Collaborative Data Acquisition for UAV-Aided IoTs based on Time-Balancing Scheduling

Mingyuan Ren, Xiuwen Fu, *Member, IEEE,* Pasquale Pace, *Member, IEEE,* Gianluca Aloi, *Member, IEEE,* Giancarlo Fortino, *Fellow, IEEE,*

***Abstract*—The emergence of the Internet of Things (IoT) has revolutionized various domains by enabling seamless connectivity and real-time data exchange between connected IoT devices. However, in sparse deployment scenarios where sensor nodes are sparsely distributed, ensuring low data delivery latency becomes a significant challenge. Our research aims to address this issue by utilizing unmanned aerial vehicles (UAVs) to support IoT networks. In the existing UAV-aided IoT systems, all UAVs are required to return to the base station to deliver data, which results in significant data delivery latency. To overcome this limitation, we propose a collaborative data acquisition model that uses air-to-air data relay between UAVs. By leveraging the mobility and agility of UAVs, the proposed system facilitates efficient data relay between sensor nodes and the base station. To further optimize the performance of the system, we present a time-balancing scheduling data acquisition (TSDA) scheme. This scheme combines a centripetal-based relay pairing method for UAVs to achieve seamless data relay and a joint scheduling scheme to minimize the hovering time during data delivery. Through extensive simulations, we demonstrate that the proposed TSDA scheme can achieve lower data delivery latency in sparse deployment scenarios compared to existing data acquisition schemes. In addition, the joint scheduling scheme can significantly reduce the hovering time of UAVs so that the collaborative relaying advantage can be better exploited.**

***Index Terms*—UAV-aided IoTs; delivery latency; data acquisition; time-balancing; joint scheduling**

I. Introduction

THE Internet of Things (IoT) is a technology that enables seamless interaction between the physical and digital realms of human existence [1]–[3]. It has broad applications in sectors such as industry [4]–[8], healthcare [9], [10], and transportation [11]. In IoT systems, the primary function of sensor nodes (SNs) is to efficiently deliver data to the base station (BS), enabling the system to respond quickly to unforeseen events. However, achieving low data delivery latency becomes a significant challenge when SNs are distributed in remote or unattended regions, especially in remote regions with limited infrastructure. In such scenarios, the use of unmanned aerial vehicles (UAVs) as mobile relay nodes has gained significant attention where traditional network infrastructure is absent or inadequate [12], [13]. We refer to such IoT systems that use UAVs to support SNs as UAV-aided IoT. This system is a promising solution to bridge the communication gap in remote regions, enabling data acquisition and delivery from SNs to the BS [14]. Moreover, this integration of UAVs and IoT brings a number of opportunities, offering improved connectivity, reduced latency and optimized data freshness.

In the IoT era, the freshness of information is a core factor for accurate user decision-making [15]. For example, in emergency response scenarios such as disaster management or sudden accidents, relying on outdated data can lead to misguided rescue efforts and potentially serious consequences. Consequently, the age of information (AoI), a metric that measures the freshness of IoT data, is crucial in assessing the data acquisition efficiency of UAV-aided IoT systems [16], [17]. The AoI takes into account factors such as delivery latency, data waiting time at SNs and data dwell time at the BS. A low AoI indicates that the information is fresh and the system responds quickly. Existing research on UAV-aided IoTs typically divides the task area into subareas equal to the number of UAVs. Each UAV follows an optimized flight path to acquire data within its assigned subarea. This approach ensures that there is no task overlap. However, it also presents a challenge: the UAVs responsible for gathering data from the marginal areas of the network must move to the BS to deliver data, which has a negative impact on the AoI.

For this reason, our research paper presents a collaborative data acquisition model for UAV-aided IoTs. On this basis, a time-balancing scheduling data acquisition scheme is developed to solve the collaboration problem of UAVs in the air-to-air delivery process. The main contributions in our work are summarized as follows.

- To enable the task collaboration between UAVs, a collaborative data acquisition model for UAV-aided IoTs is developed. This model involves three types of data delivery processes: ground-to-air process, air-to-air process, and air-to-ground process.
- In order to facilitate task collaboration among UAVs, a centripetal-based relay pairing method for UAVs is proposed. In this pairing method, the relay pairing relationship between UAVs is determined based on the angle and distance between UAVs and the BS.
- To reduce the hovering time of UAVs during data deliv-

M. Ren is with the Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, People's Republic of China (e-mail: renmy98@163.com);

X. Fu is with the Logistics Engineering College, Shanghai Maritime University, Shanghai 201306, People's Republic of China (e-mail: xwfu@shmtu.edu.cn);

P. Pace is with the Department of Informatics, Modeling, Electronics and Systems, University of Calabria, Italy and Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), Via P Bucci, I-87036 Arcavacata Di Rende, CS, Italy (email: p.pace@dimes.unical.it);

G. Aloi and G. Fortino are with the Department of Informatics, Modeling, Electronics and System Engineering, University of Calabria, 87036 Rende, Italy (e-mail: aloi@dimes.unical.it; fortino@dimes.unical.it);

ery, a joint scheduling scheme is proposed. The scheme allocates the hovering time of each UAV according to the difference in the length of the closed-loop flight path, and sets the take-off point and flight direction of each UAV according to the hovering time.

The remaining organization of this paper is outlined as follows. Section II briefly sums up the related works in literature. Section III describes the collaborative data acquisition model. Section IV presents the time-balancing scheduling data acquisition scheme. In Section V, simulation results are presented. Finally, the paper is concluded summarizing the main improvements.

II. Related work

Optimizing UAV flight paths to enhance system AoI has garnered a lot of attention in the field of UAV-aided IoT research. Tong *et al.* [18] proposed a collaborative SN distribution and path planning strategy for UAV-aided IoT. Liu *et al.* [19] used a unified approach combining dynamic programming and genetic algorithm (GA) to determine trajectories that are optimal in terms of both maximum AoI and average AoI. Abd-Elmagid *et al.* [20] designed a deep reinforcement learning (DRL) algorithm to minimize the weighted sum of AoI values for different processes of UAVs. Ahani *et al.* [21] developed an algorithm based on graph labeling for reducing the AoI of SNs. Yi *et al.* [22] investigated the issue of the age-optimal data gathering in IoT networks supported by UAVs, utilizing DRL as their method of study. Tong *et al.* [23] proposed a DRL algorithm to reduce the AoI and packet loss ratio of SNs. Liu *et al.* [24] designed a novel optimization strategy for path planning of UAVs that aimed to achieve age-optimal solutions by combining a clustering algorithm with GA. Jia *et al.* [25] proposed a solution framework based on dynamic programming to obtain the optimal path and data acquisition strategy. Ferdowsi *et al.* [26] developed a novel neural combinatorial-based DRL algorithm to reduce the AoI of SNs. Wang *et al.* [27] introduced a heuristic-based path planning algorithm aimed at minimizing the system AoI. The main idea of the algorithm is to direct the UAV to the SN with the highest AoI. Li *et al.* [28] designed two learning algorithms based on the Sarsa and value-decomposition network to plan the path of UAVs, minimizing the long-term return, which was characterized by the AoI of SNs, power consumption of UAVs, and penalty for inappropriate actions. Deng *et al.* [29] presented a path planning algorithm using the spiral method. The primary objective of this algorithm was to minimize the time required to complete the mission by efficiently planning the shortest flight path for the UAV while ensuring coverage of all SNs. Xu *et al.* [30] developed an improved grey wolf optimizer algorithm for an optimized multi-UAV cooperative path planning. Zhu *et al.* [31] designed a novel algorithm framework based on the weighted A* to minimize the total AoI of the acquired data by the UAV from the ground IoT network. In this algorithm, the selection of hovering points of the UAV and the visiting order to these hovering points were jointly optimized. Oubbati *et al.* [32] designed a multi-Agent DRL method that aimed to jointly optimize the trajectories of UAV teams and minimize the expected AoI. Liao *et al.* [33] proposed a multi-objective mixed integer linear programming with a flow-based constraint set to jointly optimize the average AoI and the aggregate energy consumption. Liu *et al.* [34] designed a deep Q network-based scheme for speedy preparation and path planning of UAVs. Wu *et al.* [35] proposed a comprehensive framework based on artificial intelligence for UAV path planning. The framework included a clustering module to determine optimal hovering positions for the UAV. A neural trajectory solver is used to generate the flight path that minimizes the AoI. Huang *et al.* [36] proposed an AoI-sensitive data acquisition scheme based on Hamilton path to achieve better AoI performance. This scheme used multiple UAVs to form a Hamiltonian loop to complete the data acquisition. While this approach incorporated the concept of air-to-air data relay, it restricted each UAV to establishing a relay pairing relationship with only one other UAV. This limitation somewhat reduced the flexibility of UAV data relay. We summarize the comparison between our work and the existing literature in Table I.

The literature review clearly indicates that the main trend in existing research is to treat UAVs as independent individuals, without any data relaying or task collaboration between them. The main challenge with this approach is that the UAVs tasked with covering the marginal task area have to move considerable distances to deliver data to the BS. This has a serious impact on the average system AoI. The collaborative relay mode, with its air-to-air data relay, provides an effective solution to this problem. In this mode, the UAV located at a significant distance from the BS (i.e., far-end UAV) does not need to return directly to the BS. Instead, the far-end UAV uses encounter opportunities to deliver data to a UAV nearer to the BS (i.e., near-end UAV), which then relays the data to the BS. Although there have been few studies that adopt air-to-air data relay, these studies do not support flexible pairing of UAVs, which to some extent limits the advantages of air-to-air data relay. The following three research challenges related to this delivery process need to be addressed:

1) In the UAV-aided IoTs, we first need to assign reasonable task areas to UAVs. When dividing task areas, we need to consider two factors. On one hand, to improve data acquisition efficiency, the task area of each UAV should avoid overlapping. Overlapping task areas can cause unnecessary flight paths for UAVs, resulting in a waste of data acquisition time. On the other hand, to provide sufficient opportunities for data relay of UAVs, the data acquisition task for each UAV in the divided task areas should be as balanced as possible. If a UAV undertakes too many data acquisition tasks, the overall system performance degrades due to the prolonged data acquisition time of that UAV. In this work, we select the data acquisition nodes using a centroid-based algorithm and generate closed-loop flight paths of UAVs with balanced data acquisition tasks using GA.

2) Once the task areas have been divided, the next challenge is determining which UAVs should act as data forwarders. Improperly assigning UAVs to forward data can result in ineffective air-to-air data relay and may even result in data not reaching the BS. Therefore, we need to rationalize the

TABLE I
COMPARISON BETWEEN OUR WORK AND THE EXISTING LITERATURE

Reference	Single UAV	Multi UAV	Air-to-air relay	Energy constraints	Flexible pairing
[18]	✓				
[19]	✓				
[20]	✓			✓	
[21]	✓			✓	
[22]	✓			✓	
[23]	✓			✓	
[24]	✓				
[25]	✓			✓	
[26]	✓				
[27]		✓		✓	
[28]		✓		✓	
[29]	✓			✓	
[30]		✓		✓	
[31]	✓			✓	
[32]		✓		✓	
[33]	✓			✓	
[34]	✓			✓	
[35]	✓				
[36]		✓	✓	✓	
Our work		✓	✓	✓	✓

pairing relationships between UAVs to ensure that data can be successfully delivered through feasible routes to the BS. In this work, we use a centripetal-based relay pairing method to establish relay pairing relationships between UAVs. The method establishes the relay relationship based on the distance and angle between UAVs and the BS, ensuring efficient transmission of data from the far-end UAVs to the BS through the relay of the near-end UAVs.

3) How to provide suitable opportunities for paired UAVs to meet and forward data in a timely manner is the core issue. Therefore, in the flight planning of UAVs, we need to consider the hovering time, flight direction and path modification of UAVs. In this work, we use the diffusion direction selection method to determine take-off points and flight directions of UAVs in a certain order. In this method, we take the BS as the center and first determine the take-off point and flight direction of the near-end UAV. We then gradually diffuse outward to determine the take-off point and flight direction of the far-end UAV based on the relay pairing relationship, achieving timely data relay between paired UAVs.

