

RESEARCH

Open Access



Device association and trajectory planning for UAV-assisted MEC in IoT: a matching theory-based approach

Xinjun Zhang¹, Guopeng Zhang^{1*} , Kezhi Wang² and Kun Yang^{3,4}

*Correspondence:
gpzhang@cumt.edu.cn

¹The School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

²Department of Computer Science, Brunel University, Uxbridge, London, Middlesex UB8 3PH, UK

³The School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China

⁴The Yangtze Delta Region Institute, University of Electronic Science and Technology of China, Quzhou 324000, China

Abstract

Unmanned aircraft vehicles (UAVs)-enabled mobile edge computing (MEC) can enable Internet of Things devices (IoT) to offload computing tasks to them. Considering this, we study how multiple aerial service providers (ASPs) compete with each other to provide edge computing services to multiple ground network operators (GNOs). An ASP owning multiple UAVs aims to achieve the maximum profit from providing MEC service to the GNOs, while a GNO operating multiple IoTs aims to seek the computing service of a certain ASP to meet its performance requirements. To this end, we first quantify the conflicting interests of the ASPs and GNOs by using different profit functions. Then, the UAV scheduling and resource allocation is formulated as a multi-objective optimization problem. To address this problem, we first solve the UAV trajectory planning and resource allocation problem between one ASP and one GNO by using the Lagrange relaxation and successive convex optimization (SCA) methods. Based on the obtained results, the GNOs and ASPs are then associated in the framework based on the matching theory, which results in a weak Pareto optimality. Simulation results show that the proposed method achieves the considerable performance.

Keywords: Unmanned aerial vehicle, Mobile edge computing, Internet of Things, Matching theory

1 Introduction

With the rapid development of Internet of Things (IoT), the network terminals or user equipment is expected to deal with a large number of computational-intensive and delay-sensitive applications in the future [1]. To solve the problem of computing power shortage of IoT devices (IoT), mobile edge computing (MEC) [2] has been proposed to make IoT devices to offload part of the computing tasks to the high-performance computing servers in network edge, such as base stations. However, in some areas (such as farms, pastures and forests), the lack of power and communication infrastructure leads to deployment of edge servers difficult [3]. In recent years, small UAV [4] has been applied to carry microbase station or edge servers to help provide service to the above-mentioned areas.

Most of existing works [3, 5, 6] regard UAVs as the components of ground networks, such as maximizing network throughput [5], minimizing task completion time [6], or reducing the energy consumption of IoTDs [3, 7]. However, few papers considered the business model of the UAVs, which may be run by different organizations or companies. For example, an air service provider (ASP) may focus on offering services to ground network operators (GNOs) to maximize its profit, while a GNO operating an IoT network may need to buy the service from an ASP to achieve their goals. However, ASPs and GNOs have different targets, which may form a conflict of interest environment.

From the above perspective, this paper investigates the UAV scheduling and resource allocation between multiple ASPs and GNOs. Each of the GNOs deploys multiple IoTDs to cover a specific area. The IoTDs generate computing tasks at a certain time interval and can offload part of the tasks to reduce energy consumption and task completion time. Each of the ASPs operates one UAV to provide computing service to a GNO during a scheduling cycle. The aim of this paper is to coordinate the resource demand and supply between the ASPs and GNOs. To this end, we use price as a signal to associate the GNOs and ASPs in the framework based on the matching theory [8]. This leads to a weak Pareto optimality and obtains the UAV flight trajectory, UAV-and-IoTD association, as well as communication and computational resource allocation that achieve the benefits of both the GNOs and ASPs. The main contribution of this paper is summarized as follows.

- 1 By quantifying the conflicting interests of the ASPs and GNOs as different profit functions, the UAV scheduling and resource allocation is formulated as a multi-objective optimization problem, where an ASP aims to reap the most profits from providing computing service to the GNOs, while a GNO aims to seek the service of a certain ASP to meet its performance requirements.
- 2 The formulated problem is a mixed-integer nonlinear multi-objective optimization problem, which is difficult to solve by traditional methods. To address this difficulty, we decompose it into multiple one-GNO to one-ASP problems and solve them by using the Lagrange relaxation and successive convex optimization (SCA) methods.
- 3 Based on the results of the one-to-one problem, the GNOs and ASPs are associated in the framework of the matching theory, which results in a weak Pareto optimality. Simulation results show that the proposed method achieves the considerable performance.

The remainder of this paper is organized as follows. Section 2 presents the related works. Section 3 describes the system model. Section 4 introduces the proposed multi-objective optimization problem. Section 5 gives the algorithm to solve the one-to-one UAV trajectory planning and resource allocation problem. Section 6 addresses the association of the GNOs with the ASPs in the framework of the matching theory. The simulation results are reported in Sect. 7. Finally, conclusions are made in Sect. 8.

2 Related works

Recently, UAVs have been widely used in the wireless power transfer (WPT) [9], IoT data collection [10], MEC [6, 11], and mobile crowd sensing (MCS) [12]. The authors in [9] presented a UAV-assisted WPT to ensure the sufficient energy of IoTDs that

supports their basic operation. Zhan and Zeng [10] minimized the task completion time of a multi-UAV system by jointly optimizing the UAV trajectory and the wake-up scheduling of the IoTDs. Hu et al. [6] proposed a UAV-assisted MEC system in which a set of UAVs work together to minimize the overall task completion time of the ground users by jointly optimizing the ratio of offloaded tasks, the trajectories of the UAVs, and the user scheduling strategies. Xu et al. [11] studied UAV-assisted MEC systems from the perspective of the physical layer security, where the minimum security computing capacity maximization problem for both time division multiple access (TDMA) and non-orthogonal multiple access (NOMA) schemes was solved by optimizing the computation and communication resources as well as the trajectories of the UAVs. Zhou et al. [12] investigated the joint task assignment and route planning problem in UAV-aided MCS systems from an energy efficiency perspective.

However, all the above works consider the overall system performance optimization, but without considering the UAV scheduling and resource allocation in the presence of a conflict of interest between the ASPs and the GNOs.

3 System model

As shown in Fig. 1, the considered UAV-assisted MEC consists of a set \mathcal{N} of N ASPs and a set \mathcal{M} of M GNOs. We assume that each ASP having one rotary wing UAV, and each GNO covers a specific ground service area (GSA). Here we use ASP and UAV, as well as GNO and GSA interchangeably. A set \mathcal{K}_m of K_m IoTDs are distributed in GSA m , $\forall m \in \mathcal{M}$, and the GSAs do not overlap with each other. The IoTDs are fixed on the ground, and the coordinate of IoTD k_m ($\forall k_m \in \mathcal{K}_m$) in GSA m is given as $w_{k_m} = (x_{k_m}, y_{k_m}, z_{k_m} = 0)$.

A discrete-time model is considered, and the duration of a time slot is represented by τ . The scheduling of the UAVs is performed in a periodic manner, and each period is defined as a *mission cycle* lasting for a set \mathcal{T} of T consecutive time slots. In the mission cycle, each UAV can serve only one GSA, and each GSA can be served by only one UAV. Let $\theta_{n,m} \in \{0, 1\}$ denote the association between UAV n ($\forall n \in \mathcal{N}$) and GSA m , where $\theta_{n,m} = 1$ represents UAV n is associated with GSA m . Otherwise, $\theta_{n,m} = 0$. Therefore, we have the following *association constraints* for the UAVs and GSAs.

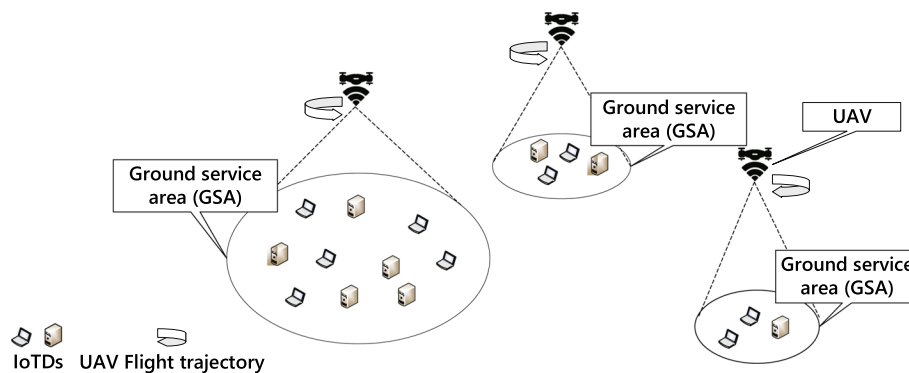


Fig. 1 UAV-assisted MEC in IoT

$$\sum_{m \in \mathcal{M}} \theta_{n,m} = 1, \quad \forall n \in \mathcal{N}, \tag{1}$$

and

$$\sum_{n \in \mathcal{N}} \theta_{n,m} = 1, \quad \forall m \in \mathcal{M}. \tag{2}$$

Suppose that the UAV n is scheduled to serve GSA m in a mission cycle. The computing task of IoTD k_m is represented as (L_{k_m}, C_{k_m}) , where L_{k_m} (in bits) denotes the size of the task input data, and C_{k_m} (in CPU cycles/bit) denotes the number of CPU cycles required for computing 1 bit of the data. During any slot $t, \forall t \in \mathcal{T}$ in the mission cycle, the coordinate of UAV n is given by $q_{n,m}[t] = (x_{n,m}[t], y_{n,m}[t], z_{n,m}[t] = H)$, where $H > 0$ is the flight altitude of the UAV. Then, the flight trajectory of the UAV during the mission cycle is given by $Q_{n,m} = \{q_{n,m}[1], \dots, q_{n,m}[t], \dots, q_{n,m}[T]\}$.

3.1 Communication model

In order to offload part of the computing task to a UAV for remote execution, an IoTD should first upload the task input data to the UAV via wireless links. Since there exists a strong line-of-sight (LoS) link, the channel gain from IoTD k_m to UAV n in slot t is given by $g_{n,k_m}[t] = \frac{g_0}{\|q_{n,m}[t] - w_{k_m}\|^2}$ [13], where g_0 represents the received power at the reference distance of 1 m.

Let B and p_{k_m} denote the channel bandwidth and transmission power of IoTD k_m , respectively. The transmission rate from IoTD k_m to UAV n in slot t is given as [13]

$$R_{n,k_m}[t] = B \log_2 \left(1 + \frac{g_{n,k_m}[t] p_{k_m}}{\sigma^2} \right), \quad \forall t \in \mathcal{T}. \tag{3}$$

where σ^2 is the noise power at the receiver of a UAV.

3.2 Local computing model

The IoTD can perform the remaining task by using its local computing power, whereas it can offload part of the computing task to a UAV for remote execution. Let $l_{k_m}[t]$ (in bits) denote the size of the task data being processed by IoTD k_m locally in slot t . Let $f_{k_m}[t]$ denote the CPU frequency of IoTD k_m to process the $l_{k_m}[t]$ task bits in slot t . Then we have

$$l_{k_m}[t] = \frac{f_{k_m}[t] \tau}{C_{k_m}}, \quad \forall t \in \mathcal{T}. \tag{4}$$

Assuming that the dynamic voltage and frequency scaling (DVFS) technique [3] is used to improve the energy efficiency of the IoTDs. The energy consumption of IoTD k_m in slot t for local computing is given by [3]

$$E_{k_m}^C[t] = \gamma_m (f_{k_m}[t])^3 \tau, \quad \forall t \in \mathcal{T}. \tag{5}$$

where γ_m is the coefficient depending on the chip architecture. Denote the maximum CPU frequency of IoTD k_m by $f_{k_m}^{\max}$. The following *maximum computing power constraints* should be satisfied by IoTD k_m in any slot t .

$$0 \leq f_{k_m}[t] \leq f_{k_m}^{\max}, \quad \forall t \in \mathcal{T}. \tag{6}$$

3.3 Edge computing model

When a large number of IoTDs simultaneously offload their computing tasks to a UAV, the UAV-carried server can work in parallel (in an assembly line manner) to reduce the overall task completion time. That is, after receiving the task data of an IoTD in any slot $t, 1 \leq t \leq T - 1$, the UAV can perform the task on behalf of the IoTD in any slot $(t + i), 1 \leq i \leq T - t$, as shown in Fig. 2.

The task offloading from IoTD k_m in GSA m to UAV n will span the following two phases.

