Number and Operation Time Minimization for Multi-UAV Enabled Data Collection System with Time Windows

Shuai Shen, *Student Member, IEEE,* Kun Yang, *Senior Member, IEEE,* Kezhi Wang, Guopeng Zhang, and Haibo Mei

***Abstract*—In this paper, we investigate multiple unmanned aerial vehicles (UAVs) enabled data collection system in Internet of Things (IoT) networks with time windows, where multiple rotary-wing UAVs are dispatched to collect data from time constrained terrestrial IoT devices. We aim to jointly minimize the number and the total operation time of UAVs by optimizing the UAV trajectory and hovering location. To this end, an optimization problem is formulated considering the energy budget and cache capacity of UAVs as well as the data transmission constraint of IoT devices. To tackle this mix-integer non-convex problem, we decompose the problem into two subproblems: UAV trajectory and hovering location optimization problems. To solve the first subproblem, an modified ant colony optimization (MACO) algorithm is proposed. For the second subproblem, the successive convex approximation (SCA) technique is applied. Then, an overall algorithm, termed MACO-based algorithm, is given by leveraging MACO algorithm and SCA technique. Simulation results demonstrate the superiority of the proposed algorithm.**

***Index Terms*—Time window, UAV trajectory, location optimization, multi-UAV enabled system.**

I. Introduction

INTERNET of Things (IoT) is being applied in many fields for improving the system efficiency and performance, including public safety, smart agriculture, smart grid, and transportation [1], etc. In these IoT networks, the global controllers or gateways can be applied to connect massive devices and collect data from them for further processing [2] [3]. However, the increasing demand for data transmission and the growing density of IoT devices lead to the exponential growth of information exchange in IoT networks [4], which causes great difficulty in data collection. Moreover, the wide-spatial distributed nature along with the diverse requirements of IoT devices also challenge the data collection in IoT networks. Fortunately, the improvement of unmanned aerial vehicle (UAV) makes it a promising technology to alleviate above issues.

Owing to the great mobility and on-demand service ability [5], UAV can serve as a supplement of traditional terrestrial communication systems to improve the coverage range, device access and transmission quality. Because of the significant improvement brought by UAV to the performance of IoT networks, the UAV-enabled systems have received considerable attention recently [6]–[8]. The UAV-enabled data collection system is one of the important issues. In terms of the number of UAVs, the researched systems can be divided into two types: single-UAV and multi-UAV systems. In these two systems, researchers mainly focused on energy efficiency [9]–[16], throughput maximization [14], [17]–[19] and system operation time minimization [14], [20]–[22]. From the aspects of these research directions, the related work in single-UAV and multi-UAV systems is described, respectively, as follows.

1) *Single-UAV system:* Considering the UAV energy budget and data collection requirements, Zhan and Lai minimized the maximum energy consumption of all devices [9]. In two different data collection protocols, Zeng *et al.* minimized the total UAV energy consumption via different methods [10]. In [11], a new UAV-assisted IoT network was proposed to address the high energy consumption issue, in which the UAV is not only a data collector but also a device positioning anchor node. While satisfying data collection and UAV traveling distance constraints, Baek *et al.* maximized the minimum residual energy of sensors after data transmission [12]. Through allocating the hovering and flying time of UAV, Ye *et al.* optimized the sum-throughput of sensors, the total operation time and the total energy consumption of UAV, respectively, in a UAV-enabled wireless-powered IoT network [14]. In order to optimize the network throughput, Feng *et al.* adopted a composite channel model consisting of both large-scale and small-scale fading to depict a typical propagation environment from UAV to IoT devices [17]. Due to the high data volume

Shuai Shen is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the Yangtze Delta Region Institute (Quzhou) & School of Information and Communication Engineering , University of Electronic Science and Technology of China, Quzhou 324000, P. R. China (e-mail: shuaishen@std.uestc.edu.cn).

Kun Yang is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China, the Yangtze Delta Region Institute (Quzhou) & School of Information and Communication Engineering , University of Electronic Science and Technology of China, Quzhou 324000, P. R. China, and the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K(e-mail: kunyang@essex.ac.uk).

Kezhi Wang is with the Department of Computer and Information Science, Northumbria University, Newcastle upon Tyne, NE1 8ST, U.K. (email: kezhi.wang@northumbria.ac.uk).

Guopeng Zhang is with the School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China (email: gpzhang@cumt.edu.cn).

Haibo Mei (*Corresponding author*) is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: haibo.mei@uestc.edu.cn).

in multimedia IoT networks, Jiang *et al.* took into account the cache strategy and location placement of cache-enabled UAV while maximizing the network throughput [19]. With regard to time-constrained IoT networks where the data of IoT devices needs to be uploaded before their expiry deadline, Samir *et al.* maximized the number of IoT devices served by UAV [23].

2) *Multi-UAV system:* With regard to the data collection in the IoT networks where UAVs serve as relays, Fu *et al.* minimized the system power consumption by jointly optimizing the UAV deployment and transmission power [13], and Kim and Qiao proposed an energy-efficient cooperative multi-hop relay scheme for UAVs [15]. For reducing the total transmission power of IoT devices, Liu *et al.* studied the three-dimensional (3D) placement and resource allocation of UAVs an in uplink IoT network [16]. Regarding multi-UAV enabled nonorthogonal multiple access (NOMA) uplink transmission systems, Duan *et al.* proposed an efficient subchannel assignment algorithm to enhance the sum-rate of IoT devices [18]. Considering the completeness of collected data and the energy budget of nodes, Zhan and Zeng minimized the maximum mission completion time in two proposed collection modes [20]. In [21], a multi-UAV data collection framework was proposed, in which the flight time for data collection was minimized. Considering deterministic boundary and ambiguous boundary scenarios for UAVs, Zhang *et al.* proposed two corresponding algorithms to reduce the total flight time [22]. In order to minimize the network costs for data collection, Zhan and Zeng investigated the fundamental tradeoff between aerial cost and ground cost [24].

In UAV-enabled data collection systems, the energy budget of a UAV determines the number of IoT devices it can serve and how far it can fly. Moreover, in practical applications, multimedia data communication is widely applied in IoT networks, which relies on videos and images with high data volume [19] [25]. Therefore, the data size that UAV can store is also an indispensable factor for IoT networks. Although reference [9] minimized the energy consumption of UAV, the energy budget of UAV is still ignored. In [26]–[28], the authors considered content caching for the UAV, but the content is precached and the focus is on the downlink systems. To the best of our knowledge, there is no literature studying both the energy budget and cache capacity of UAV in data collection systems.

In addition, the timeliness of data collection is also vital for some IoT applications demanding up-to-date information, such as emergency rescue, disaster monitoring and target tracking [23]. Moreover, some IoT devices only have limited memory. The data would be overwritten by new data if it is not extracted in time. Furthermore, the IoT devices may not complete information gathering while the UAV arrives at them. Therefore, the IoT device is assumed to have a time window, within which the data transmission must be done. In [23], the IoT devices are also time-constrained, but the UAV collects data while flying. This is not an energy efficient mode. The reason is as follows.

In the existing research, two UAV data collection schemes are mainly adopted: fly-hover-collect scheme and fly-collect scheme [20]. In fly-hover-collect scheme, the UAV only collects data while hovering. In the other scheme, the UAV collects data while flying. Hence, the IoT devices do not need to wait for the arrival of the UAV, which may leads to less data collection latency. However, they have to be operating all the time since they do not know when the UAV flies over and communicates with them. By contrast, the IoT devices in fly-hover-collect scheme can stay off until the UAV reaches them. Therefore, the fly-hover-collect scheme is more energy efficient for IoT devices. In practical applications, most IoT devices are deployed outdoor and battery powered. The huge energy consumption of the synchronization between UAVs and IoT devices will exhaust the device battery easily. Moreover, the operation time of the IoT devices determines the network lifetime. Hence, we adopt the fly-hover-collect scheme in this paper.

In most of the research on UAV-enabled data collection systems, the number of UAVs is fixed. However, in some scenarios, the fixed number of UAVs may not be able to complete the mission due to the energy and cache constraints. Furthermore, from the perspective of network overhead, the purchase and maintenance cost of UAV account for a large proportion in practical applications. Therefore, unnecessary UAVs will result in excessive cost. To the best of our knowledge, no literature has taken into consideration the number of UAVs while designing the UAV trajectory. Although Wang *et al.* minimized the number of UAVs in [29], the locations of UAVs are fixed.

Different from the above work, this paper investigates the data collection in multi-UAV enabled systems with time windows while considering both the energy budget and cache capacity of UAVs, as well as the data transmission constraint of IoT devices. The data collection is operated in the fly-hover-collect scheme proposed by [20]. For reducing the network overhead and data collection duration, we minimize the number and the total operation time of UAVs jointly by optimizing the UAV trajectory and hovering location. The main contributions of this paper are summarized below.

