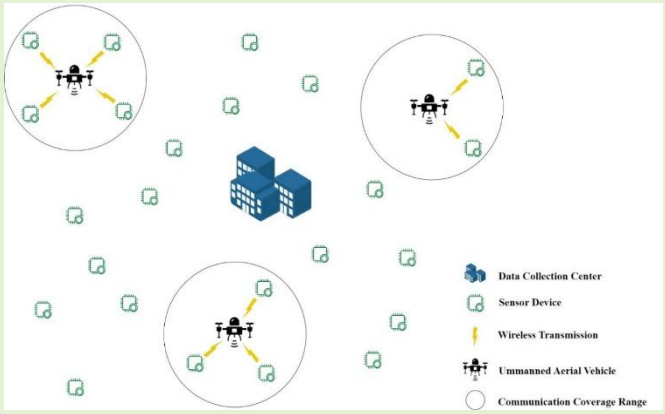
Enhanced Multi-UAV-Assisted Data Transmission Control Strategy in Wireless Sensor Networks

Jau-Yang Chang, Song-Shyong Chen, and Kai-Hua Chen

***Abstract*—The utilization of unmanned aerial vehicles (UAVs) for collecting data from sensing devices (SDs) in areas with limited or inadequate communication infrastructure is a critical concern. In large-scale sensing areas, UAVs can improve the efficiency of collecting data from ground SDs and reduce the time needed to complete data collection tasks for wireless sensor networks (WSNs). We consider the positions of ground SDs to plan the mission of multiple UAVs for data collection. By flying UAVs to suitable hover positions for data collection, ground SDs can transmit data over shorter distances, reducing power consumption for transmission and extending the mission of sensing. Considering the power consumption during UAV flight and hovering, designing and developing efficient transmission control strategy for UAVs is crucial and necessary. In this paper, we analyze and explore the relationship between the deployment positions of SDs, UAVs hover positions, construction costs, and flight distances. For the planning of UAVs hover positions, we employ a centroid-based algorithm in the search for suitable UAV hover positions. In the planning of multiple UAVs data collection areas, we propose a clustering mechanism and strategy based on clustering centers. For UAVs flight path planning, we use an algorithm to avoid path reversals, calculating appropriate flight paths. Additionally, we design a transmission routing mechanism for ground SDs to extend sensing tasks. The goal is to achieve reasonable construction costs for UAVs, appropriate data collection areas, and efficient reduction of UAVs flight distances and completion time for data collection tasks.**



***Index Terms*—Flying path, sensing devices (SDs), hovering, unmanned aerial vehicles (UAVs), wireless sensor networks (WSNs)**

I. Introduction

IN recent years, advancements in sensing, communication, and battery technologies have expanded the development and application of UAVs, enabling them to achieve high-precision positioning and obstacle avoidance. In addition, the UAV can enable real-time data transmission with ground stations or other equipment through its communication capabilities. With advancements in battery technology, the endurance of UAVs has significantly increased, allowing them to complete missions over longer durations. Employing advanced and intelligent autonomous flight strategies, UAVs can autonomously navigate and perform tasks while adapting to diverse conditions and environmental changes. When UAVs are integrated with Internet of Things (IoT) systems, they enable a wide range of monitoring applications for various sensing devices (SDs). This integration unlocks numerous innovative applications across industries such as agriculture, fisheries, military, logistics, emergency response, and more, enhancing data collection, real-time monitoring, and operational efficiency. In a large-scale sensing area, the implementation of an IoT system involves deploying numerous SDs across the monitored area. These devices are strategically positioned to sense and monitor specific targets as well as environmental changes. The collected data is then transmitted wirelessly to a central data collection center (DCC) for analysis and processing. However, in certain specific or geographically challenging areas where monitoring applications are necessary, such as battlefields, disaster zones, contaminated regions, high-risk areas, or places with limited communication infrastructure, the communication range of the SDs may not reach the DCC. As a result, the monitored data may not be effectively transferred to the intended destination. Therefore, when there is a requirement to deploy a large number of SDs for monitoring event changes, utilizing the mobility of

This work was supported in part by the National Science and Technology Council, Taiwan, under Grant NSTC 113-2221-E-150-028.

Jau-Yang Chang and Kai-Hua Chen are with the Department of Computer Science and Information Engineering, National Formosa University, Hu-Wei, Yun-Lin, Taiwan. (e-mail: jychang@nfu.edu.tw; 11063107@gm.nfu.edu.tw).

Song-Shyong Chen is the Department of Electro-Optical Engineering, National Formosa University, Hu-Wei, Yun-Lin, Taiwan. (e-mail: sscheneoe@nfu.edu.tw).

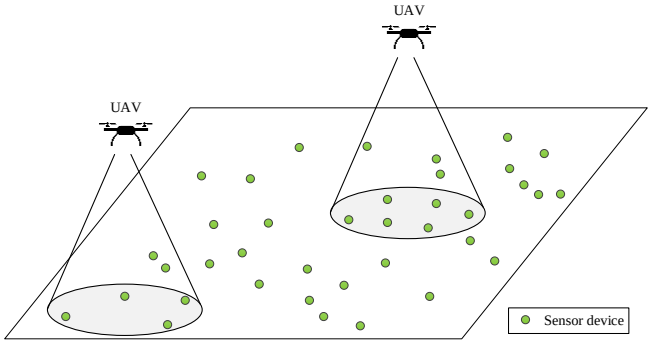


Fig. 1. Data collection by UAVs in a large-scale sensing area.

UAVs as rapid data collection relay stations can enhance data collection efficiency and reduce the time needed to complete data collection tasks [1–5].

In large-scale IoT systems, UAVs are employed to enter monitoring areas and collect data from ground SDs. This approach offers a convenient and flexible method for data collection, as illustrated in Fig. 1. The figure assumes that UAVs are equipped with automatic flight capabilities and directional wireless communication to exchange information with ground SDs within their range. Each ground SD is also equipped with array antennas, enabling it to transmit and receive data from all directions [1]. To ensure timely and accurate data collection, multiple UAVs are deployed to collect data from the ground SDs. This setup allows the ground sensors to transmit data over shorter distances, reducing data loss and power consumption. When considering the costs associated with UAV construction, flight, and hovering, it is crucial to minimize the flying distance and hovering time of the UAVs, and to avoid inefficient flight paths. Therefore, careful analysis of the sensor deployment positions and UAV hovering times is necessary. Efficient data collection relies on planning the optimal deployment positions of the sensors and the flight paths of the UAVs, considering construction costs and flight distances. Developing effective UAV transmission control strategies is essential to achieving these goals. By minimizing UAV flight distances and optimizing data collection area planning, the time required to complete data collection tasks can be reduced. Thus, designing efficient transmission control strategies for multiple autonomous flying vehicles in extensive sensing environments becomes a complex and critical research challenge.

In this paper, we introduce a centroid-based overlap coverage (CBOC) strategy for designing transmission control strategies for multiple UAVs. This approach aims to balance the construction costs of the UAVs while optimizing data collection areas, efficiently minimizing both the UAVs' flight distances and the time required to complete data collection tasks. The unique contributions and novelty of this paper are detailed as follows:

- Optimization of UAV Hovering Positions: Traditional approaches assign hovering positions directly above each SD, resulting in a high number of hovering positions and increased flight distances. By calculating centroids, the CBOC strategy reduces redundant hovering and better balances coverage.
- Integration of Clustering with Flight Path Planning: Existing methods often lack explicit clustering mechanisms, relying instead on heuristic-based hovering position assignments. The CBOC strategy's emphasis on clustering effectively minimizes overlapping coverage. Additionally, it integrates seamlessly with centroid-based coverage and path planning, ensuring the efficient utilization of UAV resources.
- Efficient Path Planning to Avoid Backtracking: Many existing strategies do not address the issue of backtracking explicitly. As a result, UAVs in these methods may revisit areas unnecessarily, leading to increased flight times and energy consumption. A backtracking avoidance algorithm is incorporated to optimize UAV flight paths between hovering positions. This algorithm ensures that UAVs follow the shortest and most efficient routes while avoiding previously covered areas.
- Simultaneous Data Collection from Multiple SDs: Traditional approaches focus on single-SD communication, leading to shorter hovering times per SD but requiring more overall hovering positions and flight time. The CBOC strategy enables simultaneous data collection from multiple SDs within a UAV's coverage area. By optimizing bandwidth allocation and communication, it minimizes hovering time while ensuring reliable data transmission.
- Reduction in UAV Deployment Costs: Existing methods often require more UAVs to achieve comparable coverage due to inefficient use of resources. By minimizing the number of hovering positions and reducing flight distances, the CBOC strategy decreases the number of UAVs required to complete a data collection task. This directly translates to lower deployment and operational costs.
- Energy Efficiency and Scalability: The CBOC strategy reduces UAV energy consumption by optimizing UAV positions and minimizing redundant movements. It ensures scalability by dynamically adjusting to varying SD densities and UAV constraints.

The CBOC strategy redefines efficiency in UAV-assisted WSNs by systematically addressing key challenges in hovering position optimization, path planning, and energy management. It represents a significant advancement over existing methods, offering practical benefits in deployment costs, task completion time, and scalability.

The remaining sections of the paper are organized as follows: Section II reviews related work and relevant literature. Section III describes the system model considered in this study. Section IV presents our proposed approach for enhancing data transmission control for multiple UAVs in WSNs. Section V examines and discusses the performance of the proposed method. Finally, Section VI concludes with a summary of the effectiveness of the approach.

