_k t_n^l$, where μ_k is the computing energy consumption per unit time of ground terminal k, denoted by $\mu_k = \gamma f_k^3$, and $\gamma \geqslant 0$ is the effective switched capacitance [33]. In order to cope with the diversity of task delay and energy consumption requirements, the local execution cost of task n at ground terminal k is defined as

$$Z_n^l = x_{M+1,n}(\alpha t_n^l + \beta w_c e_n^l), \tag{4}$$

where α and β is the weight of task delay and energy consumption, and $\alpha + \beta = 1$. The two weights respectively represent the sensitivity of the task to the task delay and energy consumption. If the task focuses more on reducing the task delay, the coefficient α can be increased appropriately. On the contrary, if we focus more on reducing energy consumption, we could increase the coefficient β appropriately. w_c is the coefficient that adjusts the task delay and energy consumption to the same scale.

(2) UAV task execution model. Let the position variable matrix of the UAV nodes be $\boldsymbol{u}$, The matrix element is the position vector $\boldsymbol{u}_m$ of UAV node m, which is expressed as (x_m, y_m, z_m) in rectangular coordinates. The distance $d_{m,k}$ between UAV node m and ground terminal k is expressed as $d_{m,k} = ||\boldsymbol{u}_m - \boldsymbol{u}_k||$.

Because of the high-altitude advantage of the UAV, we only consider the line-of-sight (Los) channel [34], so the channel fading is only related to the position of the UAV. Besides, the position of UAVs can be obtained by means of Global positioning system (GPS) so that the channel gain can be obtained accurately [35]. So the channel gain $g_{m,k}$ is

$$g_{m,k} = \frac{g_0^D}{{d_{m,k}}^2}, \tag{3}$$

where g_0^D denotes the channel gain with a distance of 1 m. Suppose that when multiple ground terminals are associated with the same UAV node due to task offloading, the ground to air transmission link between them adopts the statistical time multiplexing. Since there are multiple UAV nodes communicating with the ground terminal at the same time, the same bandwidth is in common. As a result, the ground terminal accessed by the same UAV node has complementary interference. For the interference of air-ground communication channel, there is only inter cell co-frequency interference.

According to the free space propagation model, the signal to interference plus noise ratio (SINR) of the ground terminal k associated with the UAV node m is

$$\mathrm{SINR}_{m,k} = \frac{p_D g_{m,k}}{\sum_{k' \neq k}^{K} p_D g_{m,k'} \Gamma(m,k') + B_D \sigma^2}, \tag{4}$$

where σ^2 denotes the Gaussian noise of the channel, p_D denotes the signal transmission power from the ground terminal to the UAV node, $g_{m,k}$ denotes the channel gain between ground terminal k and UAV node m, B_D denotes the bandwidth of UAV node for air-ground communication link. $\Gamma(m,k')$ indicates whether the ground terminal k' associates with other UAVs,

$$\Gamma(m,k') = \begin{cases} 1, & \sum\limits_{m' \neq m}^{M} \Theta(m',k') > 0, \\ 0, & \sum\limits_{m' \neq m}^{M} \Theta(m',k') = 0. \end{cases} \tag{5}$$

If $\Gamma(m,k') = 1$, ground terminal k' will interfere with the transmission link between UAV node m and ground terminal k, otherwise not. Then, the transmission rate $r_{m,k}$ between UAV node m and ground terminal k is

$$r_{m,k} = B_D \log_2(1 + \mathrm{SINR}_{m,k}). \tag{6}$$

The transmission delay of task n for offloading is

$$t_n^{\mathrm{c,tr}} = \frac{I_n}{r_{m,k}}. \tag{7}$$

The computing task capability of each UAV node is related to many internal factors, including CPU core frequency, number of CPU cores, and memory space. Assume that the computation capability (in CPU cycles per second) of UAV node m is f_m, the computing delay of task n executed at UAV node m is

$$t_n^{\mathrm{c,ex}} = \frac{\xi_n}{f_m} = \frac{\kappa I_n}{f_m}. \tag{8}$$

Therefore, the task delay of task n for offloading is

$$t_n^{\mathrm{c}} = t_n^{\mathrm{c,ex}} + t_n^{\mathrm{c,tr}}. \tag{9}$$

Generally speaking, the output data of the task is much less than the input data [23], so the transmission cost of task results is ignored. Similarly, suppose that μ_m is the computing energy consumption per unit time of UAV node m, and $\mu_m = \gamma f_m^3$. The energy consumption of offloading task n at UAV node m is

$$e_n^{\mathrm{c}} = \mu_m t_n^{\mathrm{c,ex}} + p_D t_n^{\mathrm{c,tr}}. \tag{10}$$

In order to further prolong the survival time of the network and balance the workload, the remaining energy of UAV node needs to be considered. Similar to the local task execution model, for different task requirements, the cost of task n executed in UAV node m is defined as

$$Z_n^c = \frac{e_0}{e_m}(\alpha t_n^c + \beta w_c e_n^c) x_{mn}, \tag{11}$$

where e_m and e_0 denote the remaining energy and initial energy of UAV node m. That is, the less energy left in UAV node m, the greater the cost of mobilizing UAV node m for offloaded task computing. Parameter α, β, w_c is consistent with the local task execution model.

2.3 Hovering and flying consumption model

Based on the design idea of combining multi-UAV deployment optimization with scheduling strategy, UAV nodes need to optimize deployment according to task scheduling strategy. We need to further consider the cost of UAV node position updating. Let $\boldsymbol{u}_m$ be the new position vector of UAV node m after receiving the decision of deployment optimization strategy. Let $\boldsymbol{u}_m^0$ be the original position vector of the UAV node m. We assume that UAV m maintains a uniform motion in position updating and the flight velocity is v_m. Therefore, the flight time t_m^F is described as

$$t_m^F = \frac{||\boldsymbol{u}_m - \boldsymbol{u}_m^0||}{v_m}. \tag{12}$$

According to the emergency scenario assumption, the UAV nodes need to hover for task computing. Let the flight power of UAV node m be $p^F(v_m)$, then the flight energy consumption required in T_s is

$$\begin{aligned} e_m^F &= e_m^{F,\text{fly}} + e_m^{F,\text{ho}}, \\ e_m^{F,\text{fly}} &= t_m^F p^F(v_m), \\ e_m^{F,\text{ho}} &= p^F(0)(T_s - t_m^F), \end{aligned} \tag{13}$$

where $e_m^{F,\text{fly}}$ is the flying energy consumption with v_m in flying state, $e_m^{F,\text{ho}}$ is the hovering energy consumption, and $p^F(0)$ denotes the flight power consumed by UAV nodes in hovering state. According to the flight power model of rotary-wing UAV [36], the relationship between $p^F(v_m)$ and v_m is

$$p^F(v_m) = P_0\left(1 + \frac{3v_m^2}{U_{\text{tip}}^2}\right) + P_i\left(\sqrt{1 + \frac{v_m^4}{4v_0^2}} - \frac{v_m^2}{2v_0^2}\right)^{\frac{1}{2}} + \frac{1}{2}d_0\rho s A v_m^3, \tag{14}$$

where P_0 and P_i two constants denote blade profile power and induced power respectively in hovering state, U_{tip} is the tip velocity of the rotor blade, v_0 denotes the average rotor induced velocity in hover, d_0 and s denote the fuselage resistance ratio and rotor solids respectively, ρ and A denote air density and rotor disk area respectively. So the hovering flight power of the UAV node $p^F(0) = P_0 + P_i$.

In order to balance the contradiction between delay and energy consumption during UAV movement and achieve better cost saving effect, the flight cost of UAV task execution model is designed as

$$Z_m^F = \alpha t_m^F + \beta w_F e_m^F, \tag{15}$$

where parameter α, β is consistent with the execution model. Since the flight energy consumption far exceeds the transmission and calculation losses, we use the different coefficient w_F to adjust the flight delay and flight energy consumption to the same scale.

