ASSUME: An Optimal Algorithm to Minimize UAV Energy by Altitude and Speed Scheduling

Jianping Huang, Feng Shan, *Member, IEEE*, Junzhou Luo, *Member, IEEE*, Runqun Xiong, *Member, IEEE*, and Wenjia Wu, *Member, IEEE*

***Abstract*—Uncrewed aerial vehicles (UAVs) are being widely employed in wireless communication applications, e.g., collecting data from ground nodes (GNs). Minimizing UAV energy in these applications is crucial due to the limited energy supply onboard. Unlike previous studies that assume UAVs fly at a fixed altitude and simplify the energy consumption model of UAVs, we consider the impact of varying UAV altitudes on the ground-to-air communication and utilize a general communication model for GN. Furthermore, we conduct real-world flight tests and introduce a practical speed-related flight energy consumption model of UAVs. This paper focuses on the UAV altitude-speed scheduling and GN transmission switching (UASS-GTS) problem, specifically in scenarios where the UAV flies straight for monitoring applications such as power transmission lines, roads, and water/oil/gas pipes. However, minimizing energy consumption presents challenges due to the tight coupling of altitude scheduling and speed scheduling. To tackle this, first, we develop the *looking before crossing* algorithm for speed scheduling. We then extend this algorithm by integrating altitude scheduling to propose the Altitude-Speed Scheduling of UAV for Minimizing Energy (ASSUME) algorithm, using a dynamic programming method. The ASSUME algorithm is theoretically proven to be optimal. Additionally, based on ASSUME, we propose an offline-inspired online heuristic algorithm to handle agnostic situations where GN information is not available unless flies close. Simulations indicate that the ASSUME algorithm saves an average of 26.1%–62.7% energy compared to the baseline methods, and the performance gap between the online algorithm and the offline optimal algorithm ASSUME is 22.8%.**

***Index Terms*—Uncrewed aerial vehicle, energy efficiency, speed scheduling, altitude scheduling, flight energy minimization.**

I. Introduction

UNCREWED aerial vehicles (UAVs) are being widely employed in wireless communication applications [1], [2], [3], [4], [5], [6], including wireless base stations [1], wireless relays [2], and in edge computing [3] and data collection [4], [6]. Wireless sensors and Internet of Things (IoT) devices are extensively deployed for a wide range of monitoring purposes. Data collection from such ground nodes (GNs) using UAVs is one of the most important applications, as UAVs can fly close to establish line-of-sight(LoS) energy-efficient data communications. Considerable research efforts have already been dedicated to this area [4], [6], [7], [8], [9], [10], [11], [12].

The energy consumption of UAVs is a critical issue due to the limited energy supply available on board, as a result, minimizing UAV energy is of great importance. Generally speaking, a UAV employed in data collection applications involves both horizontal and vertical flight, which indicates the energy consumption consist of two components: the energy required to change flight altitude and the energy needed to cover distance. However, most existing work consider the simplified models: 1) assuming that UAVs fly at a fixed altitude [6], [13], which overlooks the effects of varying altitudes on the performance of ground-to-air communication, and thus loses the optimization potential, and 2) utilizing either a distance-related UAV energy model [7], [14], [15], where the energy consumption is proportional to the distance that the UAV covers, or a duration-related UAV energy mode [5], [9], [16], [17], [18], [19], [20], where the energy consumption is proportional to the duration of flight. Both models fail to accurately reflect the actual energy consumption, limiting their applicability for in-depth investigations.

Distinct from the previous work, first, this paper considers how varying altitudes of UAV leads to different efficiency of communication during data collection. Inspired by the work in [21], we adopt a more general communication model for GN as illustrated in Fig. 1(b). The range of ground-to-air communication varies following the flight altitude of the UAV. Although a higher altitude increases the probability of LoS, it may simultaneously reduce the communication coverage area of the UAV, which influences the level of energy minimization achieved. This observation motivates us to delicately schedule the UAV flight altitude for energy optimization. Furthermore, we conduct a set of real-world flight tests using a multi-rotor UAV, and a practical speed-related energy consumption model is disclosed as displayed in Fig. 1(a). In this model, we observe that the UAV has the lowest power consumption at a specific speed (v^*, as depicted in the figure), while consuming more power at both higher and lower speeds. Further details regarding this model will be discussed in the following section. Such a model has been verified by a theoretical analysis work on the energy model of rotary-wing UAVs [22]. As a result, we conclude that our speed-related energy model is more practical

Received 30 June 2024; revised 18 March 2025; accepted 10 June 2025. Date of publication 23 June 2025; date of current version 3 October 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62232004, Grant 62472090, Grant 62132009, and Grant 62172091, in part by the National Natural Science Foundation of Jiangsu under Grant BK20242026, and in part by the Jiangsu Provincial Key Laboratory of Network and Information Security under Grant BM2003201. Recommended for acceptance by C. Peng. *(Corresponding author: Feng Shan.)*

The authors are with the Department of Computer Science and Engineering, Southeast University, Nanjing 211102, China (e-mail: jphuang@seu.edu.cn; shanfeng@seu.edu.cn; jluo@seu.edu.cn; rxiong@seu.edu.cn; wjwu@seu.edu.cn).

Digital Object Identifier 10.1109/TMC.2025.3581929

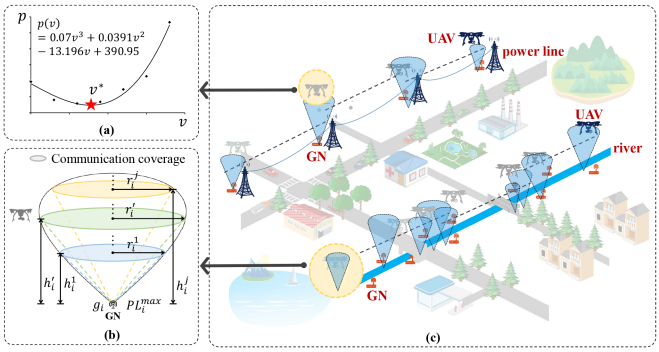


Fig. 1. The application scenarios are illustrated in (c), where a UAV is dispatched to collect data from a set of wireless sensors or IoT devices (GNs) deployed along a straight line, such as a power transmission line, a road, a water/oil/gas pipe or a river/coast. Note that we consider a practical speed-related energy model of UAV, as depicted in (a), and more general communication model of GN, as depicted in (b).

than most existing energy consumption models used for wireless communication. Based on this promising speed-related energy model, we identify an opportunity to optimize energy power through speed scheduling.

In this paper, our focus lies on the energy-efficient UAV altitude-speed scheduling and GN transmission switching (UASS-GTS) problem. Specifically, we consider scenarios where GNs are deployed along a straight line to serve various applications, including monitoring power transmission lines [23], roads [24], water/oil/gas pipes [25] or rivers/coasts [26]. Fig. 1(c) is presented as an illustration of the application scenarios. Note that we consider only the flight energy consumption and do not take into account energy consumption for wireless data transmission. This is because, according to our real-world flight tests, the power for UAVs to hover is around $400W$, while a typical LoRa/Wi-Fi wireless communication module's power consumption is 100 mW [27]/300 mW [28], which implies that the energy consumption for wireless transmission is negligibly small in comparison to the energy consumed during flight, allowing us to focus on tracing the fundamental nature of altitude and speed scheduling.

Despite incorporating both the practical communication model of GN and the speed-related energy model of UAV, the fundamental nature of altitude and speed scheduling remains evident in the following challenges posed to our problem.

1) Minimizing the energy consumption is challenging due to the tight coupling between UAV altitude scheduling and speed scheduling. On one hand, optimizing speed scheduling may frequently adjust altitude in order to achieve an optimum transmission range for GN [21], which in turn increases energy consumption of changing altitude [8]. On the other hand, optimizing altitude scheduling requires minimizing altitude adjustments or maintaining a fixed altitude, which may result in a suboptimal transmission range, thereby posing a disadvantage for UAV speed scheduling.
2) Optimizing data collection with minimum energy costs during UAV speed scheduling presents a further challenge. On one hand, flying at a slower speed allows the UAV sufficient time to collect all data. However, this slower speed consumes more energy, as indicated by our practical energy model. On the other hand, a faster speed flight may reduce the flight energy consumption, but risks incomplete data collection due to insufficient time within the transmission range.
3) The competition among these heterogeneous GNs complicates the data collection scenarios. The communication ranges of GNs overlap, and the UAV can only collect data from one GN at a time. As a result, each GN competes for UAV's duration to transmit its data. Moreover, the varying amounts of data to be transmitted and the different communication models of GNs add to the challenge of balancing this competition.

As the challenges stated above, it is hard to find a straightforward solution to the UASS-GTS problem. Despite the existence of high-quality work on energy-efficient data collection, none of them can be applied directly to address this problem. The most relevant work on speed scheduling is by Zeng et al. [22], but their formalized problem is difficult to solve because both the speed scheduling and the UAV flight trajectory need to be determined. As a result, they obtain a heuristic solution with uncertain deviations from the optimal solution. In this paper, first, we develop an algorithm named *looking before crossing* to tackle a basic case of the UASS-GTS problem, specifically focusing on scheduling the UAV's speed. Within this algorithm, we construct *virtual rooms* on a time-distance diagram to represent the spatiotemporal constraints of GNs in transmission ranges and transmission time. A trajectory crossing these virtual rooms can be uniquely mapped to a solution. By extending this basic solution, we propose the **A**ltitude-**S**peed **S**cheduling of **U**AV for **M**inimizing **E**nergy (**ASSUME**) algorithm to solve the general UASS-GTS problem, integrating joint altitude scheduling through a dynamic programming method. This approach is theoretically proven to be optimal. Additionally, based on ASSUME, we propose an offline-inspired online heuristic algorithm to handle agnostic situations where GN information is not available unless flies close. The contributions of this paper are summarized as follows.

- We derive a practical speed-related energy consumption model from our real-world flight tests. We find that there exists an optimal speed at which the power consumption of UAV is minimized, with higher and lower speeds resulting in increased power consumption. This model is distinct from most existing studies that typically assume either the distance-related or duration-related energy model.
- We formulate a novel energy minimization problem in UAV-aided data collection scenarios, named the UAV altitude-speed scheduling and GN transmission switching (UASS-GTS) problem. This problem is the first to incorporate the two practical models: the communication model of GN and the speed-related energy model of UAV.
- We develop the *looking before crossing* algorithm to solve a basic case, focusing on UAV speed scheduling. Subsequently, we extend this basic solution to propose the ASSUME algorithm for the general UASS-GTS problem, integrating joint altitude scheduling through a dynamic

TABLE I
COMPARATIVE ANALYSIS OF RELATED WORK AND OUR APPROACH

Work \ Feature	Altitude Optimization	Speed Scheduling	Optimality
ours	√	√	√
[2]–[4], [22], [29]–[31]	×	√	×
[6], [10], [13], [32]–[36]	×	×	×
[5], [37]–[39]	√	√	×
[40]–[45]	√	×	×

programming method. Additionally, we present an offline-inspired online heuristic algorithm to address the online problem where GN information is unavailable until the UAV flies close.

- We conduct extensive simulations to evaluate the performance of the proposed algorithms. The results show that the ASSUME algorithm can save an average of 26.1%–62.7% energy compared to the baseline methods, and the performance gap between the online algorithm and the ASSUME is 22.8%.

The rest of this paper is organized as follows. Section II investigates the related work. Section III presents the system model and formulates the problem. The basic case of this problem and its solution are detailed in Section IV. Following this, the ASSUME algorithm and its online algorithm are introduced in Sections V and VI, respectively. Extensive simulations are conducted in Section VII. Finally, Section VIII concludes the paper.

II. RELATED WORK

In this section, we examine closely related works concerning energy-efficient algorithms for UAV-aided data collection, UAV altitude optimization, and UAV flight energy model. To summarize these existing studies, we present a comparative table in Table I.

A. Energy-Efficient Algorithms for UAV-Aided Data Collection

Energy consumption is a critical issue in UAV-aided data collection applications, thus many researchers have focused on the energy-efficient algorithms. (1) A classical approach to optimizing energy consumption is mathematical programming. In these studies [2], [3], [4], [29], researchers typically start by theoretically formulating an optimization problem that is mathematically tractable and then employ optimization theories such as the successive convex approximation technique or quadratic optimization to find feasible solutions. However, mathematical programming often involves complex models and formulations. (2) Therefore, some researches [9], [30], [32], [37], [46], [47] propose energy-efficient algorithms based on heuristic. For example, Jia et al. [47] introduce a multi-strategy multi-objective optimization algorithm to solve the problem of minimizing the maximum task completion time and the maximum energy consumption among all UAVs. In [46] and [32], the authors design Ant Colony Optimization (ACO) based framework to solve the energy cost and completion time minimization problems in UAV-Assisted task offloading systems, satisfying the energy, deadline, location, and priority constraints. While these hybrid intelligent algorithms provide efficient solutions, their performance is not guaranteed to be optimal theoretically. (3) More recently, the emergence of policy optimization and learning-based techniques has greatly facilitated energy optimization [10], [33], [34], [35]. Zhu et al. [10], combining UAVs with LoRa in large-scale data collection, formulate a joint of energy consumption and data acquisition efficiency optimization problem, then a Deep Reinforcement Learning (DRL) based two-stage scheme is proposed to explore the optimal trajectory based on an attention-based encoder-decoder model to generate an initial trajectory. Mondal et al. [33] consider the energy-efficiency optimization of data collection system, which is reduced into an MDP model, then a Deep Deterministic Policy Gradient (DDPG) algorithm is used to learn the UAV's trajectory. Similarly, Ning et al. [34] construct an MDP model and proposed a Multi-agent Constrained DRL algorithm to learn the optimal UAV movement policy in data acquisition and trajectory planning. Although these learning-based methods performance is promising, most of them cannot guarantee global optimality, and their decision-making process lacks interpretability. More importantly, they require substantial computational resources and data for training and inference in unseen scenarios, which may introduce latency in online deployment.