III. SYSTEM MODEL

A. Data acquisition model

Our research considers the sparse network scenario where a predetermined number of SNs are randomly distributed over the given task area. The constrained wireless communication range of the low-cost SNs results in the division of the sensor network into several subnets. The SN cannot communicate with SNs outside the subnet. Since the subnets are not connected to each other, SNs cannot deliver data to the BS through ground multi-hop data delivery, for which UAVs are needed to assist SNs to relay data to the BS. Fig.1 illustrates the data delivery from the SNs to the BS, which comprises three wireless delivery types: (a) ground-to-air; (b) air-to-air; (c) air-to-ground.

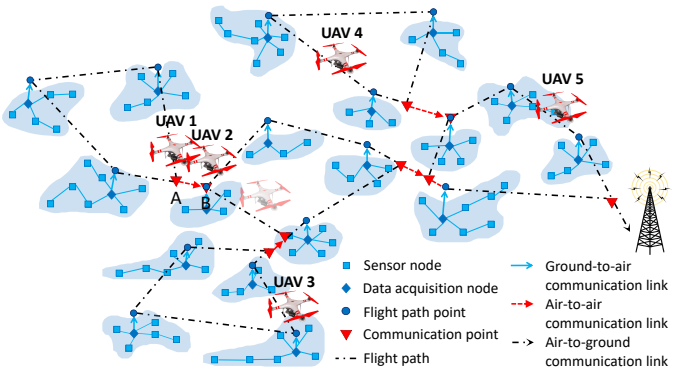


Fig. 1. Data acquisition model.

1) Ground-to-air: To guarantee the data acquisition effectiveness of UAVs, they can only establish communication with a limited number of SNs in the sensor network. Within each ground subnet, one SN is selected as the data acquisition node. Data from all SNs within the subnet is converged at the data acquisition node. The UAV can obtain the data of the subnet by establishing communication with the corresponding data acquisition node. To obtain data from all SNs, the flight paths of UAVs need to traverse all data acquisition nodes.

2) Air-to-air: For the far-end UAVs that are responsible for areas far from the BS, flying directly back to the BS can cause serious delivery latency for the data they carry. Therefore, we introduce a collaborative data relay of UAVs (i.e., air-to-air delivery process). The near-end UAV in charge of the area closer

to the BS receives the data acquired by the far-end UAV at the communication points. The function of the communication point is to ensure that when the UAV reaches this point, it can forward data through the wireless link to the designated UAV or the BS. When the two UAVs that are required to meet as planned cannot reach the communication points at the same time, we adopt the "hover and wait" approach. As shown in Fig.1, when UAV 1 is at communication point A and UAV 2 has not yet reached the designated communication point B, UAV 1 needs to hover. When UAV 2 arrives at communication point B, the two UAVs can establish communication. At this point, UAV 1 relay the data to UAV 2, and the air-to-air delivery process between the two UAVs is completed.

3) Air-to-ground: The near-end UAV is responsible for delivering data acquired by the far-end UAV to the BS. When the minimum distance between the near-end UAV and the BS exceeds the wireless communication distance, the near-end UAV is unable to deliver data to the BS [37]. In this case, we need to add additional communication points to the flight path of the near-end UAV to ensure reliable data delivery. As shown in Fig.2, the near-end UAV is unable to deliver data to the BS along the original flight path. For this reason, we add communication point A to the original flight path so that along the new flight path the UAV can send data to the BS at the communication point A.

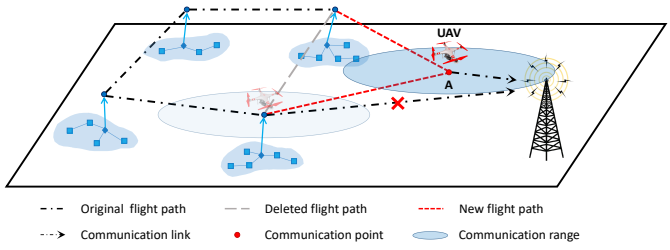


Fig. 2. Schematic diagram of the UAV communicating with the BS.

After the above delivery process, the data generated by all SNs can be delivered to the BS through the collaborative relay of UAVs.

B. AoI model

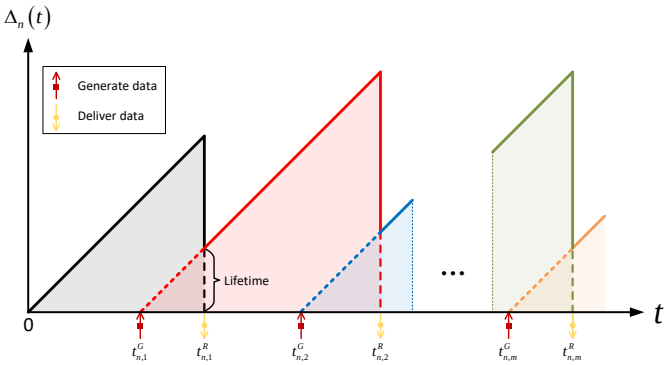


Fig. 3. The evolution of the AoI.

We present the AoI as a key performance indicator used to assess the freshness of data acquired from SNs. Specifically, the AoI is determined by the time elapsed since the last information generated by the SNs was received by the BS [38], [39]. For any given time slot t, we assume that $o_n(t)$ is the generation time when the BS receives the latest sample data from the SN n. Therefore, the AoI of the SN n can be determined as $\Delta_n(t) = t - o_n(t)$. In this work, we assume that $\Delta_n(t) = 0$ when $t = 0$. As shown in Fig.3, the AoI of the SN n at the BS either increases linearly or is determined by the latest data packet if it is successfully delivered to the BS. To illustrate with an example, as depicted in Fig.3, the first data packet is generated by the SN n at time $t^G_{n,1}$ and starts tracking its lifetime, followed by subsequent data packets at times $t^G_{n,2}, \ldots, t^G_{n,m}$. The lifetime of a data packet, which refers to the time it takes to be generated, delivered to the BS and finally replaced by the latest packet, shows a linear increase. As soon as one data packet has successfully delivered from the SN n to the BS by UAVs (i.e. $t = t^R_{n,1}, \ldots, t^R_{n,m}$), the AoI of the SN n at the BS is updated to the lifetime of this data packet. As mentioned in [38], [40], the average AoI of the SN n can be written as

$$\overline{\Delta}_n(t) = \frac{1}{T_C}\int_0^{T_C} \Delta_n(t)dt, \tag{1}$$

where T_C denotes the time to complete the data acquisition task. Considering the analysis of the average AoI of the SN, the average system AoI is calculated as follows:

$$\begin{aligned}\overline{\Delta}(t) &= \frac{1}{N_S}\sum_{n=1}^{N_S}\overline{\Delta}_n(t) \\ &= \frac{1}{N_S T_C}\sum_{n=1}^{N_S}\int_0^{T_C}\Delta_n(t)dt\end{aligned}, \tag{2}$$

where N_S represents the total number of SNs.

C. Communication channel model

We make the assumption that UAVs use line-of-sight (LoS) links as communication links between them. Furthermore, we require that the flight altitude of UAVs needs to exceed that of any obstacle to ensure unimpeded flight for all UAVs in free space area [41]. The model for path loss between UAVs is described as follows:

$$PL_{AA} = 20\log\left(\frac{4\pi f_S d_{UU}}{v_L}\right) + \eta_{LoS}, \tag{3}$$

where f_S denotes the signal frequency; v_L denotes the light speed; d_{UU} is distance between UAVs; η_{LoS} is the LoS-attenuation factor.

The ground-to-air wireless communication channel includes probabilistic LoS and non-line-of-sight (NLoS) parts. The path loss for the probabilistic LoS and NLoS parts can be described as follows:

$$L_{LoS} = 20\log\left(\frac{4\pi f_S d_{US}}{v_L}\right) + \eta_{LoS}, \tag{4}$$

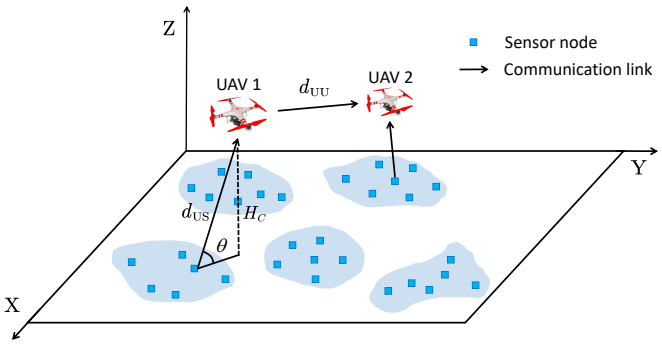


Fig. 4. Communication channel model.

$$L_{NLoS} = 20\log\left(\frac{4\pi f_S d_{US}}{v_L}\right) + \eta_{NLoS}, \tag{5}$$

where d_{US} represents the distance between the UAV and the SN and η_{NLoS} signifies the attenuation factor resulting from the NLoS connection. Taking into account this particular scenario, the probabilities associated with the presence of the LoS and NLoS parts in the ground-to-air wireless communication channel can be expressed as follows:

$$P_{LoS}(\theta) = \frac{1}{1+\psi\exp(-\beta(\theta-\psi))}, \tag{6}$$

$$P_{NLoS}(\theta) = 1 - P_{LoS}(\theta), \tag{7}$$

where the constants ψ and β are parameters that are determined based on the specific characteristics of the environment in which the system operates [42]; θ denotes the elevation angle of the communication between the SN and the UAV. According to Fig.4, we notice that $\theta = \arcsin(\frac{H_C}{d_{US}})$. H_C is the flight altitude of UAVs. The ground-to-air path loss takes into account both probabilistic LoS and NLoS parts, which is defined as

$$\begin{aligned} PL_{AG} &= L_{LoS}P_{LoS}(\theta) + L_{NLoS}P_{NLoS}(\theta) \\ &= \frac{\eta_{LoS}-\eta_{NLoS}}{1+\psi\exp(-\beta(\arcsin(\frac{H_C}{d_{US}})-\psi))} \\ &\quad + 20\log\left(\frac{4\pi f_S d_{US}}{v_L}\right) + \eta_{NLoS}. \end{aligned} \tag{8}$$

To guarantee the wireless communication quality, PL_{AA} and PL_{AG} are both less than or equal to threshold value PL_{TH}^{AA} and PL_{TH}^{AG}, respectively (i.e., $PL_{AA} \leq PL_{TH}^{AA}$, $PL_{AG} \leq PL_{TH}^{AG}$) [43]–[45]. In this work, we define the maximum communication distance between UAVs as $D_{AA} = d_{UU}|_{PL_{AA}=PL_{TH}^{AA}}$. Also, we define the maximum communication distance between the UAV and the SN as $D_{AG} = d_{US}|_{PL_{AG}=PL_{TH}^{AG}}$.

D. Energy model

The energy consumption of the UAV is divided into two main components: the energy used for communication and the energy used for propulsion. However, as the energy required for communication is comparatively small, we focus primarily on propulsion energy consumption [46], [47]. We use E_U to represent the initial energy capacity of the UAV. The propulsion power consumption can be expressed by the following formula:

$$\begin{aligned} \widetilde{P}(v_t) &= P_{BLA}\left(1+\frac{3v_t^2}{U_{TIP}^2}\right) + P_{IND}\left(\sqrt{1+\frac{v_t^4}{4v_0^4}} - \frac{v_t^2}{2v_0^2}\right)^{\frac{1}{2}} \\ &\quad + \frac{1}{2}d_0\rho s_0 A v_t^3, \end{aligned} \tag{9}$$

where P_{BLA} and P_{IND} denote the power consumed by the blade and induced power, respectively, when the UAV hovers; the UAV's speed at any given time t, is represented as v_t; U_{TIP} denotes the tip speed of the UAV; v_0 is the average rotor induced velocity during hovering. The fuselage drag ratio is denoted by d_0. The air density is represented by ρ, while s_0 refers to the solidity of the rotor. The area of the rotor disk is represented by A. We set the BS with charging function. The UAV can be efficiently recharged by returning to the BS during data acquisition. At this point, each UAV should satisfy

$$\sum_{t=1}^{T_{RB}} \widetilde{P}(v_t)\tau \leq E_U, \tag{10}$$

where T_{RB} denotes the time slot when the UAV returns to the BS ; τ denotes the time length of each slot.