- 1 *Phase 1 (Task uploading)*: In any slot t , UAV n assigns a portion $\alpha_{n,k_m}[t] \leq 1$ of the slot to IoTD k_m for task uploading. Then, the following *time-integrity constraints* should be satisfied by UAV n in any slot t .

$$\sum_{k_m \in \mathcal{K}_m} \alpha_{n,k_m}[t] \leq 1, \quad 1 \leq t \leq T - 1. \tag{7}$$

The amount of task data that IoTD k_m can upload to the UAV is given by

$$b_{n,k_m}[t] = \alpha_{n,k_m}[t] \tau R_{n,k_m}[t], \quad 1 \leq t \leq T - 1. \tag{8}$$

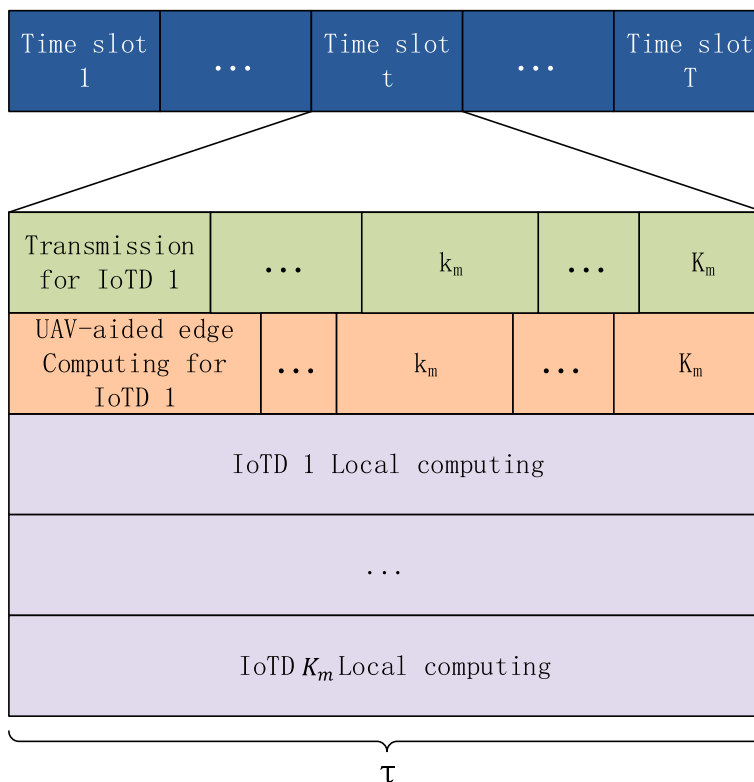


Fig. 2 The workflow of the assembly line work of UAVs

The resultant energy consumption of IoTD k_m is obtained as

$$E_{n,k_m}^{\text{Tx}}[t] = p_{k_m} \alpha_{n,k_m}[t] \tau, \quad 1 \leq t \leq T - 1. \tag{9}$$

- 2 *Phase 2 (Task execution)*: After receiving the data of an IoTD in slot $t, 1 \leq t \leq T - 1$, the UAV can perform the task in the subsequent slot $(t + i), 1 \leq i \leq T - t$. Let $0 \leq \beta_{n,k_m}[t] \leq 1$ denote the ratio of slot t allocated to IoTD k_m for task execution. The following *time-integrity constraint* should be satisfied by UAV n in any slot t .

$$\sum_{k_m \in \mathcal{K}_m} \beta_{n,k_m}[t] \leq 1, \quad 2 \leq t \leq T. \tag{10}$$

Let $c_{n,k_m}[t]$ (in bits) denote the size of task data that UAV n performs for IoTD k_m in slot t . Let $f_n[t]$ denote the CPU frequency of UAV n in slot t . Then we have

$$c_{n,k_m}[t] = \frac{\beta_{n,k_m}[t] \tau f_n[t]}{C_n}, \quad 2 \leq t \leq T. \tag{11}$$

Assuming that the DVFS technique is also used by the UAVs, the energy consumed by UAV n for executing the task of IoTD k_m in slot t is given by

$$E_{n,k_m}^{\text{C}}[t] = \gamma_n \tau (f_n[t])^3 \beta_{n,k_m}[t], \quad 2 \leq t \leq T. \tag{12}$$

where γ_n is the coefficient depending on the chip architecture. Denote the maximum CPU frequency allowed by UAV n as f_n^{max} . The following *maximum computing power constraints* should be met by any n th UAV in any t th slot

$$0 \leq f_n[t] \leq f_n^{\text{max}}, \quad 2 \leq t \leq T. \tag{13}$$

Furthermore, the following two constraints need to be considered when adopting the assembly line working mode.

- 1 *Information-causality constraints*: A UAV can process the computing task of an IoTD only after receiving the corresponding task data. Therefore, the following *information-causality constraints* should be met.

$$c_{n,k_m}[1] = 0, \quad \beta_{n,k_m}[1] = 0, \quad \text{and} \quad f_n[1] = 0, \tag{14}$$

$$b_{n,k_m}[T] = 0 \quad \text{and} \quad \alpha_{n,k_m}[T] = 0, \tag{15}$$

and

$$\sum_{i=1}^{t-1} b_{n,k_m}[i] \geq \sum_{i=2}^t c_{n,k_m}[i], \quad 2 \leq t \leq T, \quad \forall k_m \in \mathcal{K}_m. \tag{16}$$

It is worth noting that the storage capacity of the lightweight edge servers is about a few hundred Gb to a few Tb, while the program size uploaded by IoTDs is mostly several Mb. Therefore, the storage capacity limit of edge servers can be ignored.

- 2 *Task-integrity constraints*: To ensure that the tasks are fully executed (either locally or on a UAV), the following *task-integrity constraints* should be met.

$$\sum_{t=2}^T c_{n,k_m}[t] + \sum_{t=1}^T l_{k_m}[t] \geq L_{k_m}, \quad \forall k_m \in \mathcal{K}_m. \quad (17)$$

3.4 Mechanical energy consumption of UAVs

The energy consumed by the propulsion system of a UAV is far more than that consumed for task computing, and it depends mainly on the flight trajectory and velocity of the UAV. Then, assuming that the duration of a time slot τ is small enough, the flight trajectory of a UAV during each slot can be considered as a uniformly linear function. Therefore the flight velocity of a UAV in any slot t is determined by its position in slot $(t + 1)$ and its position in slot t , which is given by

$$v_n[t] = \frac{q_{n,m}[t + 1] - q_{n,m}[t]}{\tau}, \quad \forall n \in \mathcal{N}, \forall m \in \mathcal{M}. \quad (18)$$

The *maximum velocity constraints* on a UAV is given as

$$\|v_n[t]\| \leq v_n^{\max}, \quad \forall t \in \mathcal{T}. \quad (19)$$

Using Eq. (19), the energy consumed by the propulsion system of UAV n in any slot t is given by

$$E_n^F[t] = 0.5\zeta \|v_n[t]\|^2 \tau, \quad \forall t \in \mathcal{T}, \quad (20)$$

where ζ is relevant to the payload of the UAV n [6].

4 Problem formulation

In the studied UAV-enabled MEC system, the goals of each ASP and each GNO are as follows.

- 1 *The goal of a GNO:* In addition to exploit the limited energy and computing resources of the IoTDS, a GNO needs to purchase the computing service offered by an ASP to achieve its performance requirement, such as minimizing the overall task completion time, reducing the total energy consumption of the IoTDS, or achieving a compromise between both.
- 2 *The goal of an ASP:* The UAVs of different ASPs may have different mechanical properties (e.g., flying velocity, acceleration, and energy consumption) and computing power. In each scheduling cycle, the goal of an ASP is to select a specific service area and leverage the potential of the UAV to maximize its benefit.

In this section, we first define the profit functions for the ASPs and the GNOs, which separately describe their targets in the MEC system. Then, we formulate the UAV scheduling and communication-and-computation resource allocation problem as a multi-objective optimization problem. Finally, we present the main method to solve the problem.

4.1 Profit function of GNOs

When a GSA m is served by UAV n , the total energy consumption of the IoTs in GSA m is given by

$$E_{n,m}^{\text{GSA}} = \sum_{k_m \in \mathcal{K}_m} \sum_{t \in \mathcal{T}} (E_{k_m}^{\text{C}}[t] + E_{n,k_m}^{\text{Tx}}[t]), \forall m \in \mathcal{M}, \quad \forall n \in \mathcal{N}. \quad (21)$$

Let $E_{k_m}^{\text{max}}$ denote the maximum available energy of IoT k_m . Based on eqs. (5) and (9), the following *maximum available energy constraints* for the IoTs should be satisfied.

$$\sum_{t \in \mathcal{T}} (E_{k_m}^{\text{C}}[t] + E_{n,k_m}^{\text{Tx}}[t]) \leq E_{k_m}^{\text{max}}, \quad \forall k_m \in \mathcal{K}_m, \quad \forall m \in \mathcal{M}. \quad (22)$$

Therefore, a performance-adjustable profit function for a GNO is defined as

$$G_{n,m} = \varepsilon_m^1 T + \varepsilon_m^2 E_{n,m}^{\text{GSA}}, \quad \forall m \in \mathcal{M}, \quad \forall n \in \mathcal{N}, \quad (23)$$

where ε_m^1 and ε_m^2 ($0 \leq \varepsilon_m^1, \varepsilon_m^2 \leq 1$ and $\varepsilon_m^1 + \varepsilon_m^2 = 1$) are the weight parameters.

4.2 Profit function for ASPs

Let ρ_n denote the pricing of UAV n for performing 1-bit of computing tasks. When UAV n serves GSA m , the profit function of the UAV affiliated ASP is defined as

$$W_{n,m} = \rho_n \sum_{k_m \in \mathcal{K}_m} \sum_{t \in \mathcal{T}} c_{n,k_m}[t], \quad \forall m \in \mathcal{M}, \quad \forall n \in \mathcal{N}. \quad (24)$$

Let E_n^{max} denote the maximum available energy of UAV n . Based on eqs. (12) and (20), the following *maximum available energy constraints* for the UAVs should be satisfied.

$$\sum_{t \in \mathcal{T}} E_n^{\text{F}}[t] + \sum_{t \in \mathcal{T}} \sum_{k_m \in \mathcal{K}_m} E_{n,k_m}^{\text{C}}[t] \leq E_n^{\text{max}}, \quad \forall n \in \mathcal{N}. \quad (25)$$

The profit function of an ASP reflects the fact that its goal is to maximize its own benefit rather than improve the performance of the GNOs. In such a *seller market* situation with $N < M$, a UAV will choose the GNO with the highest bid for its service to serve.

4.3 System problem

A UAV aims to reap the most profits from providing MEC to the GSAs, while a GSA aims to seek the service of a UAV to meet its performance requirements. This can be formulated as a multi-objective optimization problem as given in the following problem **P1**. The variables to be optimized are as follows:

- 1 $\Theta \triangleq \{\theta_{n,m}\}$: The association between the UAVs and GSAs.
- 2 $\mathbf{A} \triangleq \{\alpha_{n,k_m}[t]\}$: The transmission time assigned by a UAV to the IoTs in each time slot.
- 3 $\mathbf{B} \triangleq \{\beta_{n,k_m}[t]\}$: The executing time assigned by a UAV to the IoTs in each time slot.
- 4 $\mathbf{F} \triangleq \{f_n[t]\}$: The computing power set by a UAV in each time slot.

- 5 $\mathbf{J} \triangleq \{f_{k_m}[t]\}$: The computing power set by an IoTD in each time slot.
 6 $\mathbf{Q} \triangleq \{Q_{n,m}\}$: The flight trajectory of a UAV.

$$(\mathbf{P1}) \max_{\Theta, \mathbf{A}, \mathbf{B}, \mathbf{F}, \mathbf{J}, \mathbf{Q}, T} \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}} \theta_{n,m} W_{n,m} \quad (26a)$$

$$\min_{\Theta, \mathbf{A}, \mathbf{B}, \mathbf{F}, \mathbf{J}, \mathbf{Q}} \sum_{n \in \mathcal{N}, T} \sum_{m \in \mathcal{M}} \theta_{n,m} G_{n,m} \quad (26b)$$

$$\text{s.t.} \quad \sum_{m \in \mathcal{M}} \theta_{n,m} = 1, \quad \forall n \in \mathcal{N}, \quad (26.1)$$

$$\sum_{n \in \mathcal{N}} \theta_{n,m} = 1, \quad \forall m \in \mathcal{M}, \quad (26.2)$$

$$0 \leq f_{k_m}[t] \leq f_{k_m}^{\max}, \quad \forall t \in \mathcal{T}, \quad (26.3)$$

$$0 \leq f_n[t] \leq f_n^{\max}, \quad \forall t \in \mathcal{T}, \quad (26.4)$$

$$\sum_{k_m \in \mathcal{K}_m} \alpha_{n,k_m}[t] \leq 1, \quad 0 \leq \alpha_{n,k_m}[t] \leq 1, \quad \forall t \in \mathcal{T}, \quad (26.5)$$

$$\sum_{k_m \in \mathcal{K}_m} \beta_{n,k_m}[t] \leq 1, \quad 0 \leq \beta_{n,k_m}[t] \leq 1, \quad \forall t \in \mathcal{T}, \quad (26.6)$$

$$c_{n,k_m}[1] = 0, \quad \beta_{n,k_m}[1] = 0, \quad f_n[1] = 0, \quad \forall k_m \in \mathcal{K}_m, \quad (26.7)$$

$$b_{n,k_m}[T] = 0 \text{ and } \alpha_{n,k_m}[T] = 0, \quad \forall k_m \in \mathcal{K}_m, \quad (26.8)$$

$$\sum_{i=1}^{t-1} b_{n,k_m}[i] \geq \sum_{i=2}^t c_{n,k_m}[i], \quad 2 \leq t \leq T, \quad \forall k_m \in \mathcal{K}_m, \quad (26.9)$$

$$\sum_{t=2}^T c_{n,k_m}[t] + \sum_{t=1}^T l_{k_m}[t] \geq L_{k_m}, \quad \forall k_m \in \mathcal{K}_m, \quad (26.10)$$

$$q_{n,m}[T] = q_{n,m}[1], \quad \forall n \in \mathcal{N}, \quad (26.11)$$

$$\|v_n[t]\| \leq v_n^{\max}, \quad \forall t \in \mathcal{T}, \quad (26.12)$$

$$\sum_{t \in \mathcal{T}} \left(E_{k_m}^C[t] + E_{n,k_m}^{\text{Tx}}[t] \right) \leq E_{k_m}^{\max}, \quad \forall k_m \in \mathcal{K}_m, \quad \forall m \in \mathcal{M}, \quad (26.13)$$

$$\sum_{t \in \mathcal{T}} E_n^F[t] + \sum_{t \in \mathcal{T}} \sum_{k_m \in \mathcal{K}_m} E_{n,k_m}^C[t] \leq E_n^{\max}, \quad \forall n \in \mathcal{N}. \quad (26.14)$$

where constraints (26.1) and (26.2) (introduced in Sec. III-A) are the *association constraints* for the UAVs and the GSAs, respectively, constraints (26.3 and (26.4) (introduced in Sec. III-B and Sec. III-C, respectively) are the *maximum computing power constraints* on a UAV and an IoTD, respectively, constraints (26.5) and (26.6) (introduced in Sec. III-B) are the *time-integrity constraints* for each of the UAVs, constraints (26.7), (26.8), and (26.9) (introduced in Sec. III-B) are the *information-causality constraints* caused by the used of the assembly line working mode, constraint (26.10) (introduced in Sec. III-B) is the *task-integrity constraint* for each of the IoTDs, constraint (26.11) indicates that at the end of a mission cycle a UAV must return to its starting point for charging and maintenance operations, constraint (26.12) (introduced in Sec. III-C) is the *maximum velocity constraint* on each of the UAVs, and constraints (26.14) and (26.13) (introduced in Sec. IV-A and Sec. IV-B, respectively) are the *maximum available energy constraints* on the UAVs and IoTDs, respectively.