II. Related Work

In recent years, research on data transmission by autonomous UAVs has focused on several key areas: planning UAV flight paths, managing power consumption during flight, and optimizing the sequence of data collection from SDs. Additionally, it explores strategies for minimizing data loss and enhancing energy efficiency. The main goal is to deploy UAVs to collect data from sensors within a specified area, which not

only facilitates data collection but also reduces the power requirements of the sensors, thereby extending their operational lifespan. In [1], the authors design the UAV's flight trajectory, hovering positions, and data upload power of the SDs by considering their transmission power and energy constraints. In [2], the authors employ a single UAV to communicate and transmit data with multiple ground SDs, aiming to satisfy the data transmission requirements of all sensors while minimizing the UAV's power consumption during flight. In [3], the authors address scenarios where UAV mobility and communication capabilities are limited in large-scale WSNs. They examine the issue of certain SDs potentially running out of storage capacity if they are unable to transmit data to the UAV for an extended period. In [4], the authors use a clustering approach to organize a large WSN into different clusters. A cluster head is chosen from each cluster based on the remaining energy of the SDs. This cluster head collects data from other members of its cluster, and a UAV then flies over the cluster head to collect the data. In [5], the authors deploy a UAV to collect data from dispersed SDs within a monitoring area of a WSN. The goal is to maximize the remaining energy of the SDs after data transmission. In [6], the authors consider the communication range of SDs and uses a low-complexity greedy algorithm to plan the UAV's flight trajectory while maintaining a fixed flight altitude. The goal is to maximize data collection from the sensors within the WSN. In [7], the authors propose an energy-efficient UAV-assisted data collection method for heterogeneous wireless sensor networks based on the remaining energy of the SDs. When the remaining energy of a SD exceeds a certain threshold, it transmits its own data along with any received data to its cluster head via single-hop transmission. If the energy is below the threshold, it only sends the data to the nearest functional SD within the cluster. The UAV then flies over the cluster head to collect the aggregated data.

In [8], the authors address the research topic of trajectory planning and data collection for multiple UAVs. They apply the multiple traveling salesman problem (MTSP) approach to plan the UAVs' hovering patterns, with the aim of minimizing the task completion time for all UAVs while ensuring that each SD can successfully complete its data transmission within the given energy budget. In [9], a minimizing data collection time (MDCT) strategy with collaborative UAVs in WSNs is proposed to address load balancing between UAVs and SDs, aiming to improve network throughput. By optimizing the data collection sequence, device power, UAV flight speed, and data collection locations, this approach enhances overall network fairness and reduces the completion time for data collection tasks using multiple UAVs. In [10], the balance between aerial and ground costs during data collection is addressed in UAV-assisted WSNs. Aerial costs refer to the flight and operational expenses of UAVs, while ground costs pertain to the energy consumption of SDs. The authors propose a mechanism to optimize UAV trajectories, sensor wake-up time allocation, and transmission power, aiming to balance costs between UAVs and SDs, thereby extending the network's lifespan while maintaining UAV endurance. In [11], a multi-UAV data collection network system is considered for a specific IoT area. The authors use the K-means algorithm to assign SDs to each UAV, followed by a multi-agent deep reinforcement learning algorithm to design the joint trajectories of the UAVs. The goal is to minimize the completion time of the data collection task while accounting for constraints such as the UAVs' maximum speed, acceleration, collision avoidance, and communication interference between UAVs. In [12], the authors address a WSN environment with energy-constrained SDs. A multi-UAV deployment algorithm is proposed to minimize the transmission power of the SDs while satisfying data rate and transmission power constraints. The algorithm is designed to allocate and position UAVs to enhance the overall energy efficiency of the network by ensuring fairness in transmission power among the SDs. In [13], the authors assume that each UAV operates at different altitudes within a grid-like environment to collect data from IoT SDs in a metropolitan urban setting. A multi-agent reinforcement learning approach is proposed that considers obstacles such as high-rise buildings, UAV power limitations, task completion time, and data collection efficiency. This approach enables the UAVs to find optimal flight trajectories that balance data collection, flight time efficiency, and navigation constraints. In [14], the authors assume that the UAV's service area covers the entire region where SDs are distributed, and the service area is divided into several sub-regions. By adjusting the number of UAVs and the size of each service area, the authors aim to optimize the amount of data collected from each SD, thereby improving the efficiency of UAV data collection tasks in WSNs. In [15], the authors explore the problem of maximizing data collection from SDs using multiple UAVs in a decentralized IoT system. Considering challenges such as no-fly zones and collision risks, the authors develop strategies for UAV scheduling and flight trajectory planning based on factors like antenna radiation patterns, path loss, signal-to-noise ratio, and maximum received signal power. In [16], the authors define distinct phases for SDs, including energy harvesting, data transmission, sleep, and data sensing, to determine whether the devices have sufficient energy to transmit data or collect new information. A scheduling algorithm is designed for multiple UAVs, with the primary goal of minimizing the total task completion time by optimizing the trajectories of all UAVs while collecting data from the SDs. In [17], the authors use multiple UAVs to collect data from all SDs in a WSN and transmit it to a remote base station. Based on the data collection areas of the SDs, horizontal and vertical flight paths are designed for the UAVs, and corresponding hovering positions with appropriate hovering times are determined. The goal is to minimize the flight time cost of the UAVs, thereby achieving efficient data collection from the SDs. In [18], the authors propose a multi-UAV-assisted data collection scheme for large agricultural areas to reduce the energy consumption of SDs during data transmission and extend their battery life. A heuristic algorithm is employed to calculate factors related to energy consumption balance, aiming to save energy for the SDs and prolong the network's lifespan. In [19], the authors propose a guided search deep reinforcement learning algorithm to enable UAVs with varying initial positions to independently perform data collection and forwarding tasks. For efficient data collection, the process is modeled as a sequential decision-making problem, with objectives of either minimizing the average age of information or maximizing the number of collected nodes based on specific environmental conditions. In [20], a deep reinforcement learning-based cooperative data collection scheme is proposed

to efficiently deliver data from buoy sensor nodes to a shipboard station in maritime data collection scenarios. To dynamically adapt UAV task areas, the authors introduce an adaptive partition algorithm based on virtual moving points. Furthermore, to minimize the time required for delivering collected data to the shipboard station, the study employs a deep reinforcement learning algorithm using a multi-agent deep deterministic policy gradient to plan the UAVs' flight paths. In [21], the authors propose an architecture for a UAV-assisted underwater IoT data collection system. By utilizing UAVs for data collection through relay cooperation, the flight time required for remote UAVs to return to the seashore base station (SBS) is reduced. To establish relay connections between UAVs, a relay matching algorithm is introduced. Additionally, a trajectory planning algorithm based on a hierarchical multi-agent deep deterministic policy gradient is developed to minimize both the flight time and data collection time of the UAVs.

Heavy data loads and extensive coverage have long posed significant challenges for big data processing in IoT systems. Mobile-edge computing (MEC) and UAV base stations (UAV-BSs) have emerged as effective solutions to tackle these challenges. Hovering UAV-BSs provide extensive and flexible service coverage. In [22], the authors propose deploying UAV-BSs as mobile MEC servers to collect data and perform initial data processing. To address the UAV path planning problem, the authors apply deep reinforcement learning to develop an online path planning algorithm. In [23], the authors consider the deployment of reconfigurable intelligent surfaces (RISs) on the ground to enhance wireless communication between UAVs and IoT devices. The authors propose a joint multi-UAV path planning and transmission scheduling algorithm aimed at maximizing the number of computing tasks successfully and timely completed by the UAVs and transmitted back to the IoT ground devices, while minimizing the total energy consumption of the UAVs. In [24], the authors address the problem of deploying UAV-BSs to provide reliable wireless communication services. The objective is to maximize the total number of covered user equipment while meeting user data rate requirements and the capacity limits of UAV-BSs. The authors propose a genetic algorithm-based 2D placement approach, where UAV-BSs are positioned to maximize user coverage while accounting for data rate distribution. UAV-BSs improve network coverage and flexibility, facilitating more efficient data collection and communication across large and diverse geographical regions [25]–[27]. Data collection from SDs in regions with limited communication infrastructure is a key area of research for autonomous aerial vehicles in IoT systems.

III. System Model

In the architecture of data collection using UAVs in WSNs, the system includes a DCC, multiple fixed SDs, and several UAVs. It is assumed that the DCC is located at the center of the sensing area, serving as the starting point for the UAVs and receiving the data collected by them. The SDs are distributed at various locations throughout the sensing area. In this work, each UAV is assumed to fly at a predetermined altitude. Due to the UAV's energy constraints, it is capable of serving only a limited number of sensor devices. These sensor devices are strategically positioned to monitor specific targets and environmental changes. When the SDs are deployed, the location information for all devices can be obtained through an initial patrol conducted by the UAV. This location data will be sent back to the DCC. Once the DCC receives the location information of the SDs, it will determine hovering positions that can fully cover all the devices based on the communication range of the UAVs. After identifying all the hovering positions and their corresponding SDs, the DCC will calculate the required hovering time for each position based on environmental parameters. The DCC plans a data collection path for the UAVs based on the hovering positions and their corresponding hovering times. Once the UAVs are assigned their data collection paths, they depart from the DCC and fly at a fixed altitude along the designated routes. Upon reaching the specified positions, the UAVs send requests to the corresponding SDs to transmit data. The SDs then upload their data to the UAVs using the assigned communication channels. After completing their data collection tasks, the UAVs return to the DCC and transmit the collected sensing data for analysis and processing. Once all UAVs have finished their data collection tasks, the overall data collection mission for the entire sensing network is considered complete.

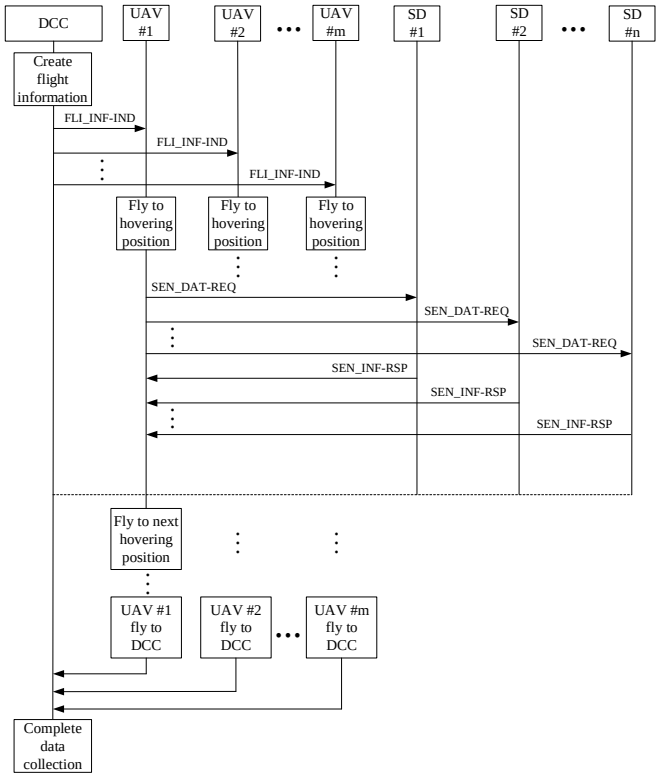


Fig. 2. Schematic diagram of the UAV data transmission control process.

