Optimizing Joint Speed and Altitude Schedule for UAV Data Collection in Low-Altitude Airspace

Yiqian Wang, *Graduate Student Member, IEEE*, Jianping Huang, *Graduate Student Member, IEEE*, Feng Shan, *Member, IEEE*, Yuming Gao, Runqun Xiong, *Member, IEEE*, and Junzhou Luo, *Member, IEEE*

***Abstract*—Low-altitude airspace in major cities across the world is increasingly congested with uncrewed aerial vehicles (UAVs) and other aircraft. Emerging technologies, innovative business models, and supportive government policies are driving the growth of the low-altitude economy, where UAVs play a crucial role. Given the limited on-board energy of UAVs, this paper investigates the Joint UAV Speed and Altitude Scheduling (JUSAS) problem for data collection from sensors deployed along power transmission lines, bridges, highways, railways, water/gas/oil pipelines, or rivers/coasts. Distinct from existing work, the paper focuses on jointly optimizing UAV speed and altitude scheduling while determining the wireless sensor collection order. It accounts for the altitude-specific sensor transmission range model and the complexities of overlapping range relationships. We first propose the *Slowest Segment First* (SSF) policy to obtain an optimal UAV speed scheduling for fixed-altitude scenarios. Building upon this, we then reformulate JUSAS as a shortest-path-type problem using our novel flight scheduling graph, solved efficiently through the *SSF-based Ant Colony Optimization* (SSF-ACO) algorithm. To handle practical scenarios without prior sensor information along the path, we develop SSF-ACO-Online for real-time scheduling. Extensive simulations demonstrate that SSF-ACO significantly outperforms four other algorithms (i.e., SSF-Only, SSF-GA, SSF-PSO, and SSF-SA) in energy efficiency, and reduces 13.11% energy consumption on average. SSF-ACO-Online achieves comparable performance with energy consumption 1.24% higher than offline counterpart in average.**

***Index Terms*—Uncrewed aerial vehicle, data collection, speed scheduling, altitude scheduling, online problem, energy efficient.**

I. Introduction

Low-altitude airspace, typically below 1,000 meters [1], is increasingly being used for commercial and civil purposes, giving rise to a rapidly emerging low-altitude economy (LAE). In many cities, the uncrewed aerial vehicles (UAVs) and electric vertical take-off and landing (eVTOL) aircraft is creating unprecedented opportunities in sectors such as logistics [2], precision agriculture [3], urban air mobility [4], and disaster response [5].

This expansion of low-altitude economic activity is supported by advances in technology, novel business models, and regulatory initiatives. For example, automated UAV docking stations (e.g., DJI's Dock [6]) allow drones to land, recharge, and resume missions autonomously, greatly extending their operational endurance. Similarly, eVTOL aircraft enable on-demand urban air mobility services, exploiting vertical take-off and landing to dramatically reduce travel time. Meanwhile, governments worldwide are developing regulatory frameworks and infrastructure to facilitate these low-altitude operations and harness the LAE's potential for innovation and economic growth [7], [8], [9].

UAVs, with their superior maneuverability, rapid deployment capabilities, and adaptive flight patterns, are indispensable to the LAE. They have been deployed for a wide range of applications, including last-mile package delivery [2], simultaneous localization and mapping [10], and forestry [3], [11]. When equipped with onboard processing units, UAVs can serve as mobile servers, providing computational services to ground user or vehicles [5], [12], [13], [14], which supply temporary computing support and alleviating computational loads.

One of the most promising applications of UAVs is in the field of data collection [15], [16], [17], [18], [19]. Ground sensor networks, which are widely deployed for environmental and infrastructure monitoring in scenarios such as bridges and railways [20], water/gas/oil pipelines [21], and rivers/coasts [11], generate massive amounts of data that require efficient and periodic collection. For example, Fig. 1 describes the case that sensors are deployed along a road to monitor traffic conditions or detect environmental changes, and a UAV is dispatched to collect data from these sensors by flying close to the sensors and establishing line-of-sight (LoS) links, which proves to be more efficient than traditional wireless sensor networks [3], [22], [23].

However, UAVs face energy constraints due to limited onboard battery capacity, necessitating careful planning of flight trajectories and speeds during data collection to minimize energy consumption. The UAV's energy consumption is primarily determined by two aspects of its flight motion. (1) Forward flight, where power consumption exhibits a convex relationship with forward speed [24], [25], [26], as illustrated in Fig. 2. There exists a unique energy-efficient speed v^*, at which the UAV

Received 26 February 2025; revised 12 June 2025; accepted 16 July 2025. Date of publication 22 July 2025; date of current version 5 November 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62232004, Grant 62472090, 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 H. Shen. *(Corresponding author: Feng Shan.)*

The authors are with the School of Computer Science and Engineering, Southeast University, Nanjing 211189, China (e-mail: yiqianwang@seu.edu.cn; jphuang@seu.edu.cn; shanfeng@seu.edu.cn; gaoyuming@seu.edu.cn; rxiong@seu.edu.cn; jluo@seu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TMC.2025.3591698, provided by the authors.

This article has supplementary downloadable material available at https://doi.org/10.1109/TMC.2025.3591698, provided by the authors.

Digital Object Identifier 10.1109/TMC.2025.3591698

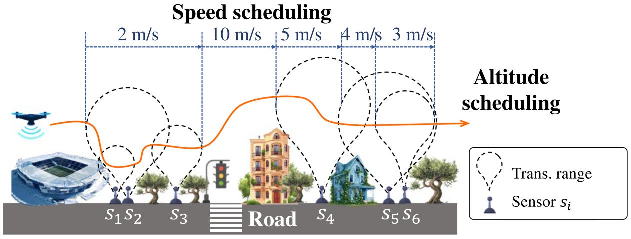


Fig. 1. Heterogeneous sensors are strategically deployed along a road. The UAV is dispatched to collect data from these sensors during the flight along the road, where the arrow curve represents the trajectory. To ensure energy efficiency, the UAV dynamically adjusts both its trajectory and flight speed, minimizing total energy consumption.

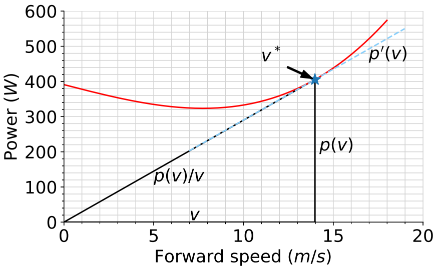


Fig. 2. A practical power consumption model when a UAV is at forward (horizontal) flight by on our previous work [25], where v^* is the most energy-efficient forward speed. See Lemma 2 for details.

consumes the minimum horizontal energy to cover any distance. (2) Vertical flight, during which cumulative altitude changes occur [26], [27].

While energy efficiency of UAVs has been extensively studied, existing studies often adopt simplified assumptions that limit their practical applicability. Some works [28], [29], [30] restrict UAV operation to a fixed altitude, overlooking the impact of altitude on LoS communication quality. Other studies [31], [32], [33], [34], [35] concentrate solely on optimizing hovering positions, failing to exploit UAV mobility. Additionally, several approaches [5], [36], [37] assume constant flight speed, neglecting the significant impact of speed variation on energy consumption. These simplified assumptions, while making the problem more tractable, often lead to suboptimal solutions in practical scenarios where both speed and altitude variations are essential for efficient data collection.

Distinct from previous work, we focus on jointly optimizing UAV altitude and speed while accounting for complex sensor coverage relationships. As illustrated in Fig. 1, we adopt an advanced air-to-ground communication model [38], [39] that captures the dynamics characteristics of data collection. In this model, the communication range exhibits a non-linear relationship with altitude, initially expanding as the UAV gains altitude owing to improved LoS probability, but eventually decreasing as the UAV approaches the sensor's maximum communication radius. This altitude-dependent coverage creates intricate spatial relationships between sensors, manifesting in various overlapping patterns: partial overlap, full containment, reverse containment, and disjointness. Moreover, these relationships become even more complex as communication range changes with altitude, creating challenges that have not been fully addressed in previous research. For example, in Fig. 1, as the altitude increases, sensor S_1 initially overlaps with S_2, but eventually becomes fully contained by S_2. Such overlapping range relationship better captures the complexity of real-world scenarios and provides practical value for UAV-assisted data collection. Capturing and fully exploiting these altitude-dependent ranges is crucial, since it enables the data from a single sensor to be collected across multiple sessions and fundamentally influences the order of data collection.

Building upon these characteristics, we formulate the Joint UAV Speed and Altitude Scheduling (JUSAS) problem to minimize energy consumption in practical scenarios. Our study specifically focuses on linear infrastructure monitoring [25], [40], where sensors are deployed along structures such as power transmission lines, highways, or pipelines, as shown in Fig. 1. In this setting, a UAV flies along a fixed path to collect data from heterogeneous sensors, each with its unique transmission range and location. The UAV collects data sequentially from one sensor at a time, must ensure complete data collection from all sensors. Accordingly, the JUSAS problem presents several challenges:

- A lower flight speed can enhance communication reliability and data collection, while a higher speed (up to the energy-efficient speed v^*) reduces energy consumption per distance traveled. This trade-off presents a complex challenge in optimizing the speed schedule.
- Each sensor has an optimal horizontal coverage transmission range at different altitudes, highlighting the advantages of dynamic altitude adjustment. However, such vertical movements consume additional energy, requiring a careful balance between maximizing horizontal coverage efficiency and minimizing vertical energy consumption.
- Due to the complex overlapping patterns of sensor data transmission ranges, the UAV is often located within the ranges of multiple sensors. Consequently, determining the optimal data collection sequence becomes highly challenging.

Although numerous studies have been conducted on UAV-assisted data collection problems aimed at minimizing energy consumption, existing approaches often address individual aspects in isolation, failing to provide a comprehensive solution to the aforementioned challenges. Our previous work [25] investigated a similar UAV speed scheduling problem for data collection, but it was limited by a fixed-altitude assumption and a simplified sensor range model, which restricted both the flexibility in determining the collection order and the support for segment-based data collection. To overcome these challenges, the main contributions of this paper are summarized as follows:

- We formulate a novel joint UAV speed scheduling and altitude scheduling (JUSAS) problem, which is the first work to integrate three critical aspects: an altitude-specific transmission range model of sensors, the intricate overlapping range relationship between sensors, and a practical UAV energy consumption model.

- We develop the *Slowest Segment First* (SSF) policy and prove its optimality for fixed-altitude scenarios. Several fundamental properties of energy-efficient speed scheduling are derived, which provide theoretical foundations for UAV altitude scheduling.
- We propose the *SSF-based Ant Colony Optimization* (SSF-ACO) algorithm for the offline JUSAS problem. Then, we extend this approach to develop SSF-ACO-Online, which adapts to dynamic real-world environments while maintaining efficient performance.
- Extensive simulations evaluate the performance of the proposed algorithms. Results indicate that SSF-ACO outperforms four other algorithms (i.e., SSF-Only, SSF-GA, SSF-PSO, and SSF-SA), and reduces 13.1% energy consumption on average. SSF-ACO-Online achieves comparable performance with energy consumption 1.24% higher than offline counterpart in average.

The rest of this paper is organized as follows: Section II gives the review of the related work. The problem under study is formulated in Section III. Section IV derives several optimal properties about UAV speed scheduling, and introduces the proposed SSF policy. Then Section V-C expands SSF to SSF-ACO algorithm to solve the general JUSAS problem. The online strategy is presented in Section VI and simulation results are presented in Section VII, followed by conclusions in Section VIII.

II. Related Work

A. UAV Altitude Scheduling

The flight altitude of UAVs significantly impacts wireless communication quality between UAVs and ground nodes. According to [38], [39], increasing altitude had two opposing effects: it improved the probability of LoS communication but reduced communication coverage radius.