4.4 The decomposition of Problem P1

It is obvious that problem **P1** is a mixed-integer nonlinear multi-objective optimization problem which is difficult to solve. To address this problem, we propose the following approach.

Step 1: Solve the following one-to-one resource allocation problem P2.

We first ignore the association between the UAVs and the GSAs, and assume that GSA m is served by UAV n . Therefore, parameter Θ and the corresponding constraints in problem **P1** can be removed. Problem **P1** is simplified into the following form

$$\begin{aligned}
 (\mathbf{P2}) \quad & \max_{\mathbf{A}_{n,m}, \mathbf{B}_{n,m}, \mathbf{F}_{n,m}, \mathbf{J}_{n,m}, \mathbf{Q}_{n,m}, T} W_{n,m} & (27a) \\
 & \min_{\mathbf{A}_{n,m}, \mathbf{B}_{n,m}, \mathbf{F}_{n,m}, \mathbf{J}_{n,m}, \mathbf{Q}_{n,m}, T} G_{n,m} & (27b) \\
 \text{s.t.} \quad & (26.3), (26.5), (26.6), (26.7), (26.8), (26.9), (26.10), (26.11), (26.12), \\
 & (26.13), (26.14). & (27.1)
 \end{aligned}$$

By solving problem **P2**, we can obtain the optimal flight trajectory of UAV n and the optimal resource allocation for the IoTDs in GSA m .

Step 2: Schedule the UAVs to the GSAs by solving the following problem P3.

Based on the obtained profits and performance of the UAVs and GSAs by solving problem **P2**, we can simplify problem **P1** into the following form

$$\begin{aligned}
 (\mathbf{P3}) \quad & \max_{\Theta} \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}} \theta_{n,m} W_{n,m} & (28a) \\
 & \min_{\Theta} \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}} \theta_{n,m} G_{n,m} & (28b) \\
 \text{s.t.} \quad & (26.1), (26.2). & (28.1)
 \end{aligned}$$

Problem **P3** addresses the issue of how to schedule the finite UAVs to the GSAs by optimizing parameter Θ , so as to achieve the goals of UAVs and GSAs. In what follows, we address problems **P2** and **P3** sequentially.

5 Solving problem P2

Problem **P2** remains a multi-objective optimization problem. We use the weighted-sum approach [14] to transform it into a single-objective optimization problem as follows.

$$\begin{aligned}
 (\mathbf{P4}) \quad & \min_{\mathbf{A}_{n,m}, \mathbf{B}_{n,m}, \mathbf{F}_{n,m}, \mathbf{J}_{n,m}, \mathbf{Q}_{n,m}, T} \mu_1 G_{n,m} - \mu_2 W_{n,m} & (29) \\
 \text{s.t.} \quad & (26.12), (26.5), (26.6), (26.3), (26.9), \\
 & (26.10), (26.7), (26.8), (26.11), (26.13), (26.14). & (29.1)
 \end{aligned}$$

where μ_1 and μ_2 are the weighting factors. The objective function of problem **P4** is expanded as

$$\begin{aligned}
 \mu_1 G_{n,m} - \mu_2 W_{n,m} = & \mu_1 \varepsilon_m^1 T + \mu_1 \varepsilon_m^2 \sum_{k \in \mathcal{K}_m} \sum_{t=1}^T \left(\gamma_{k_m} (f_{k_m}[t])^3 \tau + p_{k_m} \alpha_{n,k_m}[t] \tau \right) \\
 & - \mu_2 \rho_n \sum_{k \in \mathcal{K}_m} \sum_{t=2}^T \frac{\beta_{n,k_m}[t] \tau f_n[t]}{C_n},
 \end{aligned} \tag{26}$$

where T is an integer variable to be optimized. Next we propose a *Binary Search Algorithm* (BSA)-based method to solve problem **P4**.

- 1 *Step 1:* With any given $T = \hat{T}$, each GSA minimizes the difference between the amount of completed tasks and the target one. Let η represent the difference. We get the following problem **P5**.

$$\begin{aligned}
 \text{(P5)} \quad & \min_{\mathbf{A}_{n,m}, \mathbf{B}_{n,m}, \mathbf{F}_{n,m}, \mathbf{J}_{n,m}, \mathbf{Q}_{n,m}, \eta} \mu_1 \left(\varepsilon_m^1 \hat{T} + \varepsilon_m^2 \sum_{k \in \mathcal{K}_m} \sum_{t=1}^{\hat{T}} (\gamma_{k_m} (f_{k_m}[t])^3 \tau + p_{k_m} \alpha_{n,k_m}[t] \tau) \right) \\
 & - \mu_2 \rho_n \sum_{k \in \mathcal{K}_m} \sum_{t=2}^{\hat{T}} \frac{\beta_{n,k_m}[t] \tau f_n[t]}{C_n} + \eta \tag{31} \\
 \text{s.t.} \quad & (26.11), (26.12), (26.4), (26.3), (26.5), (26.6), (26.7), (26.8), (26.9), \\
 & (26.13), (26.14), \tag{31.1} \\
 & \sum_{t=2}^{\hat{T}} \frac{\beta_{n,k_m}[t] \tau f_n[t]}{C_n} + \sum_{t=1}^{\hat{T}} \frac{f_{k_m}[t] \tau}{C_{k_m}} + \eta \geq L_{k_m}, \forall k_m \in \mathcal{K}_m. \tag{31.2}
 \end{aligned}$$

We introduce a new variable η in the objective function (), which converts constraint (26.10) into constraint (31.2) for problem **P5**.

- 2 *Step 2:* After obtaining the solution of problem **P5**, one can determine the next operation according to the value of η . If $\eta \leq 0$, which means that the amount of the completed tasks is greater than the target one, the current \hat{T} is feasible to **P5**. Otherwise, one has to find a new feasible \hat{T} by using the BSA method.

The main difficulty to address problem **P5** is the non-convex objective function and constraint (26.9). We decompose problem **P5** into the following three subproblems **P5.1**, **P5.2** and **P5.3** and solve them sequentially.

- 1 In problem **P5.1**, the UAV trajectory $\mathbf{Q}_{n,m}$, the computing resource allocation $\mathbf{F}_{n,m}$ and $\mathbf{J}_{n,m}$ are assumed to be known. The objective is to optimize the time allocation $\mathbf{A}_{n,m}$ and $\mathbf{B}_{n,m}$.
- 2 After obtaining the time allocation $\mathbf{A}_{n,m}$ and $\mathbf{B}_{n,m}$, the computing resource allocation $\mathbf{F}_{n,m}$ and $\mathbf{J}_{n,m}$ will be optimized in problem **P5.2** with the given UAV trajectory $\mathbf{Q}_{n,m}$.
- 3 Once the time allocation $\mathbf{A}_{n,m}$ and $\mathbf{B}_{n,m}$ and the computing resource allocation $\mathbf{F}_{n,m}$ and $\mathbf{J}_{n,m}$ were known, the UAV trajectory $\mathbf{Q}_{n,m}$ is to be optimized in problem **P5.3**.

Next, we solve problems **P5.1**, **P5.2**, and **P5.3**, respectively.

5.1 The solution to problem P5.1

With the known $\mathbf{Q}_{n,m}$, $\mathbf{F}_{n,m}$, $\mathbf{J}_{n,m}$ and η , problem **P5** is simplified to **P5.1** as follows.

$$\begin{aligned}
 \text{(P5.1)} \quad & \min_{\mathbf{A}_{n,m}, \mathbf{B}_{n,m}} \mu_1 \left(\varepsilon_m^1 \hat{T} + \varepsilon_m^2 \sum_{k \in \mathcal{K}_m} \sum_{t=1}^{\hat{T}} (\gamma_{k_m} (f_{k_m}[t])^3 \tau + p_{k_m} \alpha_{n,k_m}[t] \tau) \right) \\
 & - \mu_2 \rho_n \sum_{k \in \mathcal{K}_m} \sum_{t=2}^{\hat{T}} \frac{\beta_{n,k_m}[t] \tau f_n[t]}{C_n} + \eta \tag{32} \\
 \text{s.t.} \quad & (26.5), (26.6), (26.7), (26.8), (26.9), (26.13), (26.14), (31.2). \tag{32.1}
 \end{aligned}$$

Since the objective and all the constraints are convex, problem **P5.1** is a linear program (LP) problem, which can be readily solved by applying classical optimization methods.

5.2 Solve problem P5.2

After solving problem **P5.1**, the time allocation $\mathbf{A}_{n,m}$ and $\mathbf{B}_{n,m}$ are known. With the same $\mathbf{Q}_{n,m}$ and η , the computing resource allocation $\mathbf{F}_{n,m}$ and $\mathbf{J}_{n,m}$ is further optimized in the following problem **P5.2**

$$\begin{aligned}
 \text{(P5.2)} \quad & \min_{\mathbf{F}_{n,m}, \mathbf{J}_{n,m}} \mu_1 (\varepsilon_m^1 \hat{T} + \varepsilon_m^2 \sum_{k \in \mathcal{K}_m} \sum_{t=1}^{\hat{T}} (\gamma_{k_m} (f_{k_m}[t])^3 \tau + p_{k_m} \alpha_{n,k_m}[t] \tau)) \\
 & - \mu_2 \rho_n \sum_{k \in \mathcal{K}_m} \sum_{t=2}^{\hat{T}} \frac{\beta_{n,k_m}[t] \tau f_n[t]}{C_n} + \eta \tag{33} \\
 \text{s.t.} \quad & (26.4), (26.3), (26.7), (26.9), (26.13), (26.14), (31.2). \tag{33.1}
 \end{aligned}$$

Since the objective and all the constraints of problem **P5.2** are convex with respect to $\mathbf{F}_{n,m}$ or $\mathbf{J}_{n,m}$, problem **P5.2** is a convex optimization problem. One can prove that problem **P5.2** satisfies the Karush–Kuhn–Tucker (KKT) conditions [15]. Next, we leverage the Lagrange method to solve it.

The Lagrangian function of problem **P5.2** is given by

$$\begin{aligned}
 & \mathcal{L}(\mathbf{F}_{n,m}, \mathbf{J}_{n,m}, \lambda, \vartheta, \varphi, \xi) \\
 &= \mu_1 (\varepsilon_m^1 \hat{T} + \varepsilon_m^2 \sum_{k \in \mathcal{K}_m} \sum_{t=1}^{\hat{T}} (\gamma_{k_m} (f_{k_m}[t])^3 \tau + p_{k_m} \alpha_{n,k_m}[t] \tau)) - \mu_2 \rho_n \sum_{k \in \mathcal{K}_m} \sum_{t=2}^{\hat{T}} \frac{\beta_{n,k_m}[t] \tau f_n[t]}{C_n} + \eta \\
 &+ \lambda \left(\sum_{t=1}^{\hat{T}} E_n^F[t] + \sum_{t=1}^{\hat{T}} \sum_{k_m \in \mathcal{K}_m} E_{n,k_m}^C[t] - E_n^{\max} \right) \\
 &+ \sum_{k_m \in \mathcal{K}_m} \vartheta_{k_m} \left(\sum_{t=1}^{\hat{T}} (E_{k_m}^C[t] + E_{n,k_m}^{\text{Tx}}[t]) - E_{k_m}^{\max} \right) \\
 &+ \sum_{k_m \in \mathcal{K}_m} \sum_{t=2}^{\hat{T}} \varphi_{k_m t} \left(\frac{\beta_{n,k_m}[t] \tau f_n[t]}{C_n} - b_{n,k_m}[t] \right) \\
 &+ \sum_{k_m \in \mathcal{K}_m} \xi_{k_m} \left(L_{k_m} - \sum_{t=2}^{\hat{T}} \frac{\beta_{n,k_m}[t] \tau f_n[t]}{C_n} - \sum_{t=1}^{\hat{T}} \frac{f_{k_m}[t] \tau}{C_{k_m}} - \eta \right),
 \end{aligned} \tag{27}$$

where $\lambda, \vartheta = \{\vartheta_{k_m}\}_{k_m \in \mathcal{K}_m}$, $\varphi = \{\varphi_{k_m t}\}_{k_m \in \mathcal{K}_m, 2 \leq t \leq \hat{T}}$, and $\xi = \{\xi_{k_m}\}_{k_m \in \mathcal{K}_m}$ are the non-negative Lagrange multipliers associated with constraints (26.14), (26.13), (26.9), (26.10), respectively.