In contrast to these previous techniques, in this paper, we propose an optimal algorithm for energy minimization by skillfully constructing spatial-temporal virtual rooms and flight altitude scheduling. Our algorithm not only theoretically guarantees global optimality but also features good generalization ability, low computational complexity, high interpretability, and real-time performance, without relying on complex models, intensive computational resources, or massive training data.

B. UAV Altitude Optimization

In this subsection, we investigate researches on UAV flight altitude scheduling. Some existing work assume that UAVs fly at a fixed altitude to carry out tasks [6], [13], [35], [36], which simplifies the analysis but limits practical applicability. Therefore, considering the impact of flight altitude on the communication coverage range and energy efficiency, many recent studies utilize a more practical channel communication model between UAV and GN with dynamic altitude adjustment. Confirmed by the cell radius model derived in [21], an optimum altitude of UAV can be found to optimize UAV placement issues [5], [39], [40], [41], [42], [43], [44], [45]. In the context of disaster localization, the

authors [40], [41] study the impact of UAV altitude on localization error. Albanese et al. [41] present a drone-based search and rescue solution where an optimal altitude of UAV is obtained as a trade-off between the antenna radiation pattern effect at low altitudes and the stronger path loss at high altitudes. Similarly, Ebrahimi et al. [40] suggest that higher altitude increases the probability of LoS and thus better localization accuracy but decreases the coverage area of the UAV, leading to fewer objects being localized. In a multi-antenna UAV-aided NOMA multi-user pairing system, Hoang et al. [43] also propose an algorithm to determine the optimum altitude of UAV that maximizes the throughput of system, given the number of transmission bits. To provide better coverage for high-density distribution of flash crowds, Lai et al. [42] exploit multiple UAVs placed at different altitudes, and propose data-driven 3D placement algorithms to effectively find the appropriate altitude, as well as the number and location of UAVs, to maximize the system sum rate. In data collection applications, Ko et al. [5] utilize the Newton iteration method and an aggressive method, respectively, to compute the optimum flight altitude of UAVs, tailored to varying mission requirements.

The aforementioned studies dynamically adjust UAV altitude to achieve better optimization. Hence, building on the communication model introduced in [21], our problem also incorporates UAV flight altitude scheduling into the energy optimization problem.

C. UAV Flight Energy Model

Generally speaking, the UAV flight energy model has a significant impact on the effectiveness of the UAV energy consumption algorithm. Currently, UAV flight energy models can be broadly classified into three categories: (1) Distance-related model [7], [15], [48]. Piao et al. [7] investigate an indoor CSI measurement problem using a UAV, and the UAV energy cost depends on the distance covered and the number of turns made. Additionally, Xiang et al. [15] implement a UAV delivery sensing prototype system to develop an energy consumption model that considers both the flight distance and the capacity of UAVs. (2) Duration-related model [5], [9], [19], [20]. Zhu et al. [19] adopt the duration-related energy model to minimize the energy consumption during data collection in large-scale wireless sensor networks. Zhang et al. [20] assume a constant cruising speed in a UAV-assisted IoT network and calculate energy consumption based on the accumulated duration time. In the context of large-scale IoT systems, Ma et al. [9] also leverage the duration-relate model to devise a novel data collection scheme that can balance energy minimization and data rate maximization. Ko et al. [5] design UAV trajectories for location-dependent visual coverage and estimate total energy consumption based on the basic UAV power consumption multiplied by the flying time. (3) Speed-related model [22], [31], [38], [49], [50]. Some researches indicate that both models introduced above simplify UAV energy consumption model, with the speed-related model being more practical. Morbidi et al. [49] obtain a speed-related model by leveraging the electrical model of a brushless direct current motor to design energy-efficient paths for UAVs. Ding et al. [50] focus on the UAV three-dimensional flight and also formulate a speed-related mathematical model for the propulsion energy consumption.

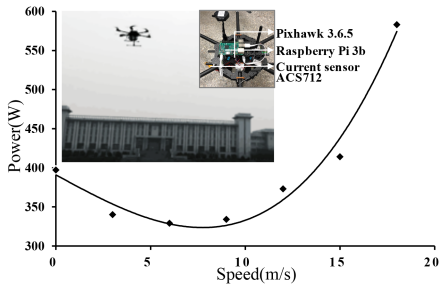


Fig. 2. A practical speed-related energy consumption model. The power consumption is related to the UAV speed via a convex function, e.g., it initially decreases and then increases as the speed increases.

However, these speed-related models only partly characterize the UAV flight energy and lack practicality, or they combine speed control with trajectory design for UAVs, complicating to solution and losing focus on the fundamental nature of speed scheduling. In this paper, we thus remodel the UAV energy power consumption through real-world experiments, and the results also be verified as sound according to the work in [22]. We will detail the experiments in the next section.

III. System Model and Problem Formulation

A. Practical Flight Energy Consumption Model

As mentioned above, traditional flight energy models have certain limitations. Therefore, we conduct a set of on-site flight tests to disclose the relationship between flight speed and power consumption. For these flight tests, we use a 2 kgs hexacopter drone, following a similar setup as the flight tests conducted in [51]. Specifically, this hexacopter drone is equipped with a flight controller Pixhawk 3.6.5 which is connected to a companion computing device, Raspberry Pi 3b single-board computer (RPi). The Pixhawk controller continuously transmitted UAV battery voltage information to the RPi using the MAVLink protocol. To monitor real-time UAV battery current values, we also install a current module ACS712. The RPi retrieve these current values through I^2C communication protocol. By collecting both the voltage and current values, we can compute the UAV power consumption.

During these flight tests, we instruct the UAV to fly along a straight line, not exceeding a distance of 1000 m, and vary the speed from 0 m/s (hover) to 18 m/s with an increment of 3 m/s. Each speed setting is repeated ten times, and the mean value is calculated to mitigate anomalies arising from individual trials. Fig. 2 illustrates a UAV during one of our flight tests.

The experiment results provide a comprehensive understanding of the correlation between flight speed and power consumption of the UAV. As demonstrated in Fig. 2, the results clearly show that neither the distance-related energy model nor the duration-related energy model holds. The flight power exhibits a convex relationship with the flight speed, more specifically,

the flight power of UAV first decreases and then increases as flight speed increases. Our findings align with the measurement results of pioneer researchers [52], and this model has been verified through theoretical analysis [22]. In conclusion, the speed-related energy consumption model is more practical and applicable compared to other existing models used for wireless communication.

B. System Model

We assume that there are n heterogeneous GNs, e.g., wireless sensors and IoT devices, distributed unevenly along a straight path. Such linear GN distributions are prevalent in real-world scenarios, including power transmission lines, roads, oil/gas pipelines, and riverbanks/coastlines, simplifying the installation and maintenance of GNs while providing efficient coverage to reduce communication interference for elongated areas. These GNs are deployed to carry out monitoring tasks and sensing data, and they are indexed based on their locations, labeled as $g_1, g_2, \ldots, g_n$. Each GN g_i has a location l_i that is determined during infrastructure deployment and regularly maintained. To collect sensed data from these GNs, a UAV is employed to fly along the path. The UAV is allowed to fly slowly or quickly, and to ascend or descend, but it can never fly backward. It is noted that the UAV is allowed to fly slowly or quickly, and to ascend or descend, but it can never fly backward.

In such data collection scenarios, the ground-to-air radio signals emitted by GNs propagate through free space, introducing inevitable loss in the communication link, known as path loss. We define the maximum allowable path loss $PL_i^{\max}$ for g_i. If the path loss between the UAV and g_i exceeds its threshold $PL_i^{\max}$, the link is considered to have failed, in other words, the data transmission process is terminated. Inspired by the work in [21], we adopt a mathematical communication model for each g_i constrained with the following definition:

$$PL_i^{\max} = \frac{\mu_{LoS} - \mu_{NLoS}}{1 + a\exp(-b[\arctan(\frac{r_i}{h_i}) - a])} + 10\log(h_i^2 + r_i^2) + M, \quad (1)$$

where μ_{LoS} and μ_{NLoS} are the mean values of the excessive path losses in LoS and non-line-of-sight (NLoS) links, respectively, $M = 20\log f + 20\log(4\pi/c) + \mu_{NLoS}$, a and b are environment parameters, and r_i and h_i are horizontal distance and vertical distance between the UAV and g_i, respectively. We assume that such communication parameters are calibrated during system setup and periodically verified to ensure the accuracy of the communication model. This communication model is depicted in Fig. 1(b). Assuming that the UAV flies at a certain altitude, h_i', and since the path loss between the UAV and g_i must be less than or equal to $PL_i^{\max}$, the transmission range is limited to a coverage disk with a radius r_i' according to (1). In this way, given the path loss threshold $PL_i^{\max}$ of g_i, a convex relationship between varying altitude h_i and radius r_i is obtained, which is visualized as resembling the shape of a balloon. This relationship reveals an extreme point where the largest radius (transmission range) and its corresponding optimum altitude can be found. Moreover, due to the heterogeneity of GNs, each g_i has a different path loss threshold $PL_i^{\max}$, mapping into a unique altitude-radius function. Accordingly, we use Fig. 3 to illustrate this system model.

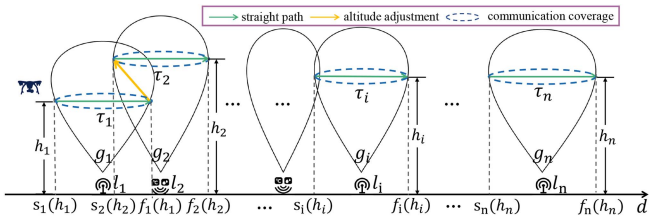


Fig. 3. Based on (1), each g_i has a *data transmission range* $(s_i(h_i), f_i(h_i))$ at altitude h_i, and requires a minimum time τ_i to upload data within this range. The UAV collects data from one GN at a time. Consequently, The UASS-GTS problem is to determine the UAV altitude-speed scheduling to ensure each g_i has enough time to transmit data while minimizing the energy consumption. However, there is a tight coupling between altitude scheduling and speed scheduling. Each GN has a preferred altitude (optimum transmission range) for speed scheduling, but frequent adjustments to these altitudes increase energy costs. Conversely, altitude scheduling aims to minimize adjustments, leading to less efficient altitudes and higher energy costs.

The UAV altitude scheduling involves designing the *flight altitude scheduling* of the UAV, where the flight altitude, h_i, over each GN, g_i, is determined, as depicted by $h_1, h_2, \ldots, h_n$ in Fig. 3. Practically, we assume that the UAV does not adjust its altitude within the transmission range of a GN and the regulated minimum and maximum flight altitudes are denoted as $H_{\min}$ and $H_{\max}$, respectively. Additionally, we define the valid maximum altitude of the communication model for g_i as $H_{\max}^i$, which is considered when $r_i \to 0$. Consequently, we have the following *altitude constraint:*

$$H_{\min} \le h_i \le H_{\max}^i. \quad (2)$$

In our scenarios, the UAV can adjust its flight altitude between any adjacent GNs, such as from h_{i-1} at GN g_{i-1} to h_i at the next GN g_i, hence the altitude adjustment energy consumption is required, denoted as E_i^h, according to the work in [8]:

$$E_i^h = \eta\Delta(h_i, h_{i-1}), i \ge 2, \quad (3)$$

where η is the correlation coefficient of altitude adjustment, and $\Delta(h_i, h_{i-1})$ is the absolute value of the difference between h_i and h_{i-1}. For instance, as depicted in Fig. 3, the UAV follows the flight path (shown in green). However, in the overlapping between g_1 and g_2, the UAV ascends (shown in orange), introducing energy consumption E_2^h, which is correlated with the altitude adjustment difference $\Delta(h_2, h_1)$. If no altitude adjustment occurs at g_i, then $\Delta(h_i, h_{i-1}) = 0$. The energy consumption for altitude scheduling over all n GNs is thus expressed as follows:

$$E_{altitude} = \eta\sum_{i=2}^{i=n} E_i^h = \eta\sum_{i=2}^{i=n}\Delta(h_i, h_{i-1}). \quad (4)$$

According to the communication model in (1), for any given flight altitude of the UAV, there exist a *data transmission range* of GN within which the UAV can collect data from GNs. For GN g_i, we denote the *data transmission range* as $(s_i(h_i), f_i(h_i))$, where $s_i(h_i)$ and $f_i(h_i)$ are the starting and ending positions of the range respectively, and are dependent on the altitude h_i.

The UAV can collect data from g_i only if its horizontal position falls between $s_i(h_i)$ and $f_i(h_i)$. Furthermore, each GN has a certain amount of data waiting to be collected. Let τ_i denote the minimal time required for the UAV to finish data collection from g_i, which can be calculated given the data amount and the data transmission rate [53]. Note that τ_i may vary between different GNs due to the heterogeneous sensed tasks. An example of the settings is given in Fig. 3. We assume each pair of adjacent GNs have an overlapping transmission range, otherwise, there is a gap between them. In this case, we can divide this problem into two smaller and independent sub-problems because the UAV scheduling before and after the gap does not affect each other.