Existing research has taken the altitude factor into account to varying extents that can be broadly categorized into three types. (1) Fixed-altitude scheduling [28], [29], [30], where the UAV flew at a pre-set fixed altitude during the entire flight. The impact of flying or hovering altitude on energy consumption and system performance was not taken into account. This approach was simple and easy to implement, but limited energy-efficient optimization potential. (2) Static deployment altitude optimization [31], [32], [33], [34], [35], which optimized the UAV altitude as a static parameter. In these works, the optimal UAV flying or hovering altitude was determined before the mission began and was maintained throughout the entire mission. Although altitude was considered, these approaches failed to exploit the mobility of UAVs. (3) Dynamic altitude scheduling [41], [42], which considered the dynamic adjustment of UAV flying altitude during the flight. These approaches aimed to adapt varying communication characteristics at different altitudes, as well as time-varying environmental conditions. Moreover, a few works [43], [44] incorporated altitude into full 3D trajectory planning, where reinforcement learning (RL) or Deep RL (DRL) techniques were often employed. Our work differs by jointly optimizing speed and altitude scheduling based on an advanced communication model from [38], [39], enabling dynamic altitude adjustments throughout the entire data collection flight.

B. UAV Speed Scheduling

The relationship between UAV horizontal power consumption and forward speed significantly impacts energy efficiency. According to [24], [25], [26], this relationship exhibited a convex property, with a unique energy-efficient speed v^* that minimized energy consumption per distance traveled.

Some studies [5], [36], [37], [45] calculated the most energy-efficient or appropriate speed, and maintained it throughout the flight. However, these approaches ignored the dynamic nature of the environment. Conversely, works such as [13], [46], [47] adjusted the UAV's speed dynamically by dividing time into slots and optimizing speed for every time slot, which led to suboptimal performance. In contrast, our approach directly optimizes the UAV's speed dynamically without time discretization, resulting in a more precise scheduling solution. Additionally, we mathematically prove the optimality of our proposed policy.

C. Optimization Algorithms for UAV-Assisted Systems

The optimization of UAV-assisted systems has attracted significant research attention, with various algorithms proposed to minimize energy consumption. These approaches can be classified into three main categories.

Mathematical optimization techniques were applied when the problem structure allowed theoretical analysis. For instance, Shan et al. [25] derived optimal policies for linear data collection by constructing temporal-spatial constraints. Other works [48], [49] employed successive convex approximation for non-convex problems. However, these methods often imposed strict assumptions on problem formulation and could be computationally intensive.

Learning-based methods, particularly deep learning (DL) and RL, gained popularity in recent years, as they learned complex parameters or action policies during the training process. In UAV-assisted mobile edge computing systems, Lin et al. [13] proposed a parametrized dueling deep Q-network to maximize the UAV's energy efficiency. The problem was formulated as a mixed-integer nonlinear programming (MINLP) problem, which facilitated the modeling of problems involving both continuous variables (e.g., trajectory) and discrete variables (e.g., data collection decision and task offloading decision). To maximize the total throughput and energy efficiency, Chen et al. [17] formulated long-term UAV-aided data collection problem as a Markov Decision Process (MDP), and addressed it by a multi-agent DRL algorithm. However, the maximum number of ground nodes that could be served in [13] and [17] was only 6 and 8, respectively, as the extremely high computational complexity became unacceptable when the number of ground nodes increased. Hao et al. [50] formulated a task offloading problem in a UAV-assisted mobile edge computing system as a MINLP, and then transformed it into a MDP. The problem was solved by a DRL algorithm. Zhong et al. [51] also applied a similar approach to address a task offloading and resource allocation problem. Despite their successful application to larger-scale problems,

TABLE I
MAJOR NOTATIONS

Notation	Explanation
D	The destination point.
s_i	i-th sensor.
h_j	j-th altitude.
$l_i, l_{i,h}$	Entering boundary point of s_i.
$r_i, r_{i,h}$	Exiting boundary point of s_i.
R_i	Transmission rate of s_i.
t_i	Required transmission duration of s_i.
$\mathbb{S}$	Data structure including all sensor information.
$f(d)$	The time when UAV reaching position d.
$v, v(d)$	Speed of UAV at position d.
$g(d)$	Altitude of UAV at position d.
$p(v)$	UAV power consumption at forward speed v.
E_f	Horizontal energy consumption of UAV.
E_v	Vertical energy consumption of UAV.
E	Total energy consumption of UAV.
b_k	k-th boundary point.
C_k	k-th cell, $C_k = (b_k, b_{k+1})$.
$U_S(i)$	$= 0$, sensor s_i is inactive; $= 1$, s_i is active.
$U_C(k)$	$= 0$, cell C_k is unavailable; $= 1$, C_k is available.
$S[k, k']$	$= (b_k, b_{k'})$, path interval between b_k and $b_{k'}$.
$D[k, k']$	Effective distance of $S[k, k']$.
$T[k, k']$	Effective time of $S[k, k']$.

challenges still remain in the context of the JUSAS problem, particularly in modeling the environment and designing the reward function, including aligning the reward signal with the objective and dealing with sparse rewards.

Heuristic algorithms were widely applied due to their greater flexibility and practical applicability. Fu et al. [15] considered a mobile crowdsensing scenario where UAVs collected data from mobile devices, and proposed a lightweight heuristic navigation algorithm to optimize UAVs' trajectory. The authors employed multiple virtual search agents to iteratively seek a satisfactory destination. Similarly, Lin et al. [52] developed a heuristic hexagon-based scheduling algorithm to improve eneryg efficiency for periodic data collection using UAVs, and also proposed an emergent node charging scheduling method to prevent node exhaustion. To reduce cost, Gong et al. [53] incorporate particle swarm optimization (PSO) algorithm to minimize the number of UAVs required to visit all sensors. Other meta heuristic algorithms included genetic-based algorithms [16] and simulated annealing (SA) algorithm [14]. However, these methods are not readily applicable to the JUSAS problem.

In our work, we propose a method to optimally solve the UAV speed scheduling based on theoretical analysis, and apply ACO algorithm to solve the UAV altitude scheduling, which does not require environmental modeling and exhibits faster convergence.

III. System Model and Problem Formulation

A. System Model

We summarize the important notations in Table I. Consider a UAV-assisted data collection system where a UAV flies over n stationary sensors to collect data, as shown in Fig. 3. The sensors are randomly distributed along a line, denoted by $s_i, i \in \{1, 2, \ldots, n\} = \mathcal{I}$. The system operates in a two-dimensional plane, where the x-axis represents the flight from the starting point at $x = 0$ to the destination point at $x = D$, and the h-axis represents the altitude, with the ground at $h = 0$. Thus, we can use (x, h) to represent any position in this plane. Practically, we assume that the regulated minimum and maximum UAV flight altitudes are denoted as $h_{\min}$ and $h_{\max}$, respectively, i.e., $h_{\min} \leq h \leq h_{\max}$. To make the problem tractable, we discretize the flight altitude into M specific levels: $h_j, j = 1, 2, \ldots, M$, with $h_{\min} \leq h_1 < h_2 < \ldots < h_M \leq h_{\max}$.

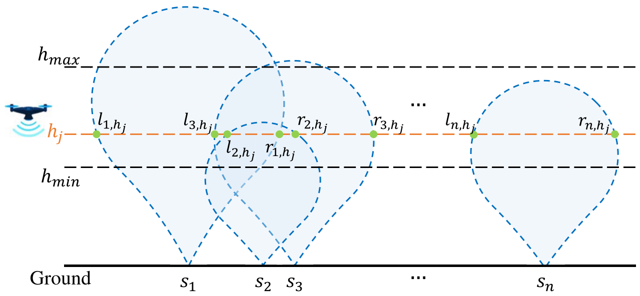


Fig. 3. An example of JUSAS problem. Each sensor has a data transmission range which varies with altitude, i.e., the blue part in the figure.

Each sensor has a balloon-shaped data transmission range as illustrated in Fig. 3. Since we assume the sensors are heterogeneous, their data transmission ranges are individually different. For any sensor s_i, we denote its boundary of the data transmission range as (l_{i,h_j}, r_{i,h_j}), where l_{i,h_j} and r_{i,h_j} are the entering and exiting boundary points intersecting with the line $h = h_j$, respectively. In other words, a UAV can only collect data from s_i when it is within the range (l_{i,h_j}, r_{i,h_j}) at flight altitude h_j. Clearly, $l_{i,h_j} < r_{i,h_j}$ always holds.

1) UAV Mobility Model: To describe the UAV's horizontal mobility, we define the following position-time function,

$$f(d) = t, \forall d \in (0, D), \tag{1}$$

indicating that the UAV reaches horizontal position $x = d$ at flight time t. The total flight completion time is given by $T = f(D)$. It is evident that $f(d)$ is continuous and monotonically increasing, as the UAV requires more time to reach farther horizontal position. Then, we have $d = f^{-1}(t)$, where $f^{-1}(\cdot)$ is the inverse function of $f(\cdot)$. Thus, the horizontal forward speed of the UAV can be denoted as $v = (f^{-1}(t))' = \frac{1}{f'(d)}$. We define the following *speed scheduling function* that specifies the UAV's horizontal flight speed for a given horizontal position d or time t,

$$v(d) = \frac{1}{f'(d)}, v(t) = (f^{-1}(t))'. \tag{2}$$

We use the altitude scheduling function $g(d)$ to represent the UAV's flight altitude at any horizontal position d. Let $\mathcal{G} = \{g_1, g_2, \ldots, g_{|\mathcal{G}|}\}$ denote the set of UAV's altitude-adjusting positions, with $g_1 < g_2 < \cdots < g_{|\mathcal{G}|}$. For ease of writing, we define $g_0 = 0$ and $g_{|\mathcal{G}|+1} = D$. Particularly, if the UAV flies at a fixed altitude during the entire flight, then $g(d)$ is a constant and $\mathcal{G} = \emptyset$. Between any two adjacent altitude-adjusting points g_e and g_{e+1}, $0 \leq e \leq |\mathcal{G}|$, the UAV flies at altitude h_{w_e}. Accordingly, $g(d)$ can be written as a piecewise function,

$$g(d) = h_{w_e}, d \in (g_e, g_{e+1}), 0 \leq e \leq |\mathcal{G}|. \tag{3}$$

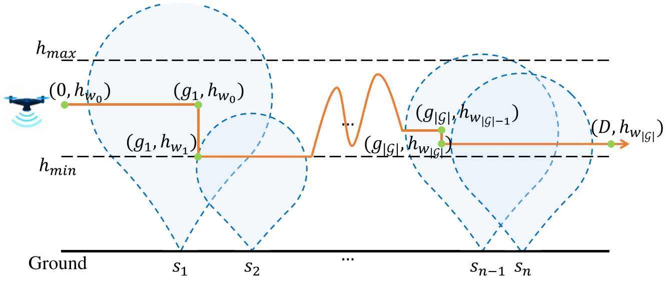

h_{max}
h_{min}
$(0, h_{w_0})$
(g_1, h_{w_0})
(g_1, h_{w_1})
...
$(g_{|\mathcal{G}|}, h_{w_{|\mathcal{G}|-1}})$
$(g_{|\mathcal{G}|}, h_{w_{|\mathcal{G}|}})$
$(D, h_{w_{|\mathcal{G}|}})$
Ground
s_1
s_2
...
s_{n-1}
s_n

Fig. 4. An example of path polyline.

Based on the above definition, the result flight path forms a polyline as illustrated in Fig. 4, referred to as the *path polyline*. It is a polyline that connects the altitude-adjusting points in the order: $(g_e, h_{w_e}) \to (g_{e+1}, h_{w_e}) \to (g_{e+1}, h_{w_{e+1}})$. Thereby, in our scenario, a path polyline is feasible only if it enables the UAV to collect data from all sensors.