The Lagrangian dual function of problem **P5.2** is given by

$$\begin{aligned}
 \mathcal{D}(\lambda, \vartheta, \varphi, \xi) &= \min_{\mathbf{F}_{n,m}, \mathbf{J}_{n,m}} \mathcal{L}(\mathbf{F}_{n,m}, \mathbf{J}_{n,m}, \lambda, \vartheta, \varphi, \xi) \\
 & \text{s.t. (26.4), (26.3), (26.7)}.
 \end{aligned} \tag{28}$$

According to the KKT conditions [15], the solution of problem **P5.2** can be obtained by solving the Lagrangian function and the Lagrangian dual function iteratively.

Letting the derivations of the Lagrangian function $\mathcal{L}(\mathbf{F}_{n,m}, \mathbf{J}_{n,m})$ with respect to $\mathbf{F}_{n,m}$ and $\mathbf{J}_{n,m}$ equal to zero, respectively, we can obtain the solution as

$$f_n^*[t] = \left[\sqrt{\frac{\sum_{k_m \in \mathcal{K}_m} ((\mu_2 \varepsilon_m^2 + \xi_{k_m} - \varphi_{k_m t})^+ \beta_{k_m}[t])}{3 \lambda \gamma_n C_n \sum_{k_m \in \mathcal{K}_m} \beta_{k_m}[t]}} \right]_0^{f_n^{\max}}, \quad 2 \leq t \leq \hat{T}, \tag{36.1}$$

$$f_{k_m}^*[t] = \left[\sqrt{\frac{\xi_{k_m}}{3 C_{k_m} \gamma_{k_m} (\mu_1 \varepsilon_m^2 + \vartheta_{k_m})}} \right]_0^{f_{k_m}^{\max}}, \quad 1 \leq t \leq \hat{T}. \tag{36.2}$$

where $[x]^+$ denotes the operation $\max(x, 0)$, and $[f^*]_0^{\max}$ denotes the operation $\min(\max(f^*, 0), f^{\max})$.

We can substitute the obtained $f_n^*[t]$ and $f_{k_m}^*[t]$ into Lagrangian dual function (28) and rewrite it as

$$\max_{\lambda, \vartheta, \varphi, \xi} \mathcal{D}(\lambda, \vartheta, \varphi, \xi) \tag{29}$$

Because dual problem (29) is convex, we can use the subgradient method to solve it.

Let $l = 1, 2, \dots$ denotes the index of the number of iterations. According to ref. [16], the dual variables are updated via iterations as

$$\lambda^{(l+1)} = \left[\lambda^{(l)} - s_\lambda^{(l)} \Delta \lambda^{(l)} \right]^+, \forall k_m \in \mathcal{K}_m, \quad (38.1)$$

$$\vartheta_{k_m}^{(l+1)} = \left[\vartheta_{k_m}^{(l)} - s_\vartheta^{(l)} \Delta \vartheta_{k_m}^{(l)} \right]^+, \forall k_m \in \mathcal{K}_m, \quad (38.2)$$

$$\varphi_{k_m,t}^{(l+1)} = \left[\varphi_{k_m,t}^{(l)} - s_\varphi^{(l)} \Delta \varphi_{k_m,t}^{(l)} \right]^+, \forall k_m \in \mathcal{K}_m, 2 \leq t \leq \hat{T}, \quad (38.3)$$

$$\xi_{k_m}^{(l+1)} = \left[\xi_{k_m}^{(l)} - s_\xi^{(l)} \Delta \xi_{k_m}^{(l)} \right]^+, \forall k_m \in \mathcal{K}_m, \quad (38.4)$$

where $s_\lambda^{(l)}, s_\vartheta^{(l)}, s_\varphi^{(l)}$ and $s_\xi^{(l)}$ denote the step sizes of the l^{th} iteration, and $\Delta \lambda^{(l)}, \Delta \vartheta_{k_m}^{(l)}, \Delta \varphi_{k_m,t}^{(l)}$ and $\Delta \xi_{k_m}^{(l)}$ are the subgradients of the dual function $\mathcal{D}^{(l)}(\lambda, \vartheta, \varphi, \xi)$ with respect to $\lambda, \vartheta, \varphi$ and ξ in the l^{th} iteration [16], respectively, which are given as

$$\Delta \lambda^{(l)} = \sum_{t=1}^{\hat{T}} E_n^F[t] + \sum_{t=1}^{\hat{T}} \sum_{k_m \in \mathcal{K}_m} E_{n,k_m}^C[t] - E_n^{\max}, \quad (39.1)$$

$$\Delta \vartheta_{k_m}^{(l)} = \sum_{t=1}^{\hat{T}} \left(E_{k_m}^C[t] + E_{n,k_m}^{\text{Tx}}[t] \right) - E_{k_m}^{\max}, \quad (39.2)$$

$$\Delta \varphi_{k_m,t}^{(l)} = \frac{\beta_{n,k_m}[t] \tau f_n[t]}{C_n} - b_{n,k_m}[t], \quad (39.3)$$

$$\Delta \xi_{k_m}^{(l)} = L_{k_m} - \sum_{t=2}^{\hat{T}} \frac{\beta_{n,k_m}[t] \tau f_n[t]}{C_n} - \sum_{t=1}^{\hat{T}} \frac{f_{k_m}[t] \tau}{C_{k_m}} - \eta. \quad (39.4)$$

Let $\Upsilon > 0$. The step sizes are configured as $s_\lambda^{(l)} = \Upsilon / \|\Delta \lambda^{(l)}\|$, $s_\vartheta^{(l)} = \Upsilon / \|\Delta \vartheta_{k_m}^{(l)}\|$, $s_\varphi^{(l)} = \Upsilon / \|\Delta \varphi_{k_m,t}^{(l)}\|$ and $s_\xi^{(l)} = \Upsilon / \|\Delta \xi_{k_m}^{(l)}\|$, respectively. The iterative algorithm to solve problem **P5.2** is summarized in following Algorithm 1.

Algorithm 1 The proposed algorithm to solve problem **P5.2**

- 1: $l = 0$;
 - 2: Initialize $\lambda^{(0)}, \vartheta_{k_m}^{(0)}, \varphi_{k_m,t}^{(0)}$ and $\xi_{k_m}^{(0)}$;
 - 3: $\mathcal{D}^0 = \mathcal{D}(\lambda^{(0)}, \vartheta_{k_m}^{(0)}, \varphi_{k_m,t}^{(0)}, \xi_{k_m}^{(0)})$;
 - 4: **repeat**
 - 5: Update $f_n[t]$ and $f_{k_m}[t]$ by using eqs. (36.1) and (36.2), respectively;
 - 6: Update $\mathcal{D}^{(l)}$ by using eq. (35);
 - 7: Update $\lambda^{(l)}, \vartheta_{k_m}^{(l)}, \varphi_{k_m,t}^{(l)}$ and $\xi_{k_m}^{(l)}$ by using eq. (38);
 - 8: $\mathcal{D}^* = \max(\mathcal{D}^{(1)}, \dots, \mathcal{D}^{(l-1)})$;
 - 9: $l = l + 1$;
 - 10: **until** $|\mathcal{D}^* - \mathcal{D}^{(l)}| \leq \varpi$
-

In the algorithm, $\varpi \geq 0$ is defined as the threshold for judging the convergence of the iteration. When $|\mathcal{D}^* - \mathcal{D}^{(l)}| \leq \varpi$, the algorithm terminates.

5.3 Solving problem **P5.3**

Now, the time resource allocation $\mathbf{A}_{n,m}$ and $\mathbf{B}_{n,m}$, and the computing resource allocation $\mathbf{F}_{n,m}$ and $\mathbf{J}_{n,m}$ are known; we can further optimize η and $\mathbf{Q}_{n,m}$ by solving the following problem.

$$\begin{aligned}
 (\mathbf{P5.3}) \quad & \min_{\mathbf{Q}_{n,m}, \eta} \mu_1 (\varepsilon_m^1 \hat{T} + \varepsilon_m^2 \sum_{k \in \mathcal{K}_m} \sum_{t=1}^{\hat{T}} (\gamma_{k_m} (f_{k_m}[t])^3 \tau + p_{k_m} \alpha_{n,k_m}[t] \tau)) \\
 & - \mu_2 \rho_n \sum_{k \in \mathcal{K}_m} \sum_{t=2}^{\hat{T}} \frac{\beta_{n,k_m}[t] \tau f_n[t]}{C_n} + \eta \\
 \text{s.t.} \quad & (26.9), (26.11), (26.12), (26.14), (31.2).
 \end{aligned} \quad (40)$$

$$\text{s.t.} \quad (26.9), (26.11), (26.12), (26.14), (31.2). \quad (40.1)$$

Because constraints (26.9), (26.11), (26.12), and (26.14) are related only to $\mathbf{Q}_{n,m}$ the trajectory of the UAV but not to the objective function (), we decompose problem **P5.3** into the following two subproblems.

First subproblem **P5.4** given below is to optimize the introduced variable η under constraint (31.2).

$$\begin{aligned}
 (\mathbf{P5.4}) \quad & \min_{\eta} \mu_1 \left(\varepsilon_m^1 \hat{T} + \varepsilon_m^2 \sum_{k \in \mathcal{K}_m} \sum_{t=1}^{\hat{T}} (\gamma_{k_m} f_{k_m}[t])^3 \tau + p_{k_m} \alpha_{n,k_m}[t] \tau \right) \\
 & - \mu_2 \rho_n \sum_{k \in \mathcal{K}_m} \sum_{t=2}^{\hat{T}} \frac{\beta_{n,k_m}[t] \tau f_n[t]}{C_n} + \eta, \tag{41} \\
 \text{s.t.} \quad & \sum_{t=2}^{\hat{T}} \frac{\beta_{n,k_m}[t] \tau f_n[t]}{C_n} + \sum_{t=1}^{\hat{T}} \frac{f_{k_m}[t] \tau}{C_{k_m}} + \eta \geq L_{k_m}, \forall k_m \in \mathcal{K}_m. \tag{41.1}
 \end{aligned}$$

This is a linear programming (LP) problem.

Because $\mathbf{Q}_{n,m}$ the trajectory of the UAV has no effect on the objective function (), we only need to find a feasible solution of $\mathbf{Q}_{n,m}$ satisfying constraints (26.9), (26.11), (26.12), and (26.14). To minimize the energy consumption of an UAV to complete a mission, we get problem **P5.5**, i.e., the second subproblem of problem **P5.3** as follows.

$$\begin{aligned}
 (\mathbf{P5.5}) \quad & \min_{\mathbf{Q}_{n,m}} \sum_{t \in \mathcal{T}} E_n^F[t], E_n^F[t] = 0.5 \zeta \| v_n[t] \|^2 \tau, \tag{42} \\
 \text{s.t.} \quad & (26.9), (26.11), (26.12), (26.14). \tag{42.1}
 \end{aligned}$$

It is noted that constraint (26.9) is non-convex with respect to $\mathbf{Q}_{n,m}$. Next, we address this difficulty by using the successive convex approximation (SCA) method. We first convert constraint (26.9) into the following form.

$$Z_{n,k_m}[t] = \sum_{i=2}^{\hat{T}} c_{n,k_m}[i] - \sum_{i=1}^{\hat{T}-1} \alpha_{n,k_m}[i] \tau R_{n,k_m}[i] \leq 0. \tag{30}$$

During any slot t , the coordinate of UAV n is given by $q_{n,m}[t] = (x_{n,m}[t], y_{n,m}[t], H)$. We can get the first derivatives of $Z_{n,k_m}[t]$ with respect to $x_{n,m}[t]$ and $y_{n,m}[t]$ as

$$\nabla_{x_{n,m}[t]} Z_{n,k_m}[t] = \frac{2B\tau g_0 p_{k_m} / \sigma^2}{\|q_{n,m}[t] - \omega_{k_m}\|^2 \ln 2} \frac{\alpha_{n,k_m}[t] (x_{n,m}[t] - x_{k_m})}{\|q_{n,m}[t] - \omega_{k_m}\|^2 + g_0 p_{k_m} / \sigma^2}, \tag{31}$$

and

$$\nabla_{y_{n,m}[t]} Z_{n,k_m}[t] = \frac{2B\tau g_0 p_{k_m} / \sigma^2}{\|q_{n,m}[t] - \omega_{k_m}\|^2 \ln 2} \frac{\alpha_{n,k_m}[t] (y_{n,m}[t] - y_{k_m})}{\|q_{n,m}[t] - \omega_{k_m}\|^2 + g_0 p_{k_m} / \sigma^2}, \tag{32}$$

respectively.