The UAV speed scheduling is represented by the *speed scheduling function*, denoted as $v(t)$, determining the UAV flight speed at any given time t. The position of the UAV, e.g., the distance from the initial point of this part of horizontal flight on the x-axis, can be denoted as $d(t)$. We assume the UAV collects data from GNs when it is flying, whether it flies horizontally or changes altitude, but it collects from one GN at a time because they share the same communication channel. Considering the distribution of the transmission ranges, it is evident that the UAV collects data from GNs sequentially, e.g., following the order of their indices. Let t'_i and t_i be the starting and finishing time for the collection of g_i, respectively, where $i = 1, \ldots, n$. When the UAV flies horizontally, we have $t'_i = t_{i-1}$. However, if the UAV adjusts its altitude from g_{i-1} to g_i, then $t'_i > t_{i-1}$. To ensure that the data collection is within the transmission range of GN, it is necessary to impose the following *range constraint:*

$$s_i(h_i) \leq d(t'_i) < d(t_i) \leq f_i(h_i), \quad \forall i. \tag{5}$$

Furthermore, τ_i is the minimal transmission time required for the UAV to finish data collection from g_i, so we have the following *completion constraint*,

$$t_i - t'_i \geq \tau_i, \quad \forall i. \tag{6}$$

Let $p(v)$ represent the flight power consumption for the UAV at a given flight speed v. We define the energy consumption of the UAV during speed scheduling as E_{speed}, which can be calculated by the following equation:

$$E_{speed} = \int_{t_0}^{t_n} p(v(t))\,\mathrm{d}t. \tag{7}$$

We denote the total UAV energy consumption, including the energy for both speed scheduling and altitude scheduling, as E_{all}. This can be computing using the following equation:

$$E_{all} = E_{altitude} + E_{speed}. \tag{8}$$

C. Problem Formulation

We are now ready to define the problem.

Definition 1 (UASS-GTS problem): Given a set of GNs and models mentioned above, the UAV altitude-speed scheduling and GN transmission switching (UASS-GTS) problem aims to find the *flight altitude scheduling* $h_i, i = 0, 1, \ldots, n$, *speed scheduling function* $v(t)$, *transmission switching times* $t_i, i = 0, 1, \ldots, n$, such that the energy consumption in (8) is minimized while *altitude constraint* (2), *range constraint* (5) and *completion constraint* (6) are satisfied.

TABLE II
KEY NOTATION

Notation	Definition
n	The number of GNs
g_i	The i-th GN
l_i	The location of g_i
PL_i^{max}	The maximum path loss threshold of g_i
$[s_i(h_i), f_i(h_i)]$	The data transmission range of g_i at altitude h_i
H_{min}, H_{max}	The minimum and maximum flight altitude
H_{max}^i	The maximum flight altitude of g_i
η	The correlation coefficient of altitude adjustment
$\Delta(h_i, h_{i-1})$	The difference between h_i and h_{i-1}
E_i^h	The energy consumption for altitude scheduling of g_i
t_i, t'_i	The starting time and finishing time for collecting g_i
τ_i	The minimal data collection time required of g_i
$p(v)$	The flight power consumption with speed v
E_{speed}	The UAV energy consumption of flight
$E_{altitude}$	The UAV energy consumption of altitude adjustment

IV. SOLUTION TO BASIC-UASS-GTS PROBLEM

The UASS-GTS problem is quite complicated as there is a tight coupling between the altitude scheduling and speed scheduling of UAV. On one hand, optimizing speed scheduling requires the UAV to adjust its altitude for optimal *data transmission range* of GN, which results in increased energy consumption due to altitude adjustment. Conversely, optimizing for altitude scheduling maintains the UAV at a minimal change or fixed altitude, preventing it from being in the optimum *data transmission range* and thus failing to minimize the energy cost of speed scheduling.

Intuitively, we simplify the UASS-GTS problem by first addressing a basic case, formally referred to as the basic-UASS-GTS problem, where the UAV flies at a fixed altitude, denoted as H_u. As a result, in this basic problem, we only consider the UAV speed scheduling and GN transmission switching. Let $H_{\max}^{\min}$ denote the minimum of the maximum altitudes across all communication models, defined as $H_{\max}^{\min} = \min_{1 \leq i \leq n} H_{\max}^i$. For brevity, let the *data transmission range* of g_i at altitude H_u, $(s_i(H_u), f_i(H_u))$, abbreviated as (s_i, f_i), and a visual representation is provided in Fig. 4. We then define the basic-UASS-GTS problem as follows:

Definition 2 (basic-UASS-GTS problem): A UASS-GTS problem is called a basic-UASS-GTS problem if it satisfies the following condition: during the entire data collection, the *flight altitude scheduling* of the UAV maintains it at a fixed altitude, denoted as H_u, constrained within the interval, $H_{\min} \leq H_u \leq H_{\max}^{\min}$, to ensure the completion of all data collection.

In this section, we only focus on addressing the basic-UASS-GTS problem by mapping it to the crossing-the-rooms problem and then providing some interesting optimal properties for it.

A. Crossing-the-Rooms Problem and Some Optimal Properties

1) Crossing-the-Rooms *Problem:* We first introduce the time-distance diagram. On a time-distance diagram, any point

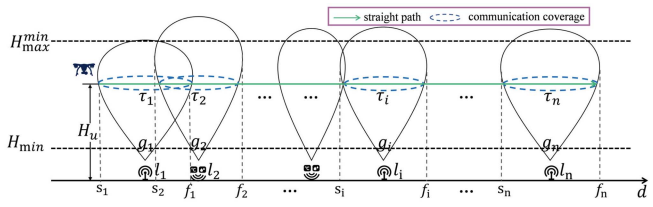


Fig. 4. The basic-UASS-GTS problem simplifies the UASS-GTS problem by assuming the UAV flies at a fixed altitude H_u within $[H_{\min}, H_{\max}^{\min}]$ to complete all data collection. It aims to schedule the UAV's speed to ensure each g_i has enough time to upload data within its *data transmission range* (s_i, f_i), while minimizing the UAV flight energy. However, slower speeds allow more data transmission time but consumes more energy, while faster speed may save energy but risks incomplete data collection. Therefore, an optimal trade-off is essential.

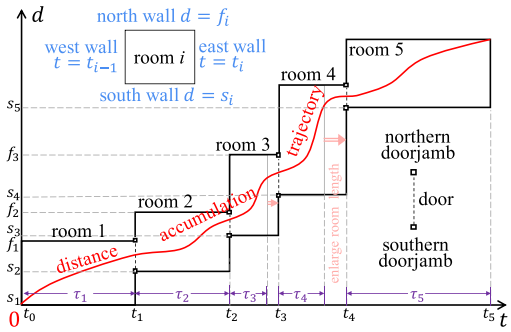


Fig. 5. The *distance accumulation trajectory* $d(t)$ crosses the rooms. For g_i, we draw a rectangle (virtual) Room i: north wall $d = f_i$, south wall $d = s_i$, west wall $t = t_{i-1}$, east wall $t = t_i$, while $t_i - t_{i-1} \geq \tau_i$. A feasible trajectory $d(t)$ must cross through all these rooms and pass through doors. The *crossing-the-rooms* problem essentially asks two questions: (1) how to construct the rooms, specifically how to determine the length of each room, (2) how to design the trajectory crossing all the rooms and passing through doors, such that the UAV energy is minimized.

(t, d) represents reaching position d at time t. Originally, at $t = 0$, the UAV is at $d = 0$. Therefore, the *distance function* $d(t)$ can be plotted as a curve on this diagram, named as *distance accumulation trajectory*, which starts at the origin (0,0). The *speed scheduling function* $v(t)$ is essentially the slope of the *distance accumulation trajectory*. Hence, finding the optimal UAV speed scheduling is equivalent to finding the optimal distance accumulation trajectory on a time-distance diagram.

As shown in Fig. 5, such a trajectory is not free to move on the diagram; it has constraints. These constraints arise from the *range constraint* and the *completion constraint* imposed on the *speed scheduling function* $v(t)$. More specifically, the *range constraint* indicates any GN g_i has a spatial constraint on the transmission range, confined between $d = s_i$ and $d = f_i$, and the *completion constraint* indicates a temporal constraint on the required transmission times, between $t = t_{i-1}$ and $t = t_i$, where $t_i - t_{i-1} \geq \tau_i$. Consequently, for g_i, we draw a rectangle Room i: the north wall at $d = f_i$, the south wall at $d = s_i$, the west wall at $t = t_{i-1}$, the east wall at $t = t_i$. Since t_{i-1} and t_i are the *collection starting time* and *collection finish time* for GN g_i, between which the UAV must fly within transmission range (s_i, f_i), so the distance accumulation trajectory must be within such rectangle region. Therefore, we construct a serial of virtual rooms for GNs, and Room i is with width (Y-axis) $(f_i - s_i)$ and length (X-axis) $(t_i - t_{i-1})$. Between two adjacent rooms, Room i and Room $i + 1$, there is a *Door* i connecting them. The door is at $t = \sum_{j=1}^{i} \tau_j$ and with size $(s_{i+1} - f_i)$, as shown in Fig. 5. A feasible *distance accumulation trajectory* must cross through all rooms and pass through doors.

The *crossing-the-rooms* problem essentially asks: (1) how to construct the rooms, specifically how to determine the length of each room, and (2) how to design the trajectory crossing all the rooms and passing through doors, such that the UAV energy is minimized. Note that the shape of this trajectory directly influences the energy consumption due to the convex relationship between flight speed and power consumption.

Solving the *crossing-the-rooms* problem is equivalent to solving the basic-UASS-GTS problem. This is because the former corresponds to asking the GN transmission switching times, whereas the latter corresponds to asking the UAV speed scheduling function. We will introduce the *looking before crossing* rooms algorithm to optimally solve the crossing-the-rooms problem which is inspired by the data flow model in [54]. The solution is uniquely mapped to the original basic-UASS-GTS problem.

2) Some Optimal Properties: An immediate lemma follows directly from the above discussion.

Lemma 1: A feasible *distance accumulation trajectory* must be within the rooms.

We want to find amongst all feasible trajectories the one with the minimal energy consumption.

Lemma 2: For any two given time intervals, the UAV consumes the minimum energy if and only if a common flight speed is used for both time intervals (if allowed).

Proof: Suppose there are two time intervals with duration τ_x and τ_y. The UAV flight speeds are constant inside each duration, and they are v_x and v_y, respectively. We need to show that using the average speed $\bar{v} = \frac{v_x\tau_x + v_y\tau_y}{\tau_x + \tau_y}$ in both intervals is more energy efficient than using any two different speed, $v_x \neq v_y$.

$$p\left(\frac{v_x\tau_x + v_y\tau_y}{\tau_x + \tau_y}\right) = p\left(\frac{\tau_x}{\tau_x + \tau_y}v_x + \frac{\tau_y}{\tau_x + \tau_y}v_y\right) \overset{(a)}{<} \frac{\tau_x}{\tau_x + \tau_y}p(v_x) + \frac{\tau_y}{\tau_x + \tau_y}p(v_y). \quad (9)$$

The inequation (a) is because of the convex property of the function $p(v)$ and the fact that $v_x \neq v_y$. Hence, we have

$$(\tau_x + \tau_y)p\left(\frac{v_x\tau_x + v_y\tau_y}{\tau_x + \tau_y}\right) < \tau_x p(v_x) + \tau_y p(v_y),$$

which clearly shows that using a common speed can reduce energy consumption. □

We have the following theory on the trajectory as a direct result of Lemma 2.

Theorem 1: The optimal trajectory is straight between any two points, as long as it is feasible.

Any non-straight trajectory between two points can be *straighten* to have the same slope, i.e., speed, to save energy. This method is called *straightening*.

From Fig. 2, it can be easily seen that there is a speed at which the power consumption is the lowest. Let such speed be denoted as $v^{\#}$. Note that, flying at $v^{\#}$ only means that the UAV consumes the minimum power. The flight energy consumption is not necessarily minimized, because the speed $v^{\#}$ may be slow and result in a long duration to cover a certain distance. Therefore, the next lemma introduces a speed v^* that minimizes the energy consumption.

Lemma 3: For any given flight distance, the UAV consumes the minimum energy if and only if it flies at speed $v = v^*$, where v^* is a constant as long as $p(v)$ is given and fixed.

Proof: Assume the given flight position interval is (d_x, d_y), hence the flight distance is $L = d_y - d_x$. Let t be the time spent to cover this distance. According to Lemma 2, the UAV consumes the minimum energy only when flying at a constant speed, $v = \frac{L}{t}$. So, the total energy consumption $E_0 = tp(v) = \frac{p(v)}{v}L$.