2.4 Problem formulation

Based on the above system model, we take the sum of the local task execution cost of tasks in ground terminals, the UAV task execution cost and the flight cost in UAV nodes as the total system cost, which is composed of task execution delay and energy consumption of ground terminals and UAV nodes. In the system cost, we can find that the task delay and energy consumption required for offloading are not only related to the task scheduling strategy x_{mn}, but also related to the position $\boldsymbol{u}_m = (x_m, y_m, z_m)$ and velocity v_m of UAV node m. Therefore, the optimization problems is formulated as follows:

$$\begin{aligned}
\text{P1}: \ & \min_{\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}} f = \sum_{m=1}^{M}\sum_{n=1}^{N}(Z_n^l + Z_n^c + Z_m^F), && (16)\\
\text{s.t.} \ & \sum_{m}^{M} x_{mn} + x_{M+1,n} = 1, \ \forall m, n, && (16\text{a})\\
& e_m^{F,\text{fly}} < e_{\max}^{\text{fly}}, \ \forall m, && (16\text{b})\\
& \sum_{n}^{N} \xi_n x_{mn} \leqslant c_m, \ \forall m, && (16\text{c})\\
& h_{\min} < z_m < h_{\max}, \ \forall m, && (16\text{d})\\
& x_{mn}, x_{M+1,n} \in \{0,1\}, \ \forall m, n, && (16\text{e})\\
& v_m \leqslant v_{\max}, \ \forall m. && (16\text{f})
\end{aligned}$$

Constraint (16a) indicates that the task can be offloaded at most once; constraint (16b) represents that the flight energy of UAV node needs to be limited, where $e_{\max}^{\text{fly}}$ is the maximum flying energy of the UAV nodes; constraint (16c) indicates that the offloading task received by the UAV node cannot exceed its remaining computing resources; constraints (16d) and (16f) represents that the flight altitude of UAV node needs to be in $[h_{\min}, h_{\max}]$ and the velocity needs to be smaller than $v_{\max}$; constraint (16e) indicates that x_{mn} and $x_{M+1,n}$ are binary.

The optimization problem P1 is a mixed integer nonlinear programming problem, which is also a non-convex optimization problem. In order to solve this problem efficiently, we divide problem P1 into two sub-problems, the task scheduling sub-problem and the UAVs' deployment problem. On this basis, we propose a joint optimization algorithm to solve problem P1.

3 Joint optimization algorithm

In order to solve the primal problem P1 proposed in Section 2, we propose an effective low complexity algorithm in this section. Specifically, we first decompose the optimization problem to two sub-problems. Then, we solved them by proposing a swap matching based task scheduling algorithm and a SCA based multi-UAV deployment algorithm. On this basis, the joint optimization algorithm is proposed by iterating these two algorithms.

3.1 Swap matching based task scheduling algorithm

Given the location variable $\boldsymbol{u}$ and velocity variable $\boldsymbol{v}$, problem P1 becomes a scheduling strategy sub-problem, given by

$$\begin{aligned}
\text{P2}: \ & \min_{\boldsymbol{x}} \sum_{n=1}^{N}\sum_{m=1}^{M} x_{M+1,n}(\alpha t_n^l + \beta w_c e_n^l) + \frac{e_0}{e_m}(\alpha t_n^c + \beta w_c e_n^c) x_{mn},\\
\text{s.t.} \ & \ (16\text{a}), (16\text{c}), (16\text{e}).
\end{aligned} \tag{17}$$

According to the system model, one UAV node can accept multiple tasks offloading, but one task can only be offloaded to one UAV node. It is necessary to develop a multi-task scheduling algorithm in the case of different computing ability and real-time energy consumption of UAV nodes with the amount of different tasks. Based on the fact that the swap matching algorithm can not only ensure the stability of a pairing but also have low complexity and fast convergence [37], we choose it to optimize the task scheduling.

Let computing device set $\mathcal{M} = \{1, \ldots, M+1\}$ and task set $\mathcal{N} = \{1, \ldots, N\}$ be finite and disjoint. When the UAV node m is assigned to task n, a matching pair (m, n) is formed. We take the system cost as the basis for establishing the preference list. Preference list records the system cost of one party to the other in the match. The matching exchange between the two sides is based on the preference list.

As a result, for the sub-problem, each UAV node has its own preference list of each task, and each task has its own preference list for each UAV node similarly. If the ranking position of UAV node m in the preference list of task n is higher than UAV node m', it indicates that the system cost of offloading task n at UAV node m is less than UAV node m', expressed as $m \succ_n m'$. During the iteration, the preference lists of both agents meet the following conditions:

• Both agents can compare and select the matching selection through their own preference list without being unable to select.

• Preference selection is transitive, that is, if $m \succ_n m'$, $m' \succ_n m''$, then $m \succ_n m''$.

Since the UAV node can accept the offloading of multiple tasks, but the tasks can only be offloaded to one UAV node at most, the task scheduling strategy is optimized as a one to many matching problem. Set the matching function denotes as φ, it is defined as follows [37]:

Definition 1. • $|\varphi(n)| = 1,\ \forall n \in \mathcal{N}$;

• $|\varphi(m)| \leqslant I_V,\ \forall m \in \mathcal{M}$, where I_V is the maximum number of tasks that UAV nodes can perform under the constraints of computing resources and energy consumption;

• $\varphi(n) = m \Leftrightarrow \varphi(m) = n$.

According to the Definition 1, some UAV nodes may not be assigned tasks, but the tasks must be executed. Since there is communication interference between tasks during task scheduling, sub-problem P2 is a one-to-many matching problem with external characteristics. Since the income of one matching pair will be affected by other matching pairs, the concept of stability cannot be defined directly. We give the definition of swap matching $\varphi_{mn}^{m'n'}$ as

$$\varphi_{mn}^{m'n'} = \{\varphi \backslash \{(m,n),(m',n')\} \cup \{(m,n'),(m',n)\}\}, \tag{18}$$

where $\varphi(n) = m \Leftrightarrow \varphi(m) = n$.

Based on the above definition of swap matching, the definition of bilateral swap stable matching is given below.

Definition 2. If and only if there is no blocking pair, the matching φ is a bilateral stable matching state, where the blocking pair (m,m') is defined as

• $\varphi(m) = n,\ \varphi(m') = n'$;

• $\forall x \in \{m,n,m',n'\},\ U_x(\varphi_{mn}^{m'n'}) \leqslant U_x(\varphi)$;

• $\exists x \in \{m,n,m',n'\},\ U_x(\varphi_{mn}^{m'n'}) < U_x(\varphi)$.

$U_x(\varphi)$ is the system cost of element x under the matching φ, The characteristics of (n,n') shows that if the swap matching is successful, the system cost of all elements will not increase, and the system cost of at least one element will decrease. With the success of swap matching, when there is no blocking pair in the system, it indicates that the bilateral matching has reached stability. For either agent, it is impossible to find a matching agent that can make the preference cost smaller from the optional list.

In order to describe the external characteristics, the preference cost function U is established according to the task delay and energy consumption of the optimization problem P2, given as

$$U(m,n) = \begin{cases} \alpha t_n^l + \beta w_c e_n^l,\ m = M+1, \\ \dfrac{e_0}{e_m}(\alpha t_n^c + \beta w_c e_n^c),\ m \neq M+1. \end{cases} \tag{19}$$

The specific steps of the algorithm are described as follows:

• During initialization, since the matching state between UAV nodes and ground terminals is not established, we use signal to noise ratio (SNR) instead of SINR in the preference cost function U for initialization assignment, where $\mathrm{SNR}_{m,k} = \frac{p_D g_{m,k}}{B_D \sigma^2}$. Namely, the cost of interference between ground terminals is not considered. The deferred acceptance algorithm (DAA) is used to establish the initial matching state between the UAV node and the task. Each task sends a request to the UAV node with the highest priority in the preference list, and each UAV node receives the task with the highest priority in the requested task according to the preference list. When the computing resources of each UAV node are full or unmatched tasks are rejected by all UAV nodes, the DAA algorithm ends.

• Update the preference cost list according to (19). Based on the initialized matching state, for any task n, if there is another task $n' \in N$ $(n' \neq n)$ that will make the system cost decrease, the swapping will happen. It will stop until the system cost stabilizes.

Therefore, the swap matching based task scheduling algorithm is summarized in Algorithm 1.

Algorithm 1 Swap matching based task scheduling algorithm

```
Require: The SINR in the preference cost $U$ is replaced by SNR for initialization assignment. And we initialize the matching
    state $S_0$ between UAV nodes and ground terminalS based on DAA algorithm, let $S = S_0$, number of iterations $t = 1$.
 1: repeat
 2:     for $n \in \{1, \ldots, N\}$ do
 3:         Find a task $n' \in N(n' \neq n)$;
 4:         if $(n, n')$ is a blocking pair, after the swap, the UAV nodes still meets the resource constraints then
 5:             Swap $(n, n')$;
 6:         else
 7:             Keep existing matching status;
 8:         end if
 9:         Update the preference cost list according to (19);
10:     end for
11: until No blocking pair in the matching;
12: return The task scheduling variable matrix $\boldsymbol{x}$.
```

In the following, we present the complexity and convergence analysis of Algorithm 1.

