Zhang *et al. J Wireless Com Network* (2025) 2025:2
https://doi.org/10.1186/s13638-024-02425-w

EURASIP Journal on Wireless Communications and Networking

RESEARCH **Open Access**

# Divisive hierarchical clustering for energy saving and latency reduction in UAV-assisted WSANs

Xuan Zhang[1,2], Yong-Long Wang[2] and Heejung Byun[3*]

*Correspondence: heejungbyun@suwon.ac.kr

[1] Department of Computer Science, The University of Suwon, Hwaseong 18323, South Korea
[2] School of Physics and Electrical Engineering, Linyi University, Linyi 276000, China
[3] Department of Information and Telecommunications Engineering, The University of Suwon, Hwaseong 18323, South Korea

**Abstract**

In response to the harsh and limited conditions prevalent in remote areas, wireless sensor and actuator networks (WSANs) play an essential role in Internet-of-Things systems by monitoring and interacting with unattended environments. However, the sensors employed by the majority of WSANs are powered by batteries, ensuring the efficient use and conservation of energy is vital for guaranteeing network connectivity and efficiency. To address this challenge, we proposed a divisive hierarchical clustering method based on K-means++ to organize the sensors. The intra-class distance of the cluster is fully taken into account to achieve the balance and full utilization of node energy. Furthermore, we utilize unmanned aerial vehicles (UAVs) for simultaneous data collection and develop a modified improved partheno genetic algorithm incorporating the Davies–Bouldin index for UAV scheduling. This approach effectively reduces network delay and balances network load. Numerical simulations demonstrate that our proposed method not only extends network lifetime but also balances energy savings and data collection latency.

**Keywords:** WSAN, Divisive hierarchical clustering, K-means++, UAVs

## 1 Introduction

The well-known concept of Internet-of-Things is defined as a dynamic environment of billions of devices with various components for data collection and exchange [1]. Wireless sensor and actuator networks (WSANs) play an essential role in the Internet-of-Things system because they can be used to monitor and interact with the physical environment, and the sensing data can be sent to a base station for further processing and analysis, establishing a link between the digital and the physical worlds [2, 3]. A large number of low-power, open-source, versatile tiny sensors are randomly deployed in the area of interest and left unattended, making their battery replacement and recharging difficult [4, 5]. When sensors form autonomous organizations, the sensor closest to the base station often consumes more energy for relaying traffic resulting in sensor energy exhaustion [6], as shown in Fig. 1a. As a result, network connectivity and coverage of the target area are not guaranteed. Considering these limitations, it is crucial to develop an energy-efficient data collection approach that evenly distributes energy consumption

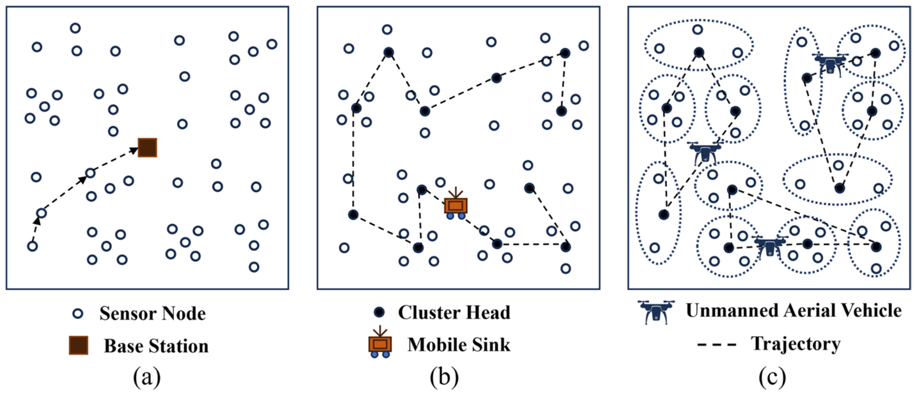


**Fig. 1** Simplified network topologies: **a** forwarding data by relaying traffic, **b** cluster head-based data gathering with single mobile sink, and **c** clustered-based data gathering with multiple mobile collectors

over the sensing field to prolong the network lifetime [7]. Furthermore, in some applications where sensing data are time-sensitive, the task of data collection must be limited to a specific time frame. As a result, designing an efficient and scalable data collection scheme capable of achieving good scalability, long network lifetime, and low data collection latency becomes a challenge.