Lemma 1 $\nabla_{x_{n,m}[t]} Z_{n,k_m}[t]$ and $\nabla_{y_{n,m}[t]} Z_{n,k_m}[t]$ are Lipschitz continuous on $x_{n,m}[t]$ and $y_{n,m}[t]$ with L_x and L_y , respectively, where L_x and L_y are the Lipschitz constants.

1 Proof

Please refer to Appendix A. \square

According to ref. [17], if the first derivative of a non-convex function is Lipschitz continuous on the variables with the corresponding Lipschitz constants and each variable is non-empty, closed, and convex, then, the non-convex function can be converted to a convex one by using the SCA method. **Lemma 1** indicates that constraint (26.9) meets this rule and thus can be converted to a convex one by using the SCA method. The specific conversion steps are as follows.

Let $l = 1, 2, \dots$ denote the index of iterations. For the resource allocation after iteration l in slot t , the location of UAV n is represented as $q_{n,m}^{(l)}[t]$ and the transmission rate from IoTD k_m to UAV n is represented as $R_{n,k_m}^{(l)}[t]$. Let $\Delta q_{n,m}^{(l)}$ denote the change in the position of UAV n between iterations l and $l + 1$. After iteration $l + 1$, the location of UAV n is given by

$$q_{n,m}^{(l+1)}[t] = q_{n,m}^{(l)}[t] + \Delta q_{n,m}^{(l)}, \tag{33}$$

and the transmission rate from IoTD k_m to the n^{th} UAV is given by

$$R_{n,k_m}^{(l+1)}[t] = B \log_2 \left(1 + \frac{g_0 p_{k_m}}{\|q_{n,m}^{(l)}[t] + \Delta q_{n,m}^{(l)} - w_{k_m}\|^2 \sigma^2} \right). \tag{34}$$

Lemma 2 With the given $q_{n,m}^{(l)}$, the definite lower bound for $R_{n,k_m}^{(l+1)}[t]$ is given by

$$R_{n,k_m}^{\text{lb}}[t] \triangleq R_{n,k_m}^{(l)}[t] - U_{n,k_m}^{(l)}[t] \left(2(q_{n,m}^{(l)}[t] - \omega_{k_m}) \right) \left(\Delta q_{n,m}^{(l)}[t] \right)^{\top} + \Delta q_{n,m}^{(l)}[t] \left(\Delta q_{n,m}^{(l)}[t] \right)^{\top}. \tag{35}$$

where $U_{n,k_m}^{(l)}[t] = \frac{2B\tau g_0 p_{k_m} / \ln 2}{\|q_{n,m}^{(l)}[t] - \omega_{k_m}\|^2 (\|q_{n,m}^{(l)}[t] - \omega_{k_m}\|^2 + g_0 p_{k_m} / \sigma^2)}$, and operation \top represents taking the transposing of a matrix.

1 Proof

Please refer to Appendix B. \square

According to ref. [18], a strong convex function to approximate constraint (26.9) can be constructed as

$$\sum_{i=2}^t c_{n,k_m}[i] - \sum_{i=1}^{t-1} \alpha_{n,k_m}[t] \tau R_{n,k_m}^{\text{lb}}[t] \leq 0, \quad 2 \leq t \leq \hat{T}. \tag{36}$$

By replacing constraint (26.9) with constraint (36), problem **P5.5** can be converted to a convex one as follows.

$$\begin{aligned} \text{(P5.6)} \quad & \min_{\mathbf{Q}_{n,m}} \sum_{t \in \mathcal{T}} E_n^{\text{F}}[t], \quad E_n^{\text{F}}[t] = 0.5\zeta \|v_n[t]\|^2 \tau, \quad (50) \\ \text{s.t.} \quad & (26.11), (26.12), (26.14), (49). \quad (50.1) \end{aligned}$$

This problem can be solved by using convex optimization tools, e.g., CVX [19]. The detail of the SCA algorithm is presented in following Algorithm 2.

Algorithm 2 The SCA based algorithm to solve problem **P5.6**

```

1: Initialize  $l = 0$ .
2: Find a feasible  $q_{n,m}^{(0)}[t]$ .
3: repeat
4:   For a given  $q_{n,m}^{(l)}[t]$ , find the optimal  $\Delta q_{n,m}^{(l)}[t]$  by solving problem P5.6;
5:    $q_{n,m}^{(l+1)}[t] = q_{n,m}^{(l)}[t] + \Delta q_{n,m}^{(l)}[t]$ ;
6:    $l = l + 1$ ;
7: until the objective function of problem P5.6 converges, or the maximum number of iterations is
   reached.
8: return  $\mathbf{Q}_{n,m}$  and the value of the objective function of problem P5.6.

```

5.4 Overall algorithm for solving problem P2

The overall algorithm for solving problem **P5** is summarized in following Algorithm 3.

Algorithm 3 The overall algorithm for solving problem **P5**.

```

1:  $l = 0$ .
2: Initialize  $\mathbf{F}_{n,m}^{(l)}$ ,  $\mathbf{J}_{n,m}^{(l)}$ ,  $\mathbf{Q}_{n,m}^{(l)}$  and  $\eta^{(l)}$ .
3: repeat
4:   With the given  $\mathbf{F}_{n,m}^{(l)}$ ,  $\mathbf{J}_{n,m}^{(l)}$ ,  $\mathbf{Q}_{n,m}^{(l)}$  and  $\eta^{(l)}$ , solve problem P5.1 in (32) by using the LP method
   and obtain  $\mathbf{A}_{n,m}^{(l)}$  and  $\mathbf{B}_{n,m}^{(l)}$ .
5:   With the known  $\mathbf{A}_{n,m}^{(l)}$ ,  $\mathbf{B}_{n,m}^{(l)}$ ,  $\mathbf{Q}_{n,m}^{(l)}$  and  $\eta^{(l)}$ , solve problem P5.2 in (33) by using Algorithm 1
   and update  $\mathbf{F}_{n,m}^l$  and  $\mathbf{J}_{n,m}^l$  to  $\mathbf{F}_{n,m}^{l+1}$  and  $\mathbf{J}_{n,m}^{l+1}$ , respectively.
6:   With the known  $\mathbf{A}_{n,m}^{(l)}$ ,  $\mathbf{B}_{n,m}^{(l)}$ ,  $\mathbf{F}_{n,m}^{(l+1)}$  and  $\mathbf{J}_{n,m}^{(l+1)}$ , solve problem P5.3 in (40) by using Algorithm 2
   and update  $\mathbf{Q}_{n,m}^{(l)}$  and  $\eta^{(l)}$  to  $\mathbf{Q}_{n,m}^{(l+1)}$  and  $\eta^{(l+1)}$ , respectively.
7:    $l = l + 1$ .
8: until the algorithm converges or reaches a certain number of iterations  $l = \hat{l}$ .

```

As problem **P2** is equivalent to problem **P4**, the overall algorithm for solving problem **P2/P4** is summarized in following Algorithm 4.

Algorithm 4 The overall algorithm based on BSA for solving problem **P2/P4**

```

1: Initialize  $T^{\min}$  and  $T^{\max}$  as the lower and upper bounds to find the optimal  $T$ .
2: repeat
3:    $\hat{T} = \left\lceil \frac{T^{\min} + T^{\max}}{2} \right\rceil$ ;
4:   With the given  $\hat{T}$ , obtain  $\mathbf{A}_{n,m}$ ,  $\mathbf{B}_{n,m}$ ,  $\mathbf{F}_{n,m}$ ,  $\mathbf{J}_{n,m}$ ,  $\mathbf{Q}_{n,m}$  and  $\eta$  by performing Algorithm 3.
5:   if  $\eta \geq 0$  then
6:      $T^{\min} = \hat{T}$ .
7:   else
8:      $T^{\max} = \hat{T}$ .
9:   end if
10: until  $(T^{\max} - T^{\min}) \leq 1$ .
11: Let  $T = T^{\min}$ . Obtain  $\mathbf{A}_{n,m}^*$ ,  $\mathbf{B}_{n,m}^*$ ,  $\mathbf{F}_{n,m}^*$ ,  $\mathbf{J}_{n,m}^*$ ,  $\mathbf{Q}_{n,m}^*$  and  $\eta$  by performing Algorithm 3.
12: if  $\eta \geq 0$  then
13:   Let  $T = T^{\max}$ .
14:   Obtain  $\mathbf{W}_{n,m}^*$ ,  $\mathbf{G}_{n,m}^*$ ,  $\mathbf{A}_{n,m}^*$ ,  $\mathbf{B}_{n,m}^*$ ,  $\mathbf{F}_{n,m}^*$ ,  $\mathbf{J}_{n,m}^*$ ,  $\mathbf{Q}_{n,m}^*$  by performing Algorithm 3.
15:   return  $\mathbf{W}_{n,m}^*$ ,  $\mathbf{G}_{n,m}^*$ ,  $\mathbf{A}_{n,m}^*$ ,  $\mathbf{B}_{n,m}^*$ ,  $\mathbf{F}_{n,m}^*$ ,  $\mathbf{J}_{n,m}^*$ ,  $\mathbf{Q}_{n,m}^*$  and  $T^* = T$ .
16: else
17:   return 0 (which means that no feasible solution was found).
18: end if

```

6 Solving problem P3

Given that UAV n were assigned to GSA m , the profit $G_{m,n}$ for the GSA and the profit $W_{n,m}$ for the UAV can be obtained by using proposed Algorithm 4. When all the possible profit combinations $\{G_{n,m}, W_{n,m}\}, \forall n \in \mathcal{N}, \forall m \in \mathcal{M}$ were known, we can further solve problem **P3**, which aims to schedule the finite UAVs to the GSAs by optimizing parameter Θ . Recall that $\Theta \triangleq \{\theta_{n,m}\}$ represents the association matrix of the UAVs and GSAs.

Problem **P3** is a multi-objective optimization problem, where an GSA wants to get better service with less budget by hiring a more suitable UAV, while a UAV wants to get higher profit by choosing an GSA to serve. Since the association between the UAVs and GSAs determines the profits of both parties, the *social welfare* resource allocation should be considered. In what follows, we map problem **P3** into a stable marriage matching problem and a weak Pareto optimal matching result can be obtained.

6.1 Matching problem modeling

The stable marriage problem is one of the classical models in two-side matching theory, where a set of men propose to their preferred women in order of preference, and a set of women decide whether to accept the proposal. The aim is to find a stable matching between the two sets of elements given an ordering of preferences for each element.

In problem **P3**, the UAV set and the GSA set are two separated sets of rational and selfish participants who care about only their own profits. For any UAV n , the profit $W_{n,m}$ represents the preference of UAV n to GSA m , and, for any GSA m , the profit $G_{n,m}$ represents the preference of the GSA m to UAV n . Note that all the preference values $W_{n,m}$ and $G_{n,m}$ ($\forall n \in \mathcal{N}, \forall m \in \mathcal{M}$) can be obtained by performing Algorithm 4. A GSA first sends its service request to the UAVs, and then each of the UAVs decides whether to serve it or not. The mutual choice $\Theta \triangleq \{\theta_{n,m}\}$ between the UAVs and GSAs depends on the preference values $W_{n,m}$ and $G_{n,m}$.

According to the above analysis, we can transfer problem **P3** into a two-sided matching problem as $(\mathcal{N}, \mathcal{M}, \succ_n, \succ_m)$, where \succ_n and \succ_m denote the comparison of preferences. For instance, $m_1 \succ_n m_2$ ($\forall m_1, m_2 \in \mathcal{M}$ and $m_1 \neq m_2$) represents that UAV n prefers GSA m_1 to GSA m_2 , which means

$$m_1 \succ_n m_2 \Leftrightarrow W_{n,m_1} > W_{n,m_2}. \tag{37}$$

Similarly, $n_1 \succ_m n_2$ ($\forall n_1, n_2 \in \mathcal{N}$ and $n_1 \neq n_2$) represents that GSA m prefers UAV n_1 to UAV n_2 , which means

$$n_1 \succ_m n_2 \Leftrightarrow G_{n_2,m} > G_{n_1,m}. \tag{38}$$

6.2 Algorithm designing

Based on the obtained preference values $W_{n,m}$ and $G_{n,m}$ ($\forall n \in \mathcal{N}$ and $\forall m \in \mathcal{M}$), any UAV n (or GSA m) can define the preference list as $\mathcal{P}\mathcal{L}_n^U = \{W_{n,m}^{(y)} | 1 \leq y \leq M, m \in \mathcal{M}\}$ (or $\mathcal{P}\mathcal{L}_m^G = \{G_{n,m}^{(z)} | 1 \leq z \leq N, n \in \mathcal{N}\}$), where y (in descending order) and z (in ascending order) represent the ranking of the preference values. In addition to *social*

optimality, stability is also an important measure of two-sided matching results. According to [12], the one-to-one stable matching is defined as follows.

Definition 1 If there exists two pair (n_1, m_1) and (n_2, m_2) , which satisfy $n_1 \succ_{m_2} n_2$ and $m_2 \succ_{n_1} m_1$, i.e., n_1 and m_2 have the motivation to leave their current partner and form a new pair with each other, (n_1, m_2) form a blocking pair. A two-sided matching without blocking pairs is a two-sided stable matching.