(1) Complexity. According to the system model, the number of UAV nodes is M. Considering the local execution model, the equipment available for task execution is $M + 1$ and the number of tasks is N. Hence, the algorithm complexity of DAA algorithm during initialization is $O(I_V(M + 1)N)$. We assume that the swap matching algorithm can converge in L_1. In the worst case, the complexity of the algorithm is $O(I_V(M+1)N + L_1C^2_{M+1}N)$, i.e., the number of swaps is $1+2+\cdots+M$ in each iteration. Therefore, compared with exhaustive search, this sub-algorithm achieves better performance due to the low complexity of exchange matching for the many to many matching problem.

(2) Convergence. In the final stable matching state, according to the definition, for any task n, there will be no n' that (n, n') is a blocking pair. In the process of matching swap, the total preferred cost of matching continues to decrease. Since the total scheduling cost is always non-negative, the value of the total scheduling cost has a lower bound. As thus, after a limited numbers of swap matching, the swap matching based task scheduling algorithm can converge to a stable matching state.

3.2 SCA based multi-UAV deployment algorithm

Given the scheduling strategy variable $\boldsymbol{x}$, problem P1 becomes a sub-problem of position and velocity of the UAV nodes, given by

$$\begin{aligned}&\text{P3}: \min_{\boldsymbol{u},\boldsymbol{v}} \sum_{n=1}^{N}\sum_{m=1}^{M} \frac{e_0}{e_m}(\alpha t_n^c + \beta w_c e_n^c)x_{mn} + \alpha t_m^F + \beta w_F e_m^F,\\ &\text{s.t. (16b), (16d), (16f).}\end{aligned} \tag{20}$$

It can be found that when the scheduling strategy variable $\boldsymbol{x}$ is determined, the execution time $t_n^{\mathrm{c,ex}}$ of task n is independent of the multi-UAV deployment. Therefore, the problem is equivalent to solving the minimum value of communication cost and flight cost when the UAV nodes position and velocity change, i.e.,

$$\begin{aligned}&\text{P3}': \min_{\boldsymbol{u},\boldsymbol{v}} \sum_{n=1}^{N}\sum_{m=1}^{M} \frac{e_0}{e_m}[(\alpha + \beta w_c p_D)t_n^{\mathrm{c,tr}}]x_{mn} + \alpha t_m^F + \beta w_F e_m^F,\\ &\text{s.t. (16b), (16d), (16f).}\end{aligned} \tag{21}$$

When the scheduling strategy variable $\boldsymbol{x}$ is given, the position and velocity variables between UAV nodes can be considered separately. Thus problem P3$'$ can be equivalent to the following problem:

$$\begin{aligned}\text{P3}'': &\min_{\boldsymbol{u}_m, v_m} \Omega_m(\boldsymbol{u}_m, v_m)\\ &= \sum_{n=1}^{N} \frac{e_0}{e_m}[(\alpha + \beta w_c p_D)t_n^{\mathrm{c,tr}}]x_{mn} + \alpha t_m^F + \beta w_F e_m^F,\\ &\text{s.t. (16b), (16d), (16f).}\end{aligned} \tag{22}$$

The problem P3$''$ is non-convex function, and constraints (16d) and (16f) are linear. Since P3$''$ contains multiple second-order terms about $||\boldsymbol{u}_m - \boldsymbol{u}_k||^2$ and cannot be replaced in a holistic manner, we consider using block coordinate descent to lower $\boldsymbol{u}_m$ and v_m respectively. In the constraint (16b), we can transform constraint (16b) to

$$e_m^{F,\text{fly}} - e_{\max}^{\text{fly}} < 0. \tag{23}$$

We give the following lemma.

Lemma 1. $l_m(x_m, y_m, z_m, v_m) = e_m^{F,\text{fly}} - e_{\max}^{\text{fly}}$ is a convex function with regard to variable x_m, y_m, z_m, v_m when only one of them is considered.

Proof. Because $e_{\max}^{\text{fly}}$ is a constant, it only needs to prove that $e_m^{F,\text{fly}}$ is a convex function. To this end, we rewrite $e_m^{F,\text{fly}}$ as

$$e_m^{F,\text{fly}} = t_m^F p_m^F(v_m) = \delta(x_m, y_m, z_m)\chi(v_m), \tag{24}$$

where $\delta(x_m, y_m, z_m) = ||\boldsymbol{u}_m - \boldsymbol{u}_m^0|| = \sqrt{(x_m - x_m^0)^2 + (y_m - y_m^0)^2 + (z_m - z_m^0)^2}$, $\chi(v_m) = \frac{p_m^F}{v_m} = \frac{P_0}{v_m}(1 + \frac{3v_m^2}{U_{\text{tip}}^2}) + \frac{P_i}{v_m}(\sqrt{1 + \frac{v_m^4}{4v_0^2}} - \frac{v_m^2}{2v_0^2})^{\frac{1}{2}} + \frac{1}{2}d_0\rho s A v_m^2$.

We can calculate the second-order partial derivatives of four variables in (24) separately. When only considering the position variable x_m, y_m, z_m separately, we simplify (24) as $e_m^{F,\text{fly}} = C_1\delta(x_m, y_m, z_m)$, where C_1 is a non-negative coefficient that they will not affect the concave-convex of (24). When only one of the position variables is considered, $\delta(x_m, y_m, z_m)$ is a composite function of monotonically increasing convex function $x^{\frac{1}{2}}$ and convex function $(x - x_0)^2 + a$, where x_0, a is a constant. Thus, it is a convex function.

When only considering the velocity variable v_m, we simplify (24) as $e_m^{F,\text{fly}} = C_2\chi(v_m)$, where C_2 is a non-negative coefficient, which will not affect the concave-convex of (24). By calculating the second-order partial derivative of v_m in $\chi(v_m)$, we can find that it is greater than 0 easily. In conclusion, when only one of the four variables is considered, the second-order partial derivatives of l_m under simulation conditions is non-negative. Therefore, Lemma 1 is proved.

In sub-problem P3$''$, we need to transform $\Omega_m(\boldsymbol{u}_m, v_m)$ to a convex form. To this end, we adopt the SCA algorithm [38, 39] to solve the optimization problem P3$''$.

In order to develop the SCA based multi-UAV deployment algorithm, Lemma 2 is given [28].

Lemma 2. For any continuous first-order differentiable non-convex function $\Gamma(x)$, there will be a convex function $g(x; x^t)$, which at the point of x^t, $\nabla\Gamma(x^t) = \nabla g(x; x^t)|_{x=x^t}$. The general form of $g(x; x^t)$ is

$$g(x; x^t) = \Gamma(x^t) + \nabla\Gamma(x^t)(x - x^t) + \tau||x - x^t||^2, \tag{25}$$

where $\tau > 0$, t is the number of iterations. So the problem of minimizing $\Gamma(x)$ can be converted to minimizing $g(x; x^t)$ and the x^t is updated iteratively according to the minimum y^t of $g(x; x^t)$ under constraints,

$$x^t = x^{t-1} + \alpha^t(y^t - x^{t-1}), \tag{26}$$

where α^t is the moving step weight, dynamically adjusted α^t can avoid the phenomenon of crossing the optimal value x^* of $\Gamma(x)$ due to too long step length. α^t is generally adjusted by the Armijo criteria.

Based on Lemma 2, we optimize the position variable x_m, y_m, z_m and the velocity variable v_m separately. And the optimized variables are combined into a new vectors variable $\boldsymbol{\eta}_m = (x_m, y_m, z_m, v_m)$. Let $\boldsymbol{\eta}_{mi}$ represents ith element of $\boldsymbol{\eta}_m$, the optimization problem P3$''$ can be converted to

$$\begin{aligned} \text{P3}''' : \min_{\boldsymbol{\eta}_{mi}} g_m(\boldsymbol{\eta}_{mi}) &= \Omega_m(\boldsymbol{\eta}_{mi}^t) + \nabla\Omega_m(\boldsymbol{\eta}_{mi}^t)(\boldsymbol{\eta}_{mi} - \boldsymbol{\eta}_{mi}^t) + \tau||\boldsymbol{\eta}_{mi} - \boldsymbol{\eta}_{mi}^t||^2, \\ \text{s.t. } &(16\text{b}), (16\text{d}), (16\text{f}). \end{aligned} \tag{27}$$

where $\boldsymbol{\eta}_{mi}^t$ is the given feasible points at the tth iteration.