2) Data Collection Process: To ensure efficient data collection, the UAV can only communicate with one sensor at a time due to interference and resource constraints. We introduce a *collection function* $I(d)$ to formally describe this process, where $I(d) = i$ indicates that the UAV is collecting data from sensor s_i at horizontal position d. When the UAV is not actively collecting data from any sensor, we denote this state as $I(d) = 0$. Therefore, we have the following *spatial constraint*,

$$l_{I(d),g(d)} \le d \le r_{I(d),g(d)}, \forall d \in (0, D), I(d) \in \mathcal{I}. \tag{4}$$

The probabilistic LoS channel model is often employed to estimate the expected transmission rate. Instead, we define R_i as the guaranteed minimum transmission rate of sensor s_i within its data transmission range to eliminate uncertainty of the probabilistic model. Outside this range, the UAV is not permitted to collect data from the sensor. Then, the required collection time of s_i is represented as $t_i = m_i/R_i$ so that t_i, where m_i denotes the data volume to be collected.

The UAV may collect data from sensor s_i across multiple non-overlapping intervals during its flight. We define $\mathcal{D}_i$ as the set of such collection intervals for s_i:

$$\mathcal{D}_i = \{(a, b) | I(d) = i, \forall d \in (a, b)\}, \forall i \in \mathcal{I}, \tag{5}$$

where (a, b) represents a continuous segment of the flight path during which the UAV collects data from sensor s_i. To ensure complete data collection from all sensors, the total time spent collecting from each sensor must meet the required collection time. This requirement is captured by the *completion constraint:*

$$\sum_{(a,b)\in\mathcal{D}_i} (f(b) - f(a)) \ge t_i, \forall i \in \mathcal{I}. \tag{6}$$

3) Energy Consumption Model: The total UAV energy is primarily consumed in flight and communication. Compared to flight, energy cost for communication is typically negligible [25]. Therefore, in this study, we only consider the energy optimization of flight, including horizontal and vertical flight.

For horizontal flight, the UAV power consumption exhibits a convex property with respect to forward flight speed at a fixed altitude [25], which is described by the *power-speed function* $p(v)$. Fig. 2 is an example. The maximum speed of the UAV is $v_{\max}$. Thus, the horizontal flight energy consumption can be calculated by:

$$E_f = \int_0^T p(v(t))\,\mathrm{d}t = \int_0^D p(v(d)) f'(d)\,\mathrm{d}d. \tag{7}$$

The vertical energy consumption for moving from altitude h to h' is denoted by $\varepsilon(h, h')$ [26], [27]. Thus, the total vertical energy consumption is calculated by the following equation:

$$E_v = \sum_{e=1}^{|\mathcal{G}|} \varepsilon(h_{w_e}, h_{w_{e-1}}). \tag{8}$$

Then the total flight energy consumption is $E = E_f + E_v$.

B. Problem Formulation

Given a set of sensors and models mentioned above, this work aims to minimize the total flight energy consumption by jointly optimize the speed scheduling function $v(d)$ (essentially $f(d)$), altitude scheduling function $g(d)$, and collection function $I(d)$. Accordingly, the JUSAS problem can be formulated as follows.

$$\mathbf{P}: \quad \min_{f(d),g(d),I(d)} \quad E = E_f + E_v \tag{9}$$

$$\text{s.t.} \quad 0 < v(d) \le v_{\max}, \forall d \in (0, D), \tag{9a}$$

$$g(d) \in \{h_1, \ldots, h_M\}, \forall d \in (0, D), \tag{9b}$$

$$I(d) \in \{0\} \cup \mathcal{I}, \forall d \in (0, D), \tag{9c}$$

$$\sum_{(a,b)\in\mathcal{D}_i} [f(b) - f(a)] \ge t_i, \forall i \in \mathcal{I}, \tag{9d}$$

$$l_{I(d),g(d)} \le d \le r_{I(d),g(d)}, I(d) \ne 0, d \in (0, D). \tag{9e}$$

Eqs. (9a), (9b) and (9c) specify the ranges for $f(d)$, $g(d)$ and $I(d)$, respectively. Eqs. (9d) and (9e) represent the completion constraint and the spatial constraint.

This problem is termed as the *offline* problem when all sensor information, such as data transmission ranges and data amounts, is known prior to scheduling. Conversely, it is referred to as the *online* problem when sensor information becomes available only when the UAV approaches.

IV. Slowest Segment First Policy

The JUSAS problem poses significant challenges due to the complex coupling between UAV altitude and speed scheduling. Adjusting speed requires the UAV to frequently change altitude to maintain optimal data transmission ranges for different sensors, increasing energy consumption from vertical movements. Conversely, minimizing altitude changes to conserve vertical energy can force the UAV to operate beyond optimal transmission ranges, resulting in longer flight times and higher horizontal energy costs.

In this section, we consider a special case of the JUSAS, named the UAV Speed Scheduling (USS) problem, where the UAV maintains a constant altitude, allowing us to focus on speed scheduling. We derive several interesting optimal properties of energy-efficient speed scheduling. Then, we introduce the

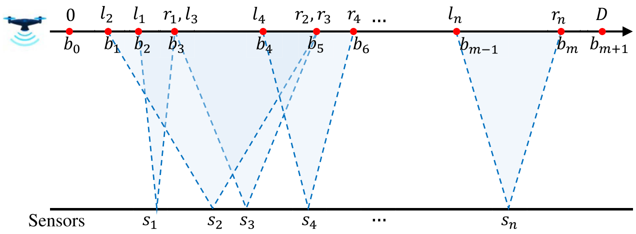


Fig. 5. An example of USS. The UAV flies from $x = 0$ to $x = D$ at a constant altitude h and collects data from all n sensors.

Slowest Segment First (SSF) policy as an optimal solution for this scenario.

A. USS Problem and Symbol Definition

A JUSAS problem is called the USS problem, if the UAV flies at a constant altitude h throughout the entire flight, i.e., $g(d) = h$ is a constant function. Mathematically, the USS problem can be formulated as follows.

$$\mathbf{P_1}: \quad \min_{f(d), I(d)} E = \int_0^D p(v(d)) f'(d) \, \mathrm{d}d,$$
$$\text{s.t.} \quad \text{(9a)}, \text{(9c)}-\text{(9e)}. \tag{10}$$

Given a fixed UAV flight altitude h, each sensor s_i has a data transmission range represented by a fixed interval (l_i, r_i) on the flight path, where $l_i = l_{i,h}$ and $r_i = r_{i,h}$ are the entering and exiting boundary points intersecting with the line h, respectively. Let m be the number of distinct boundary points, indexed as $b_k, k = 1, 2, \ldots, m$, such that $0 < b_1 < b_2 < \ldots < b_m < D$. For instance, as shown in Fig. 5, we can identify $b_1 = l_2, b_2 = l_1, b_3 = r_1 = l_3$. Particularly, we set $b_0 = 0$ and $b_{m+1} = D$. In addition, we denote the interval between two adjacent boundary points b_k and $b_{k+1}, 0 \leq k \leq m$, as a *Cell*, represented by $C_k = (b_k, b_{k+1}), k = 0, 1, \ldots, m$. Still taking Fig. 5 as an example, we have $C_0 = (0, l_2), C_1 = (l_2, l_1), C_2 = (l_1, r_1)$.

Our previous work [25] explored a similar UAV speed scheduling problem. However, it is constrained by a simplified sensor range model that transmission ranges of sensors never contain other ranges or be contained. These limitations are addressed in this paper, which introduces more dynamic and flexible solutions.

B. Properties of Optimal Policy

Before presenting the SSF policy, we will first discuss some basic optimality properties. For the purpose of this discussion, we will temporarily ignore the sensors and their data as referenced in Lemmas 1, 2, and 3.

Lemma 1: For a given flight path interval (d_x, d_y) and a required flight time T that the UAV spends, it consumes the minimum energy if and only if it flies at a constant speed $v = \frac{d_y - d_x}{T}$ from d_x to d_y.

Proof: The detailed proof is given in Appendix A, available online of the supplementary material. □

Lemma 2: For any given flight path interval, the UAV consumes the minimum energy if and only if it flies at speed v^*, where v^* is a constant if $p = p(v)$ is given and fixed.

Proof: The detailed proof is given in Appendix B, available online of the supplementary material. □

According to Lemma 2, the UAV has a unique energy-efficient speed v^* that minimizes horizontal energy consumption for any given distance. Next we introduce some properties that an optimal speed function, $v_{opt}(d)$, must possess.

Lemma 3: The optimal speed must not exceed v^*, i.e., $v_{opt}(d) \leq v^*, d \in (0, D)$.

Intuitively, whenever the UAV speed exceeds v^* with any path interval, we can adjust the speed to v^* for greater energy efficiency. This adjustment slows the UAV down, allowing more time for data collection while still ensuring that the data transmission duration remains sufficient for each sensor. Due to space constraints, we omit the proof details.

While Lemma 1 indicates a preference for the UAV to maintain a constant speed to optimize energy cost, certain conditions prompt changes in the UAV's speed. The following two lemmas describe how $v_{opt}(d)$ varies under these conditions.

Lemma 4: Any optimal speed function $v_{opt}(d), d \in (0, D)$, changes only at boundary point b_i, $1 \leq i \leq m$.

Proof: The detailed proof is given in Appendix C, available online of the supplementary material. □

Lemma 5: Any optimal speed function $v_{opt}(d), d \in (0, D)$, increases only at an exiting point, and decreases only at an entering point.

Proof: The detailed proof is given in Appendix D, available online of the supplementary material. □

Lemmas 4 and 5 imply that the UAV either decreases or maintains its speed when crossing an entering point. This is because entering a new sensor's transmission range requires the UAV to collect additional data, flying at a slower speed. Meanwhile, when the UAV crosses an exiting point and leaves a sensor's range, it may either increase or maintain its speed.

C. Segment and Effective Speed

In this subsection, we introduce some key concepts, namely *Segment* and *Effective Speed*, which are essential for understanding the optimal SSF policy.

To clarify these concepts, we first provide a brief overview of the SSF policy. The algorithm works iteratively. In each iteration, it first identifies the most constrained segment, and then determines the UAV's flight speed for that segment. Afterward, the sensors and cells belonging to that segment are marked to prepare for the next iteration, allowing the algorithm to address the problem on a reduced scale. Further details are presented in the next subsection.

Definition 1 (Segment): Given a set of sensors with data transmission ranges, a *segment* $S[k, k']$ is defined to be the interval from b_k to $b_{k'}$, $k < k'$, where b_k is an entering point and $b_{k'}$ is an exiting point.

$$S[k, k'] = \begin{cases} (b_k, b_{k'}) & \text{if } b_k < b_{k'}, \\ \varnothing & \text{otherwise.} \end{cases} \tag{11}$$

A sensor s_i *belongs* to $S[k,k']$ if and only if its data transmission range is entirely contained within $S[k,k']$, i.e., $b_k \leq l_i < r_i \leq b_{k'}$. Initially, all sensors are *active*. A sensor s_i is marked *inactive* once its data collection has been scheduled in a specific iteration. We use $U_S(i)$ to describe the status of sensors, where $U_S(i)=1$ indicates that s_i is active and $U_S(i)=0$ indicates inactive. All cells are initially *available*. A cell C_k is marked *unavailable* when the optimal speed scheduling function $v_{opt}(d), d \in (b_k, b_{k+1})$ is determined in a certain iteration. Similarly, the status of cells is represented by $U_C(k)$, where $U_C(k)=1$ indicates that C_k is available and $U_C(k)=0$ indicates unavailable. We now define some parameters which may dynamically change throughout the iterations.