Definition 1 gives the stability condition of a matching, which also indicates the weak Pareto optimality, since no one can be better off by deviating to another choice under stability condition [20]. By applying the Gale-Shapley (GS) algorithm, an effective method to achieve the stability and weak Pareto optimality (which are ensured following the proofs in [20]) for the proposed matching problem is given in following Algorithm 5. In the algorithm, lines 1 to 7 obtain the preference lists $\mathcal{P}\mathcal{L}_n^U$ and $\mathcal{P}\mathcal{L}_m^G$ for the UAVs and GSAs by performing proposed Algorithm 4. Then the GS algorithm in lines 10 to 24 builds a stable one-to-one match based on $\mathcal{P}\mathcal{L}_n^U$ and $\mathcal{P}\mathcal{L}_m^G$. In lines 10 to 12, each GSA m sends a service request to the most preferred UAV according to the preference list $\mathcal{P}\mathcal{L}_m^G$. Then, in lines 14 to 24, each UAV n chooses to accept or reject the GSA according to the preference list $\mathcal{P}\mathcal{L}_n^U$. Finally, when set $\overline{\mathcal{M}}$ is empty, a stable one-to-one matching/association $\Theta \triangleq \{\theta_{n,m}\}$ between the UAVs and GSAs is obtained.

6.3 Computational complexity of the overall algorithm

The overall algorithm to solve **P1** is Algorithm 5. We analyze its computational complexity as follows:

- 1 The first part of Algorithm 5 is to obtain preference lists of the UAVs and GSAs by executing Algorithm 4, and Algorithm 4 is essentially a binary search algorithm (BSA) which is implemented by repeatedly executing Algorithm 3.
 - (a) The first step of Algorithm 3 is to solve the linear programming problem **P5.1** by using the standard optimization solvers such as CVX. The complexity is $O(K_m^3 T^3)$ [21].
 - (b) The second step of Algorithm 3 is to solve problem **P5.2** by using Lagrange method. Let $\nu \geq 0$ denote the ordered accuracy of the subgradient method. The complexity is $O(1/\nu^2)$ [7].
 - (c) The third step of Algorithm 3 is to solve problem **P5.3** by using the SCA. Let l denote the number of iterations. The complexity is $O(lT^3)$ [22].
 - (d) Since the time complexity of BSA is $O(\log T^{\max})$, the time complexity of Algorithm 4 is given as $O(K_m^3 T^3 \log T^{\max})$. In order to obtain the preference lists of all the UAVs and GSAs, Algorithm 4 needs to be repeated NM times. Therefore, the complexity of the first part of Algorithm 5 is given as $O(NMK_m^3 T^3 \log T^{\max})$.
- 2 The second part of Algorithm 5 is based on the GS algorithm. The complexity is given by as $O(NM)$ [23].

Based on the above analysis, the computational complexity of Algorithm 5 is given as $O(NMK_m^3 T^3 \log T^{\max})$. Finally, we analyze the convergence of Algorithm 5 as follows:

- 1 In line 4 of Algorithm 5, Algorithm 4 based on the BSA can converge when $(T^{\max} - T^{\min}) \leq 1$.
- 2 In Algorithm 4, Algorithm 3 is performed repeatedly to find the lower bound of the objective function of problem **P5**. It decreases monotonically as the iterative number l increases.

According to the above analysis, the convergence of the overall algorithm Algorithm 5 can be guaranteed.

Algorithm 5 Stable matching algorithm for solving associations and overall algorithm to get final solution

```

1: Initialize  $\mathcal{P}\mathcal{L}_n^U = \emptyset$  and  $\mathcal{P}\mathcal{L}_m^G = \emptyset$ 
2: for  $\forall m \in \mathcal{M}$  do
3:   for  $\forall n \in \mathcal{N}$  do
4:     Obtain  $W_{n,m}$  and  $G_{n,m}$  by using Algorithm 4.
5:     Add the newly obtained  $W_{n,m}$  and  $G_{n,m}$  into the preference lists  $\mathcal{P}\mathcal{L}_n^U$  and  $\mathcal{P}\mathcal{L}_m^G$ , respectively, and keep  $\mathcal{P}\mathcal{L}_n^U$  and  $\mathcal{P}\mathcal{L}_m^G$  in order.
6:   end for
7: end for
8: Let  $\overline{\mathcal{M}} = \mathcal{M}$ .
9: while  $\overline{\mathcal{M}} \neq \emptyset$  do
10:  for  $\forall m \in \overline{\mathcal{M}}$  do
11:    GSA  $m$  sends a service request to its most preferred UAV in the preference list  $\mathcal{P}\mathcal{L}_m^G$ .
12:    Remove the most preferred UAV from the preference list  $\mathcal{P}\mathcal{L}_m^G$ .
13:  end for
14:  for  $\forall n \in \mathcal{N}$  do
15:    if UAV  $n$  receives a service request from GSA  $m$  and prefers this GSA to the previous matching candidate then
16:      Reject the previous matching candidate.
17:      Accept the request of GSA  $m$ .
18:      Add the previous candidate to  $\overline{\mathcal{M}}$ .
19:      Remove GSA  $m$  from  $\overline{\mathcal{M}}$ .
20:    else
21:      UAV  $n$  rejects GSA  $m$ .
22:      GSA  $m$  removes UAV  $n$  from the preference list  $\mathcal{P}\mathcal{L}_m^G$ .
23:    end if
24:  end for
25: end while
26: return the matching/association result  $\Theta \triangleq \{\theta_{n,m}\}$  between the UAVs and GSAs, and the corresponding flight trajectory  $\mathbf{Q}$  of the UAVs and resource allocation  $\mathbf{A}, \mathbf{B}, \mathbf{F}, \mathbf{J}$  for the IoTs.

```

7 Results and discussion

The performance of the proposed algorithms is evaluated from the following two aspects. Firstly, we show the impact of the task load change of GSAs on the system performance, while the flight speed and computing power of the UAVS are fixed. Then, we fix the task load of the GSAs and show the impact of the UAV power on the system performance. To illustrate the superiority of the proposed algorithm, the following two schemes are also simulated. The first is termed as the *fixed time allocation* scheme where the value of $\alpha_{n,k_m}[t]$ and $\beta_{n,k_m}[t]$ is fixed to τ/K_m [24]. The second is termed as the *full offload* scheme where each GSA offloads all the task data to a UAV for remote computing [24]. The basic simulation parameters are listed in Table 1 unless specified otherwise.

Table 1 Simulation parameters

Parameter	Symbol	Value
The channel bandwidth	B	30 MHz
The coefficient depending on the chip architecture	γ_{k_m}, γ_n	10^{-28}
The noise power at the receiver of a UAV	σ^2	-60dBm
The received power at the reference distance of 1 m	g_0	-30dB
Duration of a time slot	τ	1s
Number of CPU cycles required for UAV	C_n	1000 ~ 1200 cycles/bits
Number of CPU cycles required for IoTDS	C_k	1000 ~ 1200 cycles/bits
The flight altitude of the UAV	H	15 ~ 30 m
Maximum velocity of UAV n	v_n^{\max}	15 ~ 40 m/s
The transmission power of IoTD k_m	p_{k_m}	20 ~ 40 dBm

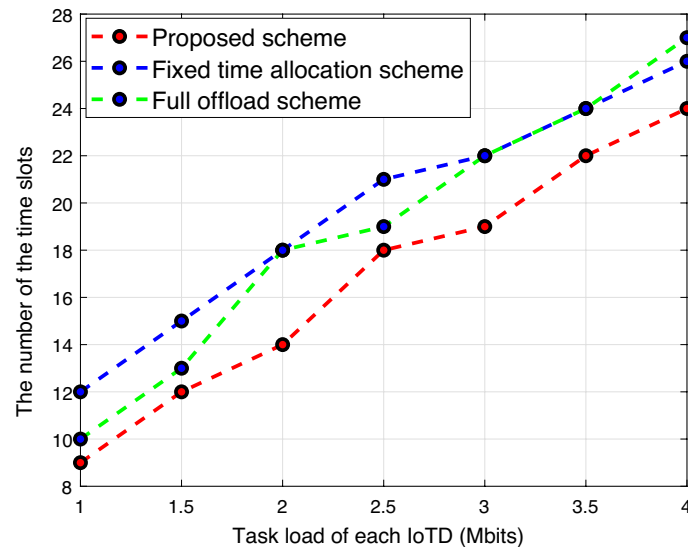


Fig. 3 The variation of required time to complete the computing tasks of the IoTDS

7.1 Effect of task load of GSAs on the system performance

A one-UAV and one-GSA system is simulated. In the GSA, six IoTDS are randomly distributed in an area of $100m \times 100m$. The size of the task input data of an IoTD is initialized to 1 Mb, and 1.2k CPU cycles are required to execute per data bit. The transmission power of each IoTD is set to 30 dBm. The computing power of the UAV is 1.3 GHz, its maximum velocity $v_n^{\max} = 20$ m/s, the flight height $H = 15$ m, and the start and end points $(100m, 100m)$. The task load of each IoTD is increased at a step of 0.5 Mb. Figure 3 shows the time required for each of the schemes to complete all the computing tasks of the system, Fig. 4 shows the energy consumption of the IoTDS in each of the schemes, Fig. 5 shows the profit obtained by the UAV in the proposed scheme, and Fig. 6 shows the task loads undertaken by the UAV and the IoTDS in the proposed scheme, respectively.

From Figs. 3 and 4, we see that as the task load of the IoTDS grows, the time and energy required for the IoTDS to complete the computing tasks increases. In all cases,

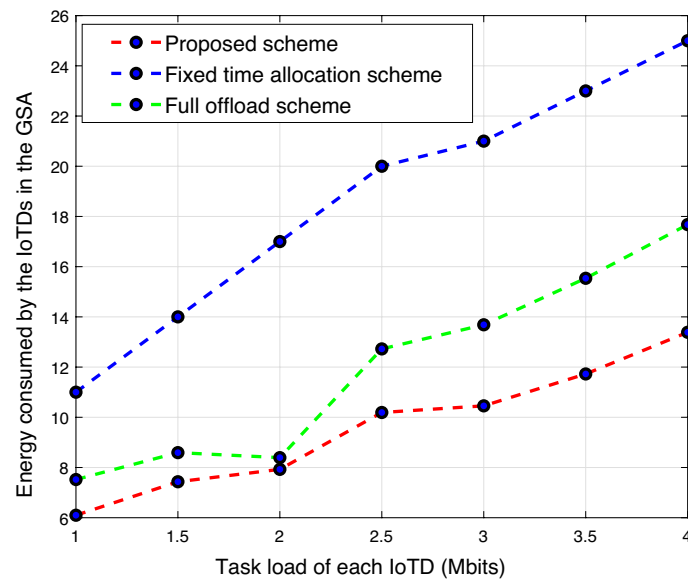


Fig. 4 The variation of energy consumption of the IoTs

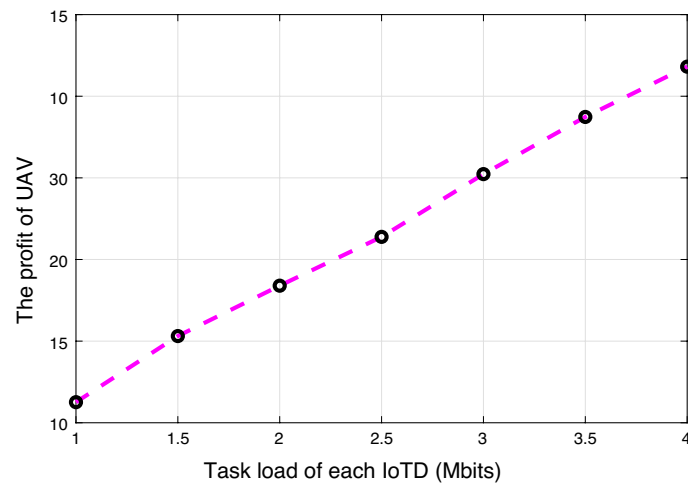


Fig. 5 The profit earned by the UAV for processing the IoTD computing tasks

the proposed scheme outperforms the other two schemes. Because the proposed scheme can flexibly allocate the computing task load to the UAVs and the IoTs, it outperforms the *full offload* scheme. Furthermore, the proposed scheme can also flexibly adjust the time length assignment of the UAVs according to their current computing load, so it outperforms the *fixed time allocation* scheme. From Fig. 5, we see that the profit obtained by the UAV increases with the added task load of the IoTs. This indicates that as the task load grows, the GSA requires the UAV to process more tasks and thus has to pay more payment to the UAV, which incentivizes the UAV to provide more computing resource. As shown in Fig. 6, this is attributed to proposed Algorithm 4 to assign more task load to the UAV when the overall task load of the GSA increases.