IV. Time-Balancing Scheduling Data Acquisition Scheme

A. TSDA scheme framework

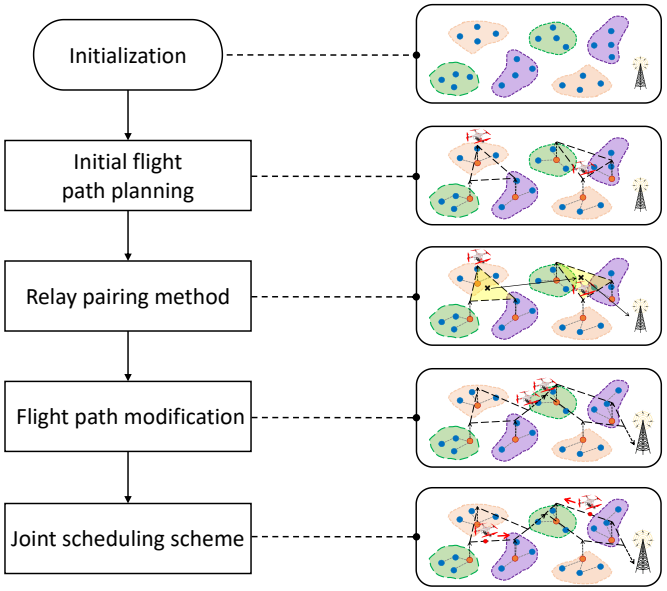


Fig. 5. Time-balancing scheduling data acquisition scheme.

To realize collaborative data acquisition by multiple UAVs, a time-balancing scheduling data acquisition scheme (TSDA) is proposed. As shown in Fig.5, in the TSDA scheme, the initial paths of UAVs that traverse all data acquisition nodes are first generated. Then, to achieve seamless data relay between UAVs, a centripetal-based relay pairing method for

UAVs is proposed. On this basis, we modify the initial flight paths depending on the relay pairing relationship. Finally, the hovering time, take-off points and flight directions of UAVs are jointly planned to ensure the timeliness of data delivery between UAVs.

B. Selection of data acquisition nodes

Considering the limited wireless communication range of SNs, we make the assumption that the sensor network is composed of N_C subnets. In order to efficiently acquire data from the subnets, we designate a specific SN as a data acquisition node from each subnet. Its role is to acquire the data generated within its respective subnet, allowing data to be acquired whenever communication is established between the UAV and the data acquisition node. To determine the data acquisition nodes for the subnets, we propose a centroid-based algorithm. This algorithm enables the selection of optimal data acquisition nodes for effective data acquisition within each subnet.

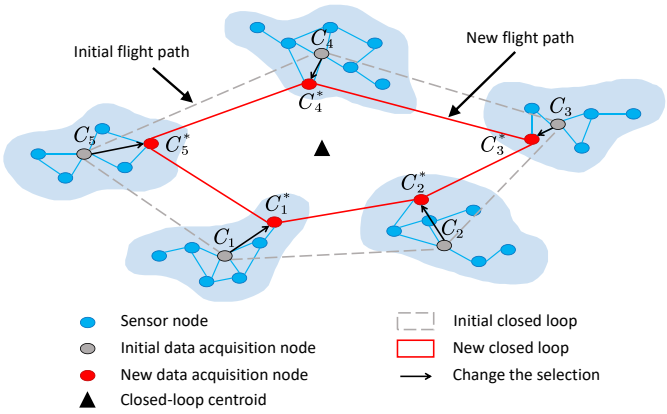


Fig. 6. Selection of data acquisition nodes.

As depicted in Fig.6, we designate the SN in each subnet that is nearest to the centroid as the initial data acquisition node, denoted by the set $\{C_1, C_2, C_3, C_4, C_5\}$. The centroid in each subnet can be calculated by

$$x_i^S = \frac{\sum_{j=1}^{n_i^S} x_{i,j}}{n_i^S}, \tag{11}$$

$$y_i^S = \frac{\sum_{j=1}^{n_i^S} y_{i,j}}{n_i^S}, \tag{12}$$

where (x_i^S, y_i^S) denotes the coordinates of centroid in subnet i; $(x_{i,j}, y_{i,j})$ denotes the coordinates of the SN j in subnet i; n_i^S denotes the number of SNs in subnet i. Based on the initial data acquisition nodes, we can generate the initial flight paths for the UAVs, as detailed in Section IV-C. These flight paths can form several closed loops. A closed loop is defined as a complete circle that a UAV can traverse starting from a point and eventually return to the same point without retracing any part of the flight path. By leveraging these closed loops delineated by the initial flight paths of UAVs, the centroid of each closed loop can be calculated by

$$x_k^C = \frac{\sum_{i=1}^{N_C} w_{i,k}^S x_i^S}{\sum_{i=1}^{N_C} w_{i,k}^S}, \tag{13}$$

$$y_k^C = \frac{\sum_{i=1}^{N_C} w_{i,k}^S y_i^S}{\sum_{i=1}^{N_C} w_{i,k}^S}, \tag{14}$$

where (x_k^C, y_k^C) denotes the coordinates of closed-loop centroid k; $w_{i,k}^S = 1$ denotes the centroid of subnet i is in the closed loop k; $w_{i,k}^S = 0$ denotes the centroid of subnet i is not in the closed loop k; Finally, we designate the SN that is nearest to the closed-loop centroid as the data acquisition node $\{C_1^*, C_2^*, C_3^*, C_4^*, C_5^*\}$, which is shown in Fig.6. By contracting all the closed loops through the above steps, we can obtain the new flight path, and the data acquisition nodes that comprise these closed loops are our finalized data acquisition nodes. The detailed procedures of data acquisition node selection are summarized in Algorithm 1. P_i^D denotes the set of SNs in subnet i and P^C denotes the set of determined data acquisition nodes.

Algorithm 1 Data acquisition nodes selection algorithm

Input: $P^D = \{P_1^D, P_2^D, \ldots, P_{N_C}^D\}$
Output: P^C

```
for each P_i^D in P^D do
    Calculate the centroid of subnet i
    Select the initial data acquisition node in subnet i
end for
Calculate centroids of closed loops
for each P_i^D in P^D do
    Update the data acquisition node in subnet i
end for
return P^C
```

C. Initial flight path planning of UAVs

In order to acquire data from all subnets, we generate initial flight paths for UAVs that can efficiently traverse all the designated data acquisition nodes. This path planning problem can be regarded as a typical multiple traveling salesman problem (MTSP). To solve the problem, we assume that N_A UAVs are deployed, the following decision variables can be defined:

$$w_{i,j,k}^U = \begin{cases} 1, \text{UAV } k \text{ flies from } p_i^C \text{ to } p_j^C \\ 0, \text{otherwise} \end{cases}, \tag{15}$$

$$w_{j,k}^C = \begin{cases} 1, \text{UAV } k \text{ traverses } p_j^C \\ 0, \text{otherwise} \end{cases}, \tag{16}$$

where p_i^C denotes the data acquisition node selected for subnet i; $d_{i,j}^C$ denotes the Euclidean distance between data acquisition

nodes p_i^C and p_j^C. On this basis, the objective function is set as follows

$$\min \left\{ \max_{1 \le k \le N_A} \left\{ \sum_{i=1}^{N_C} \sum_{j=1}^{N_C} d_{i,j}^C w_{i,j,k}^U \right\} \right\}, \tag{17}$$

$$\text{s.t.} \quad \sum_{i=1}^{N_C} w_{i,j,k}^U + \sum_{i=1}^{N_C} w_{j,i,k}^U \in \{0, 2\}, \tag{18}$$

$$\sum_{k=1}^{N_A} w_{j,k}^C = 1, \tag{19}$$

$$u_{i,k} - u_{j,k} + N_C w_{i,j,k}^U \le N_C - 1, \tag{20}$$

$$\sum_{j=1}^{N_C} w_{j,k}^C \ge N_L, \tag{21}$$

where $u_{i,k}$ is an auxiliary variable representing the order in which UAV *k* traverses the data acquisition nodes p_i^C in its flight path. The objective function (17) is used to require that the closed-loop flight path of the UAV with the longest flight path length among N_A UAVs should be as short as possible; Eq.(18) requires that the flight path of each UAV is a closed-loop path; Eq.(19) requires that each data acquisition node can only be acquired by one UAV and only once; Eq.(20) is used to constrain each UAV not to generate subtours; Eq.(21) requires each UAV to acquire at least N_L data acquisition nodes. N_L is set to 3 because the number of nodes to be traversed by the UAV to generate a closed-loop path is at least 3. By adding this constraint, the solution space of the problem can be significantly reduced.

GA has superior global search capabilities compared to other heuristic algorithms. When used as a basic method or combined with other heuristic algorithms as a hybrid algorithm, GA can produce high diversity and high-quality samples, avoiding falling into local optima prematurely and having higher chances of finding the global optimum of multimodal functions [48]–[50]. Therefore, GA is widely used to solve complex combinatorial optimization problems such as MTSP. Additionally, many studies [51]–[53] have shown that compared to other heuristic algorithms, GA can flexibly handle constraints in different scenarios when solving MTSP while maintaining high computational efficiency. Based on these two reasons, this study chooses the GA algorithm to solve the proposed path planning problem. The details of this algorithm are as follows.

1) Encoding: We use the two-part chromosome coding method to denote the pairing relationship between UAVs and data acquisition nodes [54]–[56]. This method expresses the path as a sequential arrangement of numbers where each data acquisition node is assigned to a different sequential number in advance. We organize these numbers in the order of the UAVs visit. To illustrate the concept, we consider the scenario where three UAVs are navigating through ten data acquisition nodes, as depicted in Fig.7. Chromosome I represents the flight path of the UAV through the ten data acquisition nodes. Chromosome II represents the sequential location of the breakpoints of the path that divides chromosome I into multiple segments. On the basis of the coding of chromosome I and chromosome II, the paths of UAVs are presented in Fig.7. UAV 1 starts from point 6, passes point 2 and point 5, finally returns to node 6; UAV 2 starts from point 1, passes point 3, point 9 and point 4, finally returns to point 1; UAV 3 starts from point 10, passes point 8 and point 7, finally returns to point 10. To satisfy the requirement that each UAV must traverse at least N_L data acquisition nodes in the scenario with N_C data acquisition nodes, the coding of chromosome II must satisfy the following constraint:

(a) Arrange the numbers in chromosome II in ascending order.

(b) Ensure that the difference between any two adjacent numbers is not less than N_L.

(c) The first number should be larger than or equal to N_L.

(d) The difference between N_C and the last number should be greater than or equal to N_L.

2) Fitness function: The evaluation of the population is determined by the fitness function (17). In most cases, a higher fitness value indicates a higher adaptability of an individual to the environment. In this context, the fitness value refers to the reciprocal of the maximum path distance of all UAV flight paths where the maximum path distance is smaller, implying a better path solution.

3) Selection: After evaluating the fitness value of each individual in the current population, our reproduction process for the next generation involves selecting individuals using roulette selection and elite strategy. The elite strategy first identifies the best performing individual and preserves it for the next generation. This ensures that the individual with the highest fitness value is undoubtedly carried forward to the next iteration. The remaining individuals are then selected using the roulette wheel method. The roulette selection and elite strategy not only ensure that a higher proportion of dominant individuals are retained in each iteration, but also help to prevent the algorithm from converging prematurely on a particular solution.

4) Mutation: Mutation operations play a vital role in generating new individuals and enhancing the diversity within the population. To ensure effective global search capability, a range of mutation operators are employed.