Definition 2 (Effective Distance): For any segment $S[k,k']$ such that $S[k,k'] \neq \varnothing$, *effective distance* $D[k,k']$ is the total length of all available cells contained within $S[k,k']$, which can be calculated by the following equation,

$$D[k,k'] = \sum_{k''=1}^{k'-1} (b_{k''+1} - b_{k''}) \times U_C(k''). \tag{12}$$

Definition 3 (Effective Time): For any segment $S[k,k']$ such that $S[k,k'] \neq \varnothing$, *effective time* $T[k,k']$ is the total transmission duration of all active sensors belonging to $S[k,k']$, which can be calculated by the following equation,

$$T[k,k'] = \sum_{i \in I[k,k']} t_i, \tag{13}$$

where $I[k,k'] = \{i \mid b_k \leq l_i < r_i \leq b_{k'}, U_S(i)=1, i = 1,2,\ldots,n\}$ is the set of indices of active sensors belong to $S[k,k']$.

Definitions 2 and 3 clarify that, for a given segment, the effective distance is the length of its available portion, while the effective time is the time required to collect data from the active sensors within that segment. For example, in Fig. 5, sensor s_1 belongs to $S[2,3]=(b_2,b_3)$, so $D[2,3]=b_3-b_2$, $T[2,3]=t_1$; sensor s_2, s_3 and s_4 belong to $S[1,5]$, resulting in $D[1,5]=b_5-b_1, T[1,5]=t_1+t_2+t_3$.

Lemma 6: The optimal flight time in $S[k,k']$ must equal to $T[k,k']$, if the optimal speed $v_{opt}(d)$ decreases at the entering point b_k and increases at the exiting point $b_{k'}$.

Proof: The detailed proof is given in Appendix E, available online of the supplementary material. □

Definition 4 (Effective Speed): For any segment $S[k,k']$ such that $S[k,k'] \neq \varnothing$ and $T[k,k'] > 0$, *effective speed* $V[k,k']$ is the ratio of effective distance by effective time. i.e., $V[k,k'] = D[k,k']/T[k,k']$; otherwise, $V[k,k']$ is set to $\varnothing$.

The effective speed denotes the maximum possible average speed required for data collection. We define segment $S[k,k']$ as *the Slowest Segment* when no other effective speed ($\neq \varnothing$) exceeds $V[k,k']$. Lemma 4 explains how to determine the optimal flight time for a segment, and Theorem 1 details the process for finding the optimal speed.

Theorem 1: Given a set of n sensors with models mentioned above, we have $v_{opt}(d) = V[k,k']$ for $d \in S[k,k']$ if $S[k,k']$ is the slowest segment and $V[k,k'] \leq v^*$.

Algorithm 1: Slowest-Segment-First($\mathbb{S}$).

```
$U_C(k) \leftarrow 1, U_S(i) \leftarrow 1, \forall k, i$; $E_f \leftarrow 0$;
$V[k,k'], \forall k, k' \leftarrow$ Compute-Segment-Speed($\mathbb{S}$);
Find the slowest segment $S[k,k']$;
while $v^* > V[k,k'] > 0$ and exist active sensors do
    $v_{opt}(d) \leftarrow V[k,k']$ for all available cell in $S[k,k']$;
    Mark all cells in $S[k,k']$ as unavailable;
    Mark all sensors belonging to $S[k,k']$ as inactive;
    $V[k,k'], \forall k, k' \leftarrow$ Compute-Segment-Speed($\mathbb{S}$);
    Find the slowest segment $S[k,k']$;
for $k \leftarrow 0$ to $m$ do
    if $U_C(k) = 1$ then
    $v_{opt}(d) \leftarrow v^*, d \in (b_k, b_{k+1})$;
    $E_f \leftarrow E_f + v_{opt}((b_k + b_{k+1})/2)(b_{k+1} - b_k)$;
return $E_f$;
```

Proof: We prove by contradiction. Suppose that $v_{opt}(d)$ is not equal to $V[k,k']$ in some cells contained in $S[k,k']$. We claim that there must exist at least one available cell $C_u = (b_u, b_{u+1}), \subseteq (b_k, b_{k'})$, such that UAV optimal speed $v_{opt}(d) < V[k,k']$ in cell C_u. Otherwise, we will have $v_{opt}(d) \geq V[k,k']$ in entire $S[k,k']$, which implies that the time UAV spends flying from b_k to $b_{k'}$ is less than $T[k,k']$. This contradicts the completion constraint (6). Since $v_{opt}(d) < V[k,k']$ in (b_u, b_{u+1}), we extend (b_u, b_{u+1}) to a larger interval. Let $(b_w, b_{w'}) (\supseteq (b_u, b_{u+1}))$ be the longest interval in which $v_{opt}(d) < V[k,k']$ in every cell.

Clearly, $v_{opt}(d)$ decreases at point b_w and increases at point $b_{w'}$; otherwise, $(b_w, b_{w'})$ can be extended to an even longer interval. By Lemma 5, b_w must be an entering point and $b_{w'}$ must be an exiting point. Hence $(b_w, b_{w'})$ is a segment, $S[w,w']$. Meanwhile, since $v_{opt}(d) < V[k,k'] \leq v^*$, according to Lemma 6, the time UAV spendes flying over $S[w,w']$ is equal to $T[w,w']$. Then, we must have $(b_{w'} - b_w) < V[k,k']T[w,w']$ because $v_{opt}(d) < V[k,k']$. Therefore $(b_{w'} - b_w)/T[w,w'] < V[k,k']$. However, $(b_{w'} - b_w)/T[w,w']$ is the effective speed of segment $S[w,w']$, and it is less than $V[k,k']$, which contradicts the supposition that $S[k,k']$ is the slowest segment. □

D. Slowest Segment First Policy

The core idea of SSF policy is that it identifies *the slowest segment* $S[k,k']$ among all segments, and marks all cells in $S[k,k']$ as *unavailable* and all sensors in that segment as *inactive*. The process repeats with a reduced set of sensors and cells. This iteration continues until the set is empty or the slowest effective speed $V[k,k']$ meets or exceeds v^*. For details, Algorithm 1 takes the necessary sensor information as input and returns the optimal horizontal energy consumption. For simplicity, we denote these pieces of sensor information (such as l_i and r_i) by $\mathbb{S}$.

Theorem 2: Algorithm 1 computes the optimal speed function $v_{opt}(d)$ for sensors $s_1, s_2, \ldots, s_n$.

Proof: To prove this theorem, we need to show that during any iteration of the **while** loop, if $S[k,k']$ is the slowest segment and $V[k,k'] < v^*$, then $v_{opt}(d) = V[k,k'], d \in S[k,k']$.

Algorithm 2: Compute-Segment-Speed($\mathbb{S}$).

```
$D[k,k'] \leftarrow 0, T[k,k'] \leftarrow 0, V[k,k'] \leftarrow +\infty, \forall k,k'$;
$K_{enter} \leftarrow \{k \mid b_k = l_i, \forall i\}$;
$K_{exit} \leftarrow \{k \mid b_k = r_i, \forall i\}$;
if $b_{k'} > b_k$ then
  for $k \in K_{enter}$ and $k' \in K_{exit}$ do
    for $u \leftarrow k$ to $k'-1$ do
      if $U_C(u) = 1$ then
        $D[k,k'] \leftarrow D[k,k'] + b_{u+1} - b_u$;
      for $i \leftarrow 1$ to $n$ do
        if $U_S(i)$ and $l_i \geq b_k$ and $r_i \leq b_{k'}$ then
          $T[k,k'] \leftarrow T[k,k'] + t_i$;
    if $T[k,k'] > 0$ then
      $V[k,k'] \leftarrow D[k,k']/T[k,k']$;
return $V[k,k'], \forall k,k'$;
```

We will use mathematical induction on the iterations. Theorem 1 confirms that Theorem 2 holds for the first iteration. Suppose that Theorem 2 is valid for the x-th iteration, where $x \geq 1$. Our goal is to prove that it also holds for the $(x+1)$-th iteration.

Suppose that $v_{opt}(d)$ is not equal to $V[k,k']$ in some available cells contained in the slowest segment $S[k,k']$ in x-th iteration. We claim that there must exist at least one available cell $C_u = (b_u, b_{u+1}), \subseteq (b_k, b_{k'})$, such that UAV optimal speed $v_{opt}(d) < V[k,k']$ in cell C_u. Otherwise, we will have $v_{opt}(d) \geq V[k,k']$ in every available cell contained in $S[k,k']$, which implies that the time UAV spends flying from b_k to $b_{k'}$ is less than $T[k,k']$. This contradicts the completion constraint (6). Thus, let $v_{opt}(d) < V[k,k']$ when $d \in (b_u, b_{u+1})$, we extend (b_u, b_{u+1}) to a larger interval. Let $(b_w, b_{w'}) (\supseteq (b_u, b_{u+1}))$ be the longest interval in which $v_{opt}(d) < V[k,k']$ in every cell. Note that the interval is allowed to include both available and unavailable cells, as Theorem 2 holds for the x-th iteration.

Clearly, $v_{opt}(d)$ decreases at point b_w and increases at point $b_{w'}$; otherwise, $(b_w, b_{w'})$ can be extended to a longer interval. By Lemma 5, b_w must be an entering point and $b_{w'}$ must be an exiting point. Hence, $(b_w, b_{w'})$ is a segment. Meanwhile, since $v_{opt}(d) < V[k,k'] \leq v^*$, according to Lemma 6, the time UAV spends flying over $S[w,w']$ is equal to $T[w,w']$. Note that $S[k,k']$ may contain some inactive sensors or unavailable cells, as the data of some sensors may be scheduled to be collected in during the first k iterations. In the $(x+1)$-th iteration, $D[w,w']$ and $T[w,w']$ are the effective distance and time of $S[w,w']$, respectively. That means the UAV must spend $T[w,w']$ flying over the distance $D[w,w']$. We have $D[w,w'] < V[k,k']T[w,w']$, since $v_{opt}(d) < V[k,k'], \forall d \in (b_w, b_{w'})$. Therefore, $V[w,w'] = D[w,w']/T[w,w'] < V[k,k']$, which contradicts the precondition that $S[k,k']$ is the slowest segment. □

Theorem 2 demonstrates that Algorithm 1 achieves the optimal speed schedule for the sensor set $\mathbb{S}_1$. The key difference between $\mathbb{S}_1$ and $\mathbb{S}$ is that $\mathbb{S}$ includes data transmission ranges at all altitudes, whereas $\mathbb{S}_1$ includes those at the UAV's flight altitude h. Algorithm 2 details the procedure for computing effective speed and identifying the slowest segment.

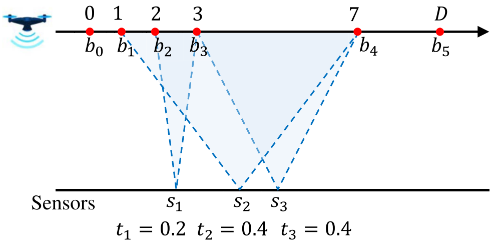


Fig. 6. An example of data collection with $n = 3$.