1) We consider practical multi-UAV enabled systems where the energy-and-cache-constrained UAVs fly among the hovering locations corresponding to IoT devices to collect data. Besides the UAV-related constraints, the time window constraint of IoT device is also considered. Under this setup, an optimization problem is formulated to jointly minimize the number and the total operation time of UAVs. Our objective is to design the UAV trajectory and the hovering location.
2) To tackle the formulated mix-integer non-convex problem, we propose an efficient algorithm to search a proper solution by metaheuristic method and successive convex approximation (SCA) technique. The problem is decomposed into two subproblems: UAV trajectory and hovering location optimization problems. The number of UAVs is determined by the binary trajectory variables in the first subproblem. However, the first subproblem is NP-hard, which is difficult to solve by exact optimization approaches. We propose a modified ant colony optimization (MACO) algorithm to solve it. As for the second subproblem, we transform it into a convex problem and

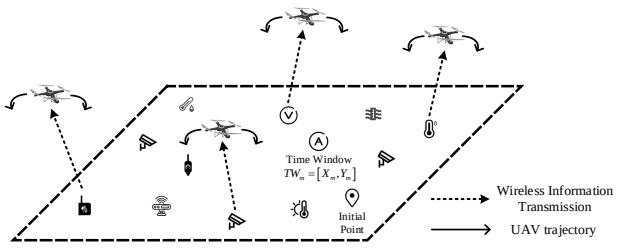


Fig. 1. Multi-UAV enabled data collection system in different IoT scenarios.

solve by SCA technique.

3) Extensive simulations are conducted to validate the performance of the proposed algorithms. The results show that the proposed algorithm outperforms other algorithms in terms of the number and the total operation time of UAVs.

The reminder of this paper is organized as follows. In Section II, the multi-UAV enabled system model with time windows is introduced and the optimization problem is formulated. The decomposition method and the proposed algorithms are presented in Section III. Section IV provides numerical results to verify the performance of our proposed algorithms. The conclusion is drawn in Section VI.

II. System Model and Problem Formulation

A. System Model

Consider K rotary-wing UAVs in set $\mathcal{K}$ are ready to be dispatched to collect data from M IoT devices in set $\mathcal{M}$ on the ground, as shown in Fig. 1. We aim to obtain the minimum number and operation time of UAVs that can complete the mission. For simplicity, K is assumed to be a greatly large constant which is much greater than the number of UAVs that are actually dispatched. The cardinalities of the UAV sets and the IoT device sets are K and M, respectively, i.e. $|\mathcal{K}| = K$ and $|\mathcal{M}| = M$. The UAVs are assumed to take off from the initial point, and then fly at a fixed altitude H to collect data before landing back in a given time limit. The index of the initial point is denoted as 0. It is also assumed that the decision is made by the global controller in the initial point before the UAVs take off, which knows the locations and the time windows of the IoT devices, the energy budget and cache capacity of the UAVs, as well as the channel state information (CSI) between the UAVs and IoT devices in advance. For ease of analysis, the initial point is seen as a special device with no data to upload and the set of all the devices (initial point and IoT devices) is denoted as $\mathcal{D} = \{0, 1, 2, ..., M\}$.

In this system, we adopt the fly-hover-collect scheme proposed by [20]. It is assumed that each IoT device has a corresponding hovering location for UAVs. The UAVs fly at maximum speed from one hovering location to another, and only collect data from one device while hovering at the corresponding hovering location. The association status and sequence of the k-th UAV among the devices is denoted as $\alpha_{i,j}[k]$, which indicates that the k-th UAV flies from the i-th hovering location to the j-th hovering location and then communicates with the j-th device if $\alpha_{i,j}[k] = 1$; otherwise, $\alpha_{i,j}[k] = 0$. In this case, the flow constraint can be expressed as

$$\alpha_{i,j}[k] \in \{0,1\}, \forall k \in \mathcal{K}, \forall i,j \in \mathcal{D}, \tag{1}$$

$$\sum_{j=1}^{M} \alpha_{0,j}[k] = 1, \forall k \in \mathcal{K}, \tag{2}$$

$$\sum_{i=1}^{M} \alpha_{i,0}[k] = 1, \forall k \in \mathcal{K}, \tag{3}$$

$$\sum_{i=0}^{M} \alpha_{i,h}[k] = \sum_{j=0}^{M} \alpha_{h,j}[k], \forall h \in \mathcal{D}, \forall k \in \mathcal{K}, \tag{4}$$

where constraints (2) and (3) state that the UAVs take off from the initial point at the beginning and return to it at the end. Constraint (4) guarantees that each UAV leaves a device if and only if it arrives at that device. It is also assumed that each IoT device only uploads its data to one UAV, then we have

$$\sum_{k=1}^{K} \sum_{i=0}^{M} \alpha_{i,j}[k] = 1, \forall j \in \mathcal{M}. \tag{5}$$

Without loss of generality, we consider a 3D Cartesian coordinate in this model. The horizontal coordinate of the i-th device and the corresponding hovering location are denoted as $\mathbf{s}_i = (x_i^g, y_i^g)$ and $\mathbf{q}_i = (x_i^h, y_i^h)$ $(i \in \mathcal{D})$, respectively. Obviously, the hovering location for the UAV at the initial point is given by

$$\mathbf{q}_0 = \mathbf{s}_0, \tag{6}$$

and the corresponding collection time is zero. By defining the distance between the location of the i-th device and the corresponding hovering location of UAV as d_i, we have

$$d_i = \sqrt{H^2 + \|\mathbf{q}_i - \mathbf{s}_i\|^2}. \tag{7}$$

As illustrated in [30]–[32], the communication links between the UAVs and the IoT devices are dominated by line-of-sight (LoS) links. Furthermore, the Doppler effect due to the UAV mobility is assumed to be perfectly compensated at the receivers. Accordingly, the channel power gain from the i-th device to the UAV at the corresponding hovering location follows the free-space path loss model, which can be expressed as

$$h_i = \rho_0 d_i^{-2} = \frac{\rho_0}{H^2 + \|\mathbf{q}_i - \mathbf{s}_i\|^2}, \tag{8}$$

where ρ_0 represents the channel power gain at the reference distance $d_0 = 1$m.

In the fly-hover-collect scheme, the distance between the UAVs and the overlap of the UAVs' data collection time are changeable. This needs a dynamic bandwidth allocation to avoid interference, which is unrealistic. Therefore, for avoiding the interference between UAVs, we assume that the UAVs equally share the same bandwidth B (in Hz) through frequency division multiplexing (FDM) and thus operate at non-overlapping frequency channel. Hence, the increase of the number of UAVs will decreases the bandwidth of each UAV. Since the fly-hover-collect scheme is adopted, the UAV only communicates with one IoT device at each time. Therefore,

the IoT device uses all the bandwidth of the connected UAV to upload its data. In other words, optimizing the number of UAVs is optimizing the bandwidth of each IoT device. Moreover, it is also assumed that all the IoT devices are allocated with an identical constant transmit power P. In this case, the available transmit rate of the j-th device can be expressed as

$$\begin{aligned} R_j &= \frac{B}{U}\log_2\left(1+\frac{Ph_j}{\sigma^2}\right) \\ &= \frac{B}{U}\log_2\left(1+\frac{\gamma_0}{H^2+\left\|\mathbf{q}_j-\mathbf{s}_j\right\|^2}\right), \end{aligned} \tag{9}$$

where σ^2 is the additive white Gaussian noise (AWGN) power at the UAV receiver, $\gamma_0 = \frac{P\rho_0}{\sigma^2}$ is the received signal-to-noise ratio (SNR) at the reference distance. As mentioned above, $\alpha_{0,j}[k] = 1$ means that the k-th UAV flies from initial point to the j-th hovering location and then communicates with the j-th device, i.e., the k-th UAV sets off to collect data. Therefore, $U = \sum_{j=1}^{M}\sum_{k=1}^{K}\alpha_{0,j}[k]$ in (9) denotes the number of UAVs.

In practical applications, an IoT device may have several pieces of data to upload. For the case that multiple pieces of data have a same time window, they can be seen as a single data stream. Furthermore, for the case that different pieces of data have different time windows, each piece of data can be seen as a uploading task. Then, the device can be seen as multiple devices with different single task at the same location. For easy of analysis, it is assumed that each IoT device has only one piece of data that needs to be uploaded.

In order to ensure the integrity of the uploaded data, the collection time $\tau_{k,j}^c$ of the k-th UAV at the j-th hovering location can not be less than the transmission time. Therefore, by denoting the size of data that needs to be uploaded by the j-th device as D_j ($D_0 = 0$), we have

$$D_j - \tau_{k,j}^c R_j \leq 0, \forall k \in \mathcal{K}, \forall j \in \mathcal{D}. \tag{10}$$

In addition, some IoT devices (e.g. camera) need to upload multimedia data with big size (e.g. photograph and video) which may exceeds the cache capacity of the associated UAV. In this case, for guaranteeing the data of each IoT device is collected completely, we consider the memory on the k-th UAV as C_k and we have

$$\sum_{i=0}^{M}\sum_{j=0}^{M}\alpha_{i,j}[k]D_j - C_k \leq 0, \forall k \in \mathcal{K}. \tag{11}$$