For chromosome I, we employ four specific types of mutation operations: flip, swap and slide, as illustrated in Fig.8. Prior to performing the mutation, we first randomly select a section, represented by the range from I to J. This means that the segment from position I to J within the path undergoes mutation. Furthermore, we randomly generate an insertion position P, taking into consideration the values of I and J. The path mutation operation is as follows:

a) Flip: The sequence of numbers within the segment from position I to J is reversed. Subsequently, the entire segment from I to J, in its reversed order, is inserted at the randomly generated insertion position P.

b) Swap: The number at position I is swapped with the number at position J. After the swap, the entire segment from I to J is inserted at the insertion position P.

c) Slide: The numbers within the segment from position I to J is cyclically shifted one position in the same direction.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3339136

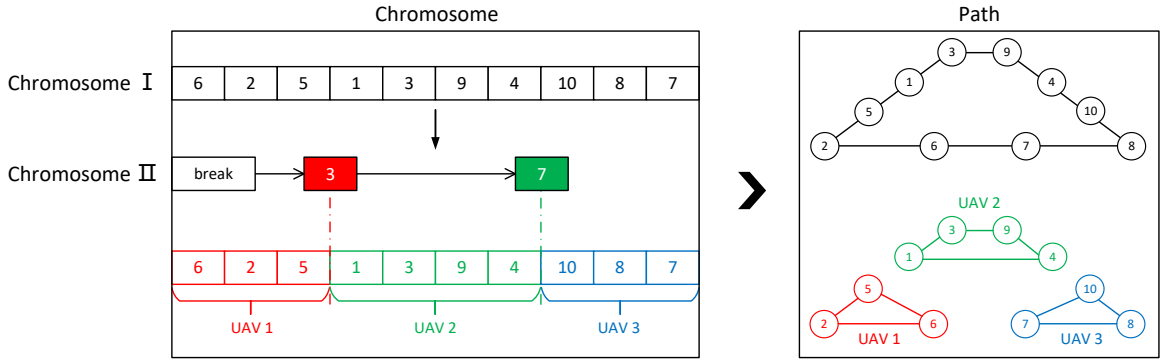


Fig. 7. Schematic diagram of two-part chromosome encoding.

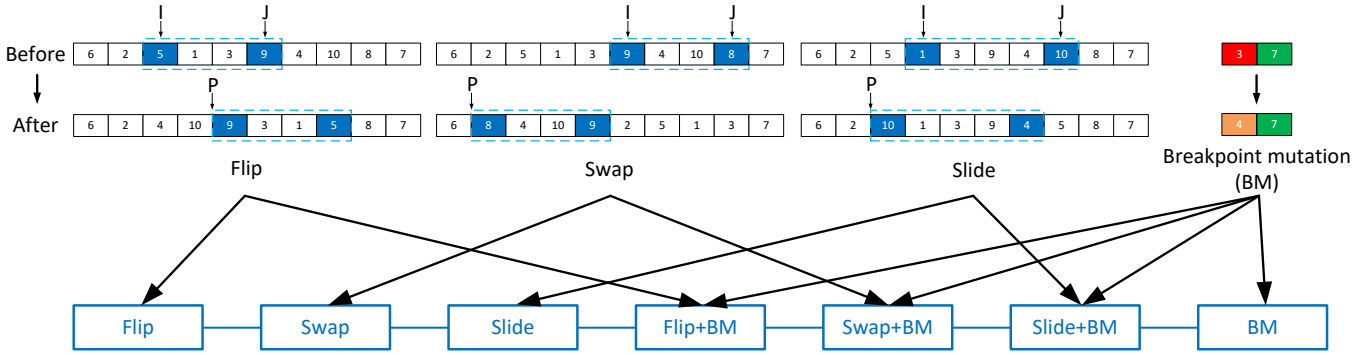


Fig. 8. Schematic diagram of chromosome mutation operator.

Following the shift, the entire segment from I to J is inserted at the insertion position P.

For chromosome II, the gene is regenerated by random regeneration under satisfying coding constraints. Therefore, when an individual is mutated, seven new individuals are created by combining the mutations of chromosome I and chromosome II, as shown in Fig.8. Among these eight individuals, the one with the highest fitness value is selected as the result of the mutation and kept in the population.

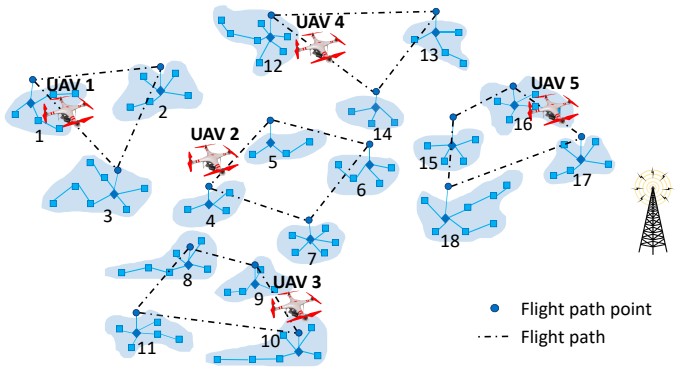


Fig. 9. Initial flight path planning of UAVs.

Fig.9 shows an example of initial flight path planning of UAVs. In this example, the far-end UAV 1 is responsible for data acquisition in subnets 1 to 3 and the near-end UAV 2 is responsible for data acquisition in subnets 4 to 7. Although all subnets in the network can be traversed at this point, there is no suitable encounter opportunity between the far-end UAV 1 and the near-end UAV 2. For this reason, the initial flight path needs to be modified to ensure that the data acquired by the far-end UAV can reach the BS via the relay of the near-end UAV.

D. Relay pairing method for UAVs

After completing the initial path planning of UAVs, the next problem we need to solve is which the near-end UAV should be used to forward the data from the far-end UAV. Since all data in IoTs is aggregated at the BS, the message routing direction is usually directed to the BS (i.e., centripetal feature). Therefore, a centripetal-based relay pairing method for UAVs is designed, the purpose of which is to avoid invalid data delivery between UAVs by reasonably pairing the far-end and the near-end UAVs.

The basic idea of the relay pairing method is: (i) in the case that the distances between all candidate pairing UAVs and the target UAV are similar, the target UAV prefers to establish a relay pairing relationship with the candidate pairing UAV with better centripetal performance; (ii) in the case that all candidate pairing UAVs have similar centripetal performance, the target UAV prefers to establish a relay pairing relationship with the candidate pairing UAV that is closer to it. Here, the mentioned centripetal performance is negatively related to the angle formed by taking the target UAV as the vertex and the candidate pairing UAV and the BS as the endpoint. The smaller the angle is, the better the centripetal performance of the candidate pairing UAV. When the angle is zero, the target

UAV, the candidate pairing UAV and the BS form a straight line. In this case, the centripetal performance of the candidate pairing UAV is the best. Based on this idea, we propose a relay pairing relationship evaluation function

$$g_{i,j} = \frac{1}{\alpha \frac{a_{i,j}}{\max\limits_{1\le j\le N_A}\{a_{i,j}\}} + (1-\alpha)\frac{d^S_{i,j}}{\max\limits_{1\le j\le N_A}\{d^S_{i,j}\}}}, \tag{22}$$

where α is the centripetal weight; $a_{i,j}$ is the angle formed by taking the centroid i as the vertex and the centroid j and the BS as the endpoint; $d^S_{i,j}$ is the distance between the centroid i and the centroid j. The centroid i denotes the centroid of the closed-loop path of UAV i, which is the average value of coordinates of all data acquisition nodes in this closed-loop path. A larger $g_{i,j}$ indicates a higher pairing degree between UAV i and UAV j. For UAV i, we can calculate the pairing degree between UAV i and other UAVs according to Eq.(22), and select the UAV with the maximum pairing degree to build relay pairing relationship with UAV i.

Algorithm 2 presents the specific process of the relay pairing method for UAVs. P^R_i denotes the set of data acquisition nodes for the closed-loop flight path of UAV i, p^B is the BS point, $P^S = \{p^S_1, p^S_2 \dots p^S_{N_A}\}$ is the set of closed-loop centroids.

Algorithm 2 Centripetal-based relay pairing method

Input: $P^R = \{P^R_1, P^R_2 \dots P^R_{N_A}\}$, p^B
Output: Relay pairing relationship

for each P^R_i **in** P^R **do**
  Calculate closed-loop centroid p^S_i from P^R_i
end for
Obtain the closest point p^S_i to the BS and establish a pairing relationship between UAV i and the BS
$P^{S*} \leftarrow P^S \backslash \{p^S_i\}$
for each p^{S*}_j **in** P^{S*} **do**
  for each p^S_k **in** P^S **do**
    Calculate $g_{j,k}$
  end for
  Obtain $\max\limits_{1\le k\le N_A}\{g_{j,k}\}$ and optimal point p^S_k
  if $\|p^{S*}_j, p^B\| > \|p^{S*}_j, p^S_k\|$ **and** $\|p^{S*}_j, p^B\| > \|p^S_k, p^B\|$ **then**
    Establish a pairing relationship between UAV j and UAV k
  else
    Establish a pairing relationship between UAV j and the BS
  end if
end for
return Relay pairing relationship

Fig.10 depicts an example of establishing relay pairing relationships for UAVs. As shown in Fig.10(a), the closed-loop centroid E is the closest point to the BS among all closed-loop centroids A, B, C, D, and E. Therefore, UAV 5 corresponding to the centroid E is selected to establish a relay pairing relationship with the BS. Through this relay pairing relationship, UAV 5 acts as the near-end UAV and is in charge for relaying data from other UAVs. For the closed-loop centroid B, the distance between centroids B and D is similar to the distance between centroids B and E, but the centripetal angle θ_4 is significantly greater than θ_3. In this case, according to Eq.(22), UAV 2 is directly paired with UAV 5. For the closed-loop centroid C, the closed-loop centroids B and D corresponding to UAV 2 and UAV 4 have the similar centripetal angle, but the distance between the centroids C and B is significantly smaller than the distance between the centroids C and D. In this case, according to Eq.(22), UAV 3 is directly paired with UAV 2. According to the proposed relay pairing method, the relay pairing relationship of all UAVs can be obtained, as shown in Fig.10(b). In this relay pairing relationship, the data from the far-end UAV 1 and UAV 3 reaches the BS via the relay of UAV 2 and UAV 5; the data from the far-end UAV 4 reaches the BS via the relay of near-end UAV 5.

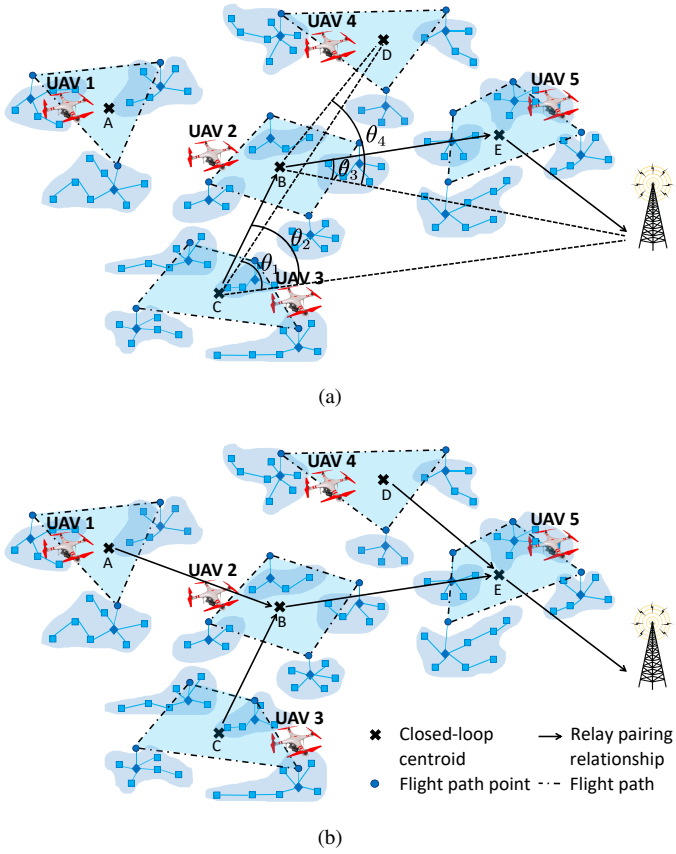


Fig. 10. Relay pairing relationships between UAVs and the BS.