Define a function $h(v) = \frac{p(v)}{v}$, so $h'(v) = \frac{1}{v}(p'(v) - \frac{p(v)}{v})$. Since $p(v)$ has its minimum value at $v^{\#}$, when $v \in (0, v^{\#})$, $p(v)$ decreases, so $p'(v) < 0$, then $h'(v) < 0$. When $v \in (v^{\#}, +\infty)$, $p(v)$ increases, so $p'(v) > 0$ but the sign of $h'(v)$ depends on function $p(v)$. More specifically, the sign of $h'(v)$ depends on which one is larger, $p'(v)$ or $\frac{p(v)}{v}$. For any given position $(v, p(v))$, $v > v^{\#}$, on the speed-power diagram, $p'(v)$ can be represented by the slope of the tangent line, and $\frac{p(v)}{v}$ can be represented by the slope of the line connecting $(v, p(v))$ to (0,0). There are two possible cases: the two slopes are equal at some point or they never become equal. In case they are equal, equation $p'(v) - \frac{p(v)}{v} = 0$ must have a solution, and let it be $v = v_m$. Hence, $h'(v) < 0$ if $v < v_m$, and $h'(v) > 0$ if $v > v_m$, and $h(v)$ has the minimum value at $v = \min\{v_m, v_{\max}\}$, where $v_{\max}$ is the maximum flight speed. In case they never are equal, $h'(v) < 0$ for $\forall v \in (0, v_{\max}]$, and $h(v)$ monotone decreases with minimum value at $v = v_{\max}$.

In conclusion, the UAV consumes the minimum energy when $v = v^*$, where v^* is equal to v_m or $v_{\max}$, depending on $p(v)$. □

We have the following theorem on the trajectory.

Theorem 2: Any point of the optimal trajectory has a slope no larger than v^*.

Proof: We only provide the sketch of the proof. Intuitively, for an interval in which the UAV speed exceeds v^*, we modify it to v^*, which is more energy-efficient. In this way, the UAV flies slower and thus more time is available to collect data from GNs, so the upload duration requirements of every GN can still be satisfied. □

B. Common Starting and Ending Position Cases

How to construct all these rooms and how to design an optimal trajectory crossing these rooms, are still very challenging questions. To obtain some useful insights, this section studies two basic cases, i.e., the common starting position case and the common ending position case.

1) The Common Starting Position Case: In this case, each GN has the same transmission range starting position, $s_1 = s_2 = \cdots = s_n = 0$. But their ending positions are different: $0 < f_1 < f_2 < .. < f_n = D$. This case is referred to as the *basic-UASS-GTS problem with common starting position*. Before we present the *looking before crossing* virtual rooms algorithm that produces the optimal trajectory, we would like to introduce some properties of the optimal trajectory. □

Lemma 4: The optimal trajectory changes its direction only by increasing the slope.

Proof: Suppose, on the contrary, the optimal trajectory changes its direction by decreasing its slope at point (t, d), from v_1 to v_2. Assume straight line between (t_1, d_1) and (t, d) is with slope v_1, while straight line between (t, d) and (t_2, d_2) is with slope v_2. Since $v_1 > v_2$, the trajectory between (t_1, d_1) and (t_2, d_2) can be *straighten* to be a straight line with slope v to save more energy according to Theorem 1. We next show that straightening such two straight lines is feasible. Because the south walls of all rooms are at $d = 0$, such straightening generally moves the trajectory towards the south, not crossing the south wall. In other words, after the modification, the UAV spends more time collecting data in (d_1, d) and less time in (d, d_2) given $v_1 > v > v_2$. Since all GNs starting positions are $d = 0$, this modification is always feasible. This is a contradiction since the optimal trajectory is modified to be even more energy efficient. □

Lemma 5: The optimal trajectory changes its direction only at a northern doorjamb.

Proof: We prove this by contradiction. Consider part of the optimal trajectory, between point (t_1, d_1) and (t_2, d_2). Assume there is only one changing point (t, d), and it is not a northern doorjamb. We then try to *straighten* these two lines. There are two cases depending on whether connecting (t_1, d_1) and (t_2, d_2) directly is feasible. In case it is feasible, then a contradiction arises since *straightening* saves energy. In case it is not feasible, then there must be at least one northern doorjamb inside the triangle of (t_1, d_1), (t_2, d_2) and (t, d). We therefore choose one as the new slope change point (t', d'). Let $\tau_1, \tau_2, \tau_1', \tau_2'$ be the old and new time spent before and after changing, i.e., $\tau_1 = t - t_1, \tau_2 = t_2 - t, \tau_1' = t' - t_1, \tau_2' = t_2 - t'$. Let v_1, v_2, v_1', v_2' be the old and new speeds before and after changing, i.e., $v_1 = (d - d_1)/\tau_1, v_2 = (d_2 - d)/\tau_2, v_1' = (d' - d_1)/\tau_1', v_2' = (d_2 - d')/\tau_2'$. Then, we must have $v_1 < v_1'$ and $v_2 > v_2'$. Since the distance covered and duration spent do not change, we have the following equations.

$$v_1\tau_1 + v_2\tau_2 = v_1'\tau_1' + v_2'\tau_2',$$
$$\tau_1 + \tau_2 = \tau_1' + \tau_2'.$$

Combine the two equations by division, we have

$$\frac{\tau_1}{\tau_1 + \tau_2}v_1 + \frac{\tau_2}{\tau_1 + \tau_2}v_2 = \frac{\tau_1'}{\tau_1' + \tau_2'}v_1' + \frac{\tau_2'}{\tau_1' + \tau_2'}v_2'.$$

By the convexity of the $p(v)$ function and $v_1 < v_2$, we have

$$\frac{\tau_1}{\tau_1 + \tau_2}p(v_1) + \frac{\tau_2}{\tau_1 + \tau_2}p(v_2) > \frac{\tau_1'}{\tau_1' + \tau_2'}p(v_1') + \frac{\tau_2'}{\tau_1' + \tau_2'}p(v_2').$$

So,

$$\tau_1 p(v_1) + \tau_2 p(v_2) > \tau_1' p(v_1') + \tau_2' p(v_2').$$

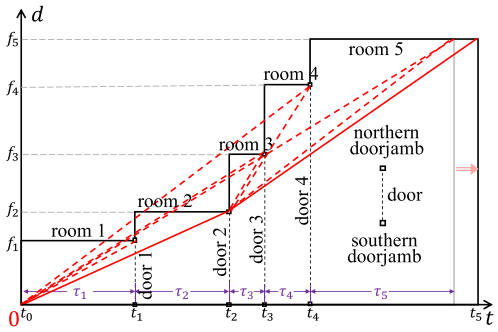


Fig. 6. The common starting position case. We first construct n rooms with minimal time as their length, i.e., Room i with length τ_i. Then, standing at the origin, we look to the east. If multiple northern doorjambs are in view, we choose the farthest one and cross the rooms to walk there. Standing at the new position, we repeatedly look east and choose the farthest doorjamb as the next stop, until we reach the destination. In case the trajectory slope is higher than v^* in some rooms, we reconstruct these rooms by enlarging the length such that the slope equals v^*.

Algorithm 1: BASIC-UASS-GTS-COSTART.

1 $k = 0, t_0 = 0, f_0 = 0$;
2 **for** $i = 1$ **to** n **do** $t_i = t_{i-1} + \tau_i$;
3 **while** $k < n$ **do**
4 $\quad k_v^n = \arg\min_{k<i\le n}(f_i - f_k)/(t_i - t_k)$;
5 $\quad v_v^n = (f_{k_v^n} - f_k)/(t_{k_v^n} - t_k)$;
6 $\quad$ **if** $v_v^n > v^*$ **then** break ;
7 $\quad$ Connect (t_k, f_k) and $(t_{k_v^n}, f_{k_v^n})$;
8 $\quad k = k_v^n$;
9 **if** $k < n$ **then**
10 $\quad t_n = (f_n - f_k)/v^* + t_k$;
11 $\quad$ Connect (t_k, f_k) and (t_n, f_n);

Thus, the optimal energy consumption is further improved, which is a contradiction. □

The optimal trajectory can thus be easily determined if all the optimal doorjambs at which the trajectory changes are known. However, how to determine these doorjambs? we present the *looking before crossing* technique in three phases.

In the first phase, set the length for each room i as τ_i, the minimal value. These rooms have their south wall at the same location $d = 0$, while north walls are different, $d = f_i$. There is no need to consider the south wall and southern doorjamb. The east wall of Room i (west wall of Room $i + 1$) is at $t = \sum_{j=1}^{i} \tau_j$, as shown in Fig. 6.

In the second phase, i.e., the main steps, the core idea is quite simple. Standing at the current position (initially at the origin), look to the east. Multiple northern doorjambs may be in view. We choose the farthest northern doorjamb in view and cross the rooms to walk there. Standing at the new position, we repeatedly look east and choose the farthest doorjamb as the next stop. Assume the northeast corner of the last room is a virtual northern doorjamb, this procedure stops until no room to cross.

In the third phase, we modify and reconstruct some rooms if the slope is higher than v^*. The room lengths are enlarged such that the slope equals v^*.

We present the formal steps of this method in Algorithm BASIC-UASS-GTS-COSTART.

These three phases are in Line 2, Line 3-8 and Line 9-11, respectively. It can be seen that the second phase, i.e., the **while** loop, is the main phase. In each **while** iteration, one piece of the trajectory is calculated by finding the next stop point. k_v^n in Line 4 is such point and v_v^n in Line 5 is the trajectory slope. After the first changing point is found, the next stop point can be computed using the same method, but from the new position. Note that if the computed speed is larger than v^* as in Line 6, the third phase begins.

Theorem 3: Algorithm BASIC-UASS-GTS-COSTART produces the optimal *distance accumulation trajectory* for the offline basic-UASS-GTS problem with a common transmission range starting position within $O(n^2)$ steps.

Proof: Since the algorithm repeats to find all trajectory pieces one by one, we prove the produced trajectory is optimal by showing its first piece is optimal. The first trajectory is set at either Line 7 or Line 11. We will prove both of them are optimal.

The farthest northern doorjamb in view k_v^n is computed in Line 4, while its corresponding speed is given v_v^n in Line 5. (1). We now prove, in case $v_v^n \le v^*$, connecting (t_k, f_k) and $(t_{k_v^n}, f_{k_v^n})$ is the first optimal trajectory piece, where $k = 0$ in the first iteration. Suppose otherwise, the first piece ends at another point (t, d). According to Lemma 5, point (t, d) must be a northern doorjamb. Let $(t, d) = (t_{k^{opt}}, f_{k^{opt}})$, for some $k^{opt} \ne k_v^n$. It is impossible $k^{opt} > k_v^n$ because the northern doorjamb beyond k_v^n is not in view, such trajectory piece will go outside rooms which are infeasible. It is impossible $k^{opt} < k_v^n$ as well, because v_v^n is the smallest by Line 4 and 5, if the first trajectory piece ends at any doorjamb before k_v^n, then there must be a following trajectory piece with a smaller slope, contradicting to Lemma 4. Hence, the first case is proved. (2). In case $v_v^n > v^*$, connecting (t_k, f_k) and (t_n, f_n) is the first optimal trajectory piece, where $k = 0$ in the first iteration. Because $t_n = (f_n - f_k)/v^* + t_k$ in Line 10, we have the slope of this trajectory piece as v^*. Besides, this trajectory piece is feasible, because, v_v^n is the smallest slope and v^* is even smaller. By Lemma 3, using v^* is optimal.

The dominant operation in this algorithm is the **while** loop. Inside the loop, the computation of k_v^n in Line 4 dominates, which takes n steps to calculate and find the minimum value. The **while** loop repeats at most n times, because in each iteration, variable k increases at least 1, and the loop terminates after $k \ge n$. Therefore the time complexity of this algorithm is $O(n^2)$.

2) The Common Ending Position Case: In this subsection, we study another basic case and derive additional properties of the optimal trajectory. In the common ending position case, each GN has the same transmission range ending position, $f_1 = f_2 = \cdots = f_n = D$. This is called the *basic-UASS-GTS problem with common ending position*.

Similar to the previous basic case, we have the following properties for the optimal trajectory. The proofs are analogous to previous ones and are left to the readers due to the space limitations. It is suggested to draw a graph for this case, similar to Fig. 6, to aid in understanding the following two lemmas.

Lemma 6: The optimal trajectory changes its direction only by decreasing the slope.

Lemma 7: The optimal trajectory changes its direction only at a southern doorjamb.

We apply the *looking before crossing* technique to find the optimal *distance accumulation trajectory* in three phases. We first construct rooms with minimal length. Then, we find the next stop by looking east and choose the farthest southern doorjamb. We then cross the rooms to walk there and repeatedly looking and crossing. Assume the northeast corner of the last room is a virtual southern doorjamb, this procedure stops until no room to cross. It is possible that in some rooms, the trajectory has a slope larger than v^* during the process. Suppose this happens for the first k rooms, then we enlarge these rooms in length such that the trajectory has a slope equal to v^*.

A formal algorithm and its proof are omitted because of space limitations. They are similar to the previous algorithm and Theorem 3.

C. Looking-Before-Crossing Algorithm for General Case

In the general case, there are no restrictions on the range of starting positions and ending positions. As a result, the optimal trajectory can change its direction by both increasing and decreasing the slope. The following lemma states how these changes occur.

Lemma 8: The optimal trajectory changes its direction only at doorjambs: increasing slope at a northern doorjamb or decreasing slope at a southern doorjamb.

This lemma is an combination of Lemma 4, 5, 6 and 7, thus the proof is similar to the previous ones and left for the readers.

The *looking before crossing* algorithm produces the optimal trajectory in three phases. In the first phase, rooms are constructed, i.e., each Room i with length τ_i, as shown in Fig. 5. Note that, each room may have different south and north walls. In the second phase, we find a walking trajectory crossing all the rooms, starting at the origin and ending at the northeast corner. In the third phase, some rooms are reconstructed by enlarging their length. Note that, when a room length is enlarged, all the rooms on the east are moved accordingly. The first and third phases have been discussed in detail in the previous section.