Besides the cache capacity, the energy budget also needs to be considered for UAV, since it restricts the flight time of UAV. The energy budget of the k-th UAV in this model is denoted as E_k. As for the power consumption of the UAVs, it mainly consists of the communication related power consumption and the propulsion power consumption. The former one primarily includes circuitry, signal reception and processing consumption, which is ignored in this paper since it is negligible compared with the propulsion power consumption. The later one is necessary for UAV to move and keep aloft, which generally depends on the flying speed and acceleration/deceleration of UAV. However, in this model, the acceleration/deceleration duration of UAV accounts for only a small part of the total operation time. Compared with the constant speed phase, the duration of the acceleration phase is negligible. Therefore, for the purpose of exposition, the energy consumption caused by the acceleration/deceleration of UAV is ignored in this paper, similar to [10] and [33]. Hence, the propulsion power consumption is a function of flying speed [10] [33] and given by

$$\begin{aligned} P\left(V\right) =& P_0\left(1+\frac{3V^2}{U_{tip}^2}\right) + P_I\left(\sqrt{1+\frac{V^4}{4v_0^4}}-\frac{V^2}{2v_0^2}\right)^{\frac{1}{2}} \\ &+ \frac{1}{2}d_0\beta s Q V^3, \end{aligned} \tag{12}$$

where $P_0 = \frac{\delta}{8}\beta s Q \Omega^3 R^3$ and $P_I = (1+\eta)\frac{G^{3/2}}{\sqrt{2\beta Q}}$ are constants, with the profile drag coefficient δ, the air density β, the rotor solidity s, the rotor disc area Q, the blade angular velocity Ω, the rotor radius R, the incremental correction factor η, and the aircraft weight G, U_{tip} represents the tip speed of the rotor speed, v_0 denotes the mean rotor induced velocity in hover, and V is the flying speed of UAV. For more details of the propulsion power consumption parameters, please refer to [10]. Obviously, while $V = 0$, equation (12) is the hovering power consumption of UAV, namely $P_k^h = P_0 + P_I$. Since the k-th UAV flies at a constant speed V_k^{max} in the fly-hover-collect scheme, the flying power consumption can be given by $P_k^f = P\left(V_k^{max}\right)$. As a result, the energy constraint can be expressed as

$$\begin{aligned} &\sum_{i=0}^{M}\sum_{j=0}^{M}\alpha_{i,j}[k]P_k^h\left(\tau_{k,j}^c + w_{k,j}\right) + \sum_{i=0}^{M}\sum_{j=0}^{M}\alpha_{i,j}[k]\tau_{k,ij}^f P_k^f \\ &- E_k \leq 0, \forall k \in \mathcal{K}, \end{aligned} \tag{13}$$

where $\tau_{k,ij}^f = \frac{\left\|\mathbf{q}_i-\mathbf{q}_j\right\|}{V_k^{\max}}$ is the time that the k-th UAV spends flying from the i-th hovering location to the j-th hovering location, and $w_{k,j}$ is the duration in which the k-th UAV waits at the j-th hovering location, which will be explained below.

In actual applications, some IoT devices have strict requirement for data collection time due to their characteristics or high demand for up-to-data information. Each of these IoT devices has a transmission time window. The open time of it denotes the time that the device completes information gathering, and the close time represents the time that the information is outdated or overwritten. Therefore, the UAVs can only collect the target data from these devices within their time windows. In this paper, the time windows are denoted as $[X_m, Y_m]$ ($i \in \mathcal{M}$), where X_m is the earliest time (ET) and Y_m is the latest time (LT) when the m-th IoT device is available. Since the UAVs is requested to come back to the initial point in a limited time, the initial point is also assumed to have a time window which is denoted as $[X_0, Y_0]$. As all the UAVs take off from the initial point, we let $X_0 = 0$. Therefore, for the devices without time limit, we set $X_i = X_0 = 0$ and $Y_i = Y_0$. If the UAV arrives before the beginning time of the device's time window, it must wait until the device is available. Meanwhile, the UAV also needs to ensure that the data collection should be finished before the end time of each

device's time window. By denoting the arrival time of the k-th UAV at the i-th hovering location as $t_{k,i}$, the time window constraints can be expressed as

$$\max\left\{X_j - t_{k,j}, 0\right\} - w_{k,j} \leq 0, \forall k \in \mathcal{K}, \forall j \in \mathcal{D}, \quad (14)$$

$$t_{k,j} + w_{k,j} + \tau^c_{k,j} - Y_j \leq 0, \forall k \in \mathcal{K}, \forall j \in \mathcal{D}, \quad (15)$$

$$\alpha_{0,j}[k]\left(\tau^f_{k,0j} - t_{k,j}\right) \leq 0, \forall k \in \mathcal{K}, \forall j \in \mathcal{D}, \quad (16)$$

$$\alpha_{i,j}[k]\left(t_{k,i} + \tau^c_{k,j} + w_{k,i} + \tau^f_{k,ij} - t_{k,j}\right) \leq 0, \forall k \in \mathcal{K}, \forall i \in \mathcal{M}, \forall j \in \mathcal{D}. \quad (17)$$

The last two constraints state that the k-th UAV can not arrive the j-th hovering location before it finishes the data collection of the i-th device if it flies from the i-th to the j-th hovering location.

B. Problem Formulation

Due to the high operational cost of UAV, it is significant to optimize the number and the operation time of the UAV systems. In [34], the network overhead is defined as the energy consumption cost of aerial and terrestrial base stations, but the cost of UAV is ignored. In [24], the aerial cost is defined as the energy consumption and the dispatching cost of UAV, but the maintenance cost of UAV is ignored. In practical applications, the energy consumption and maintenance cost are mainly related to the operation time of UAV. Therefore, we define the aerial cost as a function of the number and total operation time of UAVs, which is given by

$$O = F_1 U + F_2 L, \quad (18)$$

where $L = \sum_{i=0}^{M}\sum_{j=0}^{M}\sum_{k=1}^{K}\alpha_{i,j}[k]\left(\tau^c_{k,j} + \tau^f_{k,ij} + w_{k,j}\right)$ is the total operation time of UAVs, F_1 denotes the price for operating one UAV, and F_2 represents the price of UAV operation time measured in price/second including maintenance and charging prices.

For simplicity, the aerial cost is transformed into a normalized form. Similar to [24], the cost function measured in normalized price can be expressed as

$$O = \lambda U + L, \quad (19)$$

where $\lambda = \frac{F_1}{F_2}$ is a normalization factor for the aerial cost.

With the cost function, we aim to minimize both the number of UAVs and the total UAV operation time via optimizing the UAV trajectory $\mathbf{\Lambda} = \{\alpha_{i,j}[k], \forall k \in \mathcal{K}, \forall i, j \in \mathcal{D}\}$, the hovering location $\mathbf{Q} = \{\mathbf{q}_j, \forall j \in \mathcal{D}\}$, the arrival time $\mathbf{T} = \{t_{k,j}, \forall k \in \mathcal{K}, \forall j \in \mathcal{D}\}$, the waiting time $\mathbf{W} = \{w_{k,j}, \forall k \in \mathcal{K}, \forall j \in \mathcal{D}\}$, and the collection time $\mathbf{H} = \left\{\tau^c_{k,j}, \forall k \in \mathcal{K}, \forall i, j \in \mathcal{D}\right\}$. Thus, the optimization problem is formulated as

$$\text{P:} \min_{\mathbf{\Lambda},\mathbf{Q},\mathbf{T},\mathbf{W},\mathbf{H}} O, \quad (20)$$
$$\text{s.t. } (1)-(6), (10), (11), (13)-(17).$$

It is worth noting that the objective function decreases as the decrease of the collection time $\tau^c_{k,j}$ and the waiting time $w_{k,j}$, hence the equality of constraints (10) and (14) must hold at the optimal solution. Consequently, the collection time $\tau^c_{k,j}$ becomes a function of $\alpha_{i,j}[k]$ and $\mathbf{q}_j$, and the waiting time $w_{k,j}$ becomes a function of $t_{k,j}$, which are given by

$$\tau^c_{k,j} = \frac{D_j}{R_j}, \forall k \in \mathcal{K}, \forall j \in \mathcal{D}, \quad (21)$$

$$w_{k,j} = \max\left\{X_j - t_{k,j}, 0\right\}, \forall k \in \mathcal{K}, \forall j \in \mathcal{D}. \quad (22)$$

Consequently, problem (20) can be reformulated as

$$\min_{\mathbf{\Lambda},\mathbf{Q},\mathbf{T}} O, \quad (23)$$
$$\text{s.t. } (1)-(6), (11), (13), (15)-(17).$$

In this problem, we can see that the UAV trajectory $\mathbf{\Lambda}$ are binary variables and the hovering location $\mathbf{Q}$ as well as the arrival time $\mathbf{T}$ are continuous variables. Moreover, $\{R_j\}$ in (13), (15)-(17) are non-convex functions with respect to $\mathbf{q}_j$, which results in the non-convexity of the constraints. Therefore, problem (23) is a mixed-integer non-convex problem which is difficult to solve. In next section, we propose an efficient algorithm to obtain a proper solution of problem (23).

III. Proposed Algorithm

In this section, we decompose problem (23) into two subproblems. First, with given hovering location $\mathbf{Q}$, we solve the UAV trajectory optimization problem by the proposed MACO algorithm. Then, given the UAV trajectory $\mathbf{\Lambda}$, the hovering location $\mathbf{Q}$ is optimized through SCA technique.

A. UAV Trajectory Optimization

Given any feasible hovering location $\mathbf{Q}$, problem (23) is simplified to a UAV trajectory optimization problem as follows.

$$\min_{\mathbf{\Lambda},\mathbf{T}} O, \quad (24)$$
$$\text{s.t. } (1)-(5), (11), (13), (15)-(17).$$

Through solving problem (24), the number of UAVs can also be obtained, since $U = \sum_{j=1}^{M}\sum_{k=1}^{K}\alpha_{0,j}[k]$. However, owing to the energy and time window constraints, the binary variables $\mathbf{\Lambda}$ are coupled with the arrival time variables $\mathbf{T}$ in the objective function and the constraints (13), (15)-(17). This makes the problem still a mix-integer non-convex problem.