E. Flight path modification of UAVs

After obtaining relay pairing relationships of UAVs, there may be a situation where the paired UAVs cannot establish communication. Therefore, we need to further modify the paths of UAVs by adding additional communication points. The path modification between the far-end UAV and the near-end UAV is a typical complex problem with huge solution space. The objective function of this problem is as follows

$$\min \sum_{i=1}^{N_A}\sum_{j=1}^{N_A} w^F_{i,j} \left| d^F_i - d^F_j \right|, \tag{23}$$

$$\text{s.t. } p_{i,j}^L, p_{j,i}^L \in l_{i,j}^C, \tag{24}$$

$$D_S \leq d_{i,j}^C - (d_{i,j}^F + d_{j,i}^F) \leq D_R^{AA}, \tag{25}$$

where d_i^F is the length of modified flight path of UAV i; $w_{i,j}^F$ is the relay pairing relationship between UAVs i and j obtained by Eq.(22); $w_{i,j}^F = 1$ denotes the data of UAV i is relayed by UAV j; $w_{i,j}^F = 0$ denotes that there is no relay pairing relationship between UAV i and UAV j; $p_{i,j}^L$ denotes the communication point on the path of UAV i to establish communication with the paired relay UAV j; $p_{j,i}^L$ denotes the communication point on the path of UAV j to establish communication with the paired relay UAV i; $l_{i,j}^C$ denotes the line generated by connecting the two points closest to each other on the flight paths of UAV i and UAV j; $d_{i,j}^C$ is the length of the line $l_{i,j}^C$; $d_{i,j}^F$ is the distance between communication point $p_{i,j}^L$ and the closest data acquisition node on the initial flight path of UAV i from communication point $p_{i,j}^L$; D_S is the minimum safe separation distance between UAVs; The objective function (23) is used to require that the length of the modified path of each UAV differs as little as possible from each other and the total modified path length of all UAVs are as short as possible. Meanwhile, according to the constraints (24) and (25), the path modification should satisfy that there is no intersection and overlap between the modified flight paths of UAVs, and all UAVs can conform to the pre-defined relay pairing relationship. This problem is NP-hard. We use GA to solve it. The algorithm's specifics are outlined below.

1) Initial Population: The algorithm starts by generating an initial population of potential solutions using binary coding. Each individual in the population represents a candidate solution to the problem.Binary coding represents the individuals represented by different variable values as a distinct binary string of zeros and ones. To ensure the accuracy of the solution, we set the solution to be accurate to 2 decimal places.According to constraint (24), the corresponding variable interval $\left[X_{i,j}^{\min}, X_{i,j}^{\max}\right]$. To satisfy the accuracy of the solution, the encoded binary string adapts the string length according to the range of the interval. For this reason, we set the string length of the binary string to $N_{i,j}^B$, which should satisfy $2^{N_{i,j}^B-1} < \left(X_{i,j}^{\max} - X_{i,j}^{\min}\right) 10^2 \leq 2^{N_{i,j}^B}$. The binary string of length N_B discretises the interval into $2^{N_B} - 1$ equal regions, generating a total of $2^{N_{i,j}^B}$ discrete variables, including both ends of the interval. We randomly selected N_P individuals from these discrete variables as the initial population to ensure diversity in the solution space.

2) Selection: Once the initial population is generated, the selection process identifies individuals for reproduction based on their fitness. We use an improved roulette selection method to select individuals (see Section IV-C for details).

3) Crossover: Once the selection phase is completed, the algorithm moves on to the crossover operation. It involves combining genetic material from selected individuals to create offspring. In the case of binary coding, crossover is performed by exchanging segments between two parent individuals. The position of the crossover points is determined randomly. This exchange of genetic material introduces new combinations, potentially generating individuals with improved fitness compared to their parents.

4) Mutation: Following the crossover operation, the algorithm introduces variation through the mutation process. Mutation provides a mechanism to introduce small random changes in the genetic material of individuals. In the case of binary coding, mutation involves flipping specific bits in the individual's binary string with a predefined mutation rate. The mutation rate determines the probability of a bit being flipped. Flipping a binary string means changing the value of a bit from 0 to 1, or vice versa. By introducing these random changes, the algorithm can explore different regions of the search space and potentially escape local optima.This mutation process helps maintain diversity in the population and prevents premature convergence to suboptimal solutions.

5) Fitness function and Iteration: After crossover and variation, the fitness of the offspring individuals is evaluated using the objective function (23). In this context, the fitness value is defined as the reciprocal of the sum of the distance differences among all UAV flight paths. A smaller value of the distance differences indicates a better solution, leading to a higher fitness value. We repeat the process of selection, crossover, mutation, and replacement until we have gone through a set number of iterative generations.

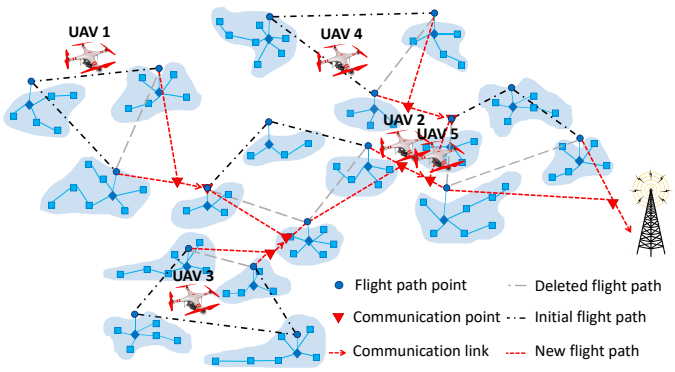


Fig. 11. Path modification between UAVs.

An example of path modification between UAVs is given in Fig.11. Before the path modification, the initial flight path of each UAV cannot effectively support air-to-air data relay. After the modification of the flight paths, the paired UAVs can establish reliable communication, while further reducing the disparity in flight path lengths. This improvement is more conducive to the following time-balancing scheduling operation.

F. Joint scheduling scheme of UAVs

To increase the efficiency of UAV data delivery by minimizing hovering time, it is essential to develop a comprehensive scheduling scheme for each UAV. This scheme involves two essential steps. First, the optimal hovering time for each UAV must be determined. Secondly, the take-off point and flight direction for the closed-loop flight path of each UAV are determined.

1) Hovering time scheduling: A hovering time scheduling method is proposed to improve the consistency of the time required for each UAV to complete a lap. The method achieves this by increasing the hovering time of the UAVs at the communication point. This method is based on two settings: (i) in the case of paired UAVs, the UAV with the shorter path length is designated to hover in order to synchronize with the UAV that has a longer path length; (ii) the larger the path length difference, the longer the UAV hovering time. Based on the above settings, for UAVs *i* and *j* that have established a relay pairing relationship, the time that UAV *i* waits for UAV *j* at communication point $P_{i,j}$ is

$$t_{i,j}^{H} = q_{i,j} t_i^{D}, \tag{26}$$

where $q_{i,j}$ is the time allocation ratio and t_i^D is the sum of the hovering time at each communication point for one lap of UAV *i*. The time allocation ratio $q_{i,j}$ can be calculated by

$$q_{i,j} = \begin{cases} \frac{w_{i,j}^F t_{i,j}^F}{\sum_{j=1}^{N_A} w_{i,j}^F t_{i,j}^F}, \sum_{j=1}^{N_A} w_{i,j}^F t_{i,j}^F > 0 \\ 0, \text{otherwise} \end{cases}, \tag{27}$$

$$t_{i,j}^F = \begin{cases} \frac{d_j^F - d_i^F}{v}, d_j^F - d_i^F > 0 \\ 0, \text{otherwise} \end{cases}, \tag{28}$$

where $t_{i,j}^F$ is the difference in path length between UAV *i* and UAV *j* for one lap of flight; v is the flight speed of UAVs. In the calculation of $t_{i,j}^F$, only the case where the path length of UAV *j* is greater than the path length of UAV *i* is regarded. For other cases, $t_{i,j}^F$ is set to 0 since UAV *i* does not need to wait for UAV *j*. The sum of hovering time t_i^D can be calculated by

$$t_i^D = \frac{\max\limits_{1 \le i \le N_A} \{d_i^F\} - d_i^F}{v}. \tag{29}$$

2) Take-off points and flight directions of UAVs: After the hovering time scheduling of UAVs, the take-off point and flight direction of each UAV also need to be configured. According to the hovering time at the communication point, the take-off point and the flight direction of the far-end UAV need to satisfy the following constraints: (i) when the UAV takes off from the take-off point, it chooses a flight direction that allows it to acquire as many data packets from the SNs as possible before reaching the communication point; (ii) at the end of the hovering time, the UAV has the ability to deliver data to another UAV that has an established relay pairing relationship.

In order to generate take-off points and flight directions that can satisfy all the above conditions, a diffusive direction selection method is proposed. The basic idea of this method is to take the BS as the center, and first confirm the take-off points and flight directions of UAVs near the BS, and gradually extend outward to determine the take-off points and flight directions of the remaining UAVs based on the relay pairing relationships. For ease of understanding, the details of this method is presented in Algorithm 3. t_i^P denotes the time consumed by UAV *i* to reach the communication point from the set take-off point in the set flight direction; p_i^T denotes the take-off point of UAV *i*; p_i^{FD} and p_i^{RD} denote the candidate take-off points for UAV *i* in the forward and reverse flight directions, respectively; n_i^{FD} denotes the number of data packets from SNs that the UAV *i* acquires in forward flight direction from p_i^{FD} to $p_{k,i}^L$; n_i^{RD} denotes the number of data packets from SNs that the UAV *i* acquires in reverse flight direction from p_i^{RD} to $p_{k,i}^L$.

Algorithm 3 Diffusive direction selection method

Input: the set of data acquisition nodes P^R, the set of hovering time for communication points $T^H = \{t_{i,j}^H, \cdots, t_{j,i}^H\}$, the set of communication points $P^L = \{p_{i,j}^L, \cdots, p_{j,i}^L\}$

Output: The set of take-off points for UAVs $P^T = \{p_1^T, p_2^T \cdots p_{N_A}^T\}$, flight directions of UAVs

for each $p_{i,j}^L$ **in** P^L **do**
  if $p_{i,j}^L$ can communicate directly with the base station **then**
    $p_i^T \leftarrow p_{i,j}^L$
    Select the direction that takes longer to reach p_i^{Lmax} from p_i^T as the flight direction in both forward and reverse directions.
    $p_{TEM} \leftarrow p_{i,j}^L$; $p_{END} \leftarrow p_{i,j}^L$
    while return to p_{END} **do**
      Move from p_{TEM} in the current closed-loop path where p_{TEM} is located in the set flight direction until you reach $p_{i,k}^L$ or p_{END}
      Update t_i^P
      if the closed-loop path where $p_{k,i}^L$ is located is not set take-off point and flight direction of UAV *k* **then**
        From $p_{k,i}^L$ takes time $t_i^P + t_{i,k}^H - t_{k,i}^H$ to move in reverse and forward direction directions to get p_k^{FD} and p_k^{RD}
        if $n_k^{FD} \ge n_k^{RD}$ **then**
          $p_k^T \leftarrow p_k^{FD}$; set UAV *k* to fly in forward direction
        else
          $p_k^T \leftarrow p_k^{RD}$; set UAV *k* to fly in reverse direction
        end if
        $p_{TEM} \leftarrow p_{k,i}^L$
      else
        $p_{TEM} \leftarrow p_{k,i}^L$
      end if
    end while
  end if
end for
return P^T, flight directions of UAVs