For the convex optimization problem P3$'''$, the standard convex optimization tool CVX can be used to solve it, which generally uses the built-in interior point method (IPM) to solve the convex optimization problem. The SCA based multi-UAV deployment algorithm is summarized in Algorithm 2.

In the following, we present the complexity and convergence analysis of Algorithm 2.

(1) Complexity. The algorithm complexity analysis of the sub-problem is based on the algorithm complexity of the IPM [40]. The total number of variables is $4M$, and there are $3M$ linear constraints,

Algorithm 2 SCA based multi-UAV deployment algorithm

```
Require: Initialize the UAV nodes position matrix $\boldsymbol{u}_0$ according to the scheduling variable matrix $\boldsymbol{x}$ and UAV nodes velocity
    matrix $\boldsymbol{v}_0$. Let number of external iterations $r = 0$.
 1: repeat
 2:   for $m \in \{1, \ldots, M\}$ do
 3:     for $i \in \{1, \ldots, 4\}$ do
 4:       Initialize $t = 0$, $\boldsymbol{\eta}_{m_i}^{r+1,0} = \boldsymbol{\eta}_{m\,i}^{r}$;
 5:       repeat
 6:         $t = t + 1$;
 7:         Use IPM to solve the convex optimization problem P3''', and get $y^t$;
 8:         Solve $\alpha^t$ using the Armijo criteria;
 9:         Update $\boldsymbol{\eta}_{m_i}^{r+1,t}$ according to (26);
10:       until $\boldsymbol{\eta}_{m_i}^{r+1,t}$ is a stable convergent solution;
11:     end for
12:     $r = r + 1$;
13:   end for
14: until $\boldsymbol{\eta}_m{}^r$ is a stable convergent solution;
15: return The UAV nodes position matrix $\boldsymbol{u}_r$ and UAV nodes velocity matrix $\boldsymbol{v}_r$.
```

including 1 variable; M second-order cone-constraints, including 4 variables. Assuming that the total number of external iterations is L_2, the total complexity of the algorithm is $O(L_2 M^{2.5})$. So compared with exhaustive search, this sub-algorithm achieves better performance for the good use of gradient descent by SCA algorithm.

(2) Convergence. Since the problem solved in each internal iteration is a convex problem about $\boldsymbol{\eta}_{m\,i}$, the global optimal solution of problem P3''' can always be found by the IPM in each internal iteration so that the internal iteration can always converge. The external iteration is about the block coordinate descent of $\boldsymbol{\eta}_{m\,i}$. Since $\Omega_m(\boldsymbol{u}_m, v_m)$ is a smooth continuous function, there is always $\Omega_m(\boldsymbol{\eta}_m^{r-1}) \geqslant \Omega_m(\boldsymbol{\eta}_m^{r})$. At the same time, because of $\Omega_m(\boldsymbol{\eta}_m) > 0$, $\Omega_m(\boldsymbol{\eta}_m)$ must have a lower bound. Therefore, the proposed Algorithm 2 is guaranteed to converge.

3.3 Joint optimization algorithm

Based on the above analysis, the optimization problem is divided into two sub-problems, and the corresponding algorithms are proposed. Now, we propose a joint optimization algorithm by combining the aforementioned two algorithms in an iterative manner to obtain the solution to the primal problem P1, which is summarized in Algorithm 3.

The convergence and complexity of the joint optimization algorithm are analyzed as follows.

Algorithm 3 Joint optimization algorithm

```
Require: Initialize the task scheduling matrix $\boldsymbol{x}_0$, the UAV nodes position matrix $\boldsymbol{u}_0$, the UAV nodes velocity matrix $\boldsymbol{v}_0$. And
    input the system parameters, set the number of iterations $l = 0$, give the convergence accuracy $\varepsilon$.
1: repeat
2:   $l = l + 1$;
3:   Update the task scheduling matrix $\boldsymbol{x}_l$ according to the swap matching based task scheduling algorithm;
4:   Update the UAV nodes position matrix $\boldsymbol{u}_l$ and the UAV nodes velocity matrix $\boldsymbol{v}_l$ according to the SCA based multi-UAV
     deployment algorithm;
5: until $|f(\boldsymbol{x}_l, \boldsymbol{u}_l, \boldsymbol{v}_l) - f(\boldsymbol{x}_{l-1}, \boldsymbol{u}_{l-1}, \boldsymbol{v}_{l-1})| < \varepsilon$;
6: return The task scheduling matrix $\boldsymbol{x}$, the UAV nodes position matrix $\boldsymbol{u}$ and the UAV nodes velocity matrix $\boldsymbol{v}$.
```

(1) Complexity. In each iteration, Algorithms 1 and 2 are used to solve problem P2 and P3 in turn. According to the analysis of the sub-problems in the above two subsections, we can get the algorithm complexity of the two sub-algorithms. We assume that the proposed optimization algorithm can converge in G iterations. Then the complexity of the joint optimization algorithm is $O(G(I_V(M + 1)N + L_1 C_{M+1}^2 N + M^{2.5} L_2))$.

(2) Convergence. According to the convergence analysis of the the swap matching based task scheduling algorithm in the above subsection, we can obtain the following conclusions:

$$f(\boldsymbol{x}_{l-1}, \boldsymbol{u}_{l-1}, \boldsymbol{v}_{l-1}) \geqslant f(\boldsymbol{x}_l, \boldsymbol{u}_{l-1}, \boldsymbol{v}_{l-1}). \tag{28}$$

And according to the convergence analysis of the SCA based multi-UAV deployment algorithm,

$$f(\boldsymbol{x}_l, \boldsymbol{u}_{l-1}, \boldsymbol{v}_{l-1}) \geqslant f(\boldsymbol{x}_l, \boldsymbol{u}_l, \boldsymbol{v}_l). \tag{29}$$

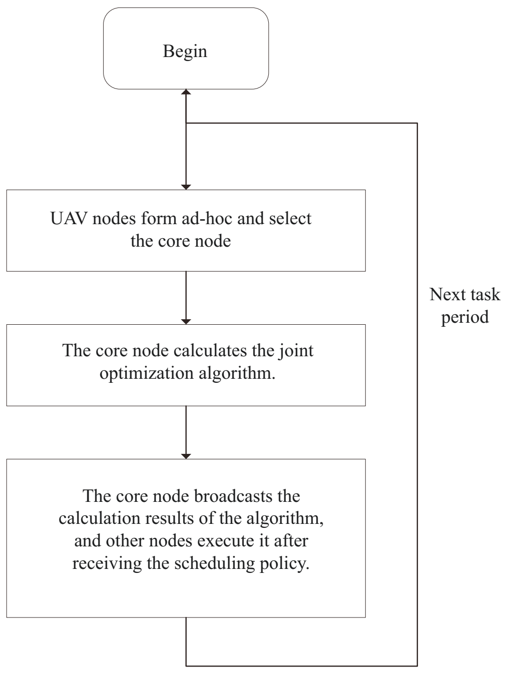
Begin
UAV nodes form ad-hoc and select the core node
The core node calculates the joint optimization algorithm.
The core node broadcasts the calculation results of the algorithm, and other nodes execute it after receiving the scheduling policy.
Next task period

Figure 2 Application flow chart of the strategy.

Since the system cost $f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) > 0$, the optimization objective function has a lower bound. So the the proposed joint optimization algorithm can converge and obtain an sub-optimal solution.

Due to the need for higher network control efficiency, and without considering the security and scalability of data, we have adopted a centralized control method to implement the joint algorithm. Since there is no cloud server in the air-ground collaboration architecture based on ad-hoc, it is necessary to select a suitable core node in the UAV nodes that can collects the resource information of other UAV nodes and task information sent by ground terminals in period T_s. And it can also compute the proposed joint optimization algorithm, which is used to assign tasks, and optimize the deployment of each UAV node. Therefore, we design the application flow as follows.

• The UAV nodes form an ad-hoc network, and then the appropriate core node can be selected according to the distributed Raft election algorithm [41], which is very suitable for the election problem of distributed networks.

• Ground terminals send task information to the nearest UAV nodes. The core node collects task information and then computes the joint optimization algorithm to reduce the system cost of offloading.

• The core node broadcasts the task scheduling strategy, deployment optimization policies to other UAV nodes. Each node updates its position and velocity according to the results of joint optimization algorithm and performs task assistance.