Fig. 2 illustrates the schematic diagram of the UAV data transmission control process. The DCC configures each UAV with flight information, including flight paths, hovering positions, and hovering times, using the Flight Information Indication message (FLY_INF-IND). When a UAV reaches its designated hovering position, it sends a Sensing Data Request message (SEN_DAT-REQ) to the associated ground SDs. This message includes the UAV's hovering position and the assigned

communication channel for each SD. The SDs then adjust their transmission power based on the distance to the UAV and respond with a Sensing Information Response message (SEN_INF-RSP), completing the data collection for that hovering position. The UAV then moves to the next hovering position and repeats this process to collect sensing data. Once the UAV has visited all its designated hovering positions, it returns to the DCC, completing its data collection task. Other UAVs follow the same procedure to collect data from their assigned areas before returning to the DCC. By effectively clustering the SDs within the UAV's collection area, the UAVs can follow distinct flight paths, minimizing overlaps and preventing multiple UAVs from collecting data in the same area. Table I outlines the parameters used in system model.

TABLE I
PARAMETERS USED IN SYSTEM MODEL

Parameter	Definition
Ts_i	Data transmission time of ground SD *i*
Da_i	Amount of data to be transmitted by ground SD *i*
$Rs_{i,j}$	Data transmission rate from ground SD *i* to the UAV *j*
B	Channel bandwidth allocated for each UAV
Nh_j	Number of ground SDs covered by the UAV *j*
Nu	Number of UAVs
$SNR_{i,j}$	Signal-to-noise ratio (SNR) between ground SD *i* and the UAV *j*
Sp_i	Transmission power of ground SD *i*
σ^2	Noise power
$APL_{i,j}$	Average path loss between ground SD *i* and UAV *j*
f_c	Carrier frequency
s	Speed of light
α	Path loss exponent
μLos	Additional path loss for the line-of-sight (LoS) scenario
$\mu NLos$	Additional path loss for the Non-line-of-sight (NLoS) scenario
H	UAV's altitude
$d_{i,j}$	Distance between the ground SD *i* and the UAV *j*
$P_{Los}^{i,j}$	LoS probability between the ground SD *i* and the UAV *j*
φ and β	Environmental constants
$\theta_{i,j}$	Elevation angle between ground SD *i* and the UAV *j*
$P_{NLos}^{i,j}$	NLoS probability between the ground SD *i* and the UAV *j*
Th_j	Hovering time of the UAV *j*
Tc	Total task completion time of the UAVs
$Th_{k,j}$	Hovering time at position *k* for the UAV *j*.
Fd_j	Distance traveled by UAV *j*
Fs_j	Flight speed of UAV *j*
Ec_j	Energy consumption of the UAV *j*
$Ec_{h,j}$	Energy consumed while the UAV *j* is hovering at each position
$Ec_{f,j}$	Energy consumed by the UAV *j* while flying between positions
P_{hover}	Power consumption during hovering
P_{fly}	Power consumption during flying
Te	Total energy consumption of the UAVs

A. Data Transmission Model

We consider a ground-to-air data transmission model in which UAVs communicate with ground-based SDs, enabling these devices to transmit data to the UAVs. This model accounts for factors such as signal attenuation, line-of-sight conditions, and environmental interference, which can significantly affect the reliability and efficiency of data transmission between the UAVs and ground SDs [1], [9], [19], [24]. Let Ts_i represent the data transmission time of ground SD *i*, which can be calculated as follows:

$$Ts_i = \frac{Da_i}{Rs_{i,j}}, \tag{1}$$

where Da_i represents the amount of data to be transmitted by ground SD *i*, and $Rs_{i,j}$ is the data transmission rate from ground SD *i* to the UAV *j*. The transmission rate $Rs_{i,j}$ is calculated as follows [9]:

$$Rs_{i,j} = \frac{B}{Nh_j + Nu}\log_2(1 + SNR_{i,j}), \tag{2}$$

where *B* is the channel bandwidth allocated for each UAV. Each SD is allocated a specific bandwidth to facilitate data transmission to the UAVs. Nh_j represents the number of ground SDs covered by the UAV *j*, and Nu is the number of UAVs. The $SNR_{i,j}$ is the signal-to-noise ratio (SNR) between ground SD *i* and the UAV *j*, which can be calculated as follows [9]:

$$SNR_{i,j} = \frac{Sp_i}{APL_{i,j} \times \sigma^2}, \tag{3}$$

where Sp_i is the transmission power of ground SD *i*, σ^2 is the noise power, and $APL_{i,j}$ represents the average path loss between ground SD *i* and UAV *j*, which can be calculated as follows [9]:

$$APL_{i,j} = \left(\frac{4\pi f_c}{s}\right)^2 \cdot {d_{i,j}}^{\alpha} \cdot (P_{Los}^{i,j} \cdot \mu Los + P_{NLos}^{i,j} \cdot \mu NLos), \tag{4}$$

where f_c is the carrier frequency, *s* is the speed of light, α is the path loss exponent, μLos represents the additional path loss for the line-of-sight (LoS) scenario, and $\mu NLos$ represents the additional path loss for the Non-line-of-sight (NLoS) scenario in free space propagation [9], [24]. If the UAV's altitude is *H*, the coordinates of ground SD *i* are (i_x, i_y), and the coordinates of the UAV *j* are (j_x, j_y), the calculations for $d_{i,j}$ is as follows:

$$d_{i,j} = \sqrt{(i_x - j_x)^2 + (i_y - j_y)^2 + H^2}. \tag{5}$$

Let $P_{Los}^{i,j}$ represent the LoS probability between the ground SD *i* and the UAV *j*, which can be expressed as follow [9], [24]:

$$P_{Los}^{i,j} = \frac{1}{1 + \varphi \times e^{-\beta(\theta_{i,j} - \varphi)}}, \tag{6}$$

where φ and β are environmental constants used to simulate potential obstructions or variations in the altitude of ground SDs, which affect the path loss. $\theta_{i,j}$ represents the elevation angle between ground SD *i* and the UAV *j*, which can be calculated as follows [9], [24]:

$$\theta_{i,j} = \frac{180}{\pi} \times sin^{-1}\left(\frac{H}{d_{i,j}}\right). \tag{7}$$

Let $P_{NLos}^{i,j}$ represent the NLoS probability between the ground SD i and the UAV, which is calculated as j [9], [24]:

$$P_{NLos}^{i,j} = 1 - P_{Los}^{i,j}. \quad (8)$$

Let Th_j be the hovering time of the UAV j, which can be expressed as:

$$Th_j = max\left\{Ts_i \text{ for all } i = 1, \dots, Nh_j\right\}. \quad (9)$$

Let Tc represent the total task completion time of the UAVs, which can be expressed as the sum of the time spent hovering and flying of UAVs. If there are V UAVs in the systems, the total task completion time can be expressed as follows:

$$Tc = \sum_{j=1}^{V}\sum_{k=1}^{N} Th_{k,j} + \frac{Fd_j}{Fs_j}, \quad (10)$$

where N is the number of hovering positions, $Th_{k,j}$ is the hovering time at position k for the UAV j. Fd_j is the distance traveled by UAV j, and Fs_j is the flight speed of UAV j.

In real-world UAV communication scenarios, the wireless signal between a ground SD and a UAV can travel along either a direct LOS path or an NLOS path obstructed by buildings, trees, or other obstacles. The ground-to-air data transmission model incorporates separate path loss factors for LOS and NLOS, enabling the model to account for these distinct propagation conditions and reflect the complexity of real-world environments. LOS paths typically experience lower path loss due to minimal obstructions, resulting in stronger signals. Conversely, NLOS paths involve signal degradation caused by reflections, diffractions, or scattering, leading to higher attenuation and weaker signals. By distinguishing between LOS and NLOS contributions, the model effectively captures the varying path loss characteristics. The likelihood of a LOS or NLOS path is influenced by the UAV's altitude and the surrounding environment (e.g., urban, suburban, or rural). Weighting the path loss components by their respective probabilities ensures that the average path loss $APL_{i,j}$ accurately represents realistic communication conditions.

B. Energy Consumption Model

In applications where UAVs are utilized for data collection, a significant amount of energy is consumed during both flight and hovering to maintain their stability, posture, and control in the air. In contrast, the energy required for receiving data is relatively small. Consequently, the energy consumption model outlined in this paper focuses exclusively on the energy used during the UAVs' flight and hovering phases.

The total energy consumption of the UAV can be calculated by considering both the hovering energy and the flying energy. Let Ec_j represent the energy consumption of the UAV j, which can be expressed as the sum of the energy used during hovering and flying:

$$Ec_j = Ec_{h,j} + Ec_{f,j}, \quad (11)$$

where $Ec_{h,j}$ is the energy consumed while the UAV j is hovering at each position. $Ec_{f,j}$ is the energy consumed by the UAV j while flying between positions. Let P_{hover} represent the power consumption during hovering. The total energy consumption of UAV j during hovering can be calculated as follows:

$$Ec_{h,j} = P_{hover} \times \sum_{k=1}^{N} Th_{k,j}. \quad (12)$$

TABLE II
PARAMETERS USED IN THE UAV HOVERING POSITION SEARCHING ALGORITHM

Parameter	Definition
UAV_positions[]	The horizontal hovering positions of the UAVs
UAV_coverage	The coverage area of the UAV at its hovering position
sensor_positions[]	The position of the ground SDs
num_sensors	The total number of ground SDs
UAV_sensor_coverage[]	The number of sensors covered by UAVs horizontal hovering position
selected_UAV_positions[]	The selected horizontal hovering positions for the UAVs
num_selected_UAV_positions	The number of selected horizontal hovering positions for the UAVs
calculate_sensor_coverage()	Calculate the number of ground SDs within the coverage area of all current UAV hovering positions
selected_UAV_positions[m]	The m-th UAV's selected hovering position
record_remove_UAV_position()	Record the selected hovering position of the UAV, and exclude all other hovering positions within the coverage area of the selected position
adjust_hover_positions()	Adjust the UAV hovering positions based on centroid concepts.
adjust_UAV_height()	Set the altitude of the UAVs on the Z-axis
HO_list[]	The three-dimensional coordinates (x, y, z) of all actual hovering positions of the UAVs.
num_hover	The number of all selected hovering positions for the UAVs