Theorem 1: Problem (24) is NP-hard.

Proof: Consider the special case that the number of UAVs is fixed at K. In this case, the collection time $\tau^c_{k,j}$ is constant.

Therefore, problem (24) reduces to minimize the total flying and waiting time as follows.

$$\min_{\mathbf{\Lambda},\mathbf{T}} \sum_{i=0}^{M}\sum_{j=0}^{M}\sum_{k=1}^{K} \alpha_{i,j}[k]\left(\tau_{k,ij}^{f}+w_{k,j}\right), \tag{25}$$

$$\text{s.t. } (1)-(5),(11),(16),$$

$$\sum_{i=0}^{M}\sum_{j=0}^{M}\alpha_{i,j}[k]P_k^h w_{k,j}+\sum_{i=0}^{M}\sum_{j=0}^{M}\alpha_{i,j}[k]\tau_{k,ij}^{f}P_k^f - E_k \le 0, \forall k\in\mathcal{K}, \tag{25a}$$

$$t_{k,j}+w_{k,j}-Y_j\le 0, \forall k\in\mathcal{K},\forall j\in\mathcal{D}, \tag{25b}$$

$$\alpha_{i,j}[k]\left(t_{k,i}+w_{k,i}+\tau_{k,ij}^{f}-t_{k,j}\right)\le 0, \forall k\in\mathcal{K},\forall i\in\mathcal{M},\forall j\in\mathcal{D}. \tag{25c}$$

It is noted that problem (25) is equivalent to vehicle routing problems with time windows (VRPTW) which is NP-hard. VRPTW aims to design the least cost routes for a fleet of vehicles to serve a set of customers. Similarly, this transportation model is established in such a way that each vehicle departs from the depot at the beginning and return back to it at the end. Each customer is served only once by exactly one vehicle within a given period, and the customer demands on one particular route must not exceed the capacity of the vehicle. In this way, we have created a NP-hard instance of problem (24). Therefore, problem (24) is also NP-hard. ■

The difference between problem (24) and VRPTW is that the number of UAVs is considered. Moreover, the number of UAVs affects the duration of data collection, since the UAVs equally share the same spectrum. The total operation time of UAVs does not always decrease with the increase in the number of UAVs. Therefore, the optimal solution of problem (24) is more difficult to obtain.

Although VRPTW is NP-hard, high-quality approximate solutions can be efficiently obtained by various heuristic algorithms. Inspired by this, we propose a modified ant colony optimization (MACO) algorithm to obtain an approximate solution of problem (24). The details of the MACO algorithm are summarized in Algorithm 1. In this algorithm, the route of each ant represents the trajectories of all UAVs dispatched in the data collection mission. For making the algorithm more understandable, we divide it into two parts to explain. The first part is the trajectory searching part of each ant, which designs the trajectory of each UAV. The main steps of this part are summarized as follows.

- *Step 1*: From the set of IoT devices that have not been visited, select the devices that make constraints (11), (13) and (15) satisfied.
- *Step 2*: If there is no satisfied device, add the initial point to the end of the trajectory, as shown in lines 6-7 of Algorithm 1. Then, the searching process is completed.
- *Step 3*: Calculate the visit probability of each satisfied device based on the present pheromone.
- *Step 4*: Choose one of them as the next device to be visited by the Roulette algorithm. Then, go to Step 1.

The second part is the main body of the algorithm, which optimize the number of UAVs. The main steps of this part are summarized as follows.

- *Step 1*: Each ant searches the trajectories for the existing UAVs in order. If there is no device that have not been visited, go to Step 3.
- *Step 2*: Add a new UAV into this mission. Due to the FDM adopted in this model, recalculate the transmit rate of each IoT device, as shown in lines 8-10 of Algorithm 1, and then go to Step 1.
- *Step 3*: After all ants finish their searches, choose the route of the ant with the minimum objective value as the optimal route of this iteration.
- *Step 4*: If the solution accuracy ϵ or the maximum number of iterations is not achieved, update the global pheromone, and then go to Step 1.

Given the solution accuracy ϵ, the computational complexity of the algorithm is roughly given by $O(\log(1/\epsilon)M^2)$.

Algorithm 1 MACO algorithm for problem (24)

Input: The number of ants N_a; Cache capacity C_k; Energy budget E_k; Data size D_j; ET of time window X_j; LT of time window Y_j;
Output: UAV trajectory $\mathbf{\Lambda}^*$;

1: **while** stopping criteria are not satisfied **do**
2: **for** $I_a = 1 : N_a$ **do**
3: Initialize the Route set of ant I_a: $\mathbf{Route}_{I_a} \leftarrow \{0\}$, the number of UAVs $NoU \leftarrow 1$, the UAV counter $C_U \leftarrow 1$, the number of visited IoT devices $NoD \leftarrow 0$;
4: **while** $NoD < M$ **do**
5: Obtain the set of devices from the remaining IoT devices that make all the constraints met;
6: **if** there is no satisfied IoT device **then**
7: Add 0 to $\mathbf{Route}_{I_a}$, update $NoU \leftarrow NoU + 1$;
8: **if** $NoU > C_U$ **then**
9: Update $C_U \leftarrow C_U + 1$, $NoU \leftarrow 1$, $\mathbf{Route}_{I_a} \leftarrow \{0\}$, $NoD \leftarrow 0$;
10: Recalculate the transmit rate of each IoT device;
11: **end if**
12: **else**
13: Calculate the visit probability of the satisfied IoT devices based on the present pheromone;
14: Choose one of them by Roulette algorithm and add it to $\mathbf{Route}_{I_a}$;
15: Update $NoD \leftarrow NoD + 1$;
16: **end if**
17: **end while**
18: **end for**
19: Obtain $\alpha_{k,ij}$ for each ant according to $\mathbf{Route}_{I_a}$, then compute the objective value V_{I_a} based on the objective function in problem (P3);
20: Update $\alpha_{k,ij}^{*}$ by $\alpha_{k,ij}$ of the ant with the minimum objective value
21: Update the global pheromone.
22: **end while**

B. Hovering Location Optimization

With any given feasible UAV trajectory $\mathbf{\Lambda}$, the number of UAVs U is confirmed. It is assumed that the set of these UAVs is $\mathcal{N}$, and the cardinality $|\mathcal{N}| = U$. Consequently, the cost function is transformed into

$$L^{'} = \sum_{i=0}^{M}\sum_{j=0}^{M}\sum_{n=1}^{U}\alpha_{i,j}[n]\left(\tau_{n,i}^{c}+\tau_{n,ij}^{f}+w_{n,j}\right), \tag{26}$$

Therefore, the hovering location $\mathbf{Q}$ can be optimized by solving the following problem.

$$\min_{\mathbf{Q},\mathbf{T}} L^{'}, \tag{27}$$

s.t. (6),

$$\sum_{i=0}^{M}\sum_{j=0}^{M}\alpha_{i,j}[n]P_n^h\left(\tau_{n,j}^c+w_{n,j}\right)+\sum_{i=0}^{M}\sum_{j=0}^{M}\alpha_{i,j}[n]\tau_{n,ij}^f P_n^f - E_n\le 0,\forall n\in\mathcal{N}, \tag{27a}$$

$$t_{n,j}+w_{n,j}+\tau_{n,j}^c-Y_j\le 0,\forall n\in\mathcal{N},\forall j\in\mathcal{D}, \tag{27b}$$

$$\alpha_{0,j}[n]\left(\tau_{n,0j}^f-t_{n,j}\right)\le 0,\forall n\in\mathcal{N},\forall j\in\mathcal{D}, \tag{27c}$$

$$\alpha_{i,j}[n]\left(t_{n,i}+\tau_{n,i}^c+w_{n,i}+\tau_{n,ij}^f-t_{n,j}\right)\le 0, \quad \forall n\in\mathcal{N},\forall i\in\mathcal{M},\forall j\in\mathcal{D}. \tag{27d}$$

It is readily to find that the waiting time $w_{n,j}$ in the objective function is not smooth at zero value. To make problem (27) more tractable, we relax it into

$$\min_{\mathbf{Q},\mathbf{T}} L^{''}, \quad \text{s.t. (6), (27a)} - \text{(27d)}, \tag{28}$$

where $L^{''}=\sum_{i=0}^{M}\sum_{j=0}^{M}\sum_{n=1}^{U}\alpha_{i,j}[n]\left(\tau_{n,i}^c+\tau_{n,ij}^f+X_j-t_{k,j}\right)$. Such relaxation implies that the objective value of problem (28) serves as a lower bound for that of problem (27).