Recent studies, such as [8–15], have shown that employing mobile data collectors can markedly ease the burden of data routing from sensors. Using mobile collectors may reduce the energy consumption of relay nodes and prevent these nodes from dying prematurely, thereby prolonging the lifetime of the network. However, if the mobile collector visits all nodes and collects data from them one by one, it may result in significant data transmission delay. Cluster heads have been employed in the network to shorten the total length of the visiting path, thereby reducing the data collection latency [16–23]. Clustering algorithms are used to group all sensors into clusters, with each cluster selecting a sensor node as the cluster head (CH) and other nodes as cluster members. Cluster member nodes transmit data to their corresponding CHs. Subsequently, CHs collect and process the data, sending information directly to the mobile sink upon its arrival. Hence, instead of visiting all sensor nodes, the mobile sink only needs to visit the CHs of each cluster, as shown in Fig. 1b, which reduces the length of the moving trajectory and, thus, the delay time. Therefore, an effective clustering algorithm is key to balancing the energy consumption of nodes in WSANs.

Although these works provide effective solutions for data collection in WSANs, their inefficiencies have been identified. In particular, in schemes where CHs are selected randomly, minimizing energy consumption while rotating CHs does not necessarily extend the network lifetime. Since the number of CHs changes dynamically, the sensor nodes are clustered dynamically, making the network's topology very unstable and the residual energy of each node unbalanced. Furthermore, in some applications, such as military and rescue, data collection is time-sensitive and must be completed in a reasonable length of time. It still consumes a lot of time, even with just one mobile sink or collector. Unmanned aerial vehicles (UAVs) have recently been introduced as a promising candidate solution for gathering sense data from WSAN sensor nodes. Multi-UAVs working simultaneously can improve data collection efficiency and further reduce the latency of the network. Network latency is the longest time UAVs spend in one round. Since UAVs are significantly more expensive than sensor nodes, selecting the optimal number of UAVs and assigning the data collection tasks of each UAV evenly is crucial to shorten the time they spend.

Based on these observations, we propose a new WSAN protocol named divisive hierarchical clustering and multiple traveling salesman problem (DHC-MTSP), which is a combination of divisive hierarchical clustering (DHC) and the MTSP-based UAVs approach. It divides sensor nodes into two clusters using K-means++ in the hierarchical clustering process to balance and reduce energy consumption between clusters. We employ multiple UAVs as mobile collectors to gather data from CHs, as shown in Fig. 1c. Then, we introduce an intelligent strategy the improved partheno genetic algorithm (IPGA) [24] for multiple UAVs, which is rooted in solving multiple traveling salesman problems (MTSP). This approach is designed to allocate data collection tasks and optimize the flight paths of UAVs to ensure efficient latency during the transmission phase of our protocol.

The main contributions of this work are as follows.

- This work proposes a novel framework termed DHC-MTSP to monitor the environment in target areas, which integrates K-means++ with divisive hierarchical clustering for the first time to embrace cluster compactness and robustness. Experimental results demonstrate that DHC-MTSP outperforms other peers in terms of inter-cluster average energy consumption, network lifespan, and data collection latency.
- By fully considering the intra-cluster distances among clustered sensors, this work utilizes the sum of squared distances between member nodes and cluster centroid as a criterion for determining whether to initially split into clusters. It effectively balances energy consumption among clusters and enhances node energy efficiency. Then, to efficiently address the network latency issue and balance the network load, we employ multiple UAVs to collect data from CHs simultaneously and develop a modified IPGA to schedule UAVs. In addition, the algorithm incorporates Davies–Bouldin index (DBI) during scheduling UAVs to determine the optimal number of UAVs, effectively saving the cost of WSAN.
- We evaluate DHC-MTSP across various datasets and across different kinds of different algorithms (grid-based and K-means-based). The results demonstrate the consistent improvements achieved by DHC-MTSP, with larger improvements for energy saving enhancement.

The rest of this paper is structured as follows. Section 2 contains a review of previous related works. Section 3 introduces definitions and problem statements. Section 4 presents the proposed DHC-MTSP. Section 5 provides a performance evaluation, and Sect. 6 concludes the paper.

## 2 Related works

Recently, WSANs have found significant utility in the realms of environmental monitoring and industrial manufacturing. With the proposed clustering and data collection methods, the performance of WSANs has been improved. Even though many protocols have been proposed for WSANs, more related studies can be summarized in this section as follows.

### 2.1 Clustering schemes for WSN

Clustering is widely used in artificial intelligence and machine learning area. For clustering, the main task in WSNs is to group sensors into well-organized clusters based on a suitable distance [25]. The proposed protocols can be classified as density, model, center, connectivity and distributed-based clustering [26], but the classification of clustering methods is not easy [27].