The high-level idea of the second phase, i.e., the main part of the *looking before crossing* algorithm, is as follows. It is obvious that if standing at the origin, we can view directly (through doors) the northeast corner, then this straight line is the optimal trajectory. However, if the view is blocked by walls, we need to find another way around it. It is clear that the view angle narrows progressively with each door. After looking through the first door, the northern boundary of the view angle is constrained by the northern doorjamb, and the southern boundary is constrained by the southern doorjamb. The view angle continues to narrow with each subsequent door until it is blocked entirely. Suppose, standing at the origin, we can see as far as Room i and part of its wall. Door i may be on the northern/southern side of the visible wall. We then walk along the northern/southern boundary of the view angle until we reach and stop at the farthest doorjamb. Standing at the new position, the *looking before crossing* strategy is repeated in the same way. Eventually, there will be no further room to cross.

Algorithm 2: BASIC-UASS-GTS-GENERAL.

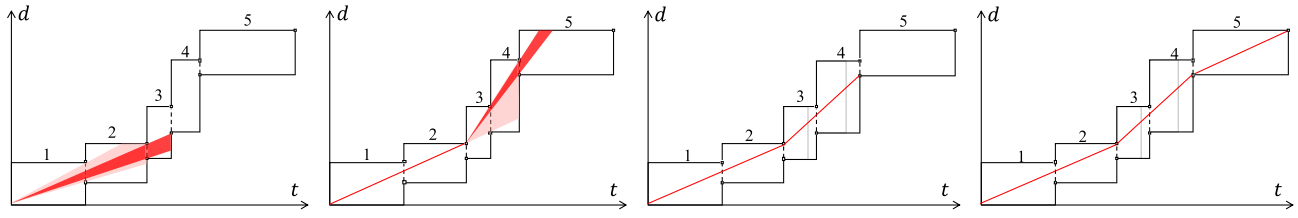
1 $k = 0, d = 0, t_0 = 0$;
2 $s_{n+1} = f_n$ // dummy for loop purpose
3 **while** $k < n$ **do**
4 $v_n^v = \infty, v_s^v = 0$;
5 **for** $j = k + 1$ **to** n **do**
6 $t_j = t_{j-1} + \tau_j, v_n^d = (f_j - d)/(t_j - t_k)$, $v_s^d = (s_{j+1} - d)/(t_j - t_k)$;
7 **if** $v_s^d > v_n^v$ **then**
8 $v_m = v_n^v, k_m = k_n^v, d_m = f_{k_n^v}$;
9 break;
10 **else if** $v_n^d < v_s^v$ **then**
11 $v_m = v_s^v, k_m = k_s^v, d_m = s_{k_s^v+1}$;
12 break;
13 **if** $v_n^v > v_n^d$ **then** $v_n^v = v_n^d, k_n^v = j$;
14 **if** $v_s^v < v_s^d$ **then** $v_s^v = v_s^d, k_s^v = j$;
15 **if** $v_n^v == v_s^v$ **then** $v_m = v_n^v, k_m = n, d_m = f_n$;
16 **if** $v_m > v^*$ **then**
17 $x = d$;
18 **for** $i = k + 1$ **to** k_m **do**
19 $t_i = \max\{(s_i - x)/v^*, \tau_i\} + t_{i-1}$;
20 $x = x + (t_i - t_{i-1})v^*$;
21 Connect (t_k, f_k) and (t_{k_m}, d_m);
22 $d = d_m, k = k_m$;

We present the formal steps of this method in Algorithm BASIC-UASS-GTS-GENERAL.

Theorem 4: Algorithm BASIC-UASS-GTS-GENERAL produces the optimal *distance accumulation trajectory* for the offline basic-UASS-GTS problem within $O(n^2)$ steps.

Proof: Similar to the proof of Theorem 3, we demonstrate the produced trajectory is optimal by showing its first piece is optimal. The first trajectory piece ends at (1) a northern doorjamb as in Line 8, (2) a southern doorjamb as in Line 11, and (3) directly at the destination as in Line 15.

The case (3) is obviously optimal. We prove (1) is optimal as well, and the case (2) is symmetry and left for the readers. In case (1), because the destination is not in direct view, then, according to Lemma 8, we know the first piece ends at a doorjamb point. It is known that in Line 8, the view angle has its northern boundary bounded by the northern doorjamb of Door k_n^v. At the same time, the southern boundary is bounded by the southern doorjamb of Door k_s^v. The southern doorjamb of the next door j is outside the view angle, on its north side. The algorithm has chosen the northern doorjamb of Door k_n^v as the ending of the first piece of trajectory. Suppose, on the contrary, that another doorjamb is the optimal ending point of the first piece. In that case, such a doorjamb must not be beyond j, because this would be infeasible (trajectory not within rooms). This doorjamb must not be a northern doorjamb before or after k_n^v, because otherwise, the trajectory can be improved. Similarly, such a doorjamb must not be a southern doorjamb before or after k_s^v. In conclusion, the only possibility is that such doorjamb is the southern doorjamb of door k_s^v. We have shown the resulting trajectory can be improved, too. To pass through door j, such trajectory must go north, crossing the northern boundary of the view angle. Let such a crossing point be (t, d). By Theorem 1, the trajectory between point (t_k, d_k) and (t, d) can improved by *straightening*. This is a contradiction.



(a) View ranges at the origin (b) New view ranges (c) Reconstruct rooms (d) The optimal trajectory

Fig. 7. *Looking before crossing* rooms. In (a), standing at the origin, the view angle through door 1 is in pink and the angle through door 2 is in red. No view through door 3, which is beyond the northern boundary of the current (red) view angle. Therefore, we walk along the northern boundary, reaching the farthest doorjamb. In (b), standing at the new position, two new view angles are in pink and red respectively. Since the northeast corner is beyond the southern boundary of the current view angle, we walk along the southern boundary to the farthest doorjamb. In (c), it can be seen that the lasted trajectory slope, in Room 3 and 4, is larger than v^*, hence we enlarge the lengths of Room 3 and 4 to reduce the slope to v^*. In (d), all room lengths and trajectories are determined.

The **while** loop repeats one most n times, because each iteration k increases at least 1, and the loop terminates after $k \geq n$. Within the **while** body, the **for** executes no more than n times. Therefore, the time complexity of this algorithm is $O(n^2)$.

An example of the proposed *looking before crossing* algorithm is illustrated in Fig. 7.

V. Solution to UASS-GTS Problem

In the previous section, we introduced the *looking before crossing* algorithm to address the basic-UASS-GTS problem, focusing on the UAV speed scheduling, which is a simplified case of the UASS-GTS problem. In this section, we propose the **A**ltitude-**S**peed **S**cheduling of **U**AV for **M**inimizing **E**nergy (**ASSUME**) algorithm to solve the UASS-GTS problem. This algorithm extends the *looking before crossing* approach by integrating joint altitude scheduling through a dynamic programming method.

From the basic-UASS-GTS problem to the UASS-GTS problem, the most critical issue is determining the *flight altitude scheduling* of the UAV to minimize the total energy consumption. Here, we use the dynamic programming technique to find the optimal *flight altitude scheduling*. Specifically, first, we discretize the altitude with a granularity of δ. According to the regulated maximum $H_{\max}$ and minimum altitude $H_{\min}$, there are $\lfloor \frac{H_{\max}-H_{\min}}{\delta} \rfloor$ adjustable altitudes. We define the sub-problem function $\mathcal{F}(i,h)$ to characterize the minimum energy consumption of the UAV collecting data from the first i GNs, with the last flight altitude of the UAV over g_i being h. In other words, if we map this procedure into a dynamic programming table, the value of each cell in i-th column and h-th row is filled with $\mathcal{F}(i,h)$. Similarly, $\mathcal{F}(j-1,h')$ can represent a sub-problem where the UAV collects data from the first $j-1$ GNs, with the last flight altitude being h'. We use $\mathcal{H}(h,h')$ to denote the energy cost of changing from altitude h' to altitude h, based on (3). Let $\mathcal{C}(j,i,h)$ denote the energy cost during the UAV speed scheduling from g_j to g_i at a fixed altitude h, which can be calculated by Algorithm BASIC-UASS-GTS-GENERAL. Note that this altitude h is constrained by $H_{\min} \leq h \leq H^k_{\max}$, for $j \leq k \leq i$, to ensure a valid altitude for data collection from g_j to g_i. Formally, the transition equation of this dynamic

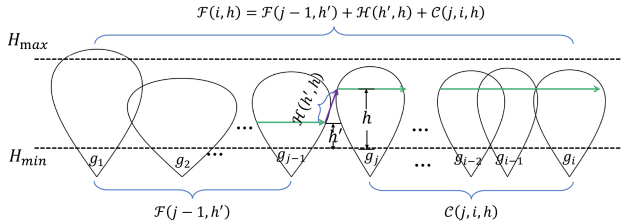


Fig. 8. The update procedure for $\mathcal{F}(i,h)$. $\mathcal{F}(i,h)$ represents the sub-problem function to characterize the minimum energy consumption of the UAV collecting data from the first i GNs, with the last flight altitude of the UAV being h. Similarly, $\mathcal{F}(j-1,h')$ denotes the sub-problem function that the first $j-1$ GNs, with the last flight altitude of the UAV being h'. $\mathcal{H}(h,h')$ is the energy cost of changing from altitude h' to altitude h, and $\mathcal{C}(j,i,h)$ is the energy cost during the UAV speed scheduling for data collection data from g_j to g_i at altitude h, as calculated by Algorithm BASIC-UASS-GTS-GENERAL.

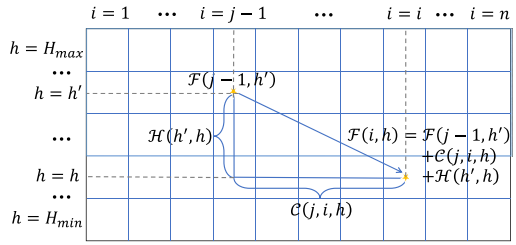


Fig. 9. A sample showcasing the update procedure for the sub-problem function $\mathcal{F}(i,h)$ by fulfilling a two-dimensional table where the row is the altitude h varying from $H_{\min}$ to $H_{\max}$ with a discretization granularity δ, and the column is the indexed GN $i, i \in \{1,\ldots,n\}$.

programming method can be written as follows:

$$\mathcal{F}(i,h) = \begin{cases} 0, & i = 0, \\ \max\limits_{H_{\min} \leq h' \leq H_{\max}, 0<j\leq i} \{\mathcal{F}(j-1,h') \\ \quad +\mathcal{H}(h,h') + \mathcal{C}(j,i,h)\}, & i > 0, \end{cases} \tag{10}$$

which indicates that if $i = 0$, $\forall h, \mathcal{F}(0,h) = 0$; otherwise, we update $\mathcal{F}(i,h)$ by iterating j from 1 to i one by one, h' from $H_{\min}$ to $H_{\max}$ in a step of δ, calculating the sum of $\mathcal{F}(j-1,h')$, $\mathcal{H}(h,h')$ and $\mathcal{C}(j,i,h)$, and selecting the minimum value. For better illustration, we demonstrate the update of $\mathcal{F}(i,h)$ in our scenarios in Fig. 8, and the corresponding dynamic programming schematic is displayed in Fig. 9. Eventually, this dynamic

Algorithm 3: ASSUME.

1 **for** $h = H_{min}$ **to** H_{max} **do**
2 | //H_{init} is the initial position of UAV
| $\mathcal{F}(1,h) = \mathcal{H}(h, H_{init}) + \mathcal{C}(1,1,h)$;
3 **for** $i = 2$ **to** n **do**
4 | **for** $h = H_{min}$ **to** H_{max} **do**
5 | | $E_{tmp} = \infty$;
6 | | **for** $j = 1$ **to** i **do**
7 | | | // ensure h is valid for all GNs from j to i
8 | | | **if** *exist* k *where* $h > H^k_{max}, j \le k \le i$ **then**
9 | | | | break;
10 | | | $E_{cur} = \mathcal{F}(j-1, h') + \mathcal{H}(h, h') + \mathcal{C}(j, i, h)$;
11 | | | $E_{tmp} = \min(E_{tmp}, E_{cur})$;
12 | | $\mathcal{F}(i,h) = E_{tmp}$;
13 return $\min_{H_{min} \le h \le H_{max}} \mathcal{F}(n,h)$;

Algorithm 4: ASSUME-ONLINE.

1 // initialize active list, position of UAV
$i = 1, AL = \emptyset, curPos = 0$;
2 **while** $curPos + c_{range} \le l_n$ **do**
3 | **while** $i \le n$ **do**
4 | | **if** $curPos + c_{range} \ge l_i$ **then**
5 | | | $AL = AL \cup \{g_i\}$;
6 | **if** *there are new GNs added to AL* **then**
7 | | Output a schedule S_{curPos} for AL with ASSUME;
8 | | **for** $\forall g_i \in AL$ **do**
9 | | | let v_i to be the speed within L_{step};
10 | | | **if** $\tau_i \cdot v_i \ge L_{step}$ **then**
11 | | | | remove g_i from AL;
12 | | | **else**
13 | | | | $\tau_i -= L_{step}/v_i$;
14 | $curPos += L_{step}$;

programming algorithm returns the optimum value $\mathcal{F}(n,h) = 0, H_{\min} \le h \le H^n_{\max}$, i.e., the minimum energy consumption. More details of this solution are shown in Algorithm ASSUME.