As can be seen from problem (28), the collection time $\tau_{n,j}^c$ in the objective function and constraints are non-convex due to the non-convexity of the transmit rate R_j with respect to $\mathbf{q}_j$. Therefore, problem (28) is still non-convex. To make this problem more tractable, we first introduce relax variables $\mathbf{\Theta}=\{\theta_{n,j},\forall n\in\mathcal{N},\forall j\in\mathcal{D}\}$, then problem (28) is reformulated as

$$\min_{\mathbf{Q},\mathbf{T},\mathbf{\Theta}} L^{'''}, \tag{29}$$

s.t. (6), (27c),

$$\sum_{i=0}^{M}\sum_{j=0}^{M}\alpha_{i,j}[n]P_n^h\left(\frac{D_j}{\theta_{n,j}}+w_{n,j}\right)+\sum_{i=0}^{M}\sum_{j=0}^{M}\alpha_{i,j}[n]\tau_{n,ij}^f P_n^f - E\le 0,\forall n\in\mathcal{K}, \tag{29a}$$

$$t_{n,j}+w_{n,j}+\frac{D_j}{\theta_{n,j}}-B_j\le 0,\forall n\in\mathcal{N},\forall j\in\mathcal{D}, \tag{29b}$$

$$\alpha_{i,j}[n]\left(t_{n,i}+\frac{D_j}{\theta_{n,j}}+w_{n,i}+\tau_{n,ij}^f-t_{n,j}\right)\le 0, \quad \forall n\in\mathcal{N},\forall i\in\mathcal{M},\forall j\in\mathcal{D}, \tag{29c}$$

$$R_j\ge\theta_{n,j},\forall n\in\mathcal{N},\forall j\in\mathcal{D}, \tag{29d}$$

where $L^{'''}=\sum_{i=0}^{M}\sum_{j=0}^{M}\sum_{n=1}^{U}\alpha_{i,j}[n]\left(\frac{D_i}{\theta_{n,i}}+\tau_{n,ij}^f+X_j-t_{k,j}\right)$.

Lemma 1: The optimal solution of problem (29) is equivalent to that of problem (28).

Proof: It can be observed that the equality of constraint (29d) must hold at the optimal solution. Otherwise, the objective function decreases as the slack variable $\theta_{k,j}$ increases while the hovering location variable $\mathbf{q}_j$ fixed, which still satisfy all other constraints. Therefore, the optimal solution can only be obtained while constraint (29d) is met with equality, which makes problem (29) equivalent to problem (28). Thus, the proof of Lemma 1 is completed. ■

However, the newly introduced constraint (29d) is non-convex due to the existence of the transmit rate R_j, which makes problem (29) still difficult to solve. Although the transmit rate R_j is neither convex nor concave with respect to the hovering location $\mathbf{q}_j$, it is convex with respect to $\|\mathbf{q}_j-\mathbf{s}_j\|^2$. According to [35], the first-order Taylor expansion of any convex function is its global lower bound. From this point, SCA technique is applied to tackle this challenge. For any given local hovering location $Q^r=\left\{\mathbf{q}_j^r,\forall j\in\mathcal{D}\right\}$ at the r-th iteration, the global lower bound of the transmit rate R_j can be expressed as

$$R_j=\frac{B}{U}\log_2\left(1+\frac{\gamma_0}{H^2+\|\mathbf{q}_j-\mathbf{s}_j\|^2}\right) \ge G_j^r-I_j^r\left(\|\mathbf{q}_j-\mathbf{s}_j\|^2-\left\|\mathbf{q}_j^r-\mathbf{s}_j\right\|^2\right)=R_j^l, \tag{30}$$

where $G_j^r=\frac{B}{U}\log_2\left(1+\frac{\gamma_0}{H^2+\left\|\mathbf{q}_j^r-\mathbf{s}_j\right\|^2}\right)$ and $I_j^r=\frac{B\gamma_0}{U\ln 2\left(H^2+\left\|\mathbf{q}_j^r-\mathbf{s}_j\right\|^2\right)\left(\gamma_0+H^2+\left\|\mathbf{q}_j^r-\mathbf{s}_j\right\|^2\right)}$ are constants.

By replacing the transmit rate R_j in constraint (29d) with its lower bound R_j^l at the r-th iteration, problem (29) is approximated as

$$\min_{\mathbf{Q},\mathbf{T},\mathbf{\Theta}} L^{'''}, \tag{31}$$

$$\text{s.t. (6), (27d), (29b)} - \text{(29c)},$$

$$R_j^l\ge\theta_{n,j},\forall n\in\mathcal{N},\forall j\in\mathcal{D}. \tag{31a}$$

From (30), it can be seen that R_j^l is concave with respect to $\mathbf{q}_j$ for the sake of the convexity of $\|\mathbf{q}_j-\mathbf{s}_j\|^2$. Therefore, problem (31) is a convex optimization problem, which can be efficiently solved by standard convex optimization tools such as CVX [36]. Note that the lower bound applied in constraint (31a) implies that the feasible solution of problem (31) is also feasible for problem (29). That is, the optimal value of problem (31) is a upper bound for that of problem (29). Therefore, an iterative algorithm for solving problem (28) is obtained by successively solving problem (31) with updated local hovering locations. The algorithm is summarized in Algorithm 2. Besides the monotonic convergence of SCA technique [37], the value and gradient of the lower bound in (30) are identical to that of the original function. Thus, it is ensured that SCA-based algorithm can converge to at least a locally optimal solution that meet the Karush-Kuhn-Tucker (KKT) conditions of problem (28) [20], [38]. As the SCA-based algorithm alternately solves the standard convex problem (31) by interior point method, the computational complexity is roughly given by $O(\log(1/\epsilon)((M+1)(2N+1))^{3.5})$.

Algorithm 2 SCA-based algorithm for problem (31)

Input: Initial hovering location $\mathbf{Q}^0$
Output: hovering location $\mathbf{Q}^*$;
1: Let iteration counter $r = 0$.
2: **repeat**
3: Solve problem (31) with given $\mathbf{Q}^r$, and denote the solution as $\mathbf{Q}^*$;
4: Update $\mathbf{Q}^{r+1} = \mathbf{Q}^*$;
5: Update $r = r + 1$;
6: **until** the discrepancy of the objective value between two successive iteration is below a given threshold ϵ.

Then, combining the solutions of MACO algorithm and SCA-based algorithm, the overall algorithm for problem is summarized in Algorithm 3. As mentioned above, the global controller in the initial point is assumed to know the states of UAVs and IoT devices, as well as the CSI between them. This algorithm has been run in the global controller before the UAVs take off. That is, it is an offline algorithm. Therefore, when the UAVs is dispatched to collect data, the number, trajectories and hovering locations of them are all predetermined.

Algorithm 3 MACO-based algorithm for problem (23)

1: Initialize the hovering locations $q_j^0 = s_j, \forall j \in \mathcal{D}$, and let the iteration counter $r = 0$.
2: Solve problem (24) with given $\mathbf{Q}^0$ by Algorithm 1, and denote the solution as $\mathbf{\Lambda}^*$;
3: With the obtained $\mathbf{\Lambda}^*$, Solve problem (31) by Algorithm 3, and denote the solution as $\mathbf{Q}^*$.

IV. Numerical Results

In this section, simulation results are presented to demonstrate the effectiveness of the proposed algorithms. We assume that the UAVs employed in this model have the identical cache capacity, energy budget and flying speed which are denoted as C, E and V^{max}, respectively. As a result, the flying and hovering power consumption as well as the flying time of the UAV are given by $P^f = P(V^{max})$, $P^h = P_0 + P_I$ and $\tau_{ij}^f = \frac{\|\mathbf{q}_i - \mathbf{q}_j\|}{V^{\max}}$, respectively. We consider that IoT devices are randomly and uniformly distributed within a 2-dimensional area of size 1km $\times$ 1km. The location of the initial point is set as $\mathbf{q}_0 = (500\text{m}, 500\text{m})$. The UAV flying altitude is fixed at $H = 100$ m and the flying speed is $V^{max} = 20$ m/s. Thus, the flying power is $P^f = 178$ W and the hovering power is $P^h = 169$ W according to (12) [10]. The cache capacity and energy budget of UAV are set as $C = 2$ Gb and $E = 1.26$ MJ, respectively. The total communication bandwidth is $B = 10$ MHz and the transmit power of the IoT device is $P = 0.01$ W. The received noise power at the UAVs is assumed to be $\sigma^2 = -110$ dBm. The channel power gain at the reference distance of 1m is set as $\rho_0 = -60$ dB. Similar to [24], the normalized constant is assumed to be $\lambda = 10000$. The main parameter settings are summarized in Table I.

In this section, we compare MACO-based algorithm with exhaustive-based algorithm, greedy-based algorithm, and random-based algorithm. In these three overall algorithms, the algorithms applied to solve the hovering location optimization problem are also SCA-based algorithm. The algorithms used to solve the UAV trajectory optimization problem in these three algorithms are exhaustive algorithm, greedy algorithm, and random trajectory algorithm, respectively. In exhaustive algorithm, all feasible trajectories for UAVs are listed and the optimal set of trajectories is chosen as the solution. In greedy algorithm, while searching for the trajectory, each UAV chooses the nearest qualified IoT device as the next device to collect data. In random trajectory algorithm, each UAV chooses the next qualified IoT device randomly. We generate results on CPU with Intel Core i5-6500 CPU @ 3.2 GHz speed, 8 GB memory ram and 64-bit windows operating system.