Heinzelman et al. [28] proposed the low energy adaptive clustering hierarchy (LEACH) scheme to get the minimal expected count of clusters. However, it cannot achieve uniform CH distribution, which results in unbalanced energy consumption of CHs due to the different number of cluster members. The hybrid energy efficient distributed (HEED) [29] considered residual energy and cost when selecting CHs. HEED achieved fairly uniform CH distribution and can produce compact clusters. Some research literature has proposed machine learning clustering algorithms on WSNs, such as the K-means algorithm. The work in [17] applied the K-means algorithm on the LEACH routing protocol prior to CH selection and achieved good results in terms of reducing energy consumption and latency time while also increasing network lifetime and throughput. The second category applies the grid algorithm to divide the target area into some subareas and sensor nodes in each grid form a cluster. A virtual grid-based dynamic routes adjustment (VGDRA) [18] strategy was proposed which divided the sensing region into equal grids to ensure balanced energy consumption among sensor nodes. The header is then selected based on the Euclidean distance from the center of each grid. The study in [19] also divided the sensing area into a logical grid of a predefined size, determined by the nominal radio range of sensor nodes. Next, the CH of each grid is selected according to the distance from the center of nodes which is calculated by the K-means algorithm. Han et al. [20] further proposed a splitting-merging scheme in which the target area is self-organized into many subareas with varying grid sizes to ensure uniform data load, considering the maximum distance between sensor nodes and the maximum number of sensor nodes in each subarea.

Biabani et al. [30] partition all sensors into various tree-like clusters utilizing the Voronoi diagram and apply efficient buffer management for data collection. The performance improvement of this work is relatively limited due to the fixed target area and communication range considered. Amutha et al. [31] adopt a combined protocol to solve the hotspot problem. In this work, the CH selection is optimized by the butterfly optimization algorithm, and the energy-saving routing is performed by the ant colony optimization algorithm. However, this study neglects the system delay resulting from the randomness of routing path selection.

Previously proposed clustering algorithms are used in unevenly distributed WSNs without guaranteeing node compactness. In real-world applications, nodes are typically randomly distributed throughout the target area. In this case, if the clustering algorithm does not consider node compactness, using a uniform clustering strategy may result in nodes spreading out within a cluster and the cluster members dying rapidly due to long intra-cluster distances.

### 2.2 Mobile data collections

When comparing data collection using a mobile collector to a static sink, several advantages emerge, including balanced energy consumption, reduced latency, and connecting disconnected regions. Shah et al. [32] investigated mobility by having a mobile collector gather data from nearly all sensors and randomly walk around the sensing region. However, in some time-sensitive applications, random walking cannot guarantee effectiveness for a specified period of time. The work in [18] proposed a VGDRA strategy by dividing the region into several cells equally to balance energy consumption between the sensor nodes. The mobile collector moves along the edge of the sensing area region and collects the data from nearby cell headers, which dynamically adjust their routings by multi-hop. In addition, the work in [33] proposed a mobile sink (MS) visiting points-based gathering scheme for WSNs. This approach selects some common visiting points from the overlapping regions of nodes, and a traveling salesman problem (TSP) is ensured to create an optimal route for MS. By considering common visiting points, this method shortens the length of the visiting path of the MS, however, it leads to high data transmission latency. Gantassi et al. [19] propose the MDC-TSP-LEACH-K scheme, which integrates K-means with the grid clustering algorithm in the CH selection course. The trajectory of the mobile data collector (MDC) is also determined based on the TSP. The data are collected by MDC directly from CHs using single-hop transmission for energy efficiency and latency reduction. A novel and efficient data collection scheme based on Shark Smell Optimization is proposed by [15], which simultaneously optimizes the number of rendezvous points and mobile collectors. This algorithm is designed to be effective in both delay-tolerant and delay-sensitive networks. Furthermore, Han et al. [20] proposed an algorithm to schedule multi-mobile elements, which effectively reduces the data collection latency. Although mobile elements were used in these works, the delay time may have been increased due to the traveling time of the mobile collector and data transmission. Therefore, in this paper, multiple MDCs are employed in the WSAN to gather data, and MTSP is exploited to shorten data transmission time for multiple MDCs.