The specific steps is also shown in Figure 2.

4 Simulation results

This section demonstrates the performance of the proposed algorithm.

Consider a square task area with a length of 500 m in which M UAV nodes and K ground terminals are distributed randomly. The UAV nodes form a self-organizing network based on the ad-hoc network architecture, and their service coverage can cover the entire task area. Considering the limitations in reality, we set the range of UAV node deployment height between 120 and 500 m [36]. According to the parameters in reality and literature [42], the specific simulation parameters are presented in Table 2.

To verify the effectiveness of the proposed joint optimization algorithm, the performance must be compared with benchmark algorithms. Benchmark algorithms should have the characteristics of most of the existing algorithms and have a certain degree of optimization performance for the proposed problem. Based on these principles, the following two algorithms are used as the benchmark algorithms for performance comparison:

Table 2 Simulation parameters

Parameter	Value
Task area size	500 m
Number of UAV nodes	$M = 5$
Deployment height range of UAV nodes	[120, 500] m
Flight velocity range of UAV nodes	[0, 20] m/s
Maximum flight energy of the UAV node	$e_{\max}^{\text{fly}} = 8 \times 10^4$ J
Number of computing tasks per ground terminal	[1, 3]
Computing resources of UAV nodes	$c_m = [16, 48]$ G CPU cycles
Ground terminal signal transmission power	$p_D = 0.5$ W
Communication bandwidth between the UAV and the ground terminal	30 Mbps
Computing rate of UAV nodes	$f_m = [3, 5]$ GHz
Computing rate of the ground terminal	$f_k = 1$ GHz
Amount of task input data	$I_n = [5, 10]$ Mbits
Coefficient of the relationship between the cycles of computation required for the task and its amount of input data	$\kappa = 1000$
CPU capacitance coefficient of UAV nodes and ground terminal	$\gamma = 10^{-27}$
Noise power spectral density	$\sigma^2 = -174$ dBm/Hz
Channel gain with a distance of 1 m	$g_0^D = -50$ dB
Blade profile power in the hovering state	$P_0 = 158.76$ W
Induced power in the hovering state	$P_i = 88.63$ W
Tip velocity of the rotor blade	$U_{\text{tip}} = 120$ m/s
Average rotor-induced velocity in hover	$v_0 = 4.03$ m/s
Fuselage resistance ratio	$d_0 = 0.3$
Rotor solids	$s = 0.05$ kg/m^3
Air density	$\rho = 1.225$ kg/m^3
Rotor disk area	$A = 0.503$ m^2
Task collaboration policy update cycle	$T_s = 300$ s
Initial energy of the UAV node	$e_0 = 4 \times 10^5$ J

(1) Heuristic algorithm (HA). Each task selects the UAV node for offloading according to the lowest system cost composed of task delay and energy consumption. When the computing resource of the selected UAV node is insufficient, the task will select the next lowest UAV node for offloading, and so on, until it is left locally. Moreover, the coordinate center point, which indicates the center position of all matched ground terminal coordinates, is taken as the optimal position of the UAV node under the condition that the flight altitude remains unchanged. Consequently, in most cases, it will reduce the transmission delay. The velocity is set to a constant of 15 m/s.

(2) Random-access and fixed position algorithm (RFA). The tasks and UAV nodes were randomly matched under the constraint. The coordinate center point of the matched ground terminal was taken as the optimal position of the UAV node under the condition that the flight altitude remains unchanged, and the velocity was set to a constant of 15 m/s.

(3) Swap matching-based task scheduling and UAV deployment algorithm (SMTDA). In map discretization, the problem of UAV node location deployment is transformed into a one-to-one matching problem. Finally, the swap-matching method was used to simultaneously iteratively optimize the task scheduling and UAV node location deployment strategy [43]. In the simulation, altitude and velocity were set as constant of 200 m and 15 m/s, respectively.

In HA, since the ground terminal can access the UAV node with the lowest system cost, the participation of an ad-hoc network can be avoided. On the contrary, because of the need for global information exchange, the proposed algorithm and SMTDA must work in the proposed self-organizing architecture using the proposed strategy. Therefore, the following comparative experiments can also further illustrate the effectiveness of our proposed architecture and algorithm:

Initially, we demonstrated the convergence of the proposed algorithm where the number of UAV nodes $M = 5$, number of ground terminals $K = 20$, and computing resources of UAV nodes $c_m = 40$ G CPU cycles. In the simulation results, the system cost is defined as the objective function in the optimization problem P1. In Figure 3, the proposed algorithm can converge within four iterations under different weights, which shows that the proposed algorithm has fast convergence and is conducive to the scheduling

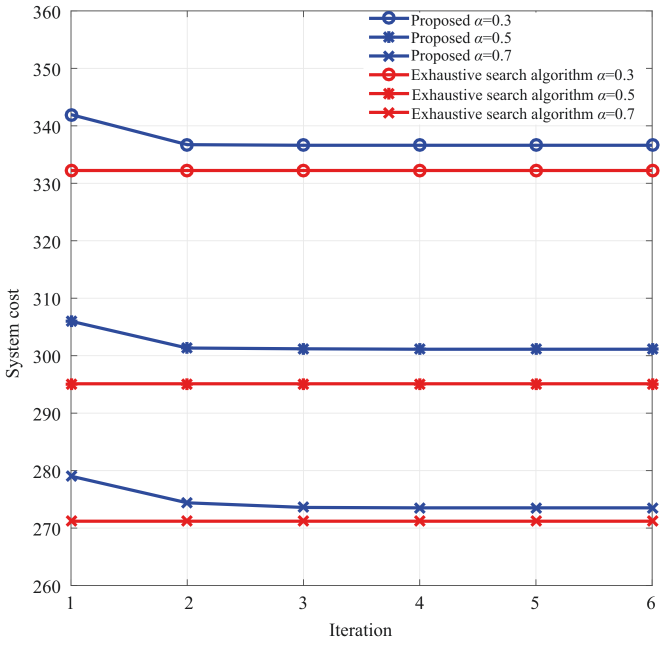

Figure 3 (Color online) The convergence of the proposed algorithm with varying weights.

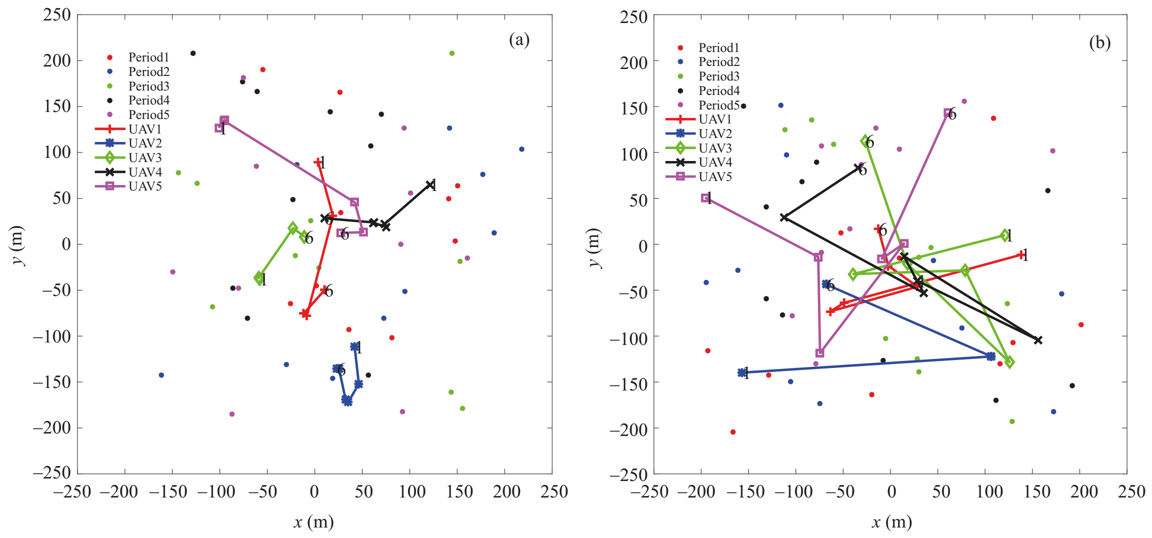

Figure 4 (Color online) Deployment of UAV nodes in different cases. (a) Delay-sensitive case; (b) energy-sensitive case.

strategy computation of UAV nodes in the distributed architecture.