Algorithm 1 UAV Hovering Position Search Algorithm

Input: UAV_positions[], UAV_coverage, sensor_positions[], num_sensors
Output: HO_list[], num_hover
Initialization: UAV_coverage[] = [∅], selected_UAV_positions[] = [∅],
num_selected_UAV_positions = 0
While num_sensors > 0
    UAV_sensor_coverage[] = calculate_sensor_coverage(UAV_positions[], UAV_coverage, sensor_positions[], num_sensors)
    If UAV_sensor_coverage[m] is max.
        selected_UAV_positions[m] is true
        num_selected_UAV_positions += 1
        num_sensors = num_sensors – UAV_sensor_coverage[m]
        record_remove_UAV_position(selected_UAV_positions[m])
    End If
End While
adjust_hover_positions(sensor_positions, selected_UAV_positions[])
HO_list[] = adjust_UAV_height(selected_UAV_positions[])
num_hover = num_selected_UAV_positions

Let P_{fly} represent the power consumption during flying. The total energy consumption of UAV j during flight can be calculated as follows:

$$Ec_{f,j} = P_{fly} \times \frac{Fd_j}{Fs_j}. \quad (13)$$

Let Te represent the total energy consumption of the UAVs, which can be expressed as follows:

$$Te = \sum_{j=1}^{V} Ec_j, \quad (14)$$

where V is the number of UAVs.

IV. Proposed Strategy

In this paper, we propose an efficient strategy for collecting data from ground SDs across large-scale areas using multiple UAVs. To accomplish this, we developed a CBOC strategy for transmission control in multi-UAV systems. For planning the UAVs' hovering positions, we employ a hovering positions search algorithm to identify suitable locations. To structure the data collection zones, we introduce a clustering mechanism that organizes UAV hovering positions around cluster centers, ensuring efficient deployment and cost-effectiveness. Additionally, a path planning algorithm is employed to prevent backtracking, ensuring that UAVs follow the most efficient routes during data collection.

A. UAV Hovering Positions Searching Algorithm

This algorithm ensures that each UAV is positioned to maximize coverage of the ground SDs while minimizing the number of UAVs required and reducing overlap between their coverage areas. By strategically placing the UAVs, we enhance data collection efficiency and reduce energy consumption throughout the mission. It is assumed that SDs are randomly distributed across a large-scale sensing area, and all positions above this area are considered potential hovering positions for the UAVs. The data transmission range of a UAV at each hovering position is defined as its coverage area. A DCC is located at the center of the sensing area, tasked with aggregating the data collected by the UAVs. The DCC also provides battery replacement and charging services for the UAVs. Additionally, the DCC is equipped with a high-performance workstation capable of running algorithms related to multi-UAV data transmission, and configuring the UAVs' hovering positions, hovering times, and flight paths. It is assumed that the positions of all SDs within the area are known. Table II outlines the relevant parameters and functions of the algorithm, and the pseudocode is presented in Algorithm 1.

In this pseudocode, the function adjust_hover_positions() is used to evaluate each hovering position for potential overlaps with other positions. If an overlap is detected, the centroid of the two overlapping positions is calculated as a new hovering position. This new position is then assessed to determine if it fully covers the SDs. If full coverage is achieved, the new position replaces the original two positions.

B. Data Collection Area Clustering Algorithm

Based on the UAVs' hovering positions, we employ the concept of cluster centers to design the data collection areas for multiple UAVs. Table III details the relevant parameters and functions of the algorithm, and the pseudocode is presented in Algorithm 2. The UAVs must fly to their designated hovering positions to collect sensing data while operating on limited battery power. After data collection, they are required to return to the DCC. The total energy consumption of multiple UAVs is calculated based on the various data collection areas. If the total energy consumption of UAVs exceeds their available battery capacity, the number of clusters in the data collection area must be increased by one. ($V = V + 1$). This adjustment indicates that the current number of UAVs is insufficient to perform the tasks at all selected hovering positions within the designated data collection area and to return to the DCC. Fig. 3 presents the

TABLE III
PARAMETERS USED IN THE DATA COLLECTION AREA CLUSTERING ALGORITHM

Parameter	Definition
selected_UAV_positions[]	The selected horizontal hovering positions for the UAVs
V	The required number of UAVs
sensor_positions[]	The position of the ground SDs
UAV_clusters[]	The clustered data collection areas for UAVs
cluster_centers[]	The positions of the cluster centers
previous_cluster_centers[]	The positions of the previous cluster centers
random_initialize_clusters(V)	Randomly determine V UAV selected hovering positions as the initial cluster center points
assign_hover_to_clusters()	Assign the UAV's selected hovering position closest to a cluster center point as a member of that cluster
update_cluster_centers()	Calculate the distances between the UAV's selected hovering positions within each cluster, and identify the position that minimizes the total distance to the new cluster center
final_clusters()	Calculate the data collection areas for the UAVs according to the clustering results
HO_list_clusters[]	The UAV's selected hovering positions within the clusters
num_hover_clusters[]	The number of UAV's selected hovering positions within the clusters

Algorithm 2 Data Collection Area Clustering Algorithm

1. **Input:** selected_UAV_positions[], sensor_positions[], V
2. **Output:** UAV_clusters [], HO_list_clusters[], num_hover_clusters[]
3. **Initialization:** cluster_centers[]=[∅], previous_cluster_centers[]=[∅]
4. cluster_centers[] = random_initialize_clusters(V)
5. **While** cluster_centers[] <> previous_cluster_centers[]
6. previous_cluster_centers[]= cluster_centers[]
7. assign_hover_to_clusters(selected_UAV_positions[], cluster_centers[])
8. cluster_centers[] = update_cluster_centers (selected_UAV_positions[], cluster_centers[])
9. **End While**
10. UAV_clusters [] = final_clusters(selected_UAV_positions[], cluster_centers[], sensor_positions[])
11. Calculate HO_list_clusters[] and num_hover_clusters[]

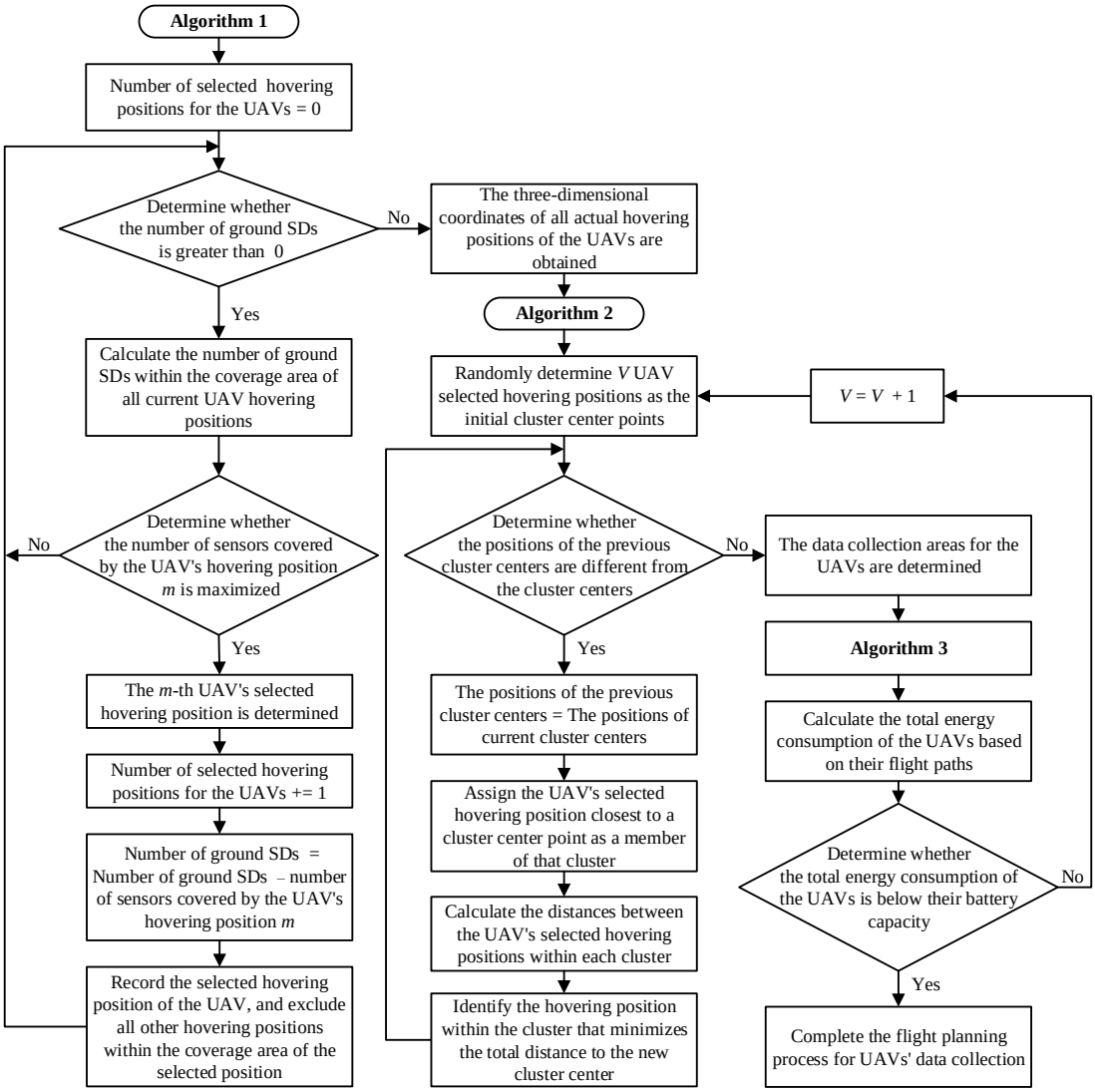


Fig. 3. Flowchart of the flight planning process for UAVs' data collection.

flowchart of the flight planning process for UAVs' data collection, clearly illustrating the determination of hovering positions and the planning of corresponding UAV data collection areas.