After setting the take-off point, flight direction and hovering time of each UAV, we can get the joint scheduling plan including the joint path planning and joint time scheduling. An example of the joint scheduling plan of UAVs is shown in Fig.12. The joint path planning of UAVs is shown in Fig.12(a), UAV 2 takes off from the take-off point P_1, arrives at the communication point P_2 and then hovers to wait for UAV 1. UAV 1 takes off from the take-off point P_3, follows reverse flight direction to the communication point P_4, and delivers data to UAV 2. UAV 2 continues to move after receiving data from UAV 1 and receives data from UAV 3 when it reaches the communication point P_5. Then, UAV 2 delivers data to

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3339136

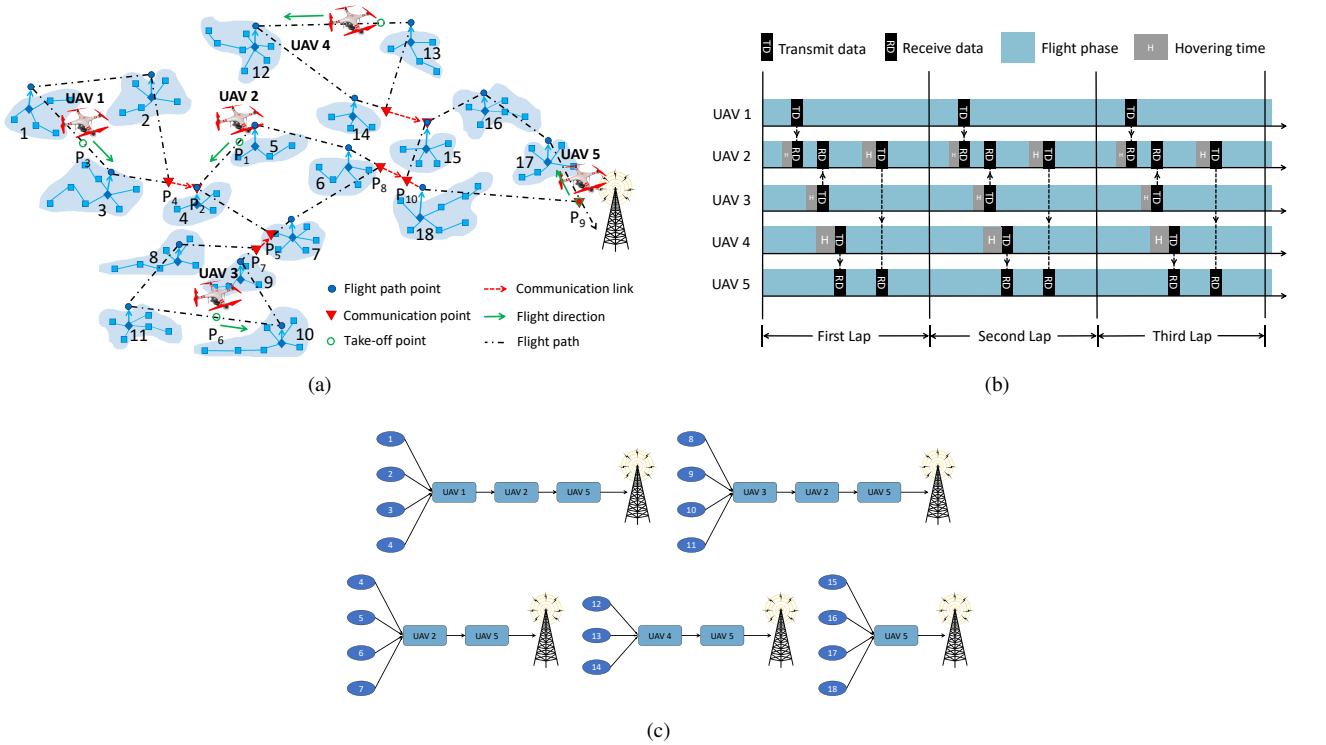


Fig. 12. The joint scheduling plan of UAVs. (a) Joint path planning of UAVs. (b) Time scheduling of UAVs. (c) Air-to-air data relay path for each subnet.

UAV 5 at the communication point P_8. Finally, UAV 5 sends the data to the BS when it returns to the communication point P_9. Fig.12(b) gives the time scheduling of UAVs. By adjusting the hovering time of each UAV, all UAVs take the same amount of time to fly one lap and complete the data relay as scheduled. After completing the joint path planning and joint time scheduling of UAVs, the air-to-air data relay path for each subnet can be confirmed, which is shown in Fig.12(c).

G. Complexity analysis

In the TSDA scheme, we first select data acquisition nodes and generate the initial flight path of UAVs by using the GA. For a sensor network divided into N_C subnets, the computational complexity of applying the GA with population size N_P for N_I iterations to solve this problem is $\mathcal{O}\left(N_P N_I N_A N_C\right)$. After that, the generated initial flight path is modified according to relay pairing relationship between UAVs. We also use the GA to solve this extreme-value problem. We still set the population size to N_P and the number of iterations to N_I. The computational complexity of the GA is $\mathcal{O}\left(N_P N_I N_A\right)$. Finally, the take-off point and flight direction of each UAV are determined by joint scheduling scheme of UAVs. The computational complexity of the proposed algorithm is $\mathcal{O}\left(N_A^2 \log\left(N_A\right)\right)$. According to the above analysis, the total computational complexity of the TSDA scheme is $\mathcal{O}\left(N_P N_I N_A N_C + N_P N_I N_A + N_A^2 \log\left(N_A\right)\right)$, which is equivalent to $\mathcal{O}\left(N_P N_I N_A N_C\right)$. The proposed TSDA scheme has a reasonable level of computational complexity, ensuring that it does not place an excessive burden on software and hardware resources. By leveraging offline calculations on the server at the BS, users have the capability to generate optimal flight paths for the UAVs and upload them in advance before commencing their data acquisition tasks. This facilitates efficient planning and execution of UAV tasks for data acquisition.

V. Experimental Results

A. Experimental setup

We executed a series of simulation by using Matlab 2021b running on a platform with 128G RAM, one Nvidia Quadro RTX 5000 and one Intel(R) Xeon(R) Gold 6230R CPU @2.10GHz with 26 cores. In the experiment, 300 SNs are randomly deployed within a 1000m×1000m region. The location of the BS is set to (1000, 500)m. The communication channel parameters are set according to the reference works in [57], [58]. We use GA to generate the initial flight paths and modified flight paths of UAVs. The recommended parameters to set the GA [51], are shown in Table II, while all simulation parameters are presented in Table III unless otherwise specified.

TABLE II
Parameter Settings of GA

Parameters	Value
Population size N_P	200
Number of iterations N_I	250
Crossover probability Q_c	1
Mutation probability Q_m	0.01

TABLE III
SIMULATION PARAMETERS

Parameter	Value
Channel parameter ψ, β	9.6, 0.28
Additional path loss under LoS η_{LoS}	1 dB
Additional path loss under NLoS η_{NLoS}	20 dB
Carrier frequency f_S	2 GHz
Threshold value $PL_{TH}^{AA}, PL_{TH}^{AG}$	34 dB, 69 dB
Max. communication distance D_{AA}, D_{AG}	50 m, 30 m
Flight altitude of UAVs H_C	25 m
Flight speed of UAV v	10 m/s
Initial energy E_U	2.2×10^5 J
Air density in ρ	1.225 kg/m^3
Tip speed U_{TIP}	120 m/s
Blade power P_{BLA}	99.66 W
Induced power P_{IND}	120.16 W
Body resistance ratio d_0	0.6
Rotor robustness s_0	0.05
Rotor disk area A	0.503 m^2
Average rotor speed at hovering v_0	4.03 m/s
Minimum safe separation distance D_S	10 m

B. Parameter analysis

Fig.13 shows the heatmap of the average AoI varying the number of UAVs and centripetal weight α. In the case where the number of deployed UAVs are relatively small ($N_A = 3$), the centripetal weight value has little effect on the average system AoI. As the number of UAVs rises, the effect of centripetal weight tends to be more evident. We can easily observe a recommended configuration parameter space. In this parameter space, the average system AoI tends to be optimum. Based on the above observations, the centripetal weight is set to 0.6 in the subsequent experiments.

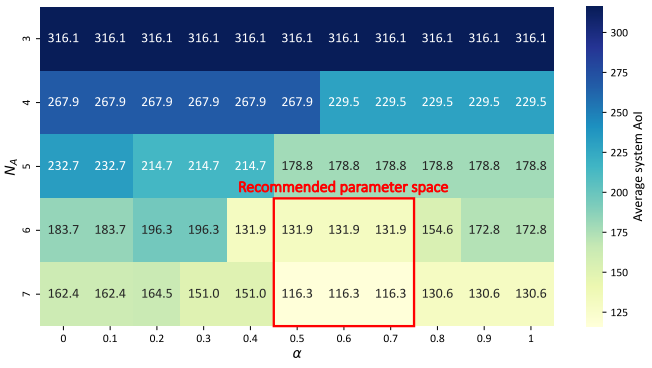
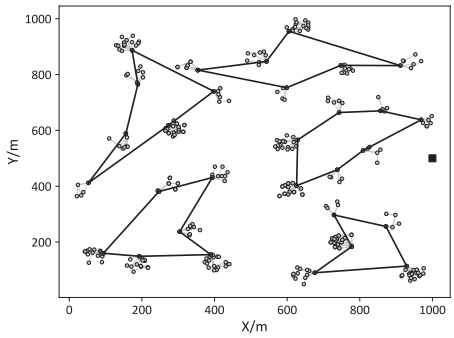
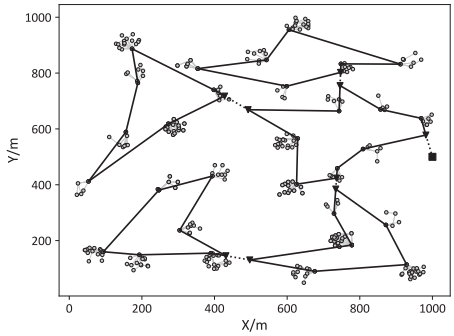


Fig. 13. The heatmap of the average system AoI varying number of UAVs and centripetal weight. ($N_C = 30$)

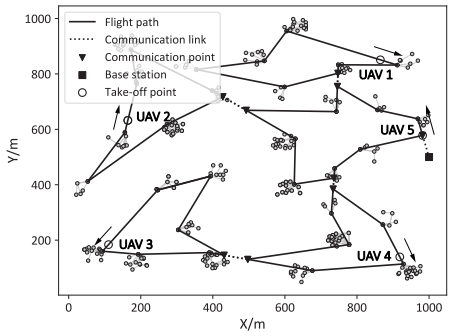
Fig.14 shows the generation process of UAV flight paths under the TSDA scheme. Fig.14(a) depicts the initial flight paths. The initial paths can traverse all subnets and the closed-loop flight path length of each UAV is approximated. Fig.14(b) shows the modified flight paths according to the centripetal-based relay pairing method. The modified path ensures that communication can be established between the paired UAVs and that the message delivery direction is directed to the BS. As shown in Fig.14(c), after determining the take-off points and flight directions of UAVs, the collaborative data acquisition model is further improved by the joint scheduling scheme.



(a)



(b)



(c)

Fig. 14. The generation process of UAV flight paths under TSDA scheme ($N_A = 5, N_C = 30$). (a) Initial flight path. (b) Modified path. (c) Final flight path

From Fig.15, we can clearly observe that for the relatively small network size ($N_C = 20$), the generated flight path length of UAVs is short, and a low average system AoI can be obtained. With the rise of N_C, although the path modification of UAVs can become more flexible, the flight path length required for UAVs to traverse all subnets increases significantly, resulting in a significant increase in the average system AoI.