Then, we demonstrated the deployment of the UAV nodes with different task requirements. Figure 4 demonstrates the deployment of UAV nodes in the delay-sensitive case ($\alpha = 0.9$) and the energy-sensitive case ($\beta = 0.9$), where the computing resources of UAV nodes $c_m = 40$ G CPU cycles and the number of ground terminals $K = 10$. The simulation results in Figures 4(a) and (b) show that most UAV nodes will choose to fly in the direction of the ground terminal concentration area after the position of ground terminals changes. Figure 4(a) also shows that to save task delay as much as possible, the flight distance of most UAV nodes is short in the delay-sensitive case. Conversely, Figure 4(b) shows that the flight distance of most UAV nodes is long in the energy-sensitive case because in the flight power model, when the flight velocity is < 20 m/s, hovering energy consumption is always greater than flight energy consumption. Thus, energy consumption should be reduced, and most UAV nodes tend to prolong flight time.

To verify the effectiveness of the proposed algorithm, we randomly set the initial positions of UAV nodes and ground terminals many times according to the Monte Carlo method. Furthermore, the performance of

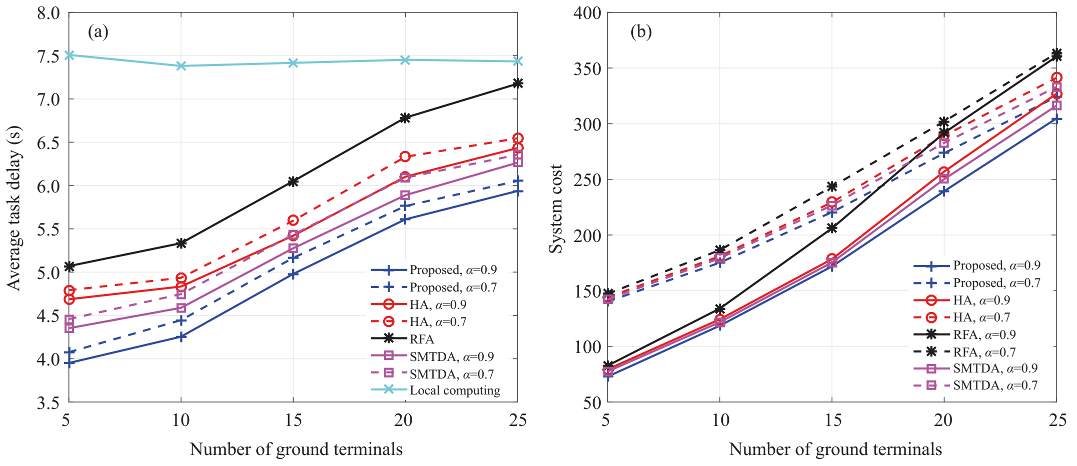

Figure 5 (Color online) Comparison with different number of ground terminals in delay-sensitive case. (a) Comparison of task delay; (b) comparison of system cost.

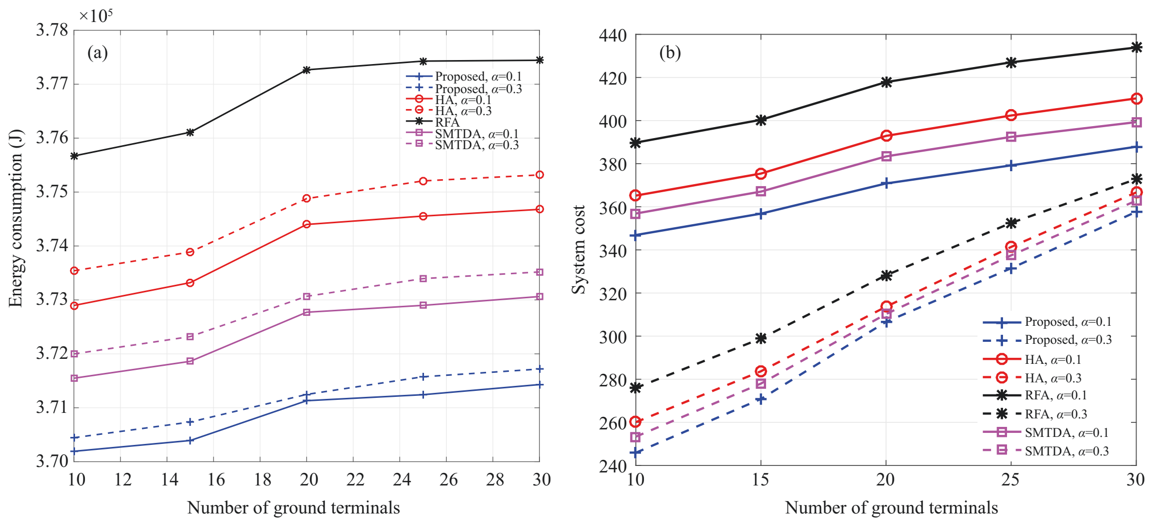

Figure 6 (Color online) Comparison with different numbers of ground terminals in the energy-sensitive case. (a) Comparison of energy consumption; (b) comparison of system cost.

the proposed algorithm was observed when the number of ground terminals and the computing resources of UAV nodes were changed.

Initially, we demonstrated the effectiveness of the proposed algorithm by varying the number of ground terminals in different delay-sensitive cases ($\alpha = 0.9$, 0.7) and energy-sensitive cases ($\beta = 0.9$, 0.7) in Figures 5–7, where the computing resources of UAV nodes $c_m = 40$ G CPU cycles.

Figure 5 shows the system performance in the delay-sensitive case. We compared the proposed algorithm and the two benchmark algorithms in the delay-sensitive case. Compared with HA and RFA, the task delay and system cost of the proposed algorithm were significantly reduced. However, with the increase of ground terminals, the task load becomes increasingly heavy, and the system cost increases gradually toward the system cost without cooperation because the computing resources and communication bandwidth of the UAV node are limited in the system.

Then, we compared the system performance by varying the number of ground terminals in the energy-sensitive case. The simulation result in Figure 6 shows that compared with HA and RFA, the energy consumption and system cost of the proposed algorithm are greatly reduced. However, by increasing the ground terminals, the task load gradually became heavy, and the system cost gradually increased.

To verify the system performance of the energy-consumption weight optimization proposed in this paper, the survival time of the UAV network was also simulated. The UAV network survival time is defined as the total execution time when the first UAV node exits the network because of insufficient energy. In addition, we added the performance of the proposed algorithm without weight optimization in the

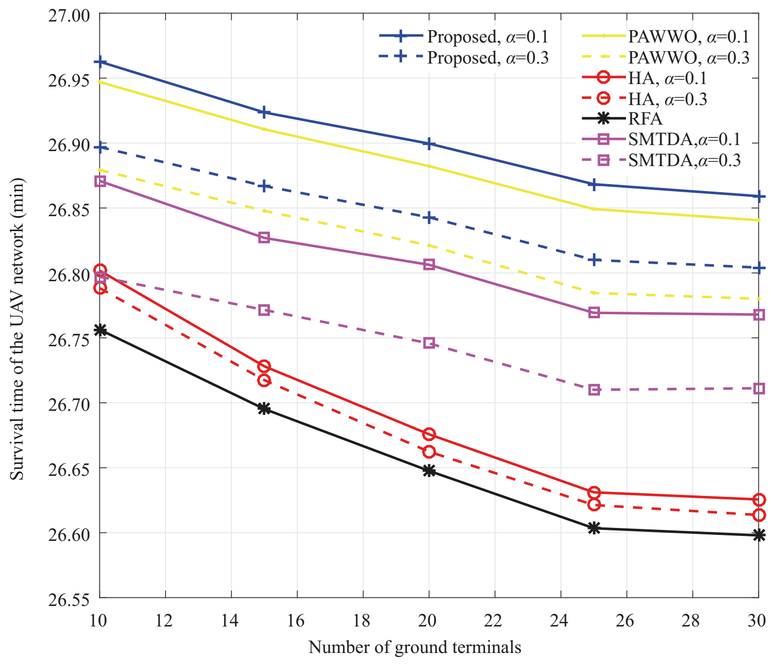

Figure 7 (Color online) Comparison of the survival time of the UAV network with different numbers of ground terminals in the energy-sensitive case.