Theorem 5: Algorithm ASSUME produces the optimal solution for the UASS-GTS problem within $O(n^4 \frac{(H_{\max}-H_{\min})}{\delta})$ steps.

Proof: In Algorithm ASSUME, we employ Algorithm BASIC-UASS-GTS-GENERAL in Line 10 to compute the energy consumption for data collection from GN g_j to g_i at a specific altitude h, $\mathcal{C}(j,i,h)$. This computation is proven to be optimal, as asserted in Theorem 4. Moreover, the energy consumption of altitude adjustments calculated by (3), $\mathcal{H}(h,h')$, is also determined. Therefore, based on the optimal properties of dynamic programming, Algorithm ASSUME can produce the optimal solution for the UASS-GTS problem.

The main body of Algorithm ASSUME includes three **For** loops, the first **For** loop iterates at most n times, the second **For** loop requires $\frac{(H_{\max}-H_{\min})}{\delta}$ iterations, and the third **For** loop also iterates at most n times. Moreover, we call Algorithm BASIC-UASS-GTS-GENERAL in Line 10, which produces the optimal solution within $O(n^2)$. Thus, the total number of steps for Algorithm ASSUME is within $O(n^4 \frac{(H_{\max}-H_{\min})}{\delta})$. □

Remarks: First, although in theoretical analysis, the dynamic programming process in Algorithm ASSUME is complexity is proportional to the number of GNs and altitude levels, during practical algorithm execution, the number of valid states typically remains below $n^2 \frac{(H_{\max}-H_{\min})}{\delta}$. This is because states $\mathcal{F}(i,h)$ that fail to meet the altitude constraint, i.e., exist k where $h > H^k_{\max}, j \le k \le i$, are discarded, which prevents the algorithm from exploring all possible states. Section VII will show the practical performance of Algorithm ASSUME that is capable of handling most real-time applications. Second, some proactive pruning strategies can be applied to further improve the performance of Algorithm ASSUME, such as eliminating the states early if h exceeds the minimum altitude between g_j and g_i. To further reduce the search space, a coarse-to-fine adaptive discretization method [55], [56], [57] can be employed, which controls the granularity of altitude discretization by dynamically adjusting the altitude varying step, i.e., δ. This effectively balances the trade-off between accuracy and computation time, which is demonstrated based on the specific requirements of the scenario.

VI. ONLINE HEURISTIC

In the previous section, Algorithm ASSUME was proposed to produce the optimal altitude and speed scheduling for the UASS-GTS problem. However, this algorithm relies on the known information of GNs, including the communication transmission model of GN and the required transmission time. In practice, the UAV does not always have access to all GN information beforehand. Therefore, in this subsection, other than the *data transmission range*, we define the *control communication range* [58], which is generally much larger than the *data transmission range* but has a slower transmission rate.

Following the work in [59], we propose an online altitude-speed scheduling algorithm. The core idea of this algorithm is that we only schedule for the GNs within the *control communication range*, and more details of the algorithm are presented in Algorithm ASSUME-ONLINE. First, the UAV keeps an active list (AL) of GNs that is initialized to empty, and the initial position of the UAV is 0 as shown in Line 1. The UAV continuously broadcasts information about the GNs, such as data amount to be transmitted, data transmission rate, location information, and communication model. At each iteration, we add these GNs that enter the *control communication range* of the UAV to AL in Line 5. Subsequently, if there are new GNs in AL compared to the last iteration, we record the information of all GNs in AL and invoke the offline algorithm, i.e., Algorithm ASSUME, to generate a schedule S_{curPos} for these GNs, as indicated in Line 7. If the new GN does not overlap with the previous one, we only need to compute for the new GN, making the process more efficient. After that, we update the data collection requirements of each GN, as written in Line 11 and 13. Specifically, let L_{step} be the movement distance of the UAV in this iteration, and v_i be the speed of GN g_i from the output schedule S_{curPos}, if all data can be collected during the time L_{step}/v_i, we remove g_i from AL, otherwise, we update the required time τ_i. We repeat the procedures in the **While** loop in Line 2 until all GNs are served.

Fig. 10 gives an example of the online heuristic algorithm. At first, the AL includes three GNs i.e., g_1, g_2, g_3, and we invoke

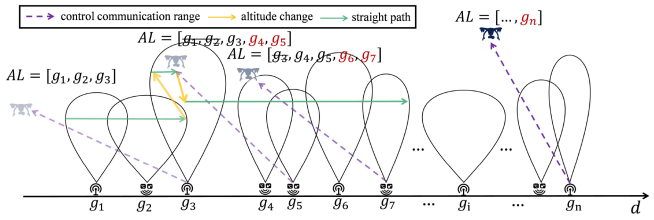


Fig. 10. An example of the online heuristic algorithm. Initially, the active list (AL) includes three GNs i.e., g_1, g_2, g_3, and we invoke the offline algorithm to compute for AL. AL changes as the UAV moves forward. Specifically, there are new GNs g_4, g_5 that fall within the control communication range of the UAV, while g_1 and g_2 finish the data collection and are removed from AL. Note that g_3 is left since its data collection is not yet complete. Repeat this procedure until the data of all GNs are collected.

the offline algorithm to compute for AL. Then the AL changes as the UAV moves forward. The new GNs g_4, g_5 fall into the *control communication range* of the UAV, while g_1 and g_2 finish the data collection, removed from AL. Note that g_3 is left since the collection is not all complete yet. Repeat this procedure until n GNs are served.

The following theorem analyzes the competitive ratio of Algorithm ASSUME-ONLINE by comparing it with the optimal offline solution, as follows.

Theorem 6: Algorithm ASSUME-ONLINE achieves a competitive ratio of $1 + 2\frac{\Delta H \eta}{c_{range}}$ compared to the optimal offline solution, where $\Delta H = H_{\max} - H_{\min}$ is the maximum flight altitude difference, η is the correlation coefficient of altitude adjustment, and c_{range} is the *control communication range*.

Proof: Let $onAlg$ be the solution from Algorithm ASSUME-ONLINE, and $offOpt$ denote the optimal offline solution from Algorithm ASSUME. We begin by constructing a modified version of Algorithm ASSUME-ONLINE, denoted as $onAlg'$. In this version, we divide the total flight path, $\mathcal{L}$, into $k = \lceil \frac{\mathcal{L}}{c_{range}} \rceil$ segments. We compute the flight scheduling for each segment, disregarding the energy cost for altitude changes between adjacent segments. Clearly, it has $onAlg \leq onAlg' + k\Delta H \eta$.

Next, we focus on a single segment $i \in [1, k]$. Let $onAlg'(i)$ and $offOpt(i)$ denote the UAV's energy consumption in segment i for $onAlg'$ and $offOpt$, respectively. Since $onAlg'$ applies Algorithm ASSUME to find the optimal solution for the GNs of each segment i, while $offOpt$ directly invokes it for all GNs along the entire flight path, we establish that $onAlg'(i) \leq offOpt(i)$. Accordingly, we can conclude that $onAlg' = \sum_{i=1}^{k} onAlg'(i) \leq \sum_{i=1}^{k} offOpt(i) = offOpt$.

Now, we add the energy consumption of altitude adjustment for $onAlg'$, which is at most $k\Delta H \eta$. Thus, we have

$$\frac{onAlg}{offOpt} \leq \frac{onAlg' + k\Delta H \eta}{onAlg'} \leq 1 + \frac{\Delta H \eta \left(\frac{\mathcal{L}}{c_{range}} + 1\right)}{p(v^*)\frac{\mathcal{L}}{v^*}}$$
$$\overset{(a)}{\leq} 1 + \frac{\Delta H \eta}{c_{range}} + \frac{\Delta H \eta}{\mathcal{L}} \overset{(b)}{\leq} 1 + 2\frac{\Delta H \eta}{c_{range}} \quad (11)$$

where inequality (a) is due to $\frac{p(v^*)}{v^*} \approx 1.05$ calculated from our real-world flight tests in Section III, and inequality (b) is due to $\mathcal{L} \geq c_{range}$. This leads us to conclude that the competitive ratio is $1 + \frac{\Delta H \eta}{c_{range}}$. Notably, when the information of GNs is completely known, i.e., $c_{range} \to \infty$, the competitive ratio reaches 1. □

TABLE III
PARAMETER SETTINGS

Parameter	Default Value	Varying Range/Step
n	30	$[40, 65]/5$
$avgDis$	$40m$	$[30m, 80m]/10m$
$avgAlt$	$130m$	$[80m, 130m]/10m$
$maxAlt$	$150m$	$[75m, 125m]/10m$
$avgDis$-τ-$ratio$	2	$[1, 6]/1$
δ	$4m$	$[2m, 7m]/1m$
c_{range}	$450m$	$[150m, 450m]/100m$

VII. SIMULATION

In this section, we conduct simulations to evaluate the performance of our proposed offline and online algorithms, *ASSUME* and *ASSUME-online*, respectively.

A. Simulation Settings

According to our real-world flight tests, we set the power-speed function $p(v)$ to be $p(v) = 0.07v^3 + 0.0391v^2 - 13.196v + 390.95$, where $v^{\#}$ and v^* can be easily calculated as $v^{\#} = 7.74$ m/s and $v^* = 13.99$ m/s. In addition, we assume that the correlation coefficient of altitude change costs is 389.15 J/m [8] and the regulated minimum altitude $H_{\min}$ is 50 m. For the parameters of the communication transmission model, we adopt the same setting as described in the work of [21]. In this simulation, we focus on studying the impact of the following key parameters: the number of GNs, the average distance between adjacent GNs, the average allowable altitude of transmission model, the allowable flight altitude of the UAV, the average ratio of the transmission rang and time required for data transmission, the granularity of altitude discretization, and the online control data transmission range, which are abbreviated as n, $avgDis$, $avgAlt$, $maxAlt$, $avgDis$-τ-$ratio$, δ, and c_{range}, respectively. To evaluate the impact of these parameters on algorithm performance, we adopt a univariate approach, altering the value of one parameter while maintaining the others at their default settings. For the sake of demonstration, the default values and varying ranges of each parameter are listed in Table III. For every parameter setting in our simulation, we randomly generate 100 instances of GNs and use the mean results for comparisons.

B. Baseline Algorithms

We compare our proposed algorithms with the following modified baseline algorithms:

- *MSMO [47]:* This approach adopts an iteration-based heuristic strategy for speed scheduling. During the t-th iteration, the UAV's speed is updated using the formula: $v^t = v^{t-1} + R_1(v^* - v^{t-1}) + R_2(1 - A)(v_{\max} - v_{\min})$, where R_1 and R_2 are random numbers. The second term guides the population towards the optimal solution, while the third term introduces random perturbations to

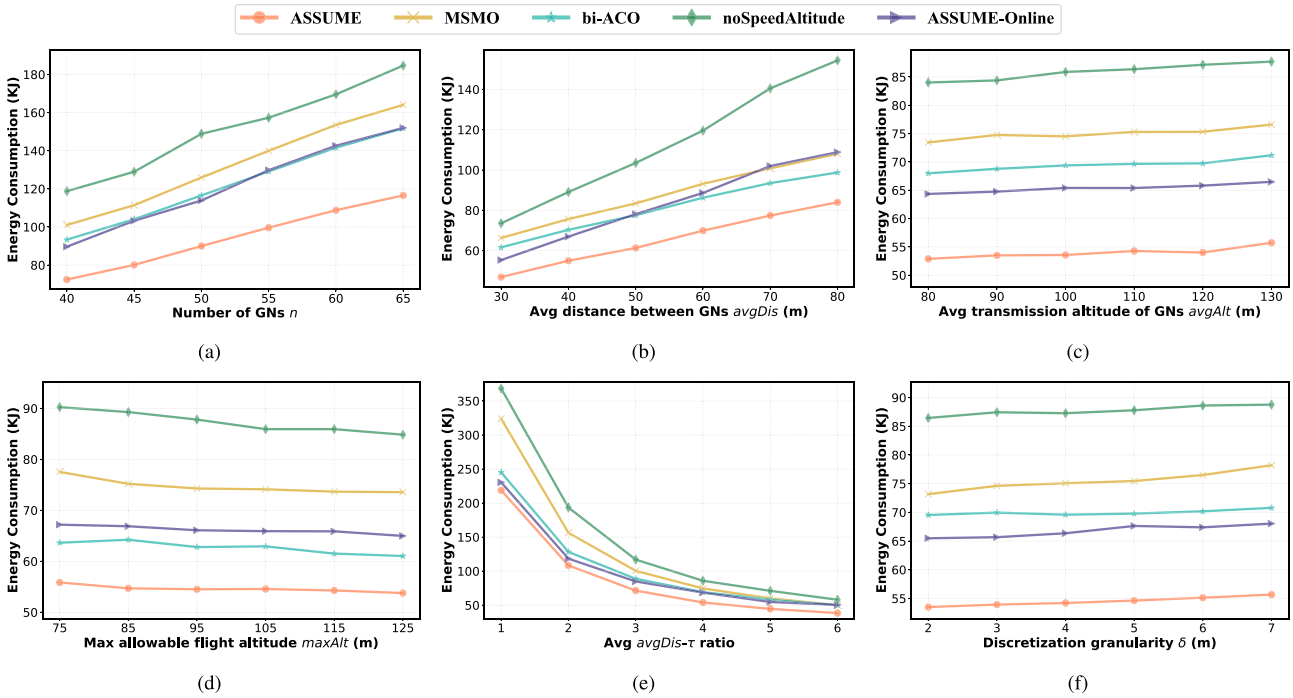


Fig. 11. Algorithm performance comparisons in UAV energy consumption. In (a), as more GNs are deployed, the energy consumption of the UAV increases across all methods. In (b), as the distance between adjacent GNs increases, the energy consumption under all methods also increases significantly across all methods. As shown in (c), there is a slight decrease in energy consumption as the average altitude grows across all methods. The results presented in (d) similarly show a slight decrease in energy consumption as the maximum allowable altitude grows across all methods. The results in (e) imply that as the average ratio increases, energy consumption shows a decreasing tendency across all methods. In (f), as the granularity increases, the energy consumption also increases across all methods. Our offline algorithm *ASSUME* saves an average of 26.1%–62.7% energy compared to the baseline methods, and the performance gap between the online algorithm and the offline optimal is 22.8%.

prevent getting trapped in local optima. This process is repeated at each possible altitude, and the altitude with the lowest energy consumption is selected.