TABLE I
Main Parameters and Settings

Parameter	Physical meaning	Value
H	UAV flying altitude	100 m
d_0	Reference distance	1 m
ρ_0	Channel power gain at the reference distance	-60 dB
B	Total communication bandwidth	10 MHz
σ^2	Additive white Gaussian power	110 dBm
P	Transmit power of IoT device	0.01 W
C	Cache capacity of UAV	2 Gb
E	Energy budget of UAV	1.26 MJ
P^f	Flying power of UAV	178 W
P^h	Hovering power of UAV	169W
V^{max}	Flying speed of UAV	20 m/s
A_0	Taking off time of UAV	0 s
B_0	Latest return time of UAV	1800 s
λ	Normalization factor for the aerial cost	10000

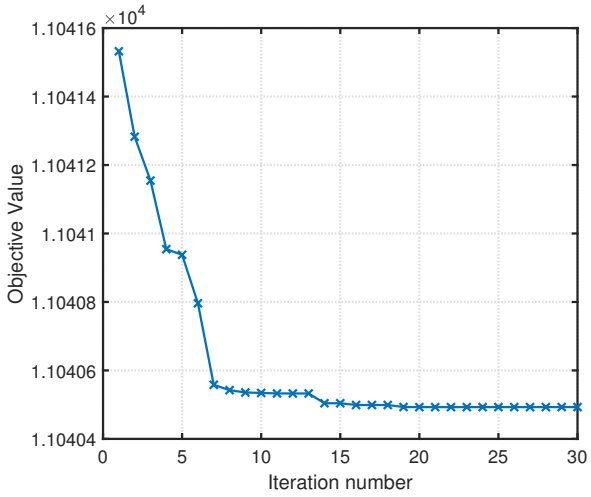
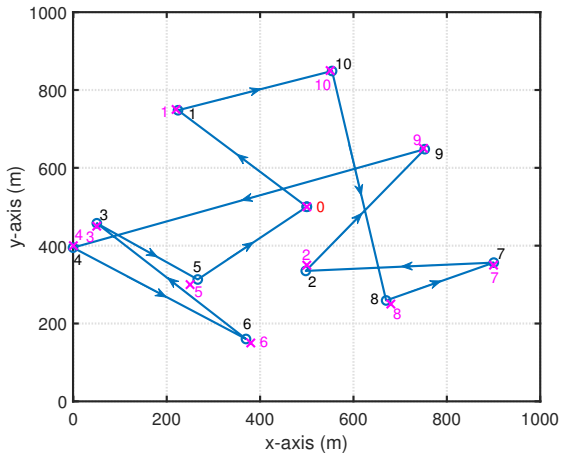


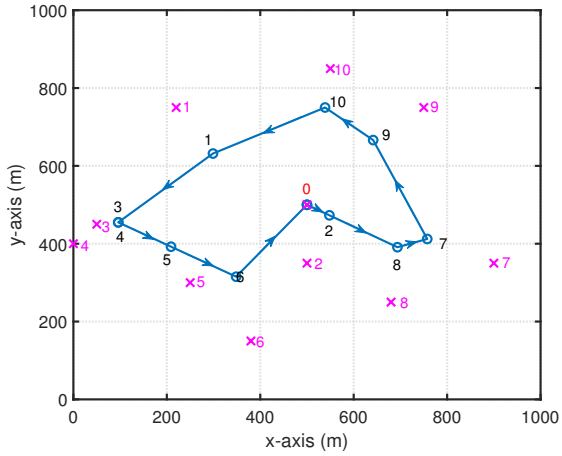
Fig. 2. Convergence of SCA-based algorithm.

First, We study the convergence of SCA-based algorithm with randomly generated IoT devices in Fig. 2. The result is averaged over 30 runs. It can be observed that the objective value optimized by the proposed algorithm decreases quickly with the increase of the number of iterations and the algorithm converges in a few iterations.

Then, for validating MACO-based algorithm and illustrating the impact of time windows on the UAV trajectory, Fig. 3 shows the obtained UAV trajectory and hovering location in the network of 10 IoT devices with and without time



(a)



(b)

Fig. 3. UAV trajectory and hovering location optimized by MACO-based algorithm in the network of 10 IoT devices: (a) with time windows; (b) without time windows. "×" denotes the location of IoT device, and "○" represents the corresponding hovering location.

windows. As it is assumed above, the UAV flies straight from one hovering location to another and collects data from the corresponding devices. Hence, the trajectory is composed only of line segments. We can see from Fig. 3(a) that intersections occur in the UAV trajectory. There are two reasons for this phenomenon. First, the time windows of IoT devices restrict the UAV trajectory, which results in the intersections. For reducing the total operation time, UAV prefers to collect data from other IoT devices and then come back rather than waiting for current IoT device's time window opening. As shown in Fig. 3(b), without time window constraint, no intersection appears in the UAV trajectory. In order to further demonstrate this, we also simulate exhaustive-based algorithm for the same IoT network and depict the result in Fig. 4. It can be seen that even for exhaustive-based algorithm applied in the IoT network with time windows, the obtained UAV trajectory still has intersections. This shows that the time window constraint can generate the intersections. Second, the solution obtained by MACO-based algorithm may not be optimal. That is, even for the IoT network without time windows, the results may still

TABLE II
MISSION COMPLETION STATE OF THE IOT NETWORK WITH 10 IOT DEVICES

IoT device index	Data size (Mb)	ET (s)	LT(s)	Arrival time (s)	Waiting time (s)
0	0	0	1800	739.9993	0
1	5	30	92	18.5444	11.4556
2	0.08	283	344	223.1766	59.8234
3	500	665	1325	677.3488	0
4	0.3	479	522	408.7366	70.2634
5	3	725	786	697.8884	27.1317
6	300	651	1131	500.9318	150.0682
7	0.9	203	260	121.5107	81.4893
8	0.05	109	170	89.3912	19.6094
9	0.009	369	420	303.1769	65.8231
10	800	0	1800	47.3270	0

have trajectory intersections. Apart from the intersections, as can be seen from Fig. 3, the distances between the hovering locations and the corresponding IoT devices in the network without time windows are farther than that in the network with time windows. That is because the UAV in the IoT network without time windows would rather hover far away from IoT devices to reduce the flying time, since it does not need to consider the waiting time and the corresponding energy consumption. On the contrary, the UAV in the IoT network with time windows prefers to fly closer to the devices rather than waiting for the time windows opening, since the closer distance leads to faster transmit rate which further reduces the upload time.

For further analysis, we show some properties of the simulated IoT network and the timings of UAV in Table II. It is noted that the UAV trajectory and hovering locations are determined jointly by the data size and time windows of IoT devices. For most of the IoT devices with high data volume such as the 1st, 3rd and 10th IoT devices, UAV is more willing to fly closer to them than that with little data volume such as the 2nd and 8th IoT devices for reducing the transmission time. However, for some IoT devices with small data size such as the 7th and 9th IoT devices, the distances between the locations of them and the corresponding hovering locations are closer than that of IoT devices with large data size such as the 5th and 6th IoT devices. The reason is that the UAV arrives before the time windows open, thus it has enough time to fly closer for faster data rate which can reduce the operation time effectively.

Moreover, in order to analyze the the trajectories and hovering locations of multiple UAVs, we also simulate the MACO-based algorithm in the network with 15 IoT devices, as shown in Fig. 5. As one can see, besides the intersections within a trajectory, the intersections also exist between the trajectories of different UAVs. There are two reasons. First, the constraints of the problem cause the intersections. The energy and cache constraints of UAVs and the time window constraints of IoT devices restrict the IoT devices that the UAVs can serve. Furthermore, the service order of the UAVs is also restricted, since they have to collect data within the constrained time of each IoT device. Second, the collision avoidance between the UAVs is not considered in this paper due to the high complexity of this problem. Since the trajectory

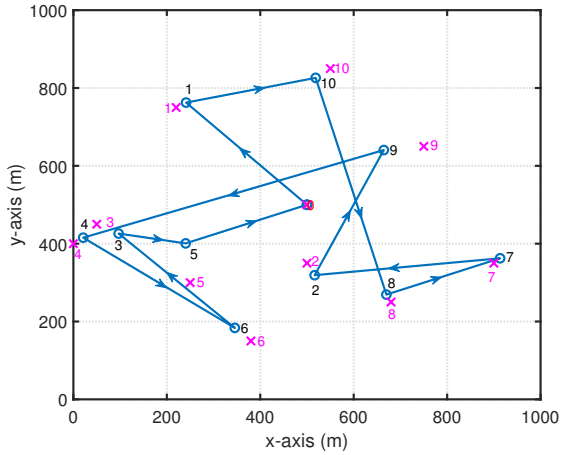


Fig. 4. UAV trajectory and hovering location optimized by exhaustive-based algorithm in the network of 10 IoT devices with time windows. "×" denotes the location of IoT device, and "○" represents the corresponding hovering location.