TABLE II
ALGORITHM PROCESS FOR EXAMPLE IN FIG. 6

Iteration	$S[k,k']$	$b_3 = r_1 = 3$	$b_4 = r_2 = r_3 = 7$
1st	$b_1 = l_2 = 1$	$D[1,3] = 2$ $T[1,3] = 0.2$ $\boldsymbol{V[1,3] = 10}$	$D[1,7] = 6$ $T[1,7] = 1$ $V[1,7] = 6$
	$b_2 = l_1 = 2$	$D[2,3] = 1$ $T[2,3] = 0.2$ $V[2,3] = 5$	$D[2,7] = 5$ $T[2,7] = 0.6$ $V[2,7] = 8.33$
	$b_3 = l_3 = 3$	$S[3,3] = \varnothing$	$D[3,7] = 4$ $T[3,7] = 0.4$ $V[3,7] = 10$
2nd	$b_1 = l_2 = 1$	$D[1,3] = 1$ $T[1,3] = 0$ $V[1,3] = +\infty$	$D[1,7] = 5$ $T[1,7] = 0.8$ $\boldsymbol{V[1,7] = 6.25}$
	$b_2 = l_1 = 2$	$D[2,3] = 0$ $T[2,3] = 0$ $V[2,3] = +\infty$	$D[2,7] = 4$ $T[2,7] = 0.4$ $V[2,7] = 10$
	$b_3 = l_3 = 3$	$S[3,3] = \varnothing$	$D[3,7] = 4$ $T[3,7] = 0.4$ $V[3,7] = 10$

To illustrate our algorithm, we present an example in Fig. 6, assuming $v^* = 6$ (m/s). We can obtain the optimal speed scheduling by calling function *Slowest-Segment-First*($\mathbb{S}$). Details are given in Table II, with the effective speed of the slowest segment highlighted in bold. Initially, all cells and sensors are marked as available and active, respectively. In the first iteration, the slowest segment is $S[2,3] = (2,3)$, so we set $v_{opt}(d) = 5$ for $d \in (2,3)$. Then C_2 is marked unavailable, and s_1 is marked inactive. In the second iteration, the slowest segment is $S[1,4] = (1,7)$, with $V[1,4] = \frac{25}{4} > v^*$, causing the **while** loop to terminate. Finally, for all remaining available cells, we set $v_{opt}(d) = v^* = 6$ for $d \in (0,2) \cup (3,D)$.

Although we have now obtained the optimal speed scheduling for offline USS, it is still necessary to determine the collection function $I(d)$, which instructs the UAV on when to collect data from specific sensor. Inspired by Earliest Deadline First policy, we propose a scheduling strategy that prioritizes data collection from the active sensor with the nearest (i.e., smallest) exiting point. For brevity, the detailed formal description of this policy is omitted.

E. Complexity Analysis of SSF Policy

In this subsection, we analyze the time complexity of the SSF algorithm in Algorithm 1. The **while** loop in Algorithm 1 terminates when all sensors become inactive. In each iteration, at least one active sensor becomes inactive as a segment must contain active sensor(s). Therefore, the loop will be executed at most n times, where n is the number of sensors. In Algorithm 2, the upper bound of the size of both K_{enter} and K_{exit} is n. Meanwhile, the maximum iteration times of both **for** loops are also n. Therefore, the complexity of SSF algorithm is $O(n^4)$.

V. SSF-Based Ant Colony Optimization Algorithm

In this section, we introduce the SSF policy to optimally address the offline USS, which is a special case of the offline JUSAS. We explore the general JUSAS that $g(d)$ is not a constant function, and propose the SSF-ACO algorithm to solve it. The overview of the SSF-ACO algorithm is as follows: (1) Convert problem $\mathbf{P}$ into $\mathbf{P}'$ with flight path discretization method; (2) Construct the flight scheduling graph (FSG) on $\mathbf{P}'$ to reformulate the problem as a shortest-path-type problem; (3) Solve $\mathbf{P}'$ based on the FSG combined with ACO algorithm.

A. Problem Conversion

To ensure UAV safety, its flight trajectory must maintain stability, avoiding frequent altitude adjustments. We divide the flight path into N evenly spaced intervals within $[0, D]$ using a granularity of $\Delta d = D/N$. The UAV is only allowed to adjust altitude at two sides of each interval, i.e., $(\varphi - 1)\Delta d$ and $\varphi\Delta d$, $\varphi = 1, 2, \ldots, N$. We take coordinate points at all flight altitudes h_j, $j = 1, 2, \ldots, M$, resulting in a total of $(N+1)M$ points, each of which is represented as $(\varphi\Delta d, h_j)$. By this way, we can convert problem $\mathbf{P}$ into $\mathbf{P}'$:

$$\mathbf{P}': \quad \min_{f(d),g(d),I(d)} \; E = \sum_{e=1}^{|\mathcal{G}|} \varepsilon(h_{\omega_e}, h_{\omega_{e-1}}) + E_f, \tag{14}$$

s.t. (9a), (9c)–(9e),

$$\mathcal{G} \subseteq \{\Delta d, 2\Delta d, \ldots, (N-1)\Delta d\}, \Delta d = D/N, \tag{14a}$$

$$h_{\omega_e} \in \{h_1, \ldots, h_M\}, \forall e \in \{0, 1, \ldots, |\mathcal{G}|\}, \tag{14b}$$

where E_f represents the horizontal energy cost calculated using Algorithm 1. Eq. (14a) indicates that the UAV can only adjust its altitude at the boundaries of the intervals, and (14b) specifies the range of values for h_{ω_e}, consistent with (9b).

Remark: Note that this flight path discretization is only used to limit the UAV's altitude-adjusting points, and does not compromise the optimality of Algorithm 1, as the coordinates of boundary points remain precise.

B. Flight Scheduling Graph Construction

In this subsection, we construct a *Flight Scheduling Graph* (FSG) and reformulate $\mathbf{P}'$ as a shortest-path-type problem, which can be efficiently solved by the proposed SSF-ACO algorithm in the next subsection.

To construct the FSG, we first treat each discretized coordinate point $(\varphi\Delta d, h_j)$, introduced in the previous subsection, as a vertex in the graph. For simplicity, we denote the vertex by (φ, j), as shown in Fig. 7. Every horizontally adjacent pair of vertices, (φ, j) and $(\varphi + 1, j')$, is connected by a directed edge, denoted by $< j, j' >_\varphi$, with a weight of $\varepsilon(h_j, h_{j'})$, which represents the vertical energy consumption for transitioning from altitude h_j to $h_{j'}$. Since the UAV flies from the starting point $x = 0$ to the destination point $x = D$, the graph consists of M starting vertices $(0, j)$ and M destination vertices (N, j) in the graph. To streamline algorithm processing, an additional dummy vertex $(-1, 1)$ as is introduced as the dummy starting vertex, with directed edges connecting $(-1, 1)$ to each of the M original starting vertices $(0, j)$, all assigned a weight of 0.

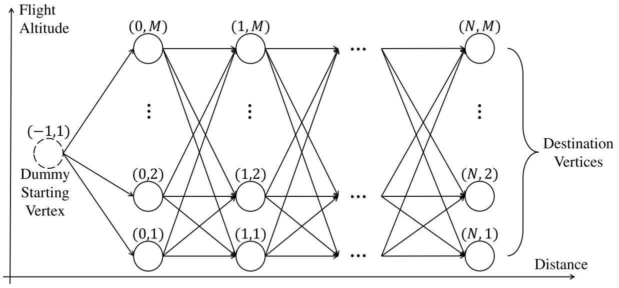


Fig. 7. Flight scheduling graph. The UAV flies from the dummy starting vertex to any of the M destination vertices in the right side. Vertices on different lines represent different flight altitudes.

We now have reformulated the original problem $\mathbf{P}'$ as a shortest-path-type problem on the FSG. The cost of a path corresponding to the total energy consumption of the UAV, which is exactly the sum of horizontal and vertical energy consumption. The vertical one equals to the sum of the weights of all edges along the path, while the horizontal one is calculated by Algorithm 1 after obtaining a complete path.

C. SSF-Based Ant Colony Optimization Algorithm

Based on the FSG, in this subsection, we design the SSF-ACO algorithm to solve the problem $\mathbf{P}'$.

According to [54], [55], the ACO algorithm is widely employed to address shortest path problems in graphs. In ACO, the ants iteratively construct solutions by probabilistically selecting edges based on pheromone trails and heuristic information.

The core idea of SSF-ACO is that, focusing on the constructed FSG, each ant departs from the virtual starting vertex, then arrives at any of destination vertices along the directed edges. The weights of edges in FSG, acting as the heuristic value, together with the pheromone matrix, guide the ants in selecting the edges. In each iteration, the solution with lowest energy consumption is selected to enhance the pheromone concentration, after naturally evaporating.

As shown in Algorithm 3, a detailed description of SSF-ACO is provided. After initializing variables, including the pheromone matrix τ, the heuristic matrix η, the maximum iteration times $iter_{\max}$, and the number of ants Z, the algorithm executes iterations. Here, Z is set to 20 to balance solution quality and computational efficiency. In each iteration, every ant traverses in the graph independently to generate an altitude scheduling function $g(d)$ by invoking Algorithm 4. Specifically, in Algorithm 4, each ant gets its current position (φ, j), and moves to next vertex randomly from the set of candidate vertices

Algorithm 3: SSF-ACO($\mathbb{S}$).

```
Initialize pheromone matrix $\tau$, heuristic matrix $\eta$;
$E_{opt} \leftarrow +\infty$, $iter \leftarrow 0$, $Z \leftarrow 20$;
while $iter < iter_{\max}$ do
    for $z \leftarrow 1$ to $Z$ do
        $g(d) \leftarrow$ Generate-Altitude-Schedule($\mathbb{S}$), $\mathbb{S}_z \leftarrow \varnothing$;
        for $i \leftarrow 1$ to $n$ do
            $g(d)$ intersects the data transmission range boundary of $s_i$ at $l_{i,g(d')}$ and $r_{i,g(d'')}$;
            $\mathbb{S}_z \leftarrow \mathbb{S}_z \cup \{l_{i,g(d')}, r_{i,g(d'')}\}$;
        Add $\{t_i, \forall i\}$ in $\mathbb{S}$ to $\mathbb{S}_z$;
        $E_f \leftarrow$ Slowest-Segment-First($\mathbb{S}_z$);
        Calculate $E_v$ by (8);
        if $E_f + E_v < E_{opt}$ then
            $E_{opt} \leftarrow E_f + E_v$, $g_{opt}(d) \leftarrow g(d)$, $v_{opt}(d) \leftarrow v(d)$;
    $\tau_{\varphi,j,j'} \leftarrow (1-\rho)\tau_{\varphi,j,j'}, \forall \varphi, j, j'$; // Pheromone evaporation
    Enhance pheromone $\tau_{\varphi,j,j'}$ based on $g_{opt}(d)$;
return $\{v_{opt}(d), g_{opt}(d), E_{opt}\}$;
```

Algorithm 4: Generate-Altitude-Schedule($\mathbb{S}$).

```
$pos \leftarrow (-1, 1)$, $\Delta d \leftarrow D/N$;
for $\varphi \leftarrow -1$ to $N-1$ do
    $J \leftarrow \{1, 2, \ldots, M\}$;
    // $(\varphi, j)$ is the ant's current position $pos$
    Compute heuristic value $\eta_{\varphi,j,j'}, \forall j' \in J$;
    Compute probability $P^{sel}_{\varphi,j,j'}, \forall j' \in J$;
    $pos \leftarrow$ Roulette-Wheel-Selection($J$);
    Record altitude schedule $g(d)$;
return $g(d)$;
```

J by a classic Roulette Wheel Selection method [56]. The selected probability $P^{sel}_{\varphi,j,j'}$ of vertex $(\varphi+1, j')$ is calculated by the following equation,

$$P^{sel}_{\varphi,j,j'} = \begin{cases} \frac{(\tau_{\varphi,j,j'})^{\alpha}(\eta_{\varphi,j,j'})^{\beta}}{\sum_{j''\in J}(\tau_{\varphi,j,j''})^{\alpha}(\eta_{\varphi,j,j''})^{\beta}} & j' \in J, \\ 0 & \text{otherwise}, \end{cases} \tag{15}$$

where τ is the 3D *pheromone matrix*, representing the experience accumulated by the ants during the exploration, $\tau_{\varphi,j,j'}$ represents the concentration of pheromone on edge $< j, j' >_{\varphi}$, η is a 3D *heuristic matrix* to guide the ants in selecting vertices, $\eta_{\varphi,j,j'}$ is the dynamically-computed heuristic of edge $< j, j' >_{\varphi}$, α and β are the exponential parameters to control the influence of pheromone and heuristic, respectively. Specifically, in *Roulette-Wheel-Selection*, vertex $(\varphi+1, j')$ is selected if $\sum_{j''=0}^{j'-1} P^{sel}_{\varphi,j,j''} \leq \text{rand}() < \sum_{j''=0}^{j'} P^{sel}_{\varphi,j,j''}$, where rand() generates a random number in [0,1).