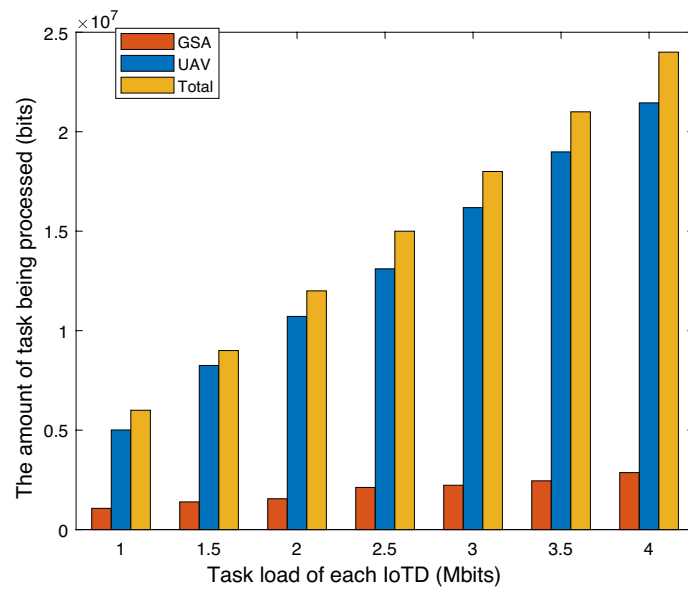


Fig. 6 The distribution of the tasks being processed by the UAV and the IoT devices

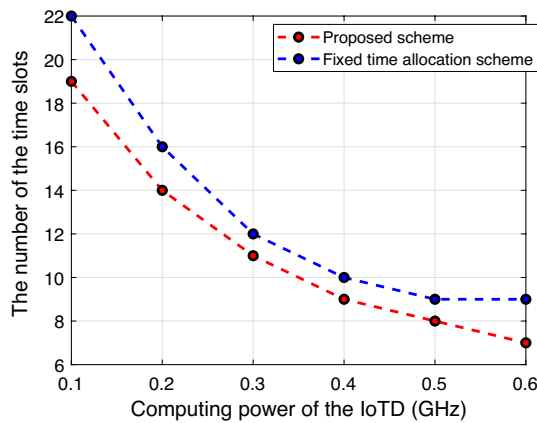


Fig. 7 The variation of required time to complete the computing tasks of the IoT devices

7.2 Effect of computing power of IoT devices on the system performance

To show the effect of the computing power of IoT devices on the system performance, we initialize the CPU frequency of the IoT devices to 0.1 GHz, and keep the task load of the IoT devices at 4 Mb, and increase the computing power of each IoT device at a step of 0.5 GHz. Figures 7 and 8 show the time and energy required for the GSA to complete all the computing tasks of the IoT devices by using the proposed scheme and the *fixed time allocation* scheme, respectively. Figure 9 shows the profit earned by the UAV, and Fig. 10 shows the task load allocated separately to the UAV and the IoT devices.

As shown in Figs. 7 and 8, as the computing power of the IoT devices increases, the time required for completing the computing tasks and the profit earned by the UAV decreases. It indicates that as the computing power of the IoT devices increases, the GSA needs less computing resource of the UAV, thus reducing the payment for the UAV.

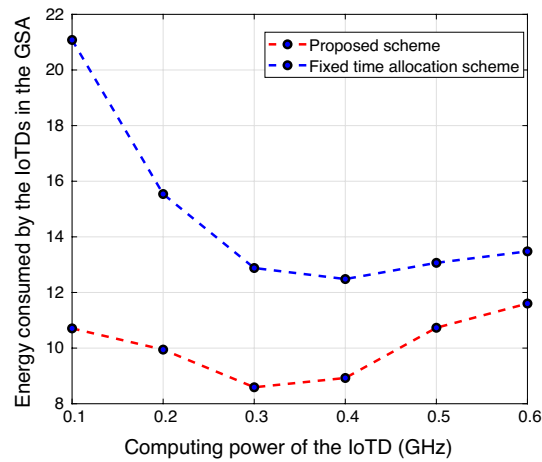


Fig. 8 The variation of energy consumption of the IoTDs

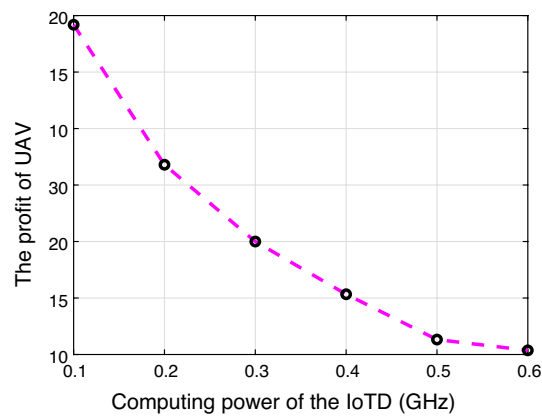


Fig. 9 The profit earned by the UAV for processing the IoTD computing tasks

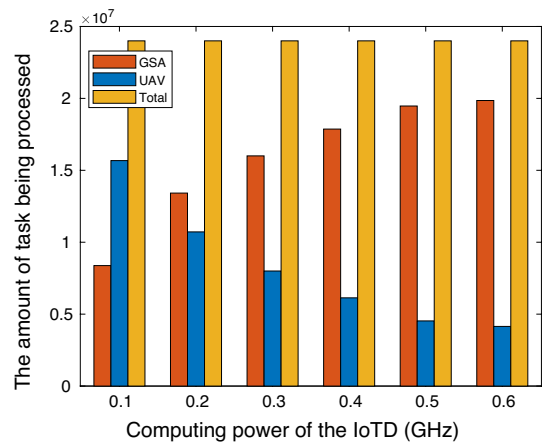


Fig. 10 The distribution of the tasks being processed by the UAV and the IoTDs

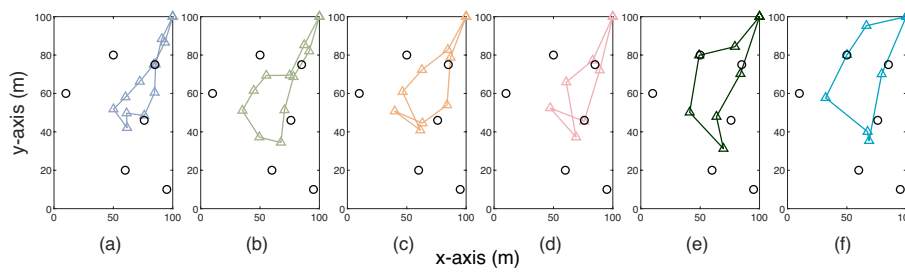


Fig. 11 The trajectory of the UAV with different flight velocities

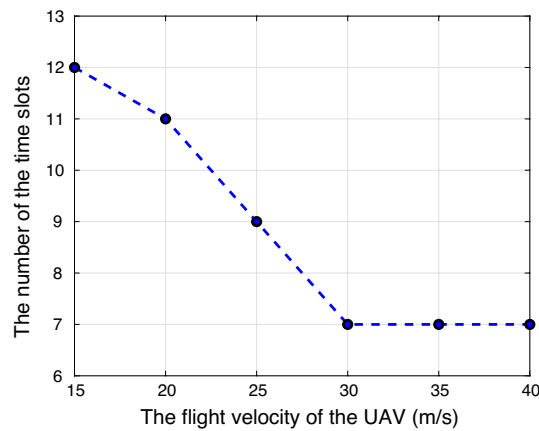


Fig. 12 The overall task completion time of UAV with different flight velocities

Again, the proposed scheme outperforms the *fixed time allocation* scheme in all the cases as it can flexibly adjust the time allocation of each UAV according to the current computing load. From Fig. 10, we can also see that as the computing power of the IoTDs increases, the task load assigned to the UAV decreases gradually.

From Fig. 8, we see that as the computing power of the IoTDs increases, the energy consumed by the IoTDs decreases first and then increases. At the beginning, the computing power of the IoTDs is weak, and the energy is mostly consumed for transmitting task data to the UAV. As the computing power of the IoTDs increases, to save costs, they tend to process the computing tasks locally rather than offload to the UAV. Therefore, the local energy consumption of the IoTDs increases gradually. The above simulation results show that proposed Algorithm 4 is able to make a reasonable distribution of the task load between the UAV and the IoTDs according to the computing power the IoTDs.

7.3 Effect of UAV performance on the system performance

We randomly placed 6 IoTDs in an area of 100 m^2 and flew one UAV to complete the computing tasks generated by the IoTDs. A total of six experiments were performed. In each of the experiments, the flight velocity of the UAV was set to $\{15, 20, 25, 30, 35, 40\}$ m/s, respectively. The optimized flight trajectory of the UAV by using Algorithm 4 is shown in Fig. 11a–f, and the task completion time with different flight velocities is shown in Fig. 12.

From Figs. 11 and 12, we see that for the same task load of the GSA, the higher the flight velocity of the UAV, the shorter the range and time to complete the mission. This indicates that a GNO has an incentive to hire high-performance UAV to serve the IoTDs, and an ASP also has the possibility to make more revenue by employing high-performance UAVs.

7.4 Matching between the UAVs and the GSAs

We divide a $200m \times 200m$ area into 5 GSAs. The numbers of IoTDs in GSAs 1 ~ 5 are set to {5, 4, 5, 3, 6}, respectively. Each of the IoTDs has the same amount of task load. Five UAVs (owned by different ASPs) are to be dispatched to serve the GSAs. The velocities of the UAVs 1 ~ 5 are set to {20, 25, 20, 25, 20} m/s, respectively, and the computing power of the UAVs 1 ~ 5 is set to {1, 0.9, 1.3, 1.1, 1.2} GHz, respectively. Figure 13 shows the profit that each UAV/GSA can achieve when matched to different GSAs/UAVs by performing Algorithm 4. Figure 14 shows the matching result and the flight trajectories of the UAVs obtained by performing Algorithm 5.

8 Conclusion

In this paper, we studied the scheduling and resource allocation problems between multiple UAVs and multiple GSAs in the presence of a conflict of interest between the ASPs and the GNOs. We formulated the competition game among the ASPs and GNOs as a multi-objective optimization problems and solved it in the framework of the matching theory. Simulation results verified that the proposed algorithms achieved the system-wide optimal scheduling of computing resources.

9 Methods/experimental

The purpose of this study is to develop a matching theory-based algorithm to solve the system-wide scheduling and resource allocation problem. The system consists of multiple ASPs and GNOs. Each ASP owns a rotary wing UAV, while each GNO operates a

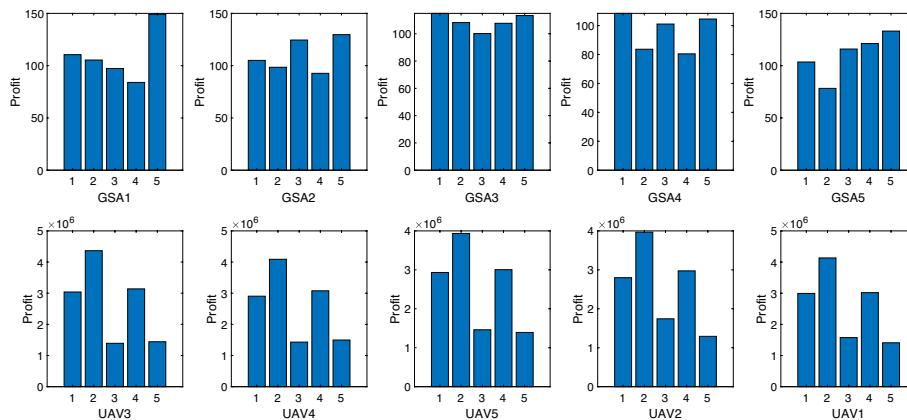


Fig. 13 The profits obtained by the UAVs/GSAs when matching to different GSAs/UAVs by performing Algorithm 4

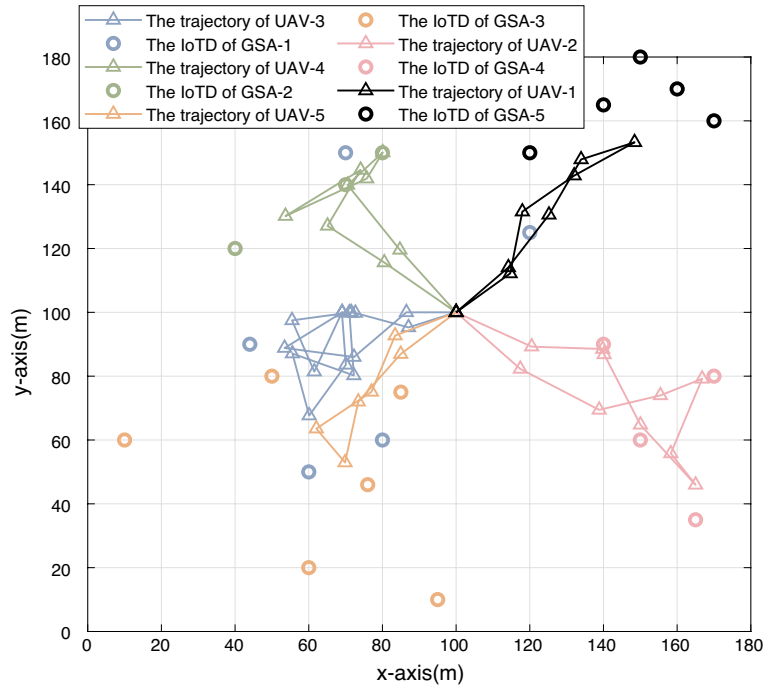


Fig. 14 The matching results between the UAVs and GSAs by performing Algorithm 5

GSA. Each GSA consists of multiple fixed location IoTDs. The proposed algorithm is applied to optimize the association of UAVs and GSAs, resource allocation, and trajectory of UAV. Then, the performance of the algorithm is analyzed experimentally.