- *bi-ACO [32]:* This approach uses $\frac{f_i - \max(s_i, f_{i-1})}{\tau_i}$ to calculate the UAV's speed over each g_i, where $f_i - \max(s_i, f_{i-1})$ is the effective transmission range of g_i and τ_i is the time required for data transmission. Then this approach incorporates an ACO strategy to optimize the altitude scheduling through feasible generation, solution division and pheromone update methods.
- *noSpeedAltitude:* Similarly, this approach calculates the UAV's speed using $\frac{f_i - \max(s_i, f_{i-1})}{\tau_i}$ for all g_i, while the flight altitude is determined using a greedy method: for each feasible altitude, we calculate the total transmission range of all GNs and select the altitude that yields the largest transmission range.

C. Results and Discussion

1) Overall Performance of Proposed Algorithms: Fig. 11 illustrates the performance of our proposed algorithms, *ASSUME* and *ASSUME-online*, against baseline algorithms across various parameter settings. In Fig. 11(a), we compare the energy consumption of different methods as the number of GNs in a straight line increases. As expected, energy consumption rises for all methods with more GNs deployed. Fig. 11(b) shows that increasing the distance between adjacent GNs also significantly increases energy consumption across all methods, as this leads to a longer total flight distance. As shown in Fig. 11(c), we reset the average altitude of the transmission model from 80 m to 130 m with a 10 m increment. The results suggest that there is a slight decrease in energy consumption as the average altitude grows across all methods. In Fig. 11(d), we vary the maximum altitude of the UAV allowed to fly from 75 m to 125 m in a 10 m step. The results similarly show a slight decrease in energy consumption as the maximum allowable altitude grows across all methods. This is because a higher flight altitude increases the probability of achieving the optimal transmission range by adjusting the altitude. As demonstrated in Fig. 11(e), we use the average ratio of transmission range to the required time for data collection to evaluate the performance of the methods. These results imply that as the average ratio increases, energy consumption shows a decreasing tendency across all methods. The reason is that increasing the average ratio is equivalent to shortening the average required time for data collection, thereby reducing energy consumption. Fig. 11(f) evaluates the algorithms by varying the discretization granularity of altitude. The results indicate that increased granularity generally leads to higher energy consumption across most methods. This is because coarser granularity makes it less likely to adjust to the most appropriate altitude to achieve the optimal transmission range, resulting in higher energy consumption.

Summary: **(1)** Across all subfigures, our proposed offline optimal algorithm, *ASSUME*, consistently exhibits the lowest

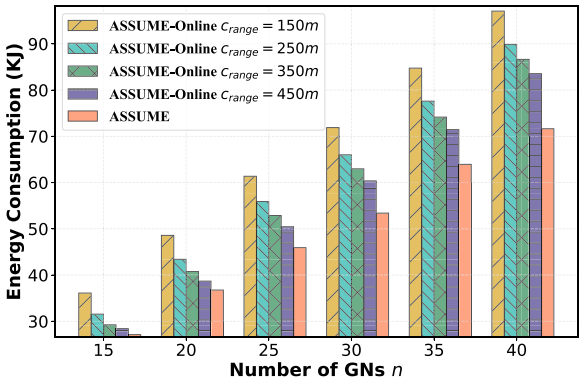


Fig. 12. The performance on energy consumption of *ASSUME-online* across varying control communication ranges and different numbers of GNs. As the range increases, the energy consumption decreases under the same number of GNs, thereby narrowing the gap with the offline results.

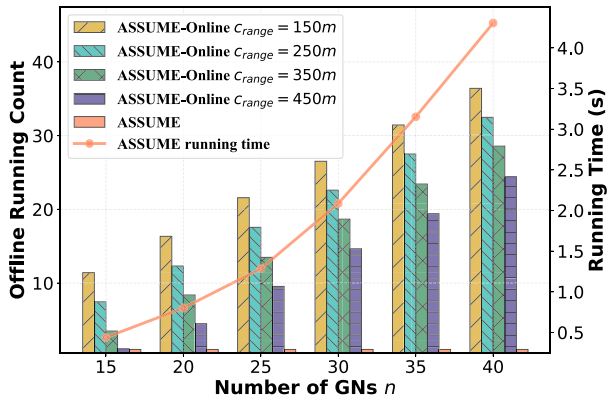


Fig. 13. The performance on offline running counts of *ASSUME-online* across varying control communication ranges and different numbers of GNs. As the number of GNs increases, the offline running time and the offline running counts of *ASSUME-online* increase, while as the range increases, the offline running counts decrease.

energy consumption, regardless of the settings. Quantitatively, *ASSUME* outperforms *MSMO*, *bi-ACO* and *noSpeedAltitude* by an average of 39.3%, 26.1% and 62.7% respectively. In addition, when the *control communication range* is 450 m, the performance gap between *ASSUME* and *ASSUME-online* is 22.8%, exceeding the theoretical lower bound of $1 + 2\frac{\Delta H\eta}{c_{range}} \approx 174$ in this simulation setting. **(2)***MSMO* employs a random search strategy to determine the optimal speed, which is less effective than our proposed speed scheduling method, *looking before crossing*. In contrast, *bi-ACO* uses a heuristic approach for scheduling UAV flight altitude, with performance heavily reliant on parameter settings, such as the distance function, making it difficult to achieve the optimal solution like *ASSUME*. Meanwhile, *noSpeedAltitude* lacks effective altitude and speed scheduling, resulting in the highest energy consumption. Therefore, *ASSUME* significantly enhances performance by integrating joint altitude and speed scheduling. These findings highlight the strong performance and practical feasibility of both the *ASSUME* and *ASSUME-online* algorithms.

2) Further Analysis of ASSUME-Online: We further conduct a set of simulations to evaluate the impact of the *control communication ranges* c_{range} on the energy consumption of the UAV. The results in Fig. 12 show a clear trend: under the same numbers of GNs, an increase in c_{range} leads to a reduction in energy consumption, thereby narrowing the gap with the optimal *ASSUME*. This is because a larger c_{range} contain more GNs at a time, reducing the frequency of dynamic changes in flight strategy. Notably, when $c_{range} = 450$ m, the *ASSUME-online* achieves the performance of *ASSUME* on average of 89.9%.

To ensure the feasibility of *ASSUME-online* in practice, we also analyze its running time by counting the running times of *ASSUME*, as shown in Fig. 13. The results show that the offline running counts of *ASSUME-online* increase with large n increases for a fixed c_{range}, while they decrease with a larger c_{range} at a fixed n. Since *ASSUME* serves as a subroutine for *ASSUME-online*, we establish its running time based on the number of GNs, n, as a baseline. As anticipated, a larger c_{range} improve performance by including more GNs. Although the running time for a single execution of *ASSUME* increases, it remains under 4.2 s, and the overall running count decreases, effectively offsetting the computational cost–an acceptable outcome for most practical applications.

Fig. 14. The efficient scheduling of ASSUME for data collection in real-world scenario.

D. ASSUME Algorithm for Real Scenario

To better reflect the effectiveness of ASSUME in real scenarios, we conduct simulation using a real-world dataset [60]. This dataset contains daily readings from various types of water sensors, such as river gauge sensors, pluviometer sensors, and aquifer piezometer sensors, placed around Catalonia in Spain. Based on the dataset, first, we filter and extract key information, including locations and types of sensors, the minimal time required for data transmission, and the transmission model corresponding to each type of sensor. We assume that an agile and flexible UAV is employed to collect sensed data from these sensors. Then we exploit our ASSUME algorithm to generate efficient energy scheduling for both altitude and speed of the UAV. As depicted in Fig. 14, we select seven representative sensors distributed along the Ebro River, denoted as $g_1, \ldots, g_7$ in their longitude order. Although the flight path from g_1 to g_7 is not a perfectly straight line, we can logically stretch it without influencing the results. Eventually, the scheduling results show, for example, that to collect data from g_7, the UAV needs to

accelerate from the previous speed of 4.50 m/s to 9.58 m/s and slightly ascend from the previous altitude of 50 m to 55 m.

E. Limitations and Future Research Directions

Limitations: From the above simulation results, while the ASSUME algorithm demonstrates significant performance improvements in data collection missions, several limitations should be acknowledged. First, our current implementation is restricted to linear GN deployments, which may not fully represent all real-world scenarios. However, this design choice enables deterministic and efficient UAV-assisted data collection scheduling for many practical applications. Second, ASSUME performs well in specific applications that depend on completely known parameters, making it less effective in dynamic environments where GN information may be incomplete or change over time. Nonetheless, our core problem formulation and optimization approach offer a solid theoretical foundation for future research. Lastly, although ASSUME produces the optimal solution with polynomial complexity, its performance may decline in very large networks or under dynamic conditions. We can mitigate this issue by adjusting the granularity of altitude discretization and implementing efficient proactive pruning strategies, which will reduce the computational overhead to a tractable level.

Future Research Directions: Looking forward, we envision several potential extensions to our work. First, a natural research direction would be adapting this algorithm to handle non-linear GN deployments and dynamic path planning like more complex intelligent transportation systems [61], [62]. Second, we will consider a more challenging scenario with mobile and changing GNs information, including node failures and imprecise transmission models due to real-world channel variations. Lastly, we will integrate our online algorithm to a real-time decision-making system to handle dynamic flight replanning, enabling more flexible and resilient operations in complex and unknown environments. These future works demonstrate how our current algorithm, while focused on a specific use case, provides a solid foundation for border applications in UAV flight scheduling optimization.

VIII. Conclusion

This paper investigates the UASS-GTS problem, where a UAV collects data from GNs deployed along a straight line with minimizing energy consumption. Unlike existing works, we focus on flight altitude and speed scheduling, i.e., control the altitude and speed of the UAV to save energy. In this paper, we adopt a general communication model of GN and a practical speed-related energy model based on our real-world flight tests. This energy model is distinct from most existing works in the literature on wireless communication, which typically assume a distance-related or duration-related energy model. We develop a novel *looking before crossing* algorithm on the time-distance diagram to schedule speed for a basic case. By extending this solution, we propose an optimal algorithm, ASSUME, for the general UASS-GTS problem by incorporating altitude scheduling through a dynamic programming method. Additionally, we present a heuristic based on the offline algorithm to address the online scenarios where the GN information is unavailable in advance. Simulations indicate that the ASSUME algorithm saves an average of 48.85%–62.7% energy compared to the baseline methods, and the performance gap between the online algorithm and the offline optimal is merely 22.8%. This study on the practical UAV flight energy model and altitude-speed scheduling not only offers practical guidelines for UAV-aided data collection in real-world scenarios, e.g., power line inspection, power grid monitoring, and border public safety patrol, but also has a theoretical contribution to the UAV flight scheduling optimization. Our future work will extend the proposed algorithm to more complex scenarios, e.g., non-linear GN deployments, mobile GNs, dynamic flight planning, and multi-UAV systems.

References

[1] X. Qi, J. Chong, Q. Zhang, and Z. Yang, "Towards cooperatively caching in multi-UAV assisted network: A queue-aware CDS-based reinforcement learning mechanism with energy efficiency maximization," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 16461–16477, May 2024.

[2] Z. He et al., "Energy minimization for UAV-enabled wireless power transfer and relay networks," *IEEE Internet Things J.*, vol. 10, no. 21, pp. 19141–19152, Nov. 2023.

[3] D. Yang, J. Wang, F. Wu, L. Xiao, Y. Xu, and T. Zhang, "Energy efficient transmission strategy for mobile edge computing network in UAV-based patrol inspection system," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5984–5998, May 2024.

[4] D. Han, T. Han, Y. Zhao, and R. Li, "UAV-aided information collection method based on joint optimization of node access and flight strategy," *IEEE Trans. Veh. Technol*, vol. 73, no. 3, pp. 4095–4106, Mar. 2024.

[5] Y.-C. Ko and R.-H. Gau, "UAV velocity function design and trajectory planning for heterogeneous visual coverage of terrestrial regions," *IEEE Trans. Mobile Comput.*, vol. 22, no. 10, pp. 6205–6222, Oct. 2023.

[6] K. Liu and J. Zheng, "UAV trajectory planning with interference awareness in UAV-enabled time-constrained data collection systems," *IEEE Trans. Veh. Technol*, vol. 73, no. 2, pp. 2799–2815, Feb. 2024.

[7] S. Piao, Z. Ba, L. Su, D. Koutsonikolas, S. Li, and K. Ren, "Automating CSI measurement with UAVs: From problem formulation to energy-optimal solution," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2019, pp. 2404–2412.