C. Flight Path Backtracking Avoidance Algorithm

After executing the data collection area clustering algorithm, the flight paths for data collection within each cluster must be planned using an algorithm designed to avoid backtracking [28]. Table IV outlines the relevant parameters and functions of the flight path backtracking avoidance algorithm, while the pseudocode is presented in Algorithm 3. The UAV data collection area clustering algorithm and the flight path backtracking avoidance algorithm are iteratively executed until the total energy consumption of the UAVs satisfies the sensing task requirements for the clustered data collection areas. The final number of UAVs determined through this process represents the necessary UAV deployment cost to meet the data collection needs of all ground SDs. Fig. 4 illustrates the flowchart of the flight path backtracking avoidance algorithm. In this diagram, TP represents the temporary position, which is initially set to the location of the DCC, marking the starting position of the UAV.

Fig. 5 illustrates the simulation diagram of the hovering positions, data collection area and flight paths for data collection by multiple UAVs. In this simulation, the sensing area is 2000 by 2000 meters, with the Z-axis representing the UAVs' hovering altitude. The filled circles indicate the locations of the 100 ground SDs, with the DCC positioned at the center of the sensing area. The UAVs hover at an altitude of 100 meters, with a coverage radius of 200 meters for each hovering position. Diamonds mark the UAVs' hovering locations, and the circles represent the data transmission coverage areas for each UAV at the selected positions. The number of UAVs is set to three, and the same color is used to represent UAVs assigned to the same cluster of SDs. This illustrates how the sensing area is divided into distinct zones to facilitate efficient data collection. The symbol ⊗ represents the UAVs, while the flight path of each UAV is shown using a unique color. This color-coding provides a clear visualization of how each UAV is tasked with collecting data from a specific cluster of sensing devices within the designated area.

TABLE IV
PARAMETERS USED IN THE FLIGHT PATH BACKTRACKING AVOIDANCE ALGORITHM

Parameter	Definition
U_initial	The initial position of the UAV, which is the location of the DCC
TP	Temporary position
HO_list_clusters[V]	The hovering positions of the UAV in cluster V
num_hover_clusters[V]	The number of hovering positions in cluster V
flight_path[V]	The flight path of the V UAVs
distances[]	The distances between the UAV's hovering positions and the temporary position
Temporary_result	The temporarily recorded UAV flight path
calculate_distances_HO_TP()	Calculate the distance between the current hovering position of the UAV and the temporary position
min_distance_HO	The hovering position of the UAV that is closest to the temporary position
find_HO_min_distance()	Select the UAV hovering position that is closest to the temporary position
path_intersect()	Calculate whether the path between the UAV's hovering position and the temporary position intersects with previously selected paths
*Temporary_result*_read()	Read the previously recorded UAV hovering positions stored in the *Temporary_result*
find_next_min_distance_HO()	Re-select the UAV hovering position with the next minimal distance to the TP
*Temporary_result*_append()	Store the UAV's hovering position in the *Temporary_result*
remove_HO()	Remove the selected UAV's hovering position
calculate_flight_path()	Calculate the UAV's flight path and distance

Algorithm 3 Flight Path Backtracking Avoidance Algorithm

1. **Input:** U_initial, HO_list_clusters[V], num_hover_clusters[V]
2. **Output:** flight_path[V]
3. **Initialization:** k = 0, distances[] = 0, *Temporary_result* = null, TP = U_initial
4. **While** k <= num_hover_clusters[V]
5. distances[] = calculate_distances_HO_TP (HO_list_clusters[V], TP)
6. min_distance_HO = find_HO_min_distance (distances[])
7. **While** path_intersect(min_distance_HO, TP)
8. TP = *Temporary_result*_read(min_distance_HO)
9. min_distance_HO = find_next_min_distance_HO(HO_list_clusters[V], TP)
10. **End While**
11. *Temporary_result*_append(min_distance_HO)
12. TP = min_distance_HO
13. remove_HO(min_distance_HO, HO_list_clusters[V])
14. $k = k + 1$
15. **End While**
16. flight_path[V] = calculate_flight_path(*Temporary_result*)

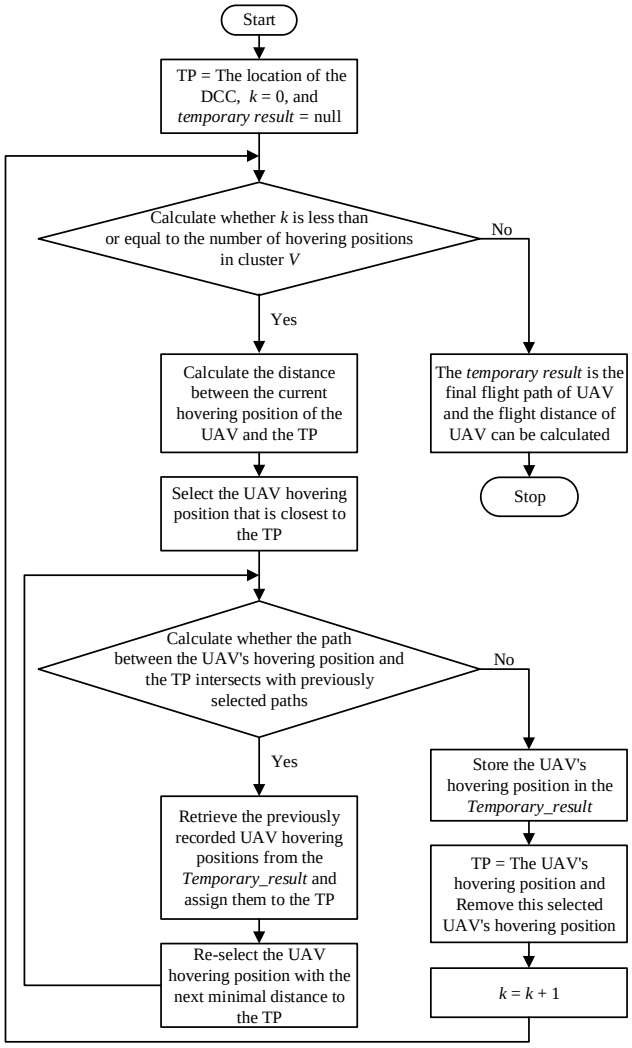


Fig. 4. Flowchart of the flight path backtracking avoidance algorithm.

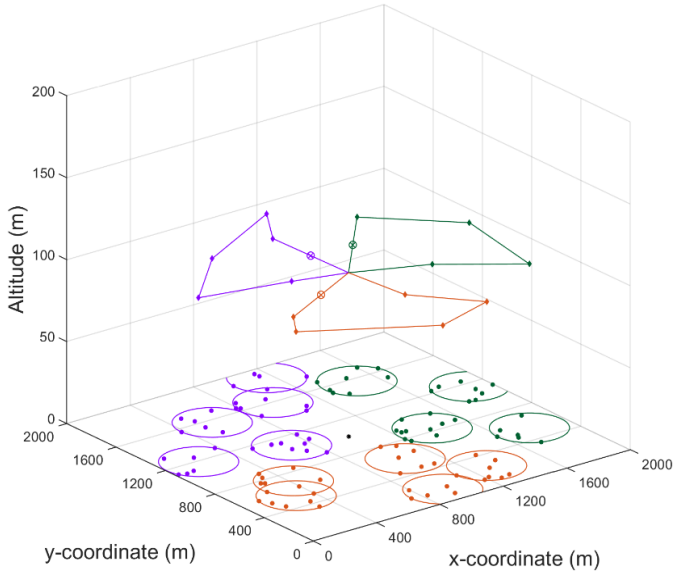


Fig. 5. Simulation diagram of the hovering positions, data collection area and flight paths for data collection by multiple UAVs.

V. Performance Analysis and Discussion

We present a detailed time complexity analysis, a comprehensive overview of the simulation model, an in-depth evaluation of the simulation results to effectively assess the performance of the proposed CBOC strategy, and a discussion of practical considerations for its implementation.

A. Time Complexities Analysis of Proposed Algorithms

Let N represent the total number of sensors (num_sensors), and M represent the number of potential UAV hovering positions (UAV_positions[]). Let S be the number of selected UAV positions. The time complexities of Algorithm 1 can be summarized as follows:

1. **Initialization**: Initializing empty arrays and counters is $O(1)$.
2. **While Loop**: The loop runs until all sensors are covered. In the worst-case scenario, one UAV position covers only one sensor in each iteration, resulting in at most N iterations. The total complexity of the loop is $O(N \times M \times N^2) = O(M \times N^2)$.
3. **Hover Position Adjustment**: Adjusting each position based on the centroid of covered sensors requires scanning N sensors. The complexity is $O(S \times N)$, where $S \leq M$.
4. **Height Adjustment**: This step adjusts the altitude for S selected UAV positions. The complexity is $O(S)$.

Since $S \leq M$, the dominant term is the complexity of the while loop. Hence, the overall time complexity of Algorithm 1 is $O(M \times N^2)$. Clearly, the quadratic dependence on N makes the algorithm computationally expensive for dense WSNs, and the linear dependence on M suggests that reducing unnecessary UAV positions could improve efficiency.

Let T represent the number of iterations required for convergence in the clustering process and V represent the required number of UAVs. The time complexities of Algorithm 2 can be summarized as follows:

1. **Initialization**: Selecting V initial cluster centers from M UAV positions. The complexity is $O(V)$.
2. **While Loop**: Each iteration of the while loop involves $O(M \times V)$ for assign_hover_to_clusters() function and $O(V \times M)$ for update_cluster_centers() function. The combined complexity per iteration is $O(M \times V) + O(V \times M) = O(M \times V)$. For T iterations, the complexity is $O(T \times M \times V)$.
3. **Post-Processing**: The final_clusters() function assigns UAV positions to their final clusters. Assuming it involves rechecking M positions and V clusters, the complexity is $O(M \times V)$. Calculating HO_list_clusters[] and num_hover_clusters[] involves V clusters and their corresponding UAV positions, resulting in $O(V \times M)$. Hence, the time complexity of Post-Processing is $O(M \times V) + O(V \times M) = O(M \times V)$.