Appendix A: The proof of Lemma 1

For easy expression, we define $x = x_{n,m}[t]$ and $H(x) = \nabla_{x_{n,m}[t]} Z_{n,k_m}[t]$. Let $[\underline{x}, \bar{x}]$ denote the range of x . Let x^{\min} and x^{\max} denote the value of x , at which $H(x)$ takes the minimum and maximum values, respectively. From Eq. (30), we know that $H(x)$ is continuously differentiable and its first derivative is bounded within $x \in [\underline{x}, \bar{x}]$. Then we have

$$H(x^{\min}) \leq H(x) \leq H(x^{\max}), \tag{39}$$

and

$$|H(x_1) - H(x_2)| \leq H(x^{\max}) - H(x^{\min}), \forall x_1, x_2 \in [\underline{x}, \bar{x}]. \tag{40}$$

Let $\Delta x = \bar{x} - \underline{x}$. Since $|x_1 - x_2| \leq \Delta x$, we can get

$$|H(x_1) - H(x_2)| \leq L_x \Delta x, \tag{41}$$

where $L_x = \frac{H(x^{\max}) - H(x^{\min})}{\Delta x}$ is the Lipschitz constant of $H(x)$.

Thus, we proved that $\nabla_{x_{n,m}[t]} Z_{n,k_m}[t]$ is Lipschitz continuous on $x_{n,m}[t]$ with the Lipschitz constant L_x . The Lipschitz continuity of $\nabla_{y_{n,m}[t]} Z_{n,k_m}[t]$ can be proved in a similar way which is omitted here.

Appendix B: The proof of Lemma 2

For easy expression, we define $a = \|q_{n,m}^{(l)}[t] - w_{k_m}\|^2$, $b = \frac{g_0 p_{k_m}}{\sigma^2}$ and $x = \|q_{n,m}^{(l+1)}[t] - w_{k_m}\|^2 - \|q_{n,m}^{(l)}[t] - w_{k_m}\|^2$, where a and b can be taken as constants.

We represent $R_{n,k_m}^{(l+1)}[t]$ as

$$f(x) = R_{n,k_m}^{(l+1)}[t] = B \log_2 \left(1 + \frac{b}{a+x} \right). \tag{42}$$

The second derivative of $f(x)$ is given by

$$f''(x) = \frac{B}{\ln 2} \left(\frac{1}{(x+a)^2} - \frac{1}{(x+a+b)^2} \right). \tag{43}$$

From Eq. (43), we know that $f''(x) \geq 0$ and $f(x)$ is convex with respect to x . Then we can use the first-order Taylor expansion of $f(x)$, e.g., $f(x_0) + f'(x_0)(x - x_0)$ to approximate $f(x)$ at any x_0 within the domain of x .

Without loss of generality, we make $x_0 = 0$ and have

$$x - x_0 = (q_{n,m}^{(l+1)}[t] - w_{k_m})(q_{n,m}^{(l+1)}[t] - w_{k_m})^\top - (q_{n,m}^{(l)}[t] - w_{k_m})(q_{n,m}^{(l)}[t] - w_{k_m})^\top, \tag{44}$$

and

$$f'(x_0) = -\frac{B}{\ln 2} \frac{b}{a+b}. \tag{45}$$

By substituting $q_{n,m}^{(l+1)}[t] = q_{n,m}^{(l)}[t] + \Delta q_{n,m}^{(l)}$ into Eq. (44), we can get

$$x - x_0 = 2(q_{n,m}^{(l)}[t] - w_{k_m})\Delta q_{n,m}^{(l)\top} + \Delta q_{n,m}^{(l)}\Delta q_{n,m}^{(l)\top} \tag{46}$$

Let $f(x_0) = R_{n,k_m}^{(l)}[t]$. Then we have

$$R_{n,k_m}^{(l+1)}[t] \geq R_{n,k_m}^{(l)}[t] - U_{n,k_m}^{(l)} \left(2(q_{n,m}^{(l)}[t] - w_{k_m})\Delta q_{n,m}^{(l)\top} + \Delta q_{n,m}^{(l)}\Delta q_{n,m}^{(l)\top} \right), \tag{47}$$

where $U_{n,k_m}^{(l)}[t] = \frac{1}{\ln 2} \frac{B g_0 p_{k_m} / \sigma^2}{(\|q_{n,m}^{(l)}[t] - w_{k_m}\|^2)(g_0 p_{k_m} / \sigma^2 + \|q_{n,m}^{(l)}[t] - w_{k_m}\|^2)}$. This completes the proof (Table 2).

Table 2 Figure title and legend

Figure number	Title of figure	Detailed legend
Figure 1	UAV-assisted MEC in IoT	Figure 1 shows an example of a UAV-assisted MEC scenario, containing multiple UAV and multiple GSA
Figure 2	The workflow of the computing and communication modules of a UAV	Figure 2 shows the discrete-time system model containing T time slots, each of which performs offloading and computation
Figure 3	The variation of required time to complete the computing tasks of the IoTDs	Figure 3 shows the variation of the time required for the system to complete all the computing tasks
Figure 4	The variation of energy consumption of the IoTDs	Figure 4 shows the variation of energy consumption of the IoTDs
Figure 5	The profit earned by the UAV for processing the IoTD computing tasks	Figure 5 shows the variation of the profit obtained by the UAV
Figure 6	The distribution of the tasks being processed by the UAV and the IoTDs	Figure 6 shows the change of computing tasks undertaken by the UAV and the IoTDs
Figure 7	The variation of required time to complete the computing tasks of the IoTDs	Figure 7 shows the variation of the time required for the GSA and UAV to complete all the computing tasks of the IoTDs
Figure 8	The variation of energy consumption of the IoTDs	Figure 8 shows the variation of the energy required for the IoTDs to complete all the computing tasks
Figure 9	The profit earned by the UAV for processing the IoTD computing tasks	Figure 9 shows the variation of the profit earned by the UAV
Figure 10	The distribution of the tasks being processed by the UAV and the IoTDs	Figure 10 shows the variation in the task load processed by the UAV and the IoTDs
Figure 11	The trajectory of the UAV with different flight velocities	Figure 11 shows the optimized flight trajectory of the UAV by using Algorithm 4
Figure 12	The overall task completion time of UAV with different flight velocities	Figure 12 shows the task completion time with different flight velocities
Figure 13	The profits of UAVs and GSAs obtained by performing Algorithm 4	Figure 13 shows the profit that each UAV/GSA can achieve when matched to different GSAs/UAVs by using Algorithm 4
Figure 14	The matching results between the UAVs and GSAs by performing Algorithm 5	Figure 14 shows the matching result for the UAVs to the GSAs as well as the flight trajectories of the UAVs in the GSAs obtained by using Algorithm 5

Abbreviations

UAV	Unmanned aircraft vehicle
MEC	Mobile edge computing
IoT	Internet of Things
ASP	Aerial service provider
GNO	Ground network operator
IoTds	Internet of Things devices
MINLP	Mixed-integer nonlinear programming
SCA	Successive convex optimization
QoS	Quality of service
WPT	Wireless power transfer
MCS	Mobile crowd sensing
NOMA	Non-orthogonal multiple access
GSA	Ground service area
LoS	Line-of-sight
DVFS	Dynamic voltage and frequency scaling
BSA	Binary Search Algorithm
LP	Linear program
KKT	Karush–Kuhn–Tucker
SCA	Successive convex approximation
GS	Gale-Shapley

Acknowledgements

Not applicable

Author contributions

All authors reviewed and edited the manuscript. All authors read and approved the final manuscript.

Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 61971421 and Grant 62132004; in part by the Quzhou Government under Grant 2021D003; and in part by the Sichuan Major Research and Development Project under Grant 22QYCX0168.

Availability of data and materials

Not applicable.

Declarations**Competing interests**

The authors declare that they have no competing interests.

Received: 14 June 2022 Accepted: 2 June 2023

Published online: 22 June 2023

References

1. T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman, D.O. Wu, Edge computing in industrial internet of things: architecture, advances and challenges. *IEEE Commun. Surv. Tutor.* **22**(4), 2462–2488 (2020). <https://doi.org/10.1109/COMST.2020.3009103>
2. F. Vhora, J. Gandhi, A comprehensive survey on mobile edge computing: Challenges, tools, applications, in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 49–55 (2020). <https://doi.org/10.1109/ICCMC48092.2020.ICCMC-0009>
3. M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, X. Shen, Energy-efficient uav-assisted mobile edge computing: resource allocation and trajectory optimization. *IEEE Trans. Veh. Technol.* **69**(3), 3424–3438 (2020). <https://doi.org/10.1109/TVT.2020.2968343>
4. Y. Liu, H.N. Dai, Q. Wang, M. Imran, N. Guizani, Wireless powering internet of things with uavs: challenges and opportunities. *IEEE Netw.* **36**(2), 146–152 (2022). <https://doi.org/10.1109/MNET.013.2000385>
5. Z. Ning, P. Dong, M. Wen, X. Wang, L. Guo, R.Y.K. Kwok, H.V. Poor, 5g-enabled uav-to-community offloading: joint trajectory design and task scheduling. *IEEE J. Sel. Areas Commun.* **39**(11), 3306–3320 (2021). <https://doi.org/10.1109/JSAC.2021.3088663>
6. Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, G.Y. Li, Joint offloading and trajectory design for uav-enabled mobile edge computing systems. *IEEE Internet Things J.* **6**(2), 1879–1892 (2019). <https://doi.org/10.1109/JIOT.2018.2878876>
7. X. Hu, K.K. Wong, K. Yang, Z. Zheng, Uav-assisted relaying and edge computing: scheduling and trajectory optimization. *IEEE Trans. Wirel. Commun.* **18**(10), 4738–4752 (2019). <https://doi.org/10.1109/TWC.2019.2928539>
8. D. Gale, L.S. Shapley, College admissions and the stability of marriage. *Am. Math. Mon.* **69**(1), 9–15 (1962)
9. X. Hu, K.K. Wong, Y. Zhang, Wireless-powered edge computing with cooperative uav: task, time scheduling and trajectory design. *IEEE Trans. Wirel. Commun.* **19**(12), 8083–8098 (2020). <https://doi.org/10.1109/TWC.2020.3019097>
10. C. Zhan, Y. Zeng, Completion time minimization for multi-uav-enabled data collection. *IEEE Trans. Wirel. Commun.* **18**(10), 4859–4872 (2019). <https://doi.org/10.1109/TWC.2019.2930190>
11. Y. Xu, T. Zhang, D. Yang, Y. Liu, M. Tao, Joint resource and trajectory optimization for security in uav-assisted mec systems. *IEEE Trans. Commun.* **69**(1), 573–588 (2021). <https://doi.org/10.1109/TCOMM.2020.3025910>
12. Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, M. Guizani, When mobile crowd sensing meets uav: energy-efficient task assignment and route planning. *IEEE Trans. Commun.* **66**(11), 5526–5538 (2018). <https://doi.org/10.1109/TCOMM.2018.2857461>
13. W. Lu, Y. Ding, Y. Gao, S. Hu, Y. Wu, N. Zhao, Y. Gong, Resource and trajectory optimization for secure communications in dual unmanned aerial vehicle mobile edge computing systems. *IEEE Trans. Ind. Inf.* **18**(4), 2704–2713 (2022). <https://doi.org/10.1109/TII.2021.3087726>
14. K. Deb, K. Deb, *Multi-objective Optimization* (Springer, Boston, 2014), pp.403–449. https://doi.org/10.1007/978-1-4614-6940-7_15
15. S. Boyd, L. Vandenberghe, *Convex Optimization* (Cambridge University Press, New York, 2004)
16. S. Boyd, L. Xiao, A. Mutapic, Subgradient methods. lecture notes of EE392o. Stanford University, Autumn Quarter **2004**, 2004–2005 (2003)
17. D. Wu, F. Wang, X. Cao, J. Xu, Wireless powered user cooperative computation in mobile edge computing systems, in *2018 IEEE Globecom Workshops (GC Wkshps)* (2018), pp. 1–7. <https://doi.org/10.1109/GLOCOMW.2018.8644186>
18. T. Zhang, Y. Xu, J. Loo, D. Yang, L. Xiao, Joint computation and communication design for uav-assisted mobile edge computing in iot. *IEEE Trans. Ind. Inf.* **16**(8), 5505–5516 (2020). <https://doi.org/10.1109/TII.2019.2948406>
19. M. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, version 2.1. (2014). <http://cvxr.com/cvx>
20. G. O'Malley, *Algorithmic Aspects of Stable Matching Problems* (University of Glasgow, Glasgow, 2007)
21. Q. Wu, R. Zhang, Common throughput maximization in uav-enabled ofdma systems with delay consideration. *IEEE Trans. Commun.* **66**(12), 6614–6627 (2018). <https://doi.org/10.1109/TCOMM.2018.2865922>
22. L. Shen, User experience oriented task computation for uav-assisted mec system, in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications* (2022), pp. 1549–1558. <https://doi.org/10.1109/INFOCOM48880.2022.9796968>
23. B. Wang, Y. Sun, S. Li, Q. Cao, Hierarchical matching with peer effect for low-latency and high-reliable caching in social iot. *IEEE Internet Things J.* **6**(1), 1193–1209 (2019). <https://doi.org/10.1109/JIOT.2018.2867617>
24. Y. Luo, W. Ding, B. Zhang, Optimization of task scheduling and dynamic service strategy for multi-uav-enabled mobile-edge computing system. *IEEE Trans. Cogn. Commun. Netw.* **7**(3), 970–984 (2021). <https://doi.org/10.1109/TCCN.2021.3051947>