Based on $g(d)$, we can obtain the information including entering and exiting points of each sensor, and call Algorithm 2 to compute the horizontal energy consumption E_f. Then, the vertical energy consumption E_v is computed by (8). Next, the solution, i.e., $\{v_{opt}(d), g_{opt}(d), E_{opt}\}$, is updated if $E_f + E_v$ is less than the current minimum consumption E_{opt}. In the end of each iteration, the algorithm maintains the pheromone matrix by evaporation and enhance operation. The algorithm terminates when the maximum iteration times $iter_{\max}$ is reached.

Algorithm 5: SSF-ACO-Online.

```
$\mathbb{S}_{ACT} \leftarrow \varnothing$, $v \leftarrow v^*$, $h \leftarrow h_1$;
while the UAV has not reached destination do
    if the UAV discovers a new sensor $s_i$ then
        $U_S(i) \leftarrow 1$, $\mathbb{S}_{ACT} \leftarrow \mathbb{S}_{ACT} + \{s_i\}$;
        Call SSF-ACO($\mathbb{S}_{ACT}$) to obtain the schedule;
    if a sensor $s_{i'} \in \mathbb{S}_{ACT}$ completes data collection then
        $U_S(i') \leftarrow 0$, $\mathbb{S}_{ACT} \leftarrow \mathbb{S}_{ACT} - \{s_{i'}\}$;
```

VI. Online Strategy

The previously proposed SSF-ACO algorithm assumes complete knowledge of sensor locations and their transmission ranges. However, in many practical scenarios, such information may only become available when the UAV flies within the detection range of sensors. This section presents an online version of SSF-ACO (SSF-ACO-Online) algorithm, an adaptation of our approach for these realistic settings.

Specifically, in the online scenario, each sensor has an additional *control communication range*, which covers a larger area than the data transmission range but operates at a slower transmission rate. The UAV keeps broadcasting 'Probe' message during the flight to detect sensors. Once receiving the 'Probe' message, the sensor sends back an 'ACK' message, which includes its position, the amount of data, and transmission characteristics. Note that each sensor sends 'ACK' message only once. This partial information availability fundamentally changes the nature of our optimization problem, requiring real-time decision-making based on local information.

The proposed online strategy is detailed in Algorithm 5. This algorithm maintains an extra *active sensor list*, denoted as $\mathbb{S}_{ACT}$, which includes sensors from which data has not yet been fully collected. Initially, $\mathbb{S}_{ACT} = \emptyset$. The UAV begins its operation at initialized altitude with speed v^*, continuously broadcasting messages to discover new sensors. Upon discovering a new sensor, the UAV performs the following steps: (1) Updates the data transmission range by setting its current horizontal position to $x = 0$, and adjusts the residual data transmission time; (2) Adds the newly-discovered sensor into $\mathbb{S}_{ACT}$, with information be recorded; (3) Invokes the offline SSF-ACO algorithm to rearrange the schedule. If the UAV completes data collection for sensor s_i, the algorithm marks this sensor as inactive (i.e., $U_S(i) = 0$), and then remove s_i from active sensor list $\mathbb{S}_{ACT}$.

VII. Simulation

In this section, the altitude-specific transmission range model of sensor is introduced. We conducted extensive simulation experiments to evaluate the performance of SSF-ACO and SSF-ACO-Online. The latency analysis of SSF-ACO-Online is provided.

TABLE III
PARAMETER SETTINGS

Parameter	Value Range	Default Value
n	{5, 10, 15, 20, 25, 30, 35, 40}	15
q	{0, 5, 10, 15, 20, 25, 30, 35, 40}	10
δ	{1, 3, 5, 10, 15, 20, 25, 30}	10
$C_{H,max}$	{40, 60, 80, 100, 120, 140}	120
$C_{W,max}$	{30, 35, 40, 45, 50, 55, 60}	45
$C_{T,max}$	{0.5, 1, 1.5, 2, 2.5, 3}	2
μ_{max}	{1, 2, 3, 4, 5, 6}	3
C_R	{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}	2

A. Altitude-Specific Transmission Range Model

Based on the widely-used approximate funciton of LoS probability, the ground-to-UAV transmission range varies with flight altitude [39]. Higher altitudes expand the UAV's field of view, increasing the likelihood of establishing a LoS link. However, this also reduces the coverage radius. Here, we use (16) to represent this radius-altitude relationship as a deterministic coverage abstraction, rather than as a probabilistic function:

$$\mu\left(\frac{x-x_0{}^2}{C_W}\right)+\left(\left(\frac{x-x_0}{C_W}\right)^2+\left(\frac{y}{C_H}\right)^2\right)^2$$
$$+2\frac{y}{C_H}\left(\left(\frac{x-x_0}{C_W}\right)^2+\left(\frac{y}{C_H}\right)^2\right)=0, \qquad (16)$$

where x_0 is the deployment position of the snesor, μ is the swell coefficient influencing the shape of the data transmission range. The width and height of the range are linearly related to the coefficients C_W and C_H, respectively. And the quantitave relationship can be found in Appendix F, available online of the supplementary material. Although real-world sensor transmission ranges lack sharply defined boundaries, a deterministic coverage abstraction offers practical benefits by aligning with the radius-altitude variation trend and improving computational tractability.

B. Simulation Settings

In the simulations, we set the power-speed function to be $p(v) = 0.07v^3 + 0.0391v^2 - 13.196v + 390.95$ (W) [25]. Let the optimal forward speed v^* and the maximum forward speed $v_{\max}$ be 13.98m/s and 18m/s, respectively. The UAV can fly at an altitude between $h_{\min} = 50$ m and $h_{\max} = 300$ m. We set $\varepsilon(h, h') = q|h - h'|$, where q is the constant coefficient of vertical energy consumption [26], [27].

To demonstrate the typical performance metrics of our algorithms, we adopt a univariate approach, adjusting one parameter at a time while keeping the others at their default values. Table III outlines the ranges and default values for key simulation parameters. For every parameter setting in our simulation, we randomly generate 100 instances of sensor deployment and use the mean results for comparisons. The parameter δ indicates the smallest increment for altitude changes, requiring that changes be multiples of δ. The altitude coefficient C_H follows a uniform distribution $C_H \sim \text{Uniform}(30, C_{H,max})$, while the width coefficient C_W follows $C_W \sim \text{Uniform}(10, C_{W,max})$, where $C_{H,max}$ and $C_{W,max}$ are the upper bound of these coefficients, respectively. $C_{T,max}$ represents the upper limit for the coefficient of required data collection time. Since sensors monitor environmental or infrastructural states along a line, the amount of data collected, i.e., the time required, is linearly related to the sensor's data transmission range radius. This linear relationship is captured by $C_T \sim \text{Uniform}(0, C_{T,max})$ The swell coefficient of the data transmission range, denoted as μ, is uniformly distributed as $\mu \sim \text{Uniform}(0, \mu_{\max})$. Lastly, C_R is the ratio of the control communication range to the data transmission range. When $C_R = 1$, control communication ranges with the corresponding data transmission ranges, preventing the UAV from acquiring sensor information until it enters the data transmission range.

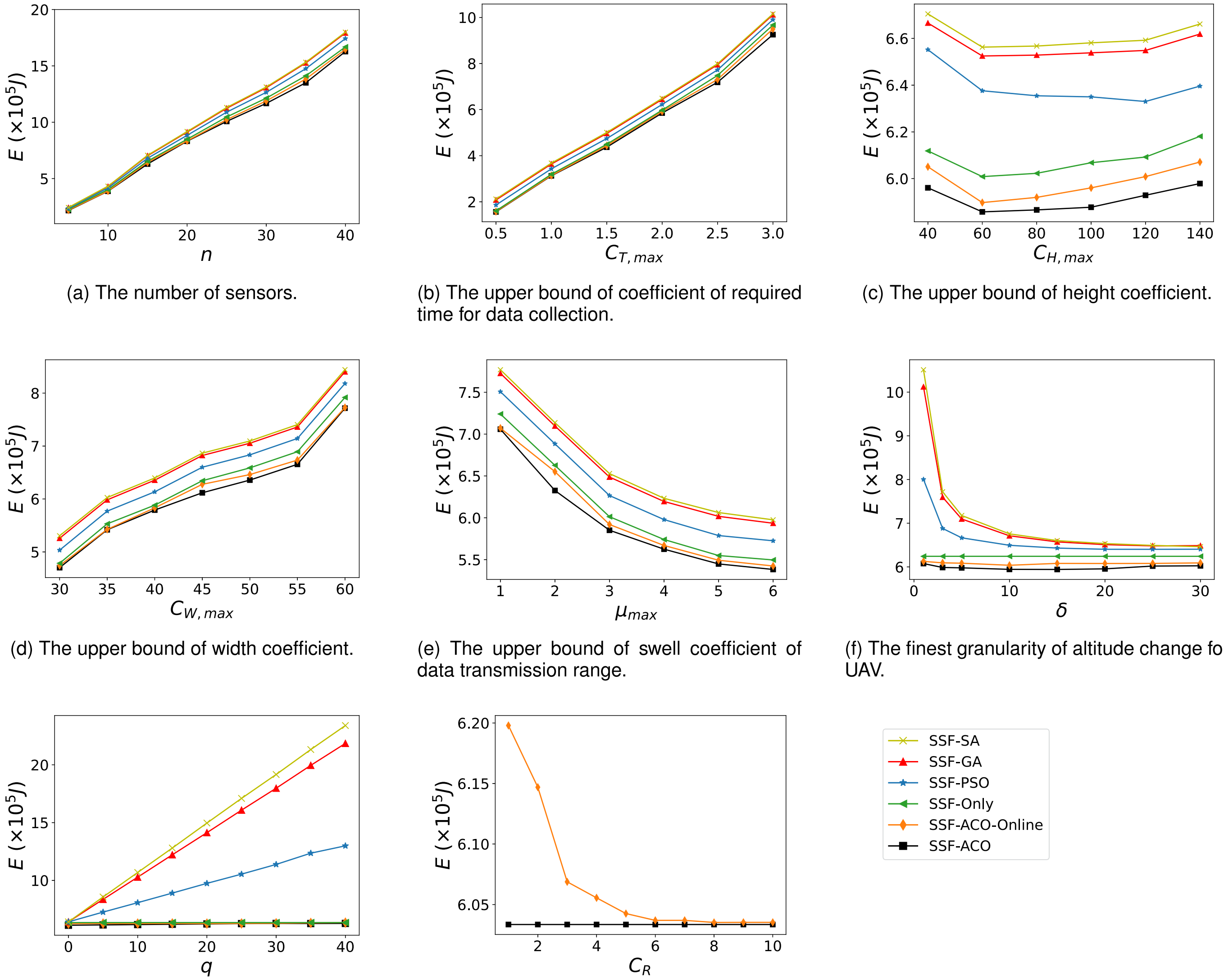
C. Baseline Algorithms

Since there are no existing algorithms that can be directly applied to the JUSAS, we design a greedy algorithm and three heuristic algorithms based on SSF policy for comparison. Below is a brief introduction to these algorithms.