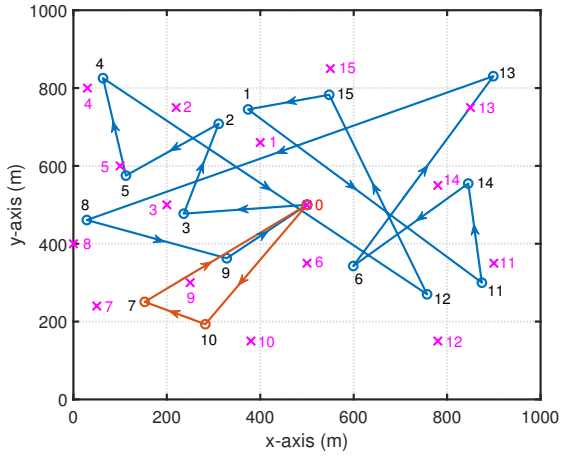
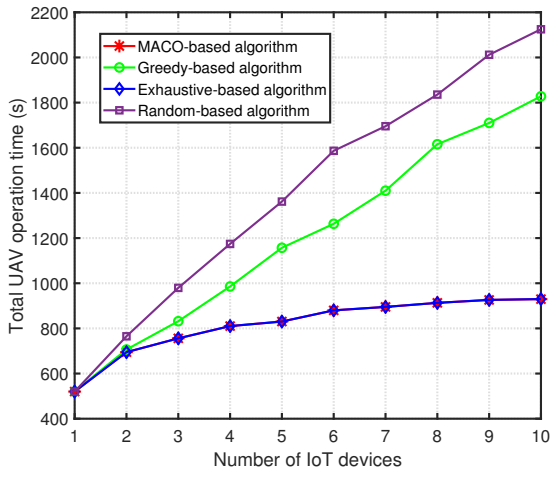


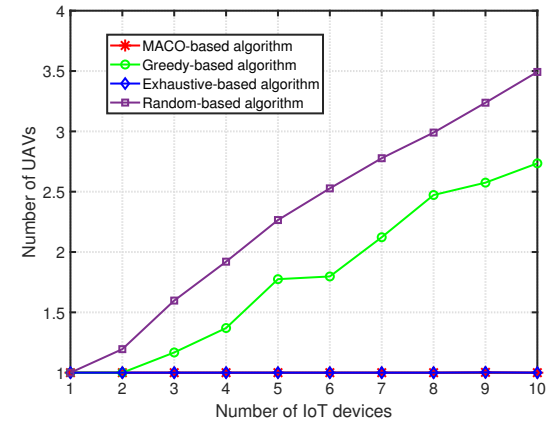
Fig. 5. UAV trajectory and hovering location optimized by MACO-based algorithm in the network of 15 IoT devices with time windows. "×" denotes the location of IoT device, and "○" represents the corresponding hovering location.

variable of UAV is discrete, the collision problem is difficult to solve by convex optimization method. In addition, the insertion of hovering location into the trajectory of any UAV may make the subsequent trajectory unable to meet the constraints. Therefore, it is also hard to design a heuristic algorithm. Although the collision avoidance of UAVs is non-trivial in this system model, it is still a significant and meaningful issue especially for the systems with numerous UAVs, which is left as our future work.

Next, we compare MACO-based algorithm with exhaustive-based algorithm, greedy-based algorithm, and random-based algorithm in small-scale IoT networks. The results with average over 400 runs are depicted in Fig. 6. Therefore, the results of the number of UAVs in Fig. 6(b) are not integers. As it can be seen, MACO-based algorithm and exhaustive-based algorithm achieve similar performance both in the total UAV operation time and the number of UAVs, and greedy-based and random-based algorithms perform much worse. For



(a)



(b)

Fig. 6. Performance comparison between different algorithms in small-scale IoT networks: (a) total UAV operation time; (b) number of UAVs.

the performance of total UAV operation time, MACO-based algorithm improves up to about 56.2% and 49.1% compared with random-based and greedy-based algorithms, respectively. As for the number of UAVs, MACO-based algorithm improves up to about 71.4% and 63.4% compared with random-based and greedy-based algorithms, respectively. In addition, from Fig. 6, we can see that the proposed MACO-based algorithm is much more stable than greedy-based and random-based algorithms.

Furthermore, we also compare the running time of MACO, exhaustive, greedy, and random trajectory algorithms in Fig. 7. It can be seen that when the number of IoT devices is not more than 8, the running time of these four algorithms are small and similar. However, when it is more than 8, the gap between exhaustive algorithm and the other three algorithms becomes large. In addition, we observe that the running time of MACO algorithm is a little bit longer than greedy and random trajectory algorithms. That is because there are not many trajectory choices in small-scale IoT networks. Therefore, the other three algorithms can finish searching quickly. However, as an iterative algorithm, the MACO algorithm still has to alternate until the stopping criteria are met. Nevertheless, the

differences between the running time of MACO algorithm and greedy as well as random trajectory algorithms are small. In general, the proposed MACO-based algorithm can achieve similar performance and stability with exhaustive-based algorithm while spending almost the same time with other two algorithms in small-scale IoT networks.

One can see from Fig. 7 that the computational complexity of the exhaustive algorithm grows exponentially with the size of the networks, which makes it unscalable for large-scale IoT networks. When the number of IoT devices reaches 11, the exhaustive algorithm already needs to run for 24 min. Therefore, owing to the long running time of exhaustive-based algorithm, we just compare the performance of MACO-based algorithm, random-based algorithm and greedy-based algorithm in large-scale IoT networks in Fig. 8. The results are averaged over 10 runs, which may also cause the results in Fig. 8(b) to be not integers. From Fig. 8, we can see that MACO-based algorithm performs better than the other algorithms both in the total UAV operation time and the number of UAVs, especially for greedy-based algorithm. As the number of IoT devices increases, the number of trajectory choices for UAVs becomes huge. However, the UAVs in random-based algorithm choose the next IoT device randomly, and the UAVs in greedy-based algorithm choose the currently best IoT device as the next device. These properties make the performance of these two algorithms deteriorate as the number of IoT devices increases. By contrast, MACO-based algorithm searches for the trajectories more comprehensively, leads to a better performance. Compared with greedy-based algorithm, MACO-based algorithm improves up to about 95.3% and 96.5% in the UAV total operation time and the number of UAVs, respectively. Meanwhile, MACO-based algorithm improves up to about 81.9% and 86.5 % in the total UAV operation time and the number of UAVs, respectively, compared with random-based algorithm. It can be observed that the solutions of random-based and greedy-based algorithms are almost infeasible in large-scale IoT networks. However, for MACO-based algorithm, the results are still considerable. Even in the networks of 200 IoT devices, MACO-based algorithm only employs about 6 UAVs to collect data, and the average operation time per UAV is about 15.4 min.

In summary, the performance of MACO-based algorithm is close to that of exhaustive-based algorithm but it spends much less time in small-scale IoT networks. As for the large-scale IoT networks, MACO-based algorithm can still achieve considerable performance compared with greedy-based algorithm and random-based algorithm.

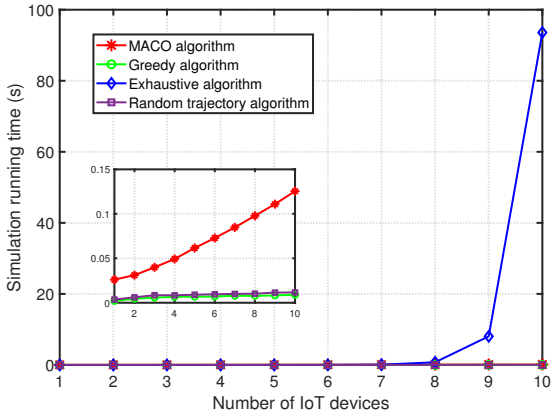
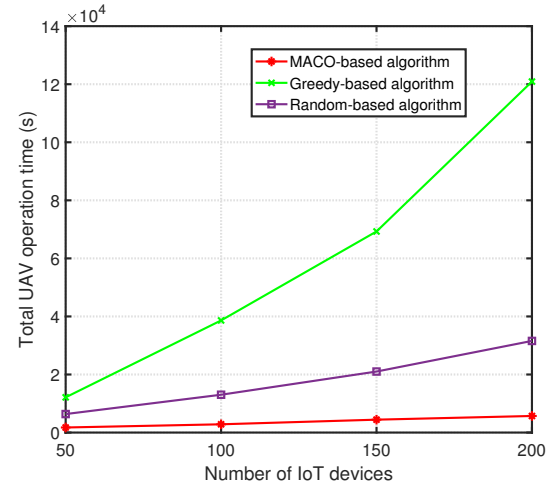
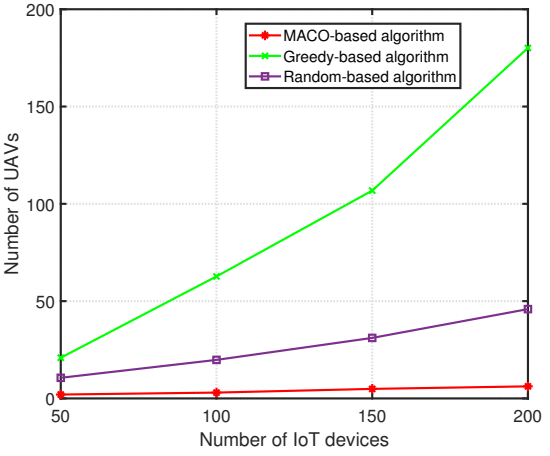


Fig. 7. Running time comparison between different algorithm in small-scale IoT networks.



(a)



(b)

Fig. 8. Performance comparison between different algorithms in large-scale IoT networks: (a) total UAV operation time; (b) number of UAVs.

V. Conclusion

In this paper, we have studied the data collection in multi-UAV enabled systems with time windows where the energy budget, cache capacity of UAVs and the transmission time of IoT devices are considered. To jointly minimize the number and the total operation time of UAVs, an optimization problem has been formulated to optimize the UAV trajectory and hovering location. We have decomposed it into two subproblems and proposed an efficient algorithm, which relies on the modified ACO algorithm and SCA technique. Numerical results have shown that the proposed MACO-based algorithm converges fast and can achieve a better performance in acceptable time compared with other benchmark solutions.

Moreover, this paper can be extended in three directions. Firstly, the machine learning may be applied as an online algorithm to help with the real-time decision [39]. Secondly, the collision avoidance between UAVs could be further considered in our future work. Finally, after the data collection, how to apply an UAV-enabled mobile edge computing (MEC) with time window could be another interesting research direction.