Since $O(T \times M \times V)$ dominates the other terms, the overall time complexity of Algorithm 2 is $O(T \times M \times V)$. The key factors affecting complexity are the number of iterations (T), the number of clusters (V), and the number of UAV positions (M). The number of iterations (T) depends on the convergence rate of the clustering process. In practice, T is typically small for centroid-based clustering algorithms. The number of clusters (V) is directly proportional to the number of UAVs, so increasing V significantly impacts runtime. A higher M increases the number of distance calculations and centroid updates. Let K represent the number of hovering positions in a given cluster (num_hover_clusters[V]). The time complexities of Algorithm 3 can be summarized as follows:

TABLE V
PARAMETERS USED IN SIMULATION MODEL

Parameter	value
Noise power (σ^2)	-110 dbm
Path loss exponent (α))	2
Environmental constant (φ)	11.95
Environmental constant (β)	0.14
Speed of light (s)	$3 \times 10^8\ m/s$
Carrier frequency (f_c)	2 GHz
Bandwidth (B)	100 MHz
Additional path loss for the LoS scenario (μLos)	3 dB
Additional path loss for the NLoS scenario ($\mu NLos$)	23 dB
Transmission power of SD (Sp_i)	0.5 W
Power consumption during hovering (P_{hover})	168.48 Wh
Power consumption during flying (P_{fly})	161.52 Wh

1. **Initialization**: Initialize counters, arrays, and variables. The complexity is $O(1)$.
2. **While Loop**: The loop runs K times, processing all hovering positions in the cluster. The time complexity is $O(K)$ for distance calculations, finding the minimum distance, and updates. In the worst case, the nested loop might iterate up to K times (e.g., if several intersections are encountered). Each iteration of the nested loop involves checking path intersections ($O(1)$) and finding the next closest hovering position ($O(K)$). The worst-case time complexity is $O(K^2)$ for all nested iterations per hovering position. Hence, the total time complexity for the While loop is $O(K \times K^2) = O(K^3)$.
3. **Final Step**: Generate the flight path based on the recorded *Temporary_result*. The time complexity is $O(K)$, which depends on the number of hovering positions K.

Since the while loop dominates, the overall worst-case complexity is $O(K^3)$. The key factor affecting complexity is the number of hovering positions. The cubic dependence on K makes this algorithm computationally expensive for clusters with a large number of hovering positions. The nested loop for resolving path intersections is the main bottleneck, as it can involve multiple recalculations for each hovering position.

B. Simulation Model and Analysis of Proposed Algorithms

To demonstrate the effectiveness of our approach, we conduct a comparative study with the existing MDCT strategy [9] and MDCTP, which is an enhanced version of MDCT. In MDCT, the goal is to minimize the time UAVs take to travel from the DCC, complete all data collection tasks, and return to the DCC. UAVs hover directly above the SDs to reduce the time spent at each hovering position. While the MDCT algorithm effectively minimizes hovering time, as the number of SDs increases, UAVs must move more frequently between devices, which increases both flight time and energy consumption. We adjust the selection process of hovering positions in MDCT to reduce flight time. In MDCTP, the process starts by identifying the SD closest to the DCC as the initial hovering position. The centroid of all SDs within the coverage area of this position is then calculated and used as the new hovering position. This process continues until the number of covered SDs and the

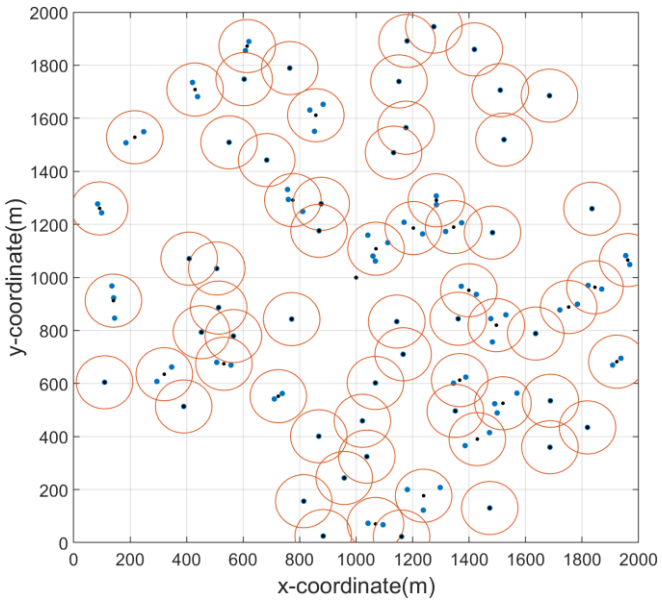


Fig. 6. Hovering positions of UAVs for MDCTP.

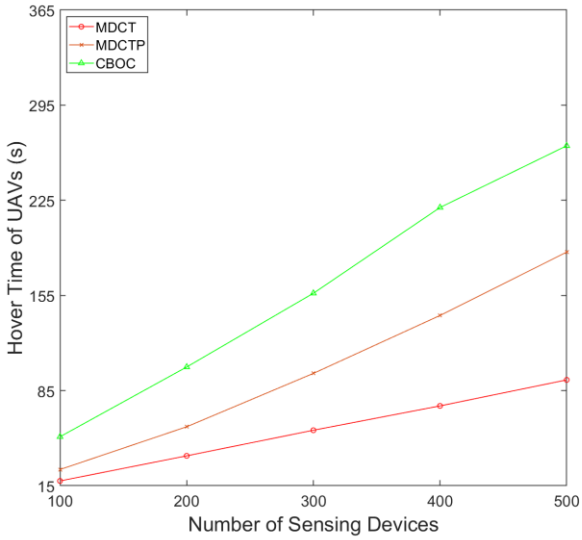


Fig. 8. Hovering time of UAVs.

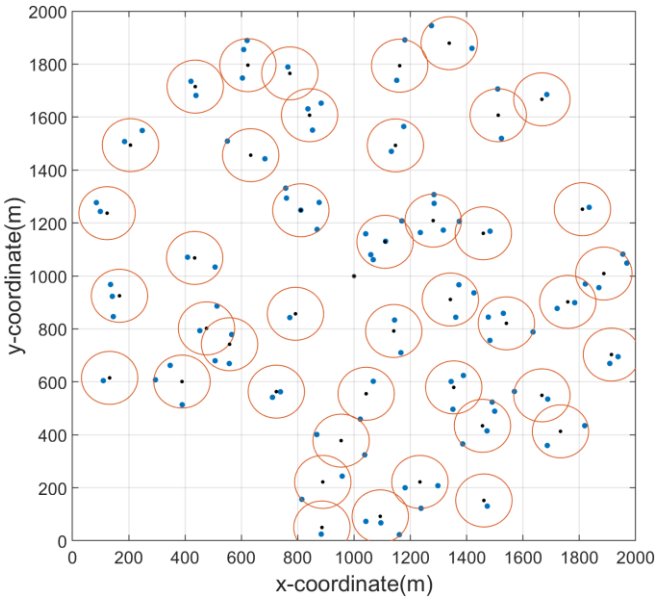


Fig. 7. Hovering positions of UAVs for CBOC.

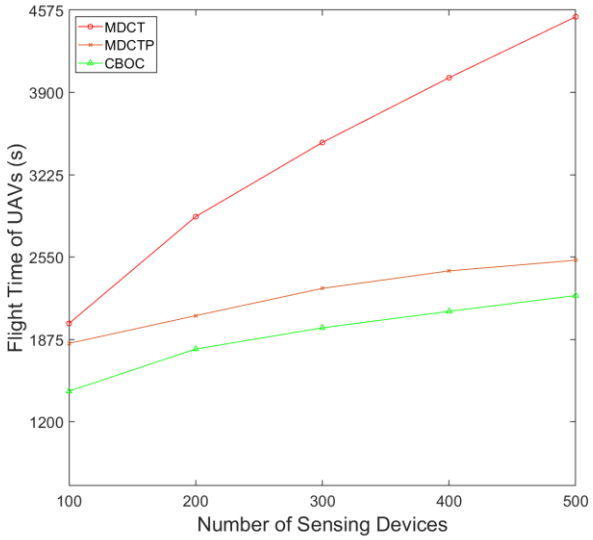


Fig. 9. Flight time of UAVs.

hovering position remain unchanged, at which point the position is recorded as the final hovering point. The covered SDs are then excluded, and the next closest SD to the current hovering position is selected as the new hovering point. This procedure is repeated until all SDs are covered. In this approach, the SDs within each hovering position's coverage area share the available bandwidth, which increases data transmission time. This leads to longer UAV hovering time and higher hovering energy consumption. However, it significantly reduces the UAVs' flight distances and overall flight energy consumption. In this analysis, we examine and discuss how varying the number of SDs impacts hovering time, flight time, task completion time, and power consumption of UAVs.

A visual interface simulator is designed to facilitate the implementation of existing strategies, ensuring a comprehensive and fair comparison. This simulator is developed using MATLAB 2020b. The assumptions related to the simulation environment are outlined as follows:

- The sensing range for each SD is bounded by a 2 km by 2 km square area in the simulation.
- The SDs are randomly distributed within the sensing area according to a Poisson process, with numbers varying from 100 to 500.
- The UAVs operate at an altitude of 100 meters [1], [8], [25], with each hovering position covering a radius of 100 meters.
- The UAV flight speed is set to 8 m/s.
- The data size for each SD is 0.5 G bits.
- The UAV is equipped with a battery capacity of 5000 mAh.
- All of the parameters used in our simulation are listed in Table V [1], [8], [9].

Taking the same SD positions into account, Figs. 6–7 illustrate the hovering position outcomes of the MDCTP and the proposed CBOC strategy when the number of SDs is set to 100. In these figures, the blue circles represent SDs, while the black circles denote the hovering positions of the UAVs. For the MDCTP strategy, the number of hovering positions is 67, whereas for the proposed CBOC strategy, the number of hovering positions is reduced to 42. This reduction in the number of hovering positions highlights the efficiency of the CBOC strategy in minimizing flight distances and improving energy consumption. To validate the effectiveness of our proposed strategy, we conducted a detailed performance analysis of three different strategies. The performance measures in this study include a table showing the average number of hovering positions of UAVs across three different strategies. Additionally, the performance measures include curves representing key metrics such as hovering time, flight time, task completion time, power consumption of UAVs, and the number of UAVs required, all plotted based on the results of one hundred simulation runs. In each simulation run, the positions of the SDs are randomly generated within the designated sensing area to ensure robustness in the results.