- *SSF-Only:* In this method, the UAV flies at a constant altitude during the whole flight, and then employs the SSF algorithm to establish the speed schedule. The flight altitude is selected as the one with the lowest energy consumption from all altitudes.
- *SSF-PSO:* This method combines SSF policy with PSO algorithm extended from [57]. Each particle's real-valued position is mapped to a path in the FSG.
- *SSF-GA:* This method integrates SSF policy with Genetic Algorithm (GA) extended from [58]. In this approach, solutions are represented as individuals within the GA and are refined through successive generations.
- *SSF-SA:* This method couples the SSF policy with the classic SA algorithm. The real-valued solution in SA is mapped to a path in the FSG.

D. Simulation Results

We evaluate the performance of all algorithms under various parameter settings, as shown in Fig. 8. **(1)** In all eight simulation experiments, the proposed SSF-ACO outperforms the other compared algorithms, and SSF-ACO-Online achieves the second-best result, demonstrating its effectiveness in online scenarios. Quantitatively, SSF-ACO reduces by 2.94%, 12.56%, 18.45% and 19.31% on average compared to the baseline SSF-Only, SSF-PSO, SSF-GA, and SSF-SA, respectively, while SSF-ACO-Online maintains an average energy cost of no more than 101.24% compared to SSF-ACO. **(2)** Observe that SSF-Only algorithm also performed well, highlighting the SSF policy's advantages in speed scheduling. In contrast, SSF-PSO, SSF-GA, and SSF-SA show noticeably weaker performance, likely due to their inefficiency in altitude scheduling. This inefficiency resulted in frequent altitude adjustments, leading to increased vertical energy consumption. **(3)** ACO's inherent suitability for shortest-path-type problems enhances its effectiveness. We utilize vertical consumption for single altitude adjustments (i.e., edge weights in the FSG) as heuristic value in ACO, promoting



(a) The number of sensors.

(b) The upper bound of coefficient of required time for data collection.

(c) The upper bound of height coefficient.

(d) The upper bound of width coefficient.

(e) The upper bound of swell coefficient of data transmission range.

(f) The finest granularity of altitude change for UAV.

(g) The vertical consumption coefficient.

(h) The radius ratio of the control communication range to the data transmission range of a sensor.

Fig. 8. Algorithm performance comparisons in UAV energy consumption.

stable flight trajectories and reducing vertical energy consumption.

From Fig. 8(a), as expected, energy consumption increases with the number of sensors across all algorithms. This trend is explained by the fact that a larger number of sensors leads to a higher energy consumption due to the increased data collection requirements. As shown in Fig. 8(b), it reveals a similar trend, where energy consumption rises with $C_{T,max}$. This is because a larger amount of data requires the UAV to fly at a slower speed to entirely collect all the data, leading to more energy consumption. In Fig. 8(c), energy consumption initially decreases and then increases as $C_{H,max}$ rises. When the widest altitude falls between $h_{\min}$ and $h_{\max}$, the UAV can efficiently collect data, reducing energy consumption. Instead, when $C_{H,max}$ is too large (small), resulting in the widest altitude being above $h_{\max}$ (below $h_{\min}$), the range that can be utilized becomes smaller, thus increasing energy consumption. Fig. 8(d) shows that energy consumption rises with $C_{W,max}$. Given that the amount of data collected is linearly related to the width of the data transmission range, a larger $C_{W,max}$ corresponds to a greater volume of data to be collected. Hence, the total flight time of the UAV increases, its average speed decreases, and energy consumption rises. From Fig. 8(e), energy consumption declines as $\mu_{\max}$ increases. The swell coefficient μ controls the shape of the data transmission range and, in combination with C_W, determines its width. When other parameters remain constant, a larger μ results in a 'thinner' data transmission range. Thus, a larger μ leads to a narrower width, a smaller amount of data to be collected, and thus lower energy consumption. Fig. 8(f) illustrates varying performances of algorithms under different UAV altitude adjustment granularity δ. The SSF-Only algorithm remains unaffected by δ since it does not adjust altitude. For SSF-ACO and SSF-ACO-Online, energy consumption initially decreases with increased δ before rising again. Both excessively

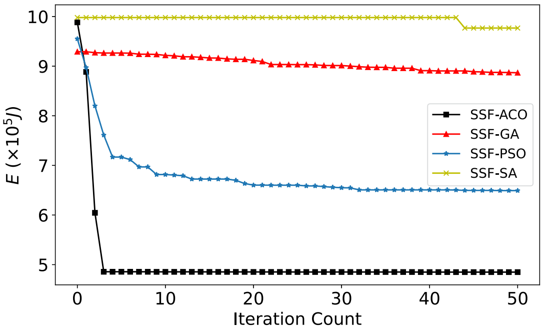


Fig. 9. Variation of optimal energy consumption over the first 50 iterations for SSF-ACO, SSF-GA, SSF-PSO, and SSF-SA.

TABLE IV
COMPUTATION TIME OF SSF-ACO AND SSF-ACO-ONLINE COMPARED WITH UAV FLIGHT DURATION IN ONLINE SCHEDULING (IN SECONDS)

n	5	10	15	20	25	30	35	40
T_f	513.73	1107.11	1502.68	2084.67	2675.47	2997.83	3519.72	4119.79
T_{off}	1.58	5.60	12.01	19.11	32.32	51.01	64.60	92.73
T_{on}	0.99	1.81	2.91	3.09	3.49	3.87	5.04	5.80
T_{on}/T_f	0.19%	0.16%	0.19%	0.15%	0.13%	0.13%	0.14%	0.14%

fine and coarse δ make energy efficiency, and the optimal δ varies by algorithm. Fine δ leads to a vast search space that can disrupt optimal solution finding, while coarse δ restricts optimization, especially in altitude scheduling. For SSF-PSO, SSF-GA, and SSF-SA, energy consumption decreases with increasing δ, which appears counterintuitive. This may might be attributed to these methods' limited search capabilities in altitude scheduling, where fine δ creates an extensive search space. In Fig. 8(g), energy consumption for all methods, except SSF-Only, increases as the vertical consumption coefficient q rises. The SSF-Only method remains unaffected by q because the UAV maintains a constant altitude. When $q = 0$, indicating no energy consumption for altitude adjustments, all methods show similar energy consumption. As q increases, SSF-ACO and SSF-ACO-Online exhibit smaller increases in energy consumption, indicating their superior stability and adaptability to variations in q by reducing altitude adjusting. In contrast, SSF-PSO, SSF-GA, and SSF-SA demonstrate larger increases, reflecting their lesser adaptability to changes in q.

In addition, we conduct simulation experiments to evaluate the performance of SSF-ACO-Online. We vary the control communication range radius C_R as multiples of the data transmission range radius and compared the online and offline algorithms, as shown in Fig. 8(h). It is evident that the offline algorithm is unaffected by C_R, as the UAV has access to all information before the task execution. For SSF-ACO-Online, when $C_R = 1$, the UAV consumes the most energy because it cannot acquire sensor information before entering the data transmission range, which coincides with the control communication range. As the control communication range expands, the UAV's energy consumption significantly decreases and gradually approaches the results of SSF-ACO.

To further analyze the convergence performance of the proposed algorithms, we compared the optimal energy consumption across the first 50 iterations for each algorithm, using the default parameters in Table III. The results are shown in Fig. 9. All algorithms are randomly initialized. SSF-ACO converged the fastest, reaching best result in just a few iterations, followed by SSF-PSO. Moreover, SSF-ACO consistently yielded significantly lower energy consumption than the other algorithms, making it more suitable for online scheduling applications. The advantage of SSF-ACO is due to its partial solution construction process, which distinguishes it from other algorithms that refine existing solutions. During the solution construction process, ACO's heuristic factors guide the selection of the next flight altitude based on the current altitude, effectively balancing vertical and horizontal energy consumption for more efficient energy use throughout the UAV's flight path.

Finally, we analyze the computational latency of SSF-ACO-Online. Table IV reports the duration of UAV flights T_f under online problem, along with the computation time of SSF-ACO-Online T_{on}, and that of SSF-ACO T_{off} in the corresponding offline problem. Across varying numbers of sensors, the computation times of SSF-ACO-Online stay under 6 seconds and constitute no more than 0.19% of the corresponding UAV flight duration. This demonstrates the suitability of SSF-ACO-Online for online applications. Since the active sensor list only maintains those sensors that have been discovered but not yet completely collected, though invoked SSF-ACO multiple times, SSF-ACO-Online has a cumulative computation time significantly shorter than that of SSF-ACO.

VIII. CONCLUSION

In this paper, we investigated the JUSAS problem with the aim of minimizing energy consumption while ensuring complete data collection from all sensors. Distinct from the existing works, we considered both a practical speed-energy model and an altitude-specific transmission range model, as well as complex overlapping range relationships. We first addressed a simplified case where the UAV flies at a constant altitude and proposed the optimal SSF policy. Then we transformed the JUSAS into a shortest-path-type problem and developed the SSF-ACO algorithm to solve it. Additionally, we extended our approach to the online setting and proposed the SSF-ACO-Online algorithm for scenarios where sensors information is only locally available. Extensive simulations showed that our proposed algorithms significantly outperform the baseline approaches, and the performance of the online algorithm is close to that of the offline. Future work will focus on energy-efficiency challenges in dynamic environments, such as mobile crowds or vehicles.

REFERENCES

[1] CAAC, "Notice on the issuance of the basic classification method for national airspace," 2023. Accessed: Jan. 09, 2025. [Online]. Available: https://www.caac.gov.cn/XXGK/XXGK/TZTG/202312/t20231221_222397.html

[2] S. Wandelt, S. Wang, C. Zheng, and X. Sun, "AERIAL: A meta review and discussion of challenges toward unmanned aerial vehicle operations in logistics, mobility, and monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, pp. 6276–6289, Jul. 2024.

[3] G. Sun, X. Zheng, J. Li, H. Kang, and S. Liang, "Collaborative WSN-UAV data collection in smart agriculture: A bi-objective optimization scheme," *ACM Trans. Sensor Netw.*, 2023, doi: 10.1145/3597025.

[4] H. Wei, B. Lou, Z. Zhang, B. Liang, F.-Y. Wang, and C. Lv, "Autonomous navigation for eVTOL: Review and future perspectives," *IEEE Trans. Intell. Veh.*, vol. 9, no. 2, pp. 4145–4171, Feb. 2024.

[5] G. Sun et al., "Joint task offloading and resource allocation in aerial-terrestrial UAV networks with edge and fog computing for post-disaster rescue," *IEEE Trans. Mobile Comput.*, vol. 23, no. 9, pp. 8582–8600, Sep. 2024.

[6] DJI Innovations, "DJI dock 3," 2025. Accessed: May 11, 2025. [Online]. Available: https://enterprise.dji.com/dock-3

[7] U. S. Congress, "H. R.302–115th Congress (2017-2018): FAA reauthorization act of 2018," 2018. Accessed: Nov. 29, 2024. [Online]. Available: https://www.congress.gov/bill/115th-congress/house-bill/302/summary/35

[8] CAAC, "CAAC issues the overall plan for the development of low-altitude flight service support system," 2018. Accessed: Nov. 29, 2024. [Online]. Available: https://www.caac.gov.cn/English/News/202305/t20230515_219191.html

[9] European Commission, "Commission Implementing Regulation (EU)2019/947 of 24 May 2019 on the rules and procedures for the operation of unmanned aircraft (text with EEA relevance.)," 2019. Accessed: Nov. 29, 2024. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX\%3A32019R0947

[10] M. Karrer and M. Chli, "Distributed variable-baseline stereo SLAM from two UAVs," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 82–88.