Fig.16 illustrates the variation in the average system AoI as the number of UAVs changes. We can easily observe that to understand that along with the increasing number of UAVs, the

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3339136

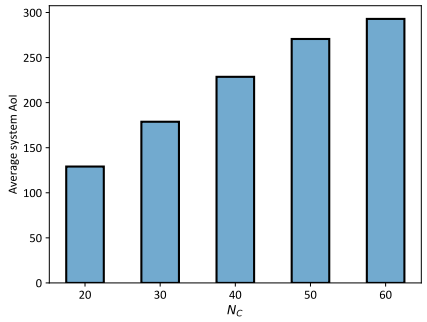


Fig. 15. The average system AoI varying number of subnets ($N_A = 5$).

data acquisition task assigned to each UAV is correspondingly reduced and is more conducive to data relay from far-end UAV.

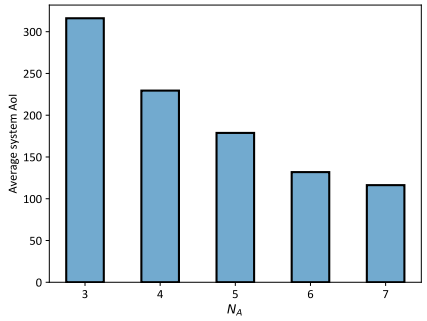


Fig. 16. The average system AoI varying number of UAVs ($N_C = 30$).

From Fig.17, we can easily observe that along with the increasing number of UAVs, the average system AoI decreases significantly in the case with joint scheduling. In contrast, in the absence of joint scheduling, the average system AoI remains at a high level as the number of UAVs increases.

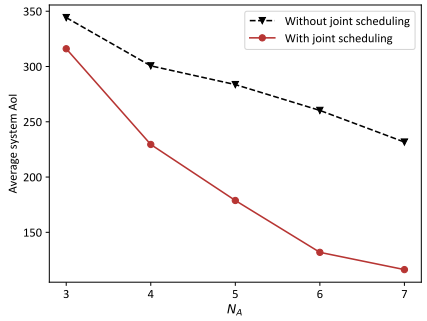
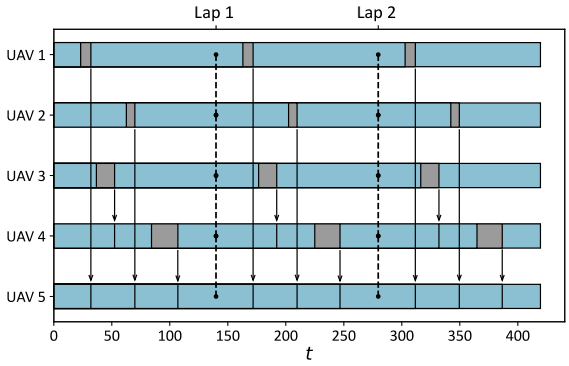
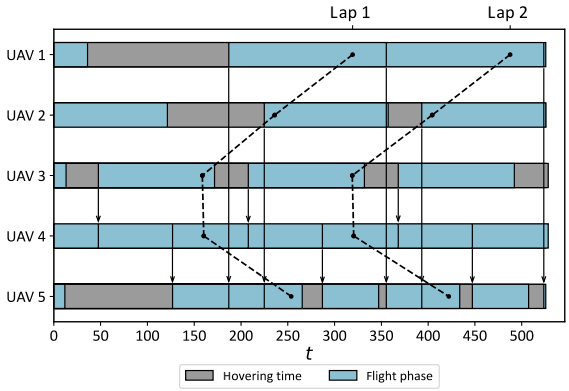


Fig. 17. Average system AoI with/without joint scheduling ($N_C = 30$).

Fig.18 presents the time states of each UAV in the two cases (i.e., with and without joint scheduling). We can find that the hovering time consumed by the UAVs during the air-to-air data relaying process is significantly reduced after using the joint scheduling scheme compared to that without it, thus being able to complete one round of data acquisition in less time. This result also reasonably explains the reason why using the joint scheduling scheme can significantly reduce the average system AoI.



(a)



(b)

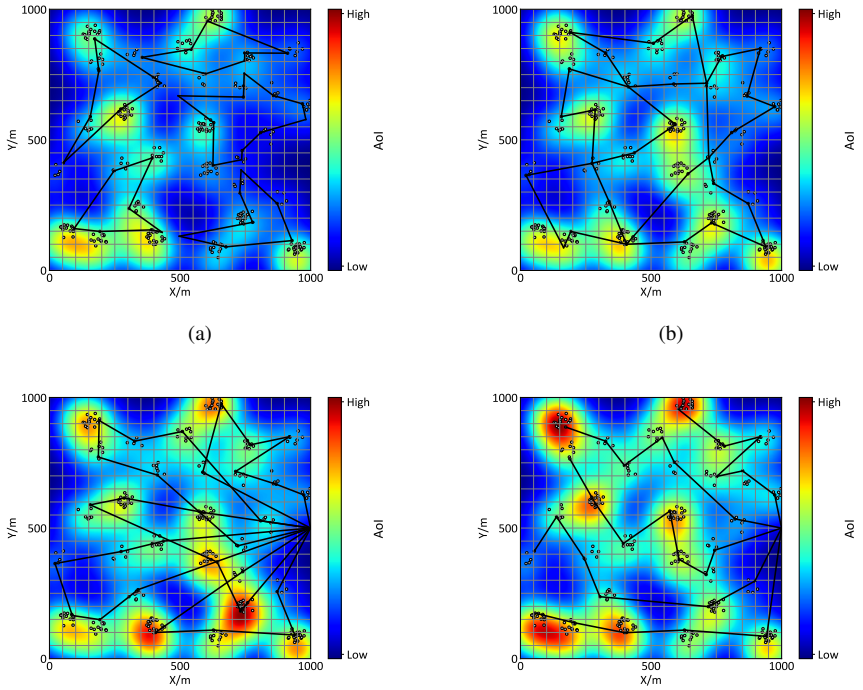
Fig. 18. Time states of each UAV with/without joint scheduling ($N_A = 5, N_C = 30$). (a) With joint scheduling scheme. (b) Without joint scheduling scheme.

C. Performance comparison

To evaluate the effectiveness of the TSDA scheme, we selected the multi-closed-loop path data acquisition scheme (MCDA) [27], the multi-trip path data acquisition scheme (MTDA) [30], and the AoI-aware data acquisition scheme (ADC) [36] as the comparison algorithms. MCDA and MTDA are the most classic UAV-aided IoT data acquisition schemes; in particular, the MCDA scheme uses a one-way flight closed-loop path and the MTDA scheme uses a round-trip flight fold path. These two schemes focus on optimizing the flight path of each UAV to improve data acquisition efficiency. The ADC scheme is the UAV-aided IoT collaborative data acquisition scheme proposed in our previous research work [36]. The ADC scheme achieves data relay among UAVs by constructing Hamiltonian paths.

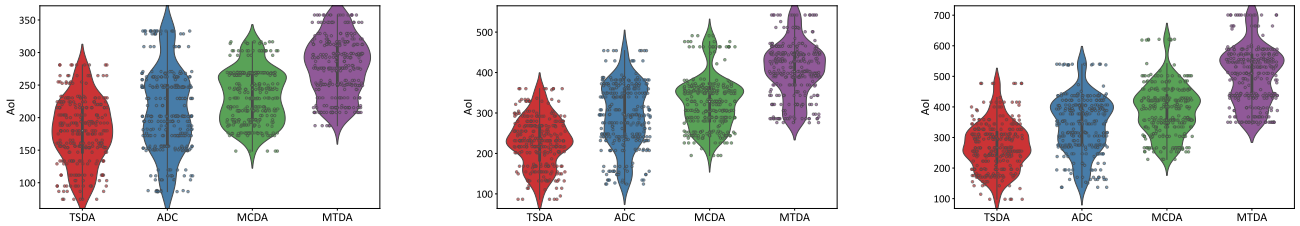
Fig.19 presents the AoI heatmap in the four data acquisition schemes. The color of the region in the figure represents the AoI value of the region. In the TSDA and ADC scheme, the far-end UAV does not required to fly to the BS through

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3339136



(a) (b) (c) (d)

Fig. 19. The AoI heatmap generated by four data acquisition schemes ($N_A = 5, N_C = 30$). (a) TSDA scheme. (b) ADC scheme (c) MCDA scheme. (d) MTDA scheme



(a) (b) (c)

Fig. 20. The AoI distribution of SNs generated by four data acquisition schemes ($N_A = 5$). (a) $N_C = 30$. (c) $N_C = 40$ (c) $N_C = 50$.

the air-to-air data relay. However, in the ADC scheme, a UAV is only allowed to establish a relay pairing with one UAV (i.e., one-to-one data relay). In the TSDA scheme, one UAV is allowed to establish pairing relationships with multiple UAVs, which makes the UAV more flexible and efficient in air-to-air data relaying. In both MCDA and MTDA schemes, UAVs are required to move to marginal regions to finalize data acquisition after accomplishing the task in the nearby region of the BS. Unlike the TSDA scheme, these two schemes involve UAV flight paths covering a broader area, resulting in increased data delivery latency and significantly higher AoI values for SNs.

In order to further explore data acquisition performance of the four data acquisition schemes in terms of AoI, we conducted statistics on the AoI distribution of the SNs in the sensor network. By observing Fig.20, we can find that the AoI distribution of SNs under the TSDA scheme exhibits significant improvements compared to the other three schemes. Taking the scenario with $N_C = 30$ as an example, the AoI values of SNs under TSDA are mostly distributed in the range of [165, 230]. In contrast, the AoI values of SNs under ADC, MCDA and MTDA are mainly distributed in the ranges of [159, 272], [170, 270] and [248, 325], respectively.

As shown in Fig.21, compared with ADC, MCDA and MTDA, the average system AoI obtained from the TSDA scheme is significantly better than that of these three schemes. Specifically, it outperforms these three schemes by 14%, 23% and 35% respectively. The performance advantage of the TSDA scheme is more prominent with the expansion of the network size. In the case where the network size is relatively small ($N_C = 30$), the UAVs do not need to perform air-to-air data relay to achieve efficient data acquisition. In this case, the performance difference between the four schemes is not evident. With the expansion of the network size, the flight paths required for UAVs to traverse each data acquisition node grows significantly. In this scenario, the collaborative relay

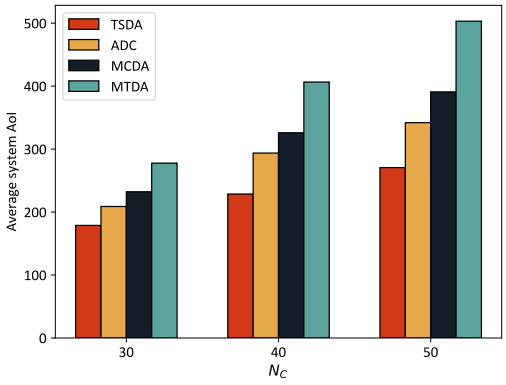


Fig. 21. Comparison of four data acquisition schemes in terms of average system AoI ($N_A = 5$).

advantages of both TSDA and ADC schemes can be fully utilized. Compared to the ADC scheme that only supports one-to-one data relay, the relay pairing strategy of the TSDA scheme enables UAVs to participate in data forwarding tasks more flexibly, thereby further reducing the average system AoI.

VI. Conclusion

In this work, we have developed a collaborative data acquisition model that utilizes collaborative data relay among UAVs to reduce data delivery latency. Building upon the proposed model, we introduce a time-balancing scheduling data acquisition scheme (TSDA) as a solution. Extensive experimental results have shown that the proposed TSDA scheme can significantly reduce the average AoI of the system compared with existing data acquisition schemes. In addition, the advantages of collaborative relaying of the TSDA scheme can be better exploited as the network size increases; moreover, the hovering time of UAVs can be significantly reduced after applying such joint scheduling scheme. We mainly focus on the air-to-air relaying problem of UAVs in UAV-aided IoT systems in obstacle-free scenarios. In the next step of our work, we focus on the multi-UAV data relaying problem in obstacle environment. How to ensure safe obstacle avoidance of UAVs while achieving efficient data relaying was the major challenge.