[8] F. Shan, J. Huang, R. Xiong, F. Dong, J. Luo, and S. Wang, "Energy-efficient general PoI-visiting by UAV with a practical flight energy model," *IEEE Trans. Mobile Comput.*, vol. 22, no. 11, pp. 6427–6444, Nov. 2023.

[9] X. Ma, M. Huang, W. Ni, M. Yin, J. Min, and A. Jamalipour, "Balancing time and energy efficiency by sizing clusters: A new data collection scheme in UAV-aided large-scale Internet of Things," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 9355–9367, Mar. 2024.

[10] Y. Zhu, B. Yang, M. Liu, and Z. Li, "UAV trajectory optimization for large-scale and low-power data collection: An attention-reinforced learning scheme," *IEEE Trans. Wireless Commun.*, vol. 23, no. 4, pp. 3009–3024, Apr. 2024.

[11] X. Gao, X. Zhu, and L. Zhai, "AoI-sensitive data collection in multi-UAV-assisted wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 8, pp. 5185–5197, Aug. 2023.

[12] X. Wang, M. Yi, J. Liu, Y. Zhang, M. Wang, and B. Bai, "Cooperative data collection with multiple UAVs for information freshness in the Internet of Things," *IEEE Trans. Commun.*, vol. 71, no. 5, pp. 2740–2755, May 2023.

[13] H. Zhang, W. Tang, and J. Peng, "Performance analysis of cooperative caching and transmission diversity in cache-enabled UAV networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 5, pp. 4411–4423, May 2024.

[14] R. Xiong and F. Shan, "Dronetank: Planning UAVs flights and sensors data transmission under energy constraints," *Sensors*, vol. 18, no. 9, 2018, Art. no. 2913.

[15] C. Xiang et al., "Reusing delivery drones for urban crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 22, no. 5, pp. 2972–2988, May 2023.

[16] A. Rahmati, X. He, I. Guvenc, and H. Dai, "Dynamic mobility-aware interference avoidance for aerial base stations in cognitive radio networks," in *Proc. IEEE Int. Conf. Comput. Commun.*, 2019, pp. 595–603.

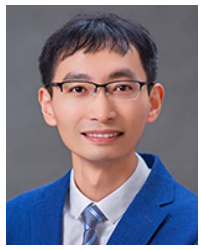
[17] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.
[18] J. Gong, T.-H. Chang, C. Shen, and X. Chen, "Flight time minimization of UAV for data collection over wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1942–1954, Sep. 2018.
[19] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and J. Henry, "UAV trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning," *IEEE Trans. Veh. Technol*, vol. 70, no. 9, pp. 9540–9554, Sep. 2021.
[20] L. Zhang, A. Celik, S. Dang, and B. Shihada, "Energy-efficient trajectory optimization for UAV-assisted IoT networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4323–4337, Dec. 2022.
[21] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.
[22] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.
[23] Y. Luo, X. Yu, D. Yang, and B. Zhou, "A survey of intelligent transmission line inspection based on unmanned aerial vehicle," *Artif. Intell. Rev.*, vol. 56, no. 1, pp. 173–201, 2023.
[24] N. Hoanh and T. V. Pham, "A multi-task framework for car detection from high-resolution UAV imagery focusing on road regions," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 11, pp. 17160–17173, Nov. 2024.
[25] L. C. Sousa et al., "Autonomous path follow UAV to assist onshore pipe inspection tasks," in *Proc. IEEE 7th Int. Conf. Robot. Autom. Eng.*, 2022, pp. 112–117.
[26] J. Jessin, C. Heinzlef, N. Long, and D. Serre, "A systematic review of UAVs for island coastal environment and risk monitoring: Towards a resilience assessment," *Drones*, vol. 7, no. 3, 2023, Art. no. 206.
[27] Lora transceivers - semtech. Accessed Jul. 30, 2019. [Online]. Available: www.semtech.com/products/wireless-rf/lora-transceivers
[28] Esp82766ex overview - espressif systems. Accessed Jul. 30, 2019. [Online]. Available: www.espressif.com/en/products/hardware/esp8266ex/overview
[29] T. Ma et al., "UAV-LEO integrated backbone: A ubiquitous data collection approach for B5G Internet of Remote Things networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3491–3505, Nov. 2021.
[30] J. Li, G. Sun, L. Duan, and Q. Wu, "Multi-objective optimization for UAV swarm-assisted IoT with virtual antenna arrays," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4890–4907, May 2024.
[31] W. Xia, Y. Zhu, L. De Simone, T. Dagiuklas, K.-K. Wong, and G. Zheng, "Multiagent collaborative learning for UAV enabled wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 9, pp. 2630–2642, Sep. 2022.
[32] Y. Wang, J. Zhu, H. Huang, and F. Xiao, "Bi-objective ant colony optimization for trajectory planning and task offloading in UAV-assisted MEC systems," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 12360–12377, Dec. 2024.
[33] A. Mondal, D. Mishra, G. Prasad, and A. Hossain, "Deep reinforcement learning for green UAV-assisted data collection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2023, pp. 1–5.
[34] Z. Ning, H. Ji, X. Wang, E. C. Ngai, L. Guo, and J. Liu, "Joint optimization of data acquisition and trajectory planning for UAV-assisted wireless powered Internet of Things," *IEEE Trans. Mobile Comput.*, vol. 24, no. 2, pp. 1016–1030, Feb. 2025.
[35] Z. Huang et al., "LI2: A new learning-based approach to timely monitoring of points-of-interest with UAV," *IEEE Trans. Mobile Comput.*, vol. 24, no. 1, pp. 45–61, Jan. 2025.
[36] F. Song, M. Deng, H. Xing, Y. Liu, F. Ye, and Z. Xiao, "Energy-efficient trajectory optimization with wireless charging in UAV-assisted MEC based on multi-objective reinforcement learning," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 10867–10884, Dec. 2024.
[37] J. Shi, P. Cong, L. Zhao, X. Wang, S. Wan, and M. Guizani, "A two-stage strategy for UAV-enabled wireless power transfer in unknown environments," *IEEE Trans. Mobile Comput.*, vol. 23, no. 2, pp. 1785–1802, Feb. 2024.
[38] C. Deng, X. Fang, and X. Wang, "Beamforming design and trajectory optimization for UAV-empowered adaptable integrated sensing and communication," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 8512–8526, Nov. 2023.
[39] C. Diaz-Vilor, A. M. Abdelhady, A. M. Eltawil, and H. Jafarkhani, "Multi-UAV reinforcement learning for data collection in cellular MIMO networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 15462–15476, Oct. 2024.
[40] D. Ebrahimi, S. Sharafeddine, P.-H. Ho, and C. Assi, "Autonomous UAV trajectory for localizing ground objects: A reinforcement learning approach," *IEEE Trans. Mobile Comput.*, vol. 20, no. 4, pp. 1312–1324, Apr. 2021.
[41] A. Albanese, V. Sciancalepore, and X. Costa-Pérez, "SARDO: An automated search-and-rescue drone-based solution for victims localization," *IEEE Trans. Mobile Comput.*, vol. 21, no. 9, pp. 3312–3325, Sep. 2022.
[42] C.-C. Lai, L.-C. Wang, and Z. Han, "The coverage overlapping problem of serving arbitrary crowds in 3D drone cellular networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 3, pp. 1124–1141, Mar. 2022.
[43] T. M. Hoang, B. C. Nguyen, H. L. T. Thanh, X. N. Tran, and P. T. Hiep, "Finite block length NOMA MU pairing UAV-enable system: Performance analysis and optimization," *IEEE Trans. Mobile Comput.*, vol. 23, no. 10, pp. 9804–9820, Oct. 2024.
[44] C.-C. Hsia, Y. Xu, J. Ren, and X. Chen, "Demo abstract: Carl: Collaborative altitude-adaptive reinforcement learning for active search with UAV swarms," in *Proc. 23rd ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2024, pp. 249–250.
[45] H.-H. Choi and J.-R. Lee, "Joint optimization of altitude and beamwidth for UAV-powered wireless sensor networks," *IEEE Trans. Veh. Technol*, vol. 72, no. 1, pp. 1279–1284, Jan. 2023.
[46] J. Zhu, X. Wang, H. Huang, S. Cheng, and M. Wu, "A NSGA-II algorithm for task scheduling in UAV-enabled MEC system," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 9414–9429, Jul. 2022.
[47] R. Jia, Q. Fu, Z. Zheng, G. Zhang, and M. Li, "Energy and time trade-off optimization for multi-UAV enabled data collection of IoT devices," *IEEE/ACM Trans. Netw.*, vol. 32, no. 6, pp. 5172–5187, Dec. 2024.
[48] Y. Liu, Q. Deng, Z. Zeng, A. Liu, and Z. Li, "A hybrid optimization framework for age of information minimization in UAV-assisted MCS," *IEEE Trans. Services Comput.*, vol. 18, no. 2, pp. 527–542, Mar./Apr. 2025.
[49] F. Morbidi, R. Cano, and D. Lara, "Minimum-energy path generation for a quadrotor UAV," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 1492–1498.
[50] R. Ding, F. Gao, and X. S. Shen, "3D UAV trajectory design and frequency band allocation for energy-efficient and fair communication: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 7796–7809, Dec. 2020.
[51] J. Modares, F. Ghanei, N. Mastronarde, and K. Dantu, "UB-ANC planner: Energy efficient coverage path planning with multiple drones," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 6182–6189.
[52] K. Karydis and V. Kumar, "Energetics in robotic flight at small scales," *Interface Focus*, vol. 7, no. 1, 2017, Art. no. 20160088.
[53] W. Wu, S. Sun, F. Shan, M. Yang, and J. Luo, "Energy-constrained UAV flight scheduling for IoT data collection with 60 GHz communication," *IEEE Trans. Veh. Technol*, vol. 71, no. 10, pp. 10 991–11 005, Oct. 2022.
[54] M. A. Zafer and E. Modiano, "A calculus approach to energy-efficient data transmission with quality-of-service constraints," *IEEE/ACM Trans. Netw.*, vol. 17, no. 3, pp. 898–911, Jun. 2009.
[55] R. Battiti, E. Amaldi, and C. Koch, "Computing optical flow across multiple scales: An adaptive coarse-to-fine strategy," *Int. J. Comput. Vis.*, vol. 6, pp. 133–145, 1991.
[56] Y. S. Akgul and C. Kambhamettu, "A coarse-to-fine deformable contour optimization framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 174–186, Feb. 2003.
[57] S. James, K. Wada, T. Laidlow, and A. J. Davison, "Coarse-to-fine Q-attention: Efficient learning for visual robotic manipulation via discretisation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13 739–13 748.
[58] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in UAV enabled wireless sensor network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 328–331, Jun. 2018.
[59] X. Ren, W. Liang, and W. Xu, "Data collection maximization in renewable sensor networks via time-slot scheduling," *IEEE Trans. Comput.*, vol. 64, no. 7, pp. 1870–1883, Jul. 2015.
[60] "Catalonia water resource daily monitoring," 2024. [Online]. Available: https://www.kaggle.com/datasets/edomingo/catalonia-water-resource-daily-monitoring
[61] S. Wandelt and C. Zheng, "Toward smart skies: Reviewing the state of the art and challenges for intelligent air transportation systems (IATS)," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 10, pp. 12943–12953, Oct. 2024.
[62] C. Peng et al., "Joint energy and completion time difference minimization for UAV-enabled intelligent transportation systems: A constrained multi-objective optimization approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 10, pp. 14040–14053, Oct. 2024.

Jianping Huang received the BS degree from the Nanjing University of Science and Technology, China, in 2019, and the MS degree in computer science from Southeast University, China in 2022. She is currently working toward the PhD degree with the School of Computer Science and Engineering, Southeast University. Her research interests are in energy consumption of UAV, UAV scheduling, and flight planning.

Feng Shan (Member, IEEE) received the PhD degree in computer science from Southeast University, Nanjing, China, in 2015. He was a visiting student with the School of Computing and Engineering, University of Missouri-Kansas City, Kansas City, MO, USA, from 2010 to 2012. He is currently an associate professor with the School of Computer Science and Engineering, Southeast University. His research interests include the areas of Internet of Things, wireless networks, swarm intelligence, and algorithm design and analysis.

Junzhou Luo (Member, IEEE) received the BSc degree in applied mathematics and the MS and PhD degrees in computer network, from Southeast University, China, in 1982, 1992, and 2000, respectively. He is a full professor with the School of Computer Science and Engineering, Southeast University. He is co-chair of IEEE SMC Technical Committee on Computer Supported Cooperative Work in Design, and he is a member of the ACM and chair of ACM SIGCOMM China. His research interests are next generation network architecture, network security, cloud computing, and wireless LAN.

Runqun Xiong (Member, IEEE) received the PhD degree in computer science from Southeast University. He was with the European Organization for Nuclear Research as a research associate for the AMS-02 experiment from 2011 to 2012. He is currently an associate professor with the School of Computer Science and Engineering, Southeast University, China, where he is involved in AMS-02 data processing with the AMS Science Operations Center. His current research interests include cloud computing, industrial Internet, and drone-based wireless communication systems. He is a member of the ACM and the China Computer Federation.



Wenjia Wu (Member, IEEE) received the BS and PhD degrees in computer science from Southeast University, Nanjing, China, in 2006 and 2013, respectively. He is currently an associate professor with the School of Computer Science and Engineering, Southeast University. His research interests include wireless and mobile networks.