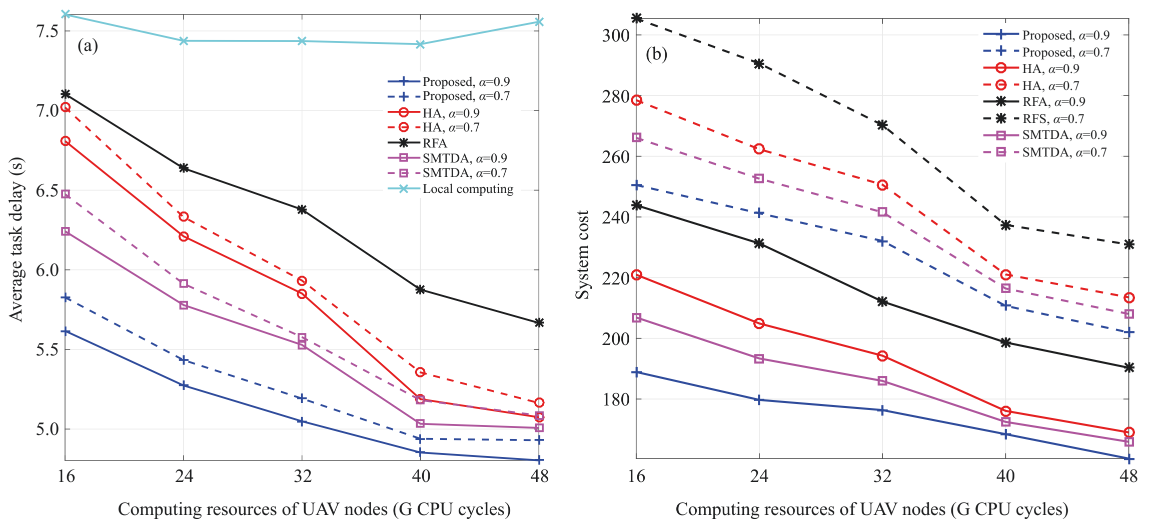

Figure 8 (Color online) Comparison with different computing resources of UAV nodes in the delay-sensitive case. (a) Comparison of task delay; (b) comparison of system cost.

simulation, which is named PAWWO in Figure 7. In the energy-sensitive case, compared with HA, RFA, and PAWWO, the UAV network survival time of the proposed algorithm has greatly improved, which shows the effectiveness of the proposed algorithm. When the number of ground terminals is small, most tasks will be offloaded to UAV nodes. Therefore, with the increase in the number of ground terminals, the survival time of the UAV network decreases rapidly. Otherwise, when there is task overloading, some tasks will be computed locally at the ground terminal without increasing the execution cost of UAV nodes. As a result, the decreasing trend of the survival time of the UAV network becomes slower.

Furthermore, we demonstrated the effectiveness of the proposed algorithm by varying the computing resources of UAV nodes in different cases in Figures 8–10, where the number of ground terminals $K = 15$.

Figure 8 compares the system performance in the delay-sensitive case. The simulation result shows that compared with HA and RFA, the task delay and system cost of the proposed algorithm are significantly reduced when the task is delay sensitive. With the increase in computing resources, the offloaded tasks gradually increased; thus, the system cost gradually decreased.

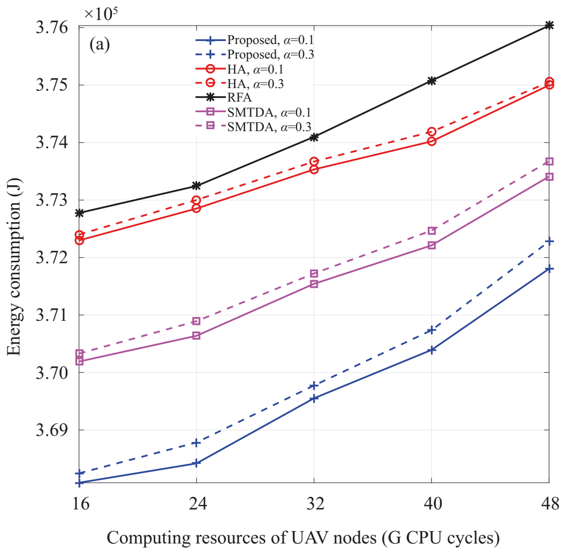
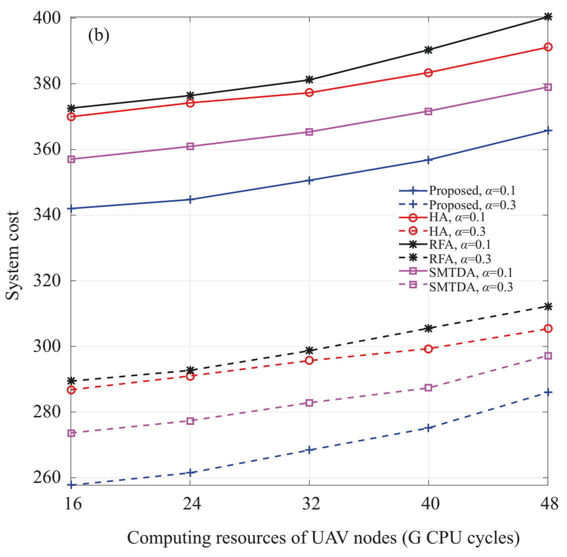
Figure 9 (Color online) Comparison with different computing resources of UAV nodes in the energy-sensitive case. (a) Comparison of energy consumption; (b) comparison of system cost.

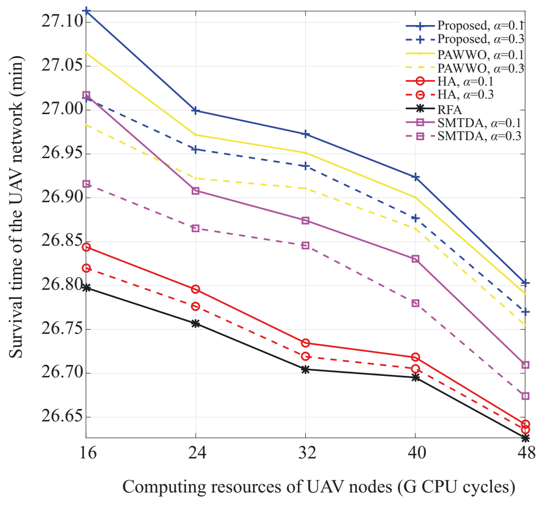
Figure 10 (Color online) Comparison of the survival time of the UAV network computing resources of UAV nodes in the energy-sensitive case.

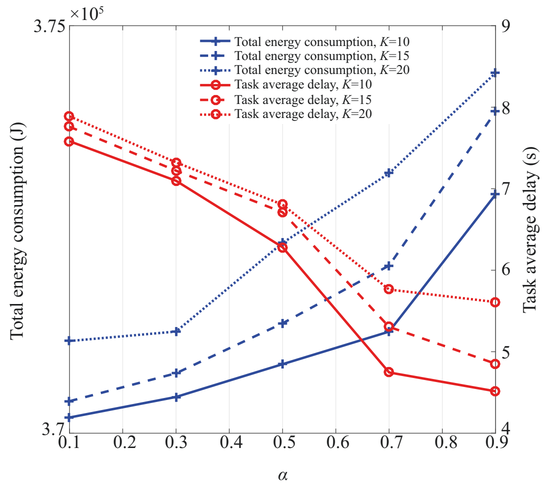
Figure 11 (Color online) Influence of different weights on system performance.

Then, we illustrated the system performance by varying the computing resources of UAV nodes in the energy-sensitive case. The simulation result Figure 9 shows that compared with HA and RFA, the energy consumption and system cost of the proposed algorithm are greatly reduced. Since the energy consumption required for task offloading to the UAV nodes is generally greater than that required for local computing, with the increase in computing resources, more and more tasks are offloaded to UAV nodes to assist in computing, resulting in the gradual increasing of the system cost. Similarly, when the computing resources change, the effectiveness of the energy-consumption weight optimization is verified in Figure 10.

Finally, we explored the effect of task delay weight α and energy weight β on system performance under different numbers of ground terminals. The simulation results in Figure 11 show that for the proposed algorithm, the task delay weight α and energy weight β can effectively control the optimization of task delay and energy consumption. It further illustrates the contradictory relationship between task delay and energy consumption, in which maximum optimization is difficult to obtain at the same time.

5 Conclusion

Owing to the broken link with the ground BS, aerial computing effectively accelerates task execution. Therefore, we studied the problem of task scheduling and multi-UAV deployment in emergency communication networks. To minimize task delay and energy consumption, we proposed a joint optimization algorithm based on task scheduling, position, and velocity optimization. The simulation results showed that compared with the benchmark algorithms, the proposed joint optimization algorithm performed better in reducing task delay and energy consumption, and prolonging network lifetime.