Table VI presents a comparison of the average number of hovering positions across the three strategies. In the MDCT strategy, UAVs hover directly above each SD to minimize hovering time, leading to the number of hovering positions being equal to the number of SDs in the area. In contrast, both the MDCTP and the proposed CBOC strategies reduce the number of hovering positions by allowing SDs within the coverage area of each hovering position to share the available bandwidth, thus minimizing the need for individual hovering positions for each SD. Fig. 8 illustrates the hovering time of UAVs across the three strategies. The MDCT strategy results in a lower hovering time for UAVs because it collects data from only one SD at a time. In contrast, both the proposed CBOC and MDCTP strategies require simultaneous data collection from multiple SDs. When bandwidth is limited, the SDs must share it, and as the number of SDs increases, the total hovering time required for data collection also increases. Fig. 9 illustrates the flight time of UAVs across three strategies. The curves indicate that the proposed CBOC strategy results in a lower flight time for UAVs. This behavior can be attributed to the fewer hovering positions, which lead to shorter flight distances. Additionally, the suitable flight paths for the UAVs are designed using the flight path backtracking avoidance algorithm, which minimizes unnecessary movements between hovering positions. This approach ensures that UAVs follow the most efficient routes, reducing travel time and enhancing overall data collection efficiency.

The selection of hovering positions greatly affects the flight and hovering time of UAVs. More hovering positions can reduce the hovering time at each location, but they increase the overall flight time. On the other hand, fewer hovering positions can decrease flight time, but they lead to longer hovering times. Fig. 10 shows the task completion time of UAVs across three strategies. Task completion time is defined as the total time required for UAVs to complete data collection from all sensing devices during a mission. It includes both the time spent hovering at designated positions and the time spent flying between those positions. Due to the larger number of hovering positions in the MDCT strategy, the flight range is limited. Considering the UAVs' constrained battery capacity, additional UAVs would be required to complete the data collection from all SDs. The curves in Fig. 10 show that the proposed CBOC strategy achieves a lower task completion time for UAVs compared to other strategies. The proposed CBOC strategy reduces task completion time for UAVs due to several contributing factors:

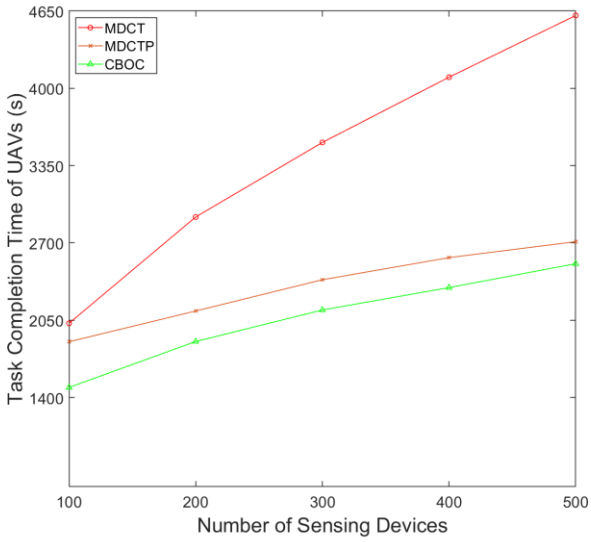


Fig. 10. Task completion time of UAVs.

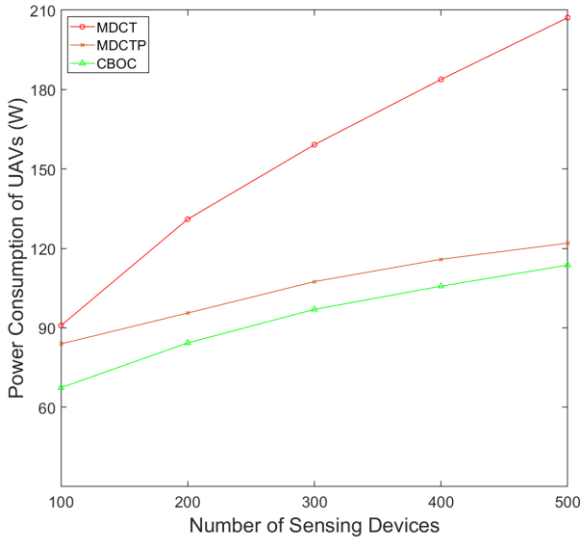


Fig. 11. Power consumption of UAVs.

TABLE VI
COMPARISON OF THE AVERAGE NUMBER OF HOVERING POSITIONS

Strategy	Number of Sensing Devices				
	100	200	300	400	500
MDCT	100	200	300	400	500
MDCTP	67	87	102	110	117
CBOC	42	67	81	92	99

1) Suitable Hovering Positions: By using centroid-based hovering positions, the UAVs can cover more SDs in each hover, reducing the number of required hovering positions and flight distances.
2) Efficient Data Collection: The strategy enables simultaneous data collection from multiple SDs, minimizing the need for repositioning and allowing for faster data transmission.
3) Balanced Task Distribution: In multi-UAV scenarios, the workload is distributed more evenly, preventing any single UAV from being overloaded with tasks, which improves overall efficiency.
4) Reduced Flight Time: Fewer hovering positions and shorter flight paths result in less flight time, contributing to the overall reduction in task completion time.

Fig. 11 shows that the power consumption of UAVs in the proposed CBOC strategy is lower than that of the MDCT and MDCTP strategies. It is evident that a lower task completion time for UAVs directly results in lower power consumption. This reduction in task completion time means less time spent on flying operations, which are primary contributors to power consumption in UAVs. In view of Figs. 10–11, the MDCT strategy results in higher task completion time and power consumption as the number of SDs increases. This is due to the larger number of hovering positions required, which leads to longer flight distances and more time spent in data collection. Consequently, the UAVs consume more power as flight time increases with the growing number of SDs. In contrast, the proposed CBOC strategy provides efficient task allocation and suitable flight paths, which minimize the energy required for data collection, leading to substantial power savings. By reducing unnecessary movements and ensuring suitable coverage with fewer hovering positions, the overall energy consumption of UAVs is significantly lowered. This not only leads to more efficient use of battery power but also extends the UAVs' operational time, improving the overall efficiency of the mission.

Fig. 12 presents a comparison of the average number of UAVs required for the three strategies, based on the results of one hundred simulation runs. The MDCT strategy involves a larger number of hovering positions, which increases the UAVs' flight distance and limits their operational range due to battery constraints. As a result, more UAVs are required to cover all the sensing devices to complete the data collection task efficiently within the available battery power. The MDCTP strategy reduces the number of hovering positions by refining the selection process used in MDCT. This modification lowers the UAVs' flight time and reduces the number of UAVs needed to complete the data collection task. The CBOC strategy obtains suitable hovering positions by clustering sensing devices based on their centroid, which reduces the number of hovering positions and flight distances. This allows UAVs to cover more devices with fewer flights, thereby reducing the total number of UAVs required.

Table VII presents specific differentiators of CBOC compared to MDCT and MDCTP. The centroid-based approach forms the cornerstone of CBOC's innovation, enabling fewer hovering positions and shorter flight distances. By integrating clustering, centroid-based coverage, and path planning, CBOC offers a comprehensive solution for UAV-assisted WSN data collection. This strategy is robust enough to handle high-density SD deployments, making it suitable for applications in agriculture, disaster response, and other large-scale IoT networks.

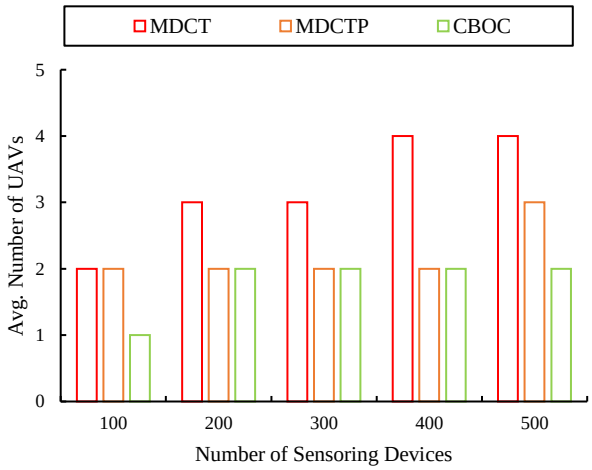


Fig. 12. Average number of UAVs required.

C. Discussion of Practical Considerations for the Proposed Strategy

To bridge the gap between the theoretical aspects of the CBOC strategy and its practical implementation, we address key considerations that may impact its real-world applicability. Below is a detailed exploration of these critical areas:

1. Communication Protocols

Efficient communication between UAVs and ground SDs is essential for the success of the CBOC strategy. Practical considerations include:

a. Selection of Protocols
- Wi-Fi (IEEE 802.11): Provides high-speed data transmission, suitable for small-scale deployments but limited by range and power consumption.
- LoRa (Long Range): Supports long-range communication with low power requirements, but limited data rates make it unsuitable for large-volume data.
- 5G: Provides ultra-low latency and high bandwidth, but its implementation may increase costs and require additional infrastructure.

b. Multi-UAV Communication
- Channel Access: Implementing a protocol like Time Division Multiple Access (TDMA) or Carrier Sense Multiple Access (CSMA) ensures collision-free communication.
- Coordination: UAVs must coordinate to prevent overlapping coverage and redundant data collection.

2. Error Correction Mechanisms

Ensuring reliable data transmission in dynamic environments is critical. This can be achieved through:

a. Forward Error Correction (FEC)
- Techniques such as Reed-Solomon coding or Low-Density Parity-Check (LDPC) codes can detect and correct errors in transmitted data.