[11] C. Dalai, M. Singh, A. S. Kumara, W. Patel, H. Patil, and R. Maranan, "Monitoring water levels in rivers with optical fiber sensors," in *Proc. 5th Int. Conf. Innov. Trends Inf. Technol.*, 2024, pp. 1–6.

[12] X. Dai, Z. Xiao, H. Jiang, and J. C. S. Lui, "UAV-assisted task offloading in vehicular edge computing networks," *IEEE Trans. Mobile Comput.*, vol. 23, no. 4, pp. 2520–2534, Apr. 2024.

[13] N. Lin, H. Tang, L. Zhao, S. Wan, A. Hawbani, and M. Guizani, "A PDDQNLP algorithm for energy efficient computation offloading in UAV-assisted MEC," *IEEE Trans. Wireless Commun.*, vol. 22, no. 12, pp. 8876–8890, Dec. 2023.

[14] Y. Liao, Y. Song, S. Xia, Y. Han, N. Xu, and X. Zhai, "Energy minimization of RIS-Assisted cooperative UAV-USV MEC network," *IEEE Internet Things J.*, vol. 11, no. 20, pp. 32490–32502, Oct. 2024.

[15] L. Fu, Z. Zhao, G. Min, W. Miao, L. Zhao, and W. Huang, "Energy-efficient 3-D data collection for Multi-UAV assisted mobile crowdsensing," *IEEE Trans. Comput.*, vol. 72, no. 7, pp. 2025–2038, Jul. 2023.

[16] A. Zhang, D. Liu, T. Li, X. Cui, and K. Zheng, "BROAD: Joint beamwidth and path optimization for energy-efficient UAV-assisted data collection," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2024, pp. 1–6.

[17] G. Chen, X. B. Zhai, and C. Li, "Joint optimization of trajectory and user association via reinforcement learning for UAV-aided data collection in wireless networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 5, pp. 3128–3143, May 2023.

[18] J. Wang et al., "Generative AI based secure wireless sensing for ISAC networks," 2024, *arXiv:2408.11398*.

[19] 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.

[20] S. Wang, X. Li, X. Sun, and C. Zheng, "A railway accident prevention system using an intelligent pilot vehicle," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 6, pp. 5170–5188, Jun. 2024.

[21] L. Yang, M. R. McKay, and X. Wang, "Hypothesis test for leakage detection in water pipelines with high-dimensional sensor signals," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2023, pp. 1–5.

[22] M. Kocakulak and I. Butun, "An overview of wireless sensor networks towards Internet of Things," in *Proc. IEEE Comput. Commun. Workshop Conf.*, 2017, pp. 1–6.

[23] M. Sadeghi Ghahroudi, A. Shahrabi, S. M. Ghoreyshi, and F. A. Alfouzan, "Distributed node deployment algorithms in mobile wireless sensor networks: Survey and challenges," *ACM Trans. Sensor Netw.*, vol. 19, no. 4, pp. 1–26, 2023.

[24] 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.

[25] F. Shan, J. Luo, R. Xiong, W. Wu, and J. Li, "Looking before crossing: An optimal algorithm to minimize UAV energy by speed scheduling with a practical flight energy model," in *Proc. IEEE Conf. Comput. Commun*, 2020, pp. 1758–1767.

[26] K. Wu, M. Feng, C. Wu, Y. Lin, and S. Lu, "Trajectory planning for multi-rotor UAV based on energy cost model," in *Proc. 41st Chin. Control Conf.*, 2022, pp. 1–6.

[27] Y. Cao, A. Wang, G. Sun, and L. Liu, "Average transmission rate and energy efficiency optimization in UAV-assisted IoT," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2023, pp. 1–6.

[28] M. Hayajneh, M. Momani, and A. Al-ghazo, "UAV coverage path optimization for collecting fine-grained distribution of air samples," in *Proc. Int. Conf. Control, Automat. Diagnosis*, 2023, pp. 1–6.

[29] M. Shao et al., "Joint passive beamforming and elevation angle-dependent trajectory design for RIS-aided UAV-enabled wireless sensor networks," in *Proc. 19th Annu. IEEE Int. Conf. Sens., Commun., Netw*, 2022, pp. 488–496.

[30] S. Yang, J. Yu, Z. Zhang, and G. Zhao, "Cooperative path planning method based on road network constraints for vehicle-mounted multi-rotor UAV swarm," in *Proc. 42nd Chin. Control Conf.*, 2023, pp. 1779–1784.

[31] I. Shoer, B. K. Gunturk, H. F. Ates, and T. Baykas, "Predicting path loss distributions of a wireless communication system for multiple base station altitudes from satellite images," in *Proc. IEEE Int. Conf. Image Process.*, 2022, pp. 2471–2475.

[32] Y. Li, H. Zhang, K. Long, and A. Nallanathan, "Exploring sum rate maximization in UAV-based multi-IRS networks: IRS association, UAV altitude, and phase shift design," *IEEE Trans. Commun.*, vol. 70, no. 11, pp. 7764–7774, Nov. 2022.

[33] Y. Deng, H. Zhang, X. Chen, and Y. Fang, "UAV-assisted multi-access edge computing with altitude-dependent computing power," *IEEE Trans. Wireless Commun.*, vol. 23, no. 8, pp. 9404–9418, Aug. 2024.

[34] M. Hui, J. Chen, L. Yang, L. Lv, H. Jiang, and N. Al-Dhahir, "UAV-Assisted mobile edge computing: Optimal design of UAV altitude and task offloading," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 13633–13647, Oct. 2024.

[35] S. Li et al., "Coverage maximization of heterogeneous UAV networks," in *Proc. IEEE 43rd Int. Conf. Distrib. Comput. Syst.*, 2023, pp. 120–130.

[36] 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.

[37] Y. Qin, Z. Zhang, X. Li, W. Huangfu, and H. Zhang, "Deep reinforcement learning based resource allocation and trajectory planning in integrated sensing and communications UAV network," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 8158–8169, Nov. 2023.

[38] H. Ni et al., "Impacts of flight altitude and UAV posture on the UAV-to-ground channel gain," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2023, pp. 1–6.

[39] 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.

[40] 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. 10991–11005, Oct. 2022.

[41] M. Samir, C. Assi, S. Sharafeddine, and A. Ghrayeb, "Online altitude control and scheduling policy for minimizing AoI in UAV-assisted IoT wireless networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 7, pp. 2493–2505, Jul. 2022.

[42] W. Cai, D. Zhang, Z. Chen, W. Luo, and Y. Tang, "Mean-field game theory based altitude control strategy for massive UAV relay-assisted mobile edge computing," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2024, pp. 1–6.

[43] Y. Cai, Z. Wei, S. Hu, C. Liu, D. W. K. Ng, and J. Yuan, "Resource allocation and 3D trajectory design for power-efficient IRS-assisted UAV-NOMA communications," *IEEE Trans. Wireless Commun.*, vol. 21, no. 12, pp. 10315–10334, Dec. 2022.

[44] D. P. Simões, H. C. Oliveira, and M. V. Garcia, "UAV 3-D path planning based on high-resolution DSM, DTM, and true orthomosaic," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 3513905.

[45] A. Khochare, F. B. Sorbelli, Y. Simmhan, and S. K. Das, "Improved algorithms for co-scheduling of edge analytics and routes for UAV fleet missions," *IEEE/ACM Trans. Netw.*, vol. 32, no. 1, pp. 17–33, Feb. 2024.

[46] R. Karmakar, G. Kaddoum, and O. Akhrif, "A novel federated learning-based smart power and 3D trajectory control for fairness optimization in secure UAV-assisted MEC services," *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 4832–4848, May 2024.

[47] X. Dai, B. Duo, X. Yuan, and M. Di Renzo, "Energy-efficient UAV communications in the presence of wind: 3D modeling and trajectory design," *IEEE Trans. Wireless Commun.*, vol. 23, no. 3, pp. 1840–1854, Mar. 2023.

[48] J. Zeng, D. Mishra, H. H. Gharakheili, and A. Seneviratne, "Secure energy efficiency fairness maximization in backscatter throughput constrained UAV-assisted data collection," in *Proc. IEEE Int. Conf. Acoust, Speech Signal Process*, 2024, pp. 9041–9045.

[49] J. Liu and H. Zhang, "Height-fixed UAV enabled energy-efficient data collection in RIS-aided wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 7452–7463, Nov. 2023.

[50] H. Hao, C. Xu, W. Zhang, S. Yang, and G.-M. Muntean, "Joint task offloading, resource allocation, and trajectory design for Multi-UAV cooperative edge computing with task priority," *IEEE Trans. Mobile Comput.*, vol. 23, no. 9, pp. 8649–8663, Sep. 2024.

[51] L. Zhong, Y. Li, M.-F. Ge, M. Feng, and S. Mao, "Joint task offloading and resource allocation for LEO satellite-based mobile edge computing systems with heterogeneous task demands," *IEEE Trans. Veh. Technol.*, vol. 74, no. 7, pp. 11337–11352, Jul. 2025, doi: 10.1109/TVT.2025.3549119.

[52] C. Lin et al., "Maximizing energy efficiency of period-area coverage with a UAV for wireless rechargeable sensor networks," *IEEE/ACM Trans. Netw.*, vol. 31, no. 4, pp. 1657–1673, Aug. 2023.

[53] H. Gong, B. Huang, and B. Jia, "Energy-efficient 3-D UAV ground node accessing using the minimum number of UAVs," *IEEE Trans. Mobile Comput.*, vol. 23, no. 12, pp. 12046–12060, Dec. 2024.

[54] S. Katiyar, N. Ibraheem, and A. Q. Ansari, "Ant colony optimization: A tutorial review," in *Proc. Nat. Conf. Adv. Power Control*, 2015, pp. 99–110.

[55] S.-H. Huang, Y.-H. Huang, C. A. Blazquez, and C.-Y. Chen, "Solving the vehicle routing problem with drone for delivery services using an ant colony optimization algorithm," *Adv. Eng. Inform.*, vol. 51, 2022, Art. no. 101536.

[56] A. Lipowski and D. Lipowska, "Roulette-wheel selection via stochastic acceptance," *Physica A: Stat. Mechan. Appl.*, vol. 391, no. 6, pp. 2193–2196, 2012.

[57] Y. Zeng, G. Guo, S. Chen, Y. Qiang, and J. Liu, "Energy-efficient data collection from UAV in WSNs based on improved PSO algorithm," *IEEE Sensors J.*, vol. 24, no. 21, pp. 35762–35774, Nov. 2024.

[58] J. Zheng, M. Ding, L. Sun, and H. Liu, "Distributed stochastic algorithm based on enhanced genetic algorithm for path planning of Multi-UAV cooperative area search," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8290–8303, Aug. 2023.

Yiqian Wang (Graduate Student Member, IEEE) received the BS degree from the Nanjing University of Posts and Telecommunications, China, in 2024. He is currently working toward the MS degree with the School of Computer Science and Engineering, Southeast University, China. His research interests include UAV trajectory planning and task scheduling.

Jianping Huang (Graduate Student Member, IEEE) 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 visited 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.

Yuming Gao received the BS degree from Southeast University, Nanjing, China, in 2021. He is currently working toward the MS degree with the School of Computer Science and Engineering, Southeast University, China. His research interests include UAV swarm scheduling and algorithm design.

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 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.

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 a co-chair of IEEE SMC Technical Committee on Computer Supported Cooperative Work in Design, and he is a member of the ACM and co-chair of the ACM China Awards Committee. His research interests are network architecture, network security, cloud computing, wireless network, and Internet of Things.