Acknowledgements This work was supported in part by Beijing Natural Science Foundation (Grant No. 4222010), National Key Research and Development Program of China (Grant No. 2019YFC1511302), and Key Technology Research Project of Jiangxi Province (Grant No. 2013AAE01007).

References

1 Munich R. Natcatservice Loss Events Worldwide 1980–2014. Munich: Munich Reinsurance, 2015
2 Stanford D K. Unmanned aircraft systems. 2009. https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470664797.fmatter
3 Zeng Y, Zhang R, Lim T J. Wireless communications with unmanned aerial vehicles: opportunities and challenges. IEEE Commun Mag, 2016, 54: 36–42
4 Wang H, Wang J, Ding G, et al. Completion time minimization with path planning for fixed-wing UAV communications. IEEE Trans Wireless Commun, 2019, 18: 3485–3499
5 Xu Y, Zhang T, Liu Y, et al. UAV-assisted MEC networks with aerial and ground cooperation. IEEE Trans Wireless Commun, 2021, 20: 7712–7727
6 Zhao N, Lu W, Sheng M, et al. UAV-assisted emergency networks in disasters. IEEE Wireless Commun, 2019, 26: 45–51
7 Sun Y, Wang T, Wang S. Location optimization and user association for unmanned aerial vehicles assisted mobile networks. IEEE Trans Veh Technol, 2019, 68: 10056–10065
8 Ueyama J, Freitas H, Faical B S, et al. Exploiting the use of unmanned aerial vehicles to provide resilience in wireless sensor networks. IEEE Commun Mag, 2014, 52: 81–87
9 Sabella D, Vaillant A, Kuure P, et al. Mobile-edge computing architecture: the role of MEC in the Internet of Things. IEEE Consumer Electron Mag, 2016, 5: 84–91
10 Du J, Cheng W, Lu G, et al. Resource pricing and allocation in MEC enabled blockchain systems: an A3C deep reinforcement learning approach. IEEE Trans Netw Sci Eng, 2022, 9: 33–44
11 Messaoudi F, Ksentini A, Bertin P. On using edge computing for computation offloading in mobile network. In: Proceedings of IEEE Global Communications Conference (GLOBECOM), 2017
12 Du J, Yu F R, Lu G, et al. MEC-assisted immersive VR video streaming over terahertz wireless networks: a deep reinforcement learning approach. IEEE Int Things J, 2020, 7: 9517–9529
13 Liao Z, Peng J, Huang J, et al. Distributed probabilistic offloading in edge computing for 6G-enabled massive Internet of Things. IEEE Int Things J, 2021, 8: 5298–5308
14 Pham Q V, Ruby R, Fang F, et al. Aerial computing: a new computing paradigm, applications, and challenges. IEEE Int Things J, 2022, 9: 8339–8363
15 Zheng J, Anpalagan A, Guizani M, et al. Guest editorial: aerial computing: drones for multi-access edge computing. IEEE Wireless Commun, 2021, 28: 10–12
16 Cheng N, Xu W, Shi W, et al. Air-ground integrated mobile edge networks: architecture, challenges, and opportunities. IEEE Commun Mag, 2018, 56: 26–32
17 Liao Z, Ma Y, Huang J, et al. HOTSPOT: a UAV-assisted dynamic mobility-aware offloading for mobile-edge computing in 3-D space. IEEE Internet Things J, 2021, 8: 10940–10952
18 Erdelj M, Natalizio E, Chowdhury K R, et al. Help from the sky: leveraging UAVs for disaster management. IEEE Pervasive Comput, 2017, 16: 24–32
19 Zhang S, Cheng W. Statistical QoS provisioning for UAV-enabled emergency communication networks. In: Proceedings of IEEE Globecom Workshops (GC Wkshps), 2019. 1–6
20 Gu J, Su T, Wang Q, et al. Multiple moving targets surveillance based on a cooperative network for multi-UAV. IEEE Commun Mag, 2018, 56: 82–89
21 Budaev D, Amelin K, Voschuk G, et al. Real-time task scheduling for multi-agent control system of UAV's group based on network-centric technology. In: Proceedings of International Conference on Control, Decision and Information Technologies (CoDIT), 2016. 378–381
22 Zhang L, Chakareski J. UAV-assisted edge computing and streaming for wireless virtual reality: analysis, algorithm design, and performance guarantees. IEEE Trans Veh Technol, 2022, 71: 3267–3275
23 Zhang T, Xu Y, Loo J, et al. Joint computation and communication design for UAV-assisted mobile edge computing in IoT. IEEE Trans Ind Inf, 2020, 16: 5505–5516
24 Zhang K, Gui X, Ren D, et al. Energy-latency tradeoff for computation offloading in UAV-assisted multiaccess edge computing system. IEEE Int Things J, 2021, 8: 6709–6719
25 Hu X, Wong K K, Yang K, et al. UAV-assisted relaying and edge computing: scheduling and trajectory optimization. IEEE Trans Wireless Commun, 2019, 18: 4738–4752
26 Wang L, Wang K, Pan C, et al. Deep reinforcement learning based dynamic trajectory control for UAV-assisted mobile edge computing. IEEE Trans Mobile Comput, 2022, 21: 3536–3550
27 Ning Z, Dong P, Wen M, et al. 5G-enabled UAV-to-community offloading: joint trajectory design and task scheduling. IEEE J Sel Areas Commun, 2021, 39: 3306–3320
28 Sun Y, Xu D, Ng D W K, et al. Optimal 3D-trajectory design and resource allocation for solar-powered UAV communication systems. IEEE Trans Commun, 2019, 67: 4281–4298
29 Cai Y, Wei Z, Li R, et al. Joint trajectory and resource allocation design for energy-efficient secure UAV communication systems. IEEE Trans Commun, 2020, 68: 4536–4553
30 Pang X, Zhao N, Tang J, et al. IRS-assisted secure UAV transmission via joint trajectory and beamforming design. IEEE Trans Commun, 2022, 70: 1140–1152

31 Tianze L, Muqing W, Min Z, et al. An overhead-optimizing task scheduling strategy for ad-hoc based mobile edge computing. IEEE Access, 2017, 5: 5609–5622

32 Haas Z J, Pearlman M R. The performance of query control schemes for the zone routing protocol. IEEE ACM Trans Netw, 2001, 9: 427–438

33 Yang Z, Pan C, Wang K, et al. Energy efficient resource allocation in UAV-enabled mobile edge computing networks. IEEE Trans Wireless Commun, 2019, 18: 4576–4589

34 Wang H, Wang J, Ding G, et al. Completion time minimization for turning angle-constrained UAV-to-UAV communications. IEEE Trans Veh Technol, 2020, 69: 4569–4574

35 Simon E P, Ros L, Hijazi H, et al. Joint carrier frequency offset and channel estimation for OFDM systems via the EM algorithm in the presence of very high mobility. IEEE Trans Signal Process, 2012, 60: 754–765

36 Zeng Y, Xu J, Zhang R. Energy minimization for wireless communication with rotary-wing UAV. IEEE Trans Wireless Commun, 2019, 18: 2329–2345

37 Bodine-Baron E, Lee C, Chong A, et al. Peer effects and stability in matching markets. In: Proceedings of the 4th International Conference on Algorithmic Game Theory, 2011

38 Scutari G, Facchinei F, Lampariello L. Parallel and distributed methods for constrained nonconvex optimization-part I: theory. IEEE Trans Signal Process, 2017, 65: 1929–1944

39 Scutari G, Facchinei F, Lampariello L, et al. Parallel and distributed methods for constrained nonconvex optimization-part II: applications in communications and machine learning. IEEE Trans Signal Process, 2017, 65: 1945–1960

40 Wang K Y, So A M C, Chang T H, et al. Outage constrained robust transmit optimization for multiuser MISO downlinks: tractable approximations by conic optimization. IEEE Trans Signal Process, 2014, 62: 5690–5705

41 Ongaro D, Ousterhout J. In search of an understandable consensus algorithm. In: Proceedings of USENIX Annual Technical Conference (USENIX ATC 14), 2014. 305–319

42 Zhou Y, Pan C, Yeoh P L, et al. Communication-and-computing latency minimization for UAV-enabled virtual reality delivery systems. IEEE Trans Commun, 2021, 69: 1723–1735

43 Zhang T, Wang Y, Liu Y, et al. Cache-enabling UAV communications: network deployment and resource allocation. IEEE Trans Wireless Commun, 2020, 19: 7470–7483