TABLE VII
SPECIFIC DIFFERENTIATORS OF CBOC COMPARED TO MDCT AND MDCTP

Aspect	MDCT	MDCTP	CBOC
Hovering Position Selection	Fixed, directly above each SD	Incremental refinement near SDs	Centroid-based, clustering SDs for maximum coverage
Clustering Mechanism	Absent	Limited	Explicit clustering for zone-based data collection
Flight Path Optimization	None, may include backtracking	Limited	Backtracking avoidance with shortest path calculation
Simultaneous SD Communication	Not supported	Supported	Enhanced multi-SD data collection with bandwidth sharing
Number of Hovering Positions	Equals the number of SDs	Reduced, but still suboptimal	Significantly reduced
Deployment Costs	High, many UAVs required	Moderate	Low, fewer UAVs needed
Energy Efficiency	Poor, due to redundant movements	Improved	Excellent, minimizes UAVs flight energy usage
Task Completion Time	Long, due to frequent repositioning	Moderate	Shortest, due to optimal hovering and flight strategies

- FEC reduces the need for retransmission but increases computational overhead on UAVs.

b. Automatic Repeat Request (ARQ)

- Retransmission of corrupted data packets ensures reliability, though it can increase latency.
- Hybrid ARQ (HARQ) combines FEC and ARQ to balance efficiency and robustness.

3. Real-Time Implementation Challenges

Real-time operation is crucial for deploying the CBOC strategy in dynamic environments. Challenges include:

a. Dynamic Environment: Weather conditions (e.g., wind, rain) can affect UAV stability and data transmission quality.

b. Computational Overhead: The algorithms are executed at the DCC prior to the UAVs' takeoff. The DCC features a high-performance workstation specifically designed to process algorithms for multi-UAV data transmission and to configure the UAVs' hovering positions, durations, and flight paths.

c. Latency: UAVs must collect data with minimal delay to support real-time data collection applications effectively.

d. Synchronization: Coordinating multiple UAVs in real time requires precise synchronization to avoid overlapping coverage and ensure efficient task allocation.

e. UAV Battery Life: Hovering and flight consume significant energy. Optimizing UAV trajectories and clustering coverage areas is crucial to extend mission duration.

This comprehensive discussion of communication protocols, error correction mechanisms, and real-time implementation challenges highlights the practical aspects critical to successfully deploying the CBOC strategy in real-world scenarios.

VI. Conclusion

In areas lacking communication infrastructure, UAVs provide an effective and efficient solution for data collection from ground SDs. This paper introduced the CBOC strategy to enhance the efficiency of multi-UAV-assisted data collection in WSNs. The CBOC strategy addresses key challenges by optimizing UAV hovering positions, clustering data collection areas, and planning efficient flight paths to minimize energy consumption and task completion time. By calculating centroid-based hovering positions, UAVs reduce the number of hovering positions and minimize flight distances. The clustering mechanism ensures balanced task allocation among UAVs, while the flight path backtracking avoidance algorithm minimizes redundant movements. Simulation results confirm that the proposed CBOC strategy significantly outperforms existing methods in reducing task completion time, flight time, and power consumption, achieving a balance between energy efficiency and operational effectiveness. This approach effectively overcomes UAV battery limitations and flight constraints while facilitating comprehensive data collection in large-scale sensing environments. As a scalable and efficient solution, the CBOC strategy provides valuable insights into UAV-assisted data collection, paving the way for significant advancements in IoT-based WSN deployments. Its potential applications span various fields, including environmental monitoring, precision agriculture, and disaster management.

In future studies, we plan to conduct real-world experiments to validate the effectiveness of the CBOC strategy. This will involve defining experimental objectives to verify the reduction in the number of UAV hovering positions, designing an experimental setup to deploy SDs with capabilities similar to those assumed in the simulations, and configuring communication protocols between UAVs and SDs. A central DCC will be established to execute the CBOC algorithms and facilitate communication with UAVs. Additionally, we will select an appropriate test area to conduct UAV data collection and validation. The experimental results will be compared and analyzed to identify the strengths and weaknesses of the CBOC strategy, as well as to assess the performance gap between real-world and simulated scenarios. Future research will also explore the integration of deep reinforcement learning to enable intelligent UAV scheduling and dynamic routing. This enhancement aims to improve adaptability to changing environmental conditions, further advancing the applicability of UAV-based solutions for WSN data collection in dynamic and challenging scenarios.

References

[1] Y. Wang, M. Chen, C. Pan, K. Wang, and Y. Pan, "Joint optimization of UAV trajectory and sensor uploading powers for UAV-assisted data collection in wireless sensor networks," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 11214–11226, July, 2022.

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

[3] X. Wang, X. Liu, C. -T. Cheng, L. Deng, X. Chen, and F. Xiao, "A joint user scheduling and trajectory planning data collection strategy for the UAV-assisted WSN," *IEEE Communications Letters*, vol. 25, no. 7, pp. 2333–2337, July 2021.

[4] 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 Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9540–9554, Sept. 2021.

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

[6] C. Zhan, Y. Zeng, and R. Zhang, "Trajectory design for distributed estimation in UAV-enabled wireless sensor network," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 10155–10159, Oct. 2018.

[7] J. Chen and J. Tang, "UAV-assisted data collection for dynamic and heterogeneous wireless sensor networks," *IEEE Wireless Communications Letters*, vol. 11, no. 6, pp. 1288–1292, June 2022.

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

[9] Y. Wang, Z. Hu, X. Wen, Z. Lu, and J. Miao, "Minimizing data collection time with collaborative UAVs in wireless sensor networks," *IEEE Access*, vol. 8, pp. 98659-98669, 2020.

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

[11] S. Xu, X. Zhang, C. Li, D. Wang, and L. Yang, "Deep reinforcement learning approach for joint trajectory design in multi-UAV IoT networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 3, pp. 3389–3394, March 2022.

[12] H. Hydher, D. N. K. Jayakody, K. T. Hemachandra, and T. Samarasinghe, "UAV deployment for data collection in energy constrained WSN system," *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2022, pp. 1–6.

[13] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert, "Multi-UAV path planning for wireless data harvesting with deep reinforcement learning," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 1171–1187, 2021.

[14] Z. Wei, Q. Chen, S. Liu, and H. Wu, "Capacity of unmanned aerial vehicle assisted data collection in wireless sensor networks," *IEEE Access*, vol. 8, pp. 162819–162829, 2020.

[15] X. Wang, M. C. Gursoy, T. Erpek, and Y. E. Sagduyu, "Learning-based UAV path planning for data collection with integrated collision avoidance," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 16663–16676, Sept. 2022.

[16] I. Benmad, E. Driouch, and M. Kardouchi, "Trajectory planning for data collection in multi-UAV assisted WSNs," *2022 IEEE 95th Vehicular Technology Conference* (VTC2022-Spring), 2022, pp. 1-6.

[17] C. Luo, M. N. Satpute, D. Li, Y. Wang, W. Chen, and W. Wu, "Fine-grained trajectory optimization of multiple UAVs for efficient data gathering from WSNs," *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 162–175, Feb. 2021.

[18] K. -V. Nguyen, C. -H. Nguyen, T. V. Do, and C. Rotter, "Efficient multi-UAV assisted data gathering schemes for maximizing the operation time of wireless sensor networks in precision farming," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 12, pp. 11664–11674, Dec. 2023.

[19] Y. Hu, Y. Liu, A. Kaushik, C. Masouros, and J. S. Thompson, "Timely data collection for UAV-based IoT networks: A deep reinforcement learning approach," *IEEE Sensors Journal*, vol. 23, no. 11, pp. 12295–12308, 1 June1, 2023.

[20] X. Fu, X. Huang, Q. Pan, P. Pace, G. Aloi, and G. Fortino, "Cooperative data collection for UAV-assisted maritime IoT based on deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 7, pp. 10333–10349, July 2024.

[21] X. Fu and S. Kang, "Deep reinforcement learning based collaborative data collection in UAV-assisted underwater IoT," *IEEE Sensors Journal*, doi: 10.1109/JSEN.2024.3493454.

[22] S. Wan, J. Lu, P. Fan, and K. B. Letaief, "Toward big data processing in IoT: Path planning and resource management of UAV base stations in mobile-edge computing system," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 5995–6009, 2020.

[23] A. V. Savkin, C. Huang and W. Ni, "Joint multi-UAV path planning and LoS communication for mobile-edge computing in IoT networks with RISs," *IEEE Internet of Things Journal*, vol. 10, no. 3, pp. 2720–2727, 1 Feb.1, 2023.

[24] X. Zhong, Y. Huo, X. Dong, and Z. Liang, "QoS-compliant 3-D deployment optimization strategy for UAV base stations," *IEEE Systems Journal*, vol. 15, no. 2, pp. 1795-1803, June 2021.

[25] Z. Zhao *et al*., "Predictive UAV base station deployment and service offloading with distributed edge learning," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 3955–3972, Dec. 2021.

[26] M. J. Sobouti *et al*., "Efficient deployment of small cell base stations mounted on unmanned aerial vehicles for the internet of things Infrastructure," *IEEE Sensors Journal*, vol. 20, no. 13, pp. 7460–7471, 1 July1, 2020.

[27] A. V. Savkin, C. Huang and W. Ni, "On-demand deployment of aerial base stations for coverage enhancement in reconfigurable intelligent surface-assisted cellular networks on uneven terrains," *IEEE Communications Letters*, vol. 27, no. 2, pp. 666-670, Feb. 2023.

[28] J.-Y. Chang, J.-T. Jeng, Y.-H. Sheu, Z.-J. Jian, and W.-Y. Chang, "An efficient data collection path planning scheme for wireless sensor networks with mobile sinks," *EURASIP journal on wireless communications and networking*, 2020, doi: 10.1186/s13638-020-01873-4.

Jau-Yang Chang received the B.S., M.S., and Ph.D. degrees in electronic engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 1994, 1996, and 2004, respectively.

He is currently a Professor with the Department of Computer Science and Information Engineering, National Formosa University, Huwei, Taiwan, where he was the Chairman of the Department of Computer Science and Information Engineering from 2018 to 2021. His research interests include areas of wireless sensor networks, Internet of Things systems, and artificial intelligence control systems.

Song-Shyong Chen received the B.S., M.S., and Ph.D. degrees in electrical engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 1992, 1994, and 2003, respectively.

He is currently an Associate Professor in the Department of Electro-Optical Engineering, National Formosa University, Huwei, Yun-Lin, Taiwan. His current research interests include areas of drone flight control, fuzzy modeling and control, robust adaptive control, and AI deep learning.

Kai-Hua Chen received the B.S. and M.S. degrees all in computer science and information engineering from National Formosa University, Huwei, Yun-Lin, Taiwan, in 2021 and 2023, respectively.

His research interests include the areas of wireless sensor networks and Internet of Things systems.