VII. Acknowledgement

This work is supported by the Science and Technology Commission of Shanghai Municipality under Grant No.22692194600, Shanghai Natural Science Foundation under Grant No.23ZR1426400, and the Italian MIUR, PRIN 2017 Project "Fluidware: a Novel Approach for Large-Scale IoT Systems" (ID: 2017KRC7KT, CUP H24I17000070001).

References

[1] C. Savaglio and G. Fortino, "A simulation-driven methodology for IoT data mining based on edge computing," *ACM Transactions on Internet Technology (TOIT)*, vol. 21, no. 2, pp. 1–22, 2021.

[2] G. Fortino, C. Savaglio, G. Spezzano, and M. Zhou, "Internet of things as system of systems: A review of methodologies, frameworks, platforms, and tools," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 223–236, 2020.

[3] G. Fortino, W. Russo, C. Savaglio, W. Shen, and M. Zhou, "Agent-oriented cooperative smart objects: From IoT system design to implementation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 11, pp. 1939–1956, 2017.

[4] H. Ko and S. Pack, "Distributed device-to-device offloading system: Design and performance optimization," *IEEE Transactions on Mobile Computing*, vol. 20, no. 10, pp. 2949–2960, 2020.

[5] H. Ko, S. Pack, and V. C. Leung, "Performance optimization of serverless computing for latency-guaranteed and energy-efficient task offloading in energy harvesting industrial IoT," *IEEE Internet of Things Journal*, 2021.

[6] G. K. Verma, B. Singh, N. Kumar, M. S. Obaidat, D. He, and H. Singh, "An efficient and provable certificate-based proxy signature scheme for IIoT environment," *Information sciences*, vol. 518, pp. 142–156, 2020.

[7] G. Xiong, F.-Y. Wang, T. R. Nyberg, X. Shang, M. Zhou, Z. Shen, S. Li, and C. Guo, "From mind to products: Towards social manufacturing and service," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 47–57, 2017.

[8] A. Hazra, P. K. Donta, T. Amgoth, and S. Dustdar, "Cooperative transmission scheduling and computation offloading with collaboration of fog and cloud for industrial IoT applications," *IEEE Internet of Things Journal*, 2022.

[9] G. Aloi, G. Fortino, R. Gravina, P. Pace, and C. Savaglio, "Simulation-driven platform for edge-based AAL systems," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 446–462, 2020.

[10] G. Horn, F. Eliassen, A. Taherkordi, S. Venticinque, B. Di Martino, M. Bücher, and L. Wood, "An architecture for using commodity devices and smart phones in health systems," in *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 2016, pp. 255–260.

[11] J. Zhang, Y. Wang, S. Li, and S. Shi, "An architecture for IoT-enabled smart transportation security system: a geospatial approach," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6205–6213, 2020.

[12] S. Han, K. Zhu, M. Zhou, and X. Liu, "Joint deployment optimization and flight trajectory planning for UAV assisted IoT data collection: A bilevel optimization approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21 492–21 504, 2022.

[13] N. Agrawal, A. Bansal, K. Singh, C.-P. Li, and S. Mumtaz, "Finite block length analysis of RIS-assisted UAV-based multiuser IoT communication system with non-linear EH," *IEEE Transactions on Communications*, vol. 70, no. 5, pp. 3542–3557, 2022.

[14] Z. Wei, M. Zhu, N. Zhang, L. Wang, Y. Zou, Z. Meng, H. Wu, and Z. Feng, "UAV-assisted data collection for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15 460–15 483, 2022.

[15] S. F. Abedin, M. S. Munir, N. H. Tran, Z. Han, and C. S. Hong, "Data freshness and energy-efficient UAV navigation optimization: A deep reinforcement learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5994–6006, 2020.

[16] H. Hu, K. Xiong, G. Qu, Q. Ni, P. Fan, and K. B. Letaief, "AoI-minimal trajectory planning and data collection in UAV-assisted wireless powered IoT networks," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1211–1223, 2020.

[17] M. A. Abd-Elmagid and H. S. Dhillon, "Average peak age-of-information minimization in UAV-assisted IoT networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 2003–2008, 2018.

[18] P. Tong, J. Liu, X. Wang, B. Bai, and H. Dai, "UAV-enabled age-optimal data collection in wireless sensor networks," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2019, pp. 1–6.

[19] J. Liu, X. Wang, B. Bai, and H. Dai, "Age-optimal trajectory planning for UAV-assisted data collection," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018, pp. 553–558.

[20] M. A. Abd-Elmagid, A. Ferdowsi, H. S. Dhillon, and W. Saad, "Deep reinforcement learning for minimizing age-of-information in UAV-assisted networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[21] G. Ahani, D. Yuan, and Y. Zhao, "Age-optimal UAV scheduling for data collection with battery recharging," *IEEE Communications Letters*, vol. 25, no. 4, pp. 1254–1258, 2020.

[22] M. Yi, X. Wang, J. Liu, Y. Zhang, and B. Bai, "Deep reinforcement learning for fresh data collection in UAV-assisted iot networks," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 716–721.

[23] P. Tong, J. Liu, X. Wang, B. Bai, and H. Dai, "Deep reinforcement learning for efficient data collection in UAV-aided internet of things," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2020, pp. 1–6.

[24] J. Liu, P. Tong, X. Wang, B. Bai, and H. Dai, "UAV-aided data collection for information freshness in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2368–2382, 2020.

[25] Z. Jia, X. Qin, Z. Wang, and B. Liu, "Age-based path planning and data acquisition in UAV-assisted IoT networks," in *2019 IEEE international conference on communications workshops (ICC Workshops)*. IEEE, 2019, pp. 1–6.

[26] A. Ferdowsi, M. A. Abd-Elmagid, W. Saad, and H. S. Dhillon, "Neural combinatorial deep reinforcement learning for age-optimal joint trajectory and scheduling design in UAV-assisted networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1250–1265, 2021.

[27] Y. Wang, Z. Hu, X. Wen, Z. Lu, and J. Miao, "Minimizing data collection time with collaborative UAVs in wireless sensor networks," *IEEE Access*, vol. 8, pp. 98 659–98 669, 2020.

[28] Z. Li, P. Tong, J. Liu, X. Wang, L. Xie, and H. Dai, "Learning-based data gathering for information freshness in UAV-assisted IoT networks," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2557–2573, 2023.

[29] L. Deng, H. Jiang, H. Xiao, Q. Zhang, Y. Luo, C. Wu, and C. Ye, "Completion time minimization for multi-antenna UAV-enabled data collection in uncorrelated rician fading," *Vehicular Communications*, vol. 37, p. 100501, 2022.

[30] C. Xu, M. Xu, and C. Yin, "Optimized multi-UAV cooperative path planning under the complex confrontation environment," *Computer Communications*, vol. 162, pp. 196–203, 2020.

[31] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and Z. Gao, "UAV trajectory planning for AoI-minimal data collection in UAV-aided IoT networks by transformer," *IEEE Transactions on Wireless Communications*, 2022.

[32] O. S. Oubbati, M. Atiquzzaman, H. Lim, A. Rachedi, and A. Lakas, "Synchronizing UAV teams for timely data collection and energy transfer by deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 6682–6697, 2022.

[33] Y. Liao and V. Friderikos, "Energy and age pareto optimal trajectories in UAV-assisted wireless data collection," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 8, pp. 9101–9106, 2022.

[34] L. Liu, K. Xiong, J. Cao, Y. Lu, P. Fan, and K. B. Letaief, "Average AoI minimization in UAV-assisted data collection with RF wireless power transfer: A deep reinforcement learning scheme," *IEEE Internet of Things Journal*, vol. 9, no. 7, pp. 5216–5228, 2021.

[35] T. Wu, J. Liu, J. Liu, Z. Huang, H. Wu, C. Zhang, B. Bai, and G. Zhang, "A novel AI-based framework for AoI-optimal trajectory planning in UAV-assisted wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 4, pp. 2462–2475, 2021.

[36] X. Huang and X. Fu, "Fresh data collection for UAV-assisted IoTs based on aerial collaborative relay," *IEEE Sensors Journal*, 2023.

[37] D. Yin, X. Yang, H. Yu, S. Chen, and C. Wang, "An air-to-ground relay communication planning method for UAVs swarm applications," *IEEE Transactions on Intelligent Vehicles*, 2023.

[38] R. Han, J. Wang, L. Bai, J. Liu, and J. Choi, "Age of information and performance analysis for UAV-aided IoT systems," *IEEE Internet of Things Journal*, vol. 8, no. 19, pp. 14 447–14 457, 2021.

[39] M. Sun, X. Xu, X. Qin, and P. Zhang, "AoI-energy-aware UAV-assisted data collection for IoT networks: A deep reinforcement learning method," *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17 275–17 289, 2021.

[40] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 2731–2735.

[41] L. A. binti Burhanuddin, X. Liu, Y. Deng, U. Challita, and A. Zahemszky, "QoE optimization for live video streaming in UAV-to-UAV communications via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 5358–5370, 2022.

[42] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[43] H. Nawaz, H. M. Ali, and A. A. Laghari, "UAV communication networks issues: a review," *Archives of Computational Methods in Engineering*, vol. 28, no. 3, pp. 1349–1369, 2021.

[44] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular UAV-to-X communications: Design and optimization for multi-UAV networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 2, pp. 1346–1359, 2019.

[45] X. Liu, X. Wang, M. Huang, J. Jia, N. Bartolini, Q. Li, and D. Zhao, "Deployment of UAV-BSs for on-demand full communication coverage," *Ad Hoc Networks*, vol. 140, p. 103047, 2023.

[46] D. Yang, Q. Wu, Y. Zeng, and R. Zhang, "Energy tradeoff in ground-to-UAV communication via trajectory design," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6721–6726, 2018.

[47] S. Ahmed, M. Z. Chowdhury, and Y. M. Jang, "Energy-efficient UAV relaying communications to serve ground nodes," *IEEE Communications Letters*, vol. 24, no. 4, pp. 849–852, 2020.

[48] J. Bi, H. Yuan, S. Duanmu, M. Zhou, and A. Abusorrah, "Energy-optimized partial computation offloading in mobile-edge computing with genetic simulated-annealing-based particle swarm optimization," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3774–3785, 2020.

[49] J. Bi, H. Yuan, J. Zhai, M. Zhou, and H. V. Poor, "Self-adaptive bat algorithm with genetic operations," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 7, pp. 1284–1294, 2022.

[50] J. Bi, H. Yuan, K. Zhang, and M. Zhou, "Energy-minimized partial computation offloading for delay-sensitive applications in heterogeneous edge networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 10, no. 4, pp. 1941–1954, 2022.

[51] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 2, pp. 601–614, 2018.

[52] P. Zhang, S. Shu, and M. Zhou, "An online fault detection method based on SVM-grid for cloud computing systems," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 2, pp. 445–456, 2018.

[53] X. Luo, Y. Zhou, Z. Liu, L. Hu, and M. Zhou, "Generalized nesterov's acceleration-incorporated, non-negative and adaptive latent factor analysis," *IEEE Transactions on Services Computing*, vol. 15, no. 5, pp. 2809–2823, 2021.

[54] H. Zhou, M. Song, and W. Pedrycz, "A comparative study of improved GA and PSO in solving multiple traveling salesmen problem," *Applied Soft Computing*, vol. 64, pp. 564–580, 2018.

[55] Z. Wang, X. Fang, H. Li, and H. Jin, "An improved partheno-genetic algorithm with reproduction mechanism for the multiple traveling salesperson problem," *IEEE Access*, vol. 8, pp. 102 607–102 615, 2020.

[56] Q. Pan and X. Fu, "Fresh data collection for UAV-assisted IoTs based on proximity-remote region collaboration," *Ad Hoc Networks*, vol. 146, p. 103182, 2023.

[57] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *2014 IEEE global communications conference*, 2014, pp. 2898–2904.

[58] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: Design, deployment and performance analysis," in *2015 IEEE global communications conference (GLOBECOM)*. IEEE, 2015, pp. 1–6.