ACKNOWLEDGMENTS

This work was supported in part by the Natural Science Foundation of China under Grant No. 61620106011, U1705263 and 61871076, and in part by the National Natural Science Foundation of China under Grant No. 61971421.

REFERENCES

[1] D. Ma, G. Lan, M. Hassan, W. Hu, and S. K. Das, "Sensing, computing, and communications for energy harvesting IoTs: A survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 1222–1250, 2020.

[2] J. Liu, H. Guo, J. Xiong, N. Kato, J. Zhang, and Y. Zhang, "Smart and resilient EV charging in SDN-enhanced vehicular edge computing networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 1, pp. 217–228, 2020.

[3] H. Guo, J. Liu, J. Ren, and Y. Zhang, "Intelligent task offloading in vehicular edge computing networks," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 126–132, 2020.

[4] X. Hu, K. Wong, and K. Yang, "Wireless powered cooperation-assisted mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2375–2388, 2018.

[5] Y. Liu, Z. Qin, Y. Cai, Y. Gao, G. Y. Li, and A. Nallanathan, "UAV communications based on non-orthogonal multiple access," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 52–57, 2019.

[6] Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao, and D. Chen, "Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 10 187–10 200, 2019.

[7] H. Guo and J. Liu, "UAV-enhanced intelligent offloading for Internet of Things at the edge," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2737–2746, 2020.

[8] X. Liu, Y. Liu, and Y. Chen, "Machine learning empowered trajectory and passive beamforming design in UAV-RIS wireless networks," *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2020.

[9] C. Zhan and H. Lai, "Energy minimization in Internet-of-Things system based on rotary-wing UAV," *IEEE Wireless Communications Letters*, vol. 8, no. 5, pp. 1341–1344, 2019.

[10] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.

[11] Z. Wang, R. Liu, Q. Liu, J. S. Thompson, and M. Kadoch, "Energy-efficient data collection and device positioning in UAV-assisted IoT," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1122–1139, 2020.

[12] J. Baek, S. I. Han, and Y. Han, "Energy-efficient UAV routing for wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1741–1750, 2020.

[13] S. Fu, Y. Tang, N. Zhang, L. Zhao, S. Wu, and X. Jian, "Joint unmanned aerial vehicle (UAV) deployment and power control for Internet of Things networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4367–4378, 2020.

[14] H. Ye, X. Kang, J. Joung, and Y. Liang, "Optimization for full-duplex rotary-wing UAV-enabled wireless-powered IoT networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 5057–5072, 2020.

[15] T. Kim and D. Qiao, "Energy-efficient data collection for IoT networks via cooperative multi-hop UAV networks," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2020.

[16] Y. Liu, K. Liu, J. Han, L. Zhu, Z. Xiao, and X. G. Xia, "Resource allocation and 3D placement for UAV-enabled energy-efficient IoT communications," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[17] W. Feng, J. Wang, Y. Chen, X. Wang, N. Ge, and J. Lu, "UAV-aided MIMO communications for 5G Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1731–1740, 2019.

[18] R. Duan, J. Wang, C. Jiang, H. Yao, Y. Ren, and Y. Qian, "Resource allocation for multi-UAV aided IoT NOMA uplink transmission systems," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7025–7037, 2019.

[19] B. Jiang, J. Yang, H. Xu, H. Song, and G. Zheng, "Multimedia data throughput maximization in Internet-of-Things system based on optimization of cache-enabled UAV," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3525–3532, 2019.

[20] C. Zhan and Y. Zeng, "Completion time minimization for multi-UAV-enabled data collection," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4859–4872, 2019.

[21] S. Alfattani, W. Jaafar, H. Yanikomeroglu, and A. Yongacoglu, "Multi-UAV data collection framework for wireless sensor networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[22] Y. Zhang, Z. Mou, F. Gao, L. Xing, J. Jiang, and Z. Han, "Hierarchical deep reinforcement learning for backscattering data collection with multiple UAVs," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[23] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, "UAV trajectory planning for data collection from time-constrained IoT devices," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 34–46, 2020.

[24] C. Zhan and Y. Zeng, "Aerial–ground cost tradeoff for multi-UAV-enabled data collection in wireless sensor networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1937–1950, 2020.

[25] K. Wang, K. Yang, and C. S. Magurawalage, "Joint energy minimization and resource allocation in C-RAN with mobile cloud," *IEEE Transactions on Cloud Computing*, vol. 6, no. 3, pp. 760–770, 2018.

[26] T. Zhang, Y. Wang, Y. Liu, W. Xu, and A. Nallanathan, "Cache-enabling UAV communications: Network deployment and resource allocation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 11, pp. 7470–7483, 2020.

[27] J. Ji, K. Zhu, D. Niyato, and R. Wang, "Joint cache placement, flight trajectory, and transmission power optimization for multi-UAV assisted wireless networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5389–5403, 2020.

[28] B. Jiang, J. Yang, H. Xu, H. Song, and G. Zheng, "Multimedia data throughput maximization in Internet-of-Things system based on optimization of cache-enabled UAV," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3525–3532, 2019.

[29] Y. Wang, Z. Ru, K. Wang, and P. Huang, "Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing," *IEEE Transactions on Cybernetics*, pp. 1–14, 2019.

[30] X. Lin, V. Yajnanarayana, S. D. Muruganathan, S. Gao, H. Asplund, H. Maattanen, M. Bergstrom, S. Euler, and Y. . E. Wang, "The sky is not the limit: Lte for unmanned aerial vehicles," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 204–210, 2018.

[31] B. Van Der Bergh, A. Chiumento, and S. Pollin, "Lte in the sky: trading off propagation benefits with interference costs for aerial nodes," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 44–50, 2016.

[32] D. W. Matolak and R. Sun, "Air–ground channel characterization for unmanned aircraft systems—part III: The suburban and near-urban environments," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 6607–6618, 2017.

[33] D.-H. Tran, T. X. Vu, S. Chatzinotas, S. ShahbazPanahi, and B. Ottersten, "Coarse trajectory design for energy minimization in UAV-enabled wireless communications with latency constraints," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9483–9496, 2020.

[34] A. Asheralieva and D. Niyato, "Distributed dynamic resource management and pricing in the IoT systems with blockchain-as-a-service and UAV-enabled mobile edge computing," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1974–1993, 2020.

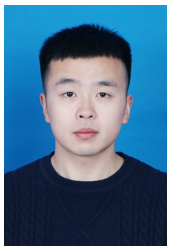
[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. USA: Cambridge University Press, 2004.

[36] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

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

[38] A. Zappone, E. Björnson, L. Sanguinetti, and E. Jorswieck, "Globally optimal energy-efficient power control and receiver design in wireless networks," *IEEE Transactions on Signal Processing*, vol. 65, no. 11, pp. 2844–2859, 2017.

[39] F. Jiang, K. Wang, L. Dong, C. Pan, W. Xu, and K. Yang, "Ai driven heterogeneous MEC system with UAV assistance for dynamic environment: Challenges and solutions," *IEEE Network*, vol. 35, no. 1, pp. 400–408, 2021.



Shuai Shen received his MSc from the School of Information and Control Engineering of China University of Mining and Technology, Xuzhou, China, in 2019. He is currently pursuing the Ph.D. degree in the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include resource efficiency of wireless communications, unmanned aerial vehicle communications and mobile edge computing.

Kun Yang received his PhD from the Department of Electronic & Electrical Engineering of University College London (UCL), UK. He is currently a Chair Professor in the School of Computer Science & Electronic Engineering, University of Essex, leading the Network Convergence Laboratory (NCL), UK. He is also an affiliated professor at UESTC, China. Before joining in University of Essex at 2003, he worked at UCL on several European Union (EU) research projects for several years. His main research interests include wireless networks and communications, data and energy integrated networks, computation-communication cooperation. He manages research projects funded by various sources such as UK EPSRC, EU FP7/H2020 and industries. He has published 150+ journal papers. He serves on the editorial boards of both IEEE and non-IEEE journals. He is a Senior Member of IEEE (since 2008) and a Member of Academia Europaea.

Kezhi Wang received his B.E. and M.E. degrees in School of Automation from Chongqing University, China, in 2008 and 2011, respectively. He received his Ph.D. degree in Engineering from the University of Warwick, U.K. in 2015. He was a Senior Research Officer in University of Essex, U.K. Currently he is a Senior Lecturer with Department of Computer and Information Sciences at Northumbria University, U.K. His research interests include wireless communications and machine learning.

Guopeng Zhang is currently a professor in the School of Computer Science and Technology, China University of Mining and Technology, China. He obtained his Ph.D. from the School of Communication Engineering at Xidian University, China in 2009. His main research interests include wireless networks, distributed machine learning, and network economics. He has published more than 70 journal and conference papers.

Haibo Mei received his BSc. and MSc. from School of Computer Science and Engineering of University of Electronic Science and Technology of China in 2005 and 2008 respectively. He received his Ph.D. degree from School of Electronic Engineering and Computer Science of Queen Mary University of London (QMUL), U.K. in 2012. He was a post doctoral research assistant in QMUL and a senior R&D engineer in Securus Software Ltd, U.K. He is currently a lecturer in School of Aeronautics and Astronautics, and a Post-doctoral in School of Communication and Information Engineering University of Electronic Science and Technology of China. His research interests include resource efficiency and self-organization of wireless communications, intelligent transportation system and mobile cloud computing.