Recently, with the evolution of UAV technologies, utilizing UAVs as mobile sinks offers unparalleled flexibility, overcoming terrain constraints and exhibiting excellent adaptability. Chen et al. [34] applied virtual grids to divide the target area into equal size and select the highest energy sensors as CHs in every subregion. Then, they proposed a direct future prediction model to schedule the UAVs. This approach aims to optimize the total information gathered by UAVs while minimizing the frequency of UAV recharging. However, this study does not address strategies for extending the network's operational lifespan. Zhu et al. [35] focus on the energy consumption of UAVs in WSNs and adopt a reinforcement learning (RL) algorithm to design the trajectory of UAVs. This method achieves the goal of reducing overall energy consumption, especially for WSNs on a large scale. Nevertheless, this work neglects the balance of energy consumption. To minimize the energy consumption and optimize the communication rate of UAVs, a successive-convex approximation method is proposed to optimize the UAV's hovering positions in [36]. However, the energy consumption of these sensor nodes presents a potential risk to the lifespan of the network.

## 3 Definitions and problem statements

A WSAN system is modeled in the target area consisting of $m$ static sensor nodes, $k$ UAVs ($k \ll m$), and a sink node. In the target area of the 2D, the sensor nodes are randomly distributed. Sensor nodes can communicate with each other or UAVs within their transmission radius $s_r$. UAVs can fly freely over the target area for data collection from sensor nodes. The target area is divided into $n$ clusters, denoted by $C = \{c_1, c_2, ..., c_n\}$. The CHs are selected in each cluster, which are denoted as $CH = \{ch_1, ch_2, ..., ch_n\}$. We designate the set of clusters for the collector indexed by $i$ as $C_i$, while its associated CH group is denoted as $CH_i$. Both $C_i$ and $CH_i$ are subsets of the sets $C$ and $CH$, respectively. The CHs aggregate data from their cluster members. Each UAV can collect data from a set of clusters by visiting their corresponding CHs. Every CH establishes a time division multiple access (TDMA) schedule and informs member nodes in their corresponding subregions. The implementation of a TDMA schedule prevents data transmission collisions among the sensor nodes, ensuring smooth and efficient communication. During its designated time slot, a single sensor node exclusively transmits data to its corresponding CH, while other remaining sensor nodes within the same cluster stay in a sleep state. The notations utilized in this work are summarized in Table 1.

The models of energy dissipation and latency of UAV-assisted networks are described in the following subsections.

**Table 1** Summary of notations

| Notation | Description |
|---|---|
| $s_r$ | The communication radius of sensor nodes |
| $E_s$ | The energy dissipation of data transmission |
| $E_r$ | The energy dissipation of data reception |
| $d_0$ | The transmission distance threshold |
| $E_{\text{elec}}$ | The energy consumption of driving the electronics |
| $l$ | The length of data |
| $D_i$ | The traveling distance of the $i$-th UAV |
| $v$ | The movement velocity of the UAV |
| $T_i$ | The time cost by the $i$-th UAV gathering data |
| $T$ | The latency of the network |
| $E_{\text{fs}}$ | Free space power loss |
| $E_{\text{mp}}$ | Multi-path power loss |
| $d$ | The distance between sender and receiver |
| $m$ | The number of sensor nodes. |
| $n$ | The number of clusters |
| $c_i$ | The $i$-th cluster |
| $ch_i$ | The cluster head in $c_i$ |
| $k$ | The number of UAVs |
| $N^*$ | The optimal number of UAVs |
| $D_t$ | The total traveling distance of all collectors |
| $D_m$ | The maximum traveling distance of the collectors |

### 3.1 Energy consumption model

According to [20], the energy loss of a sensor node primarily comprises transmission consumption $E_s$ and reception consumption $E_r$. The distance between the sender and receiver is denoted as $d$. We will use the free channel model if $d$ is below the threshold $d_0$; otherwise, the multiple hops model will be utilized. The energy consumption of sensor nodes is based on the first-order radio dissipation model [37]. The amount of energy consumed during the process of sending and receiving $l$ bits of data can be calculated using Eqs. (1) and (2).

$$E_s = \begin{cases} l \times E_{\text{elec}} + l \times E_{\text{fs}} \times d^2, & d < d_0 \\ l \times E_{\text{elec}} + l \times E_{\text{mp}} \times d^4, & d \geq d_0 \end{cases} \tag{1}$$

$$E_r = l \times E_{\text{elec}} \tag{2}$$

In the given equation, the energy consumed per bit for transmission is denoted as $E_{\text{elec}}$. The transmitter amplifier characteristics are represented by $E_{\text{fs}}$ for free space and $E_{\text{mp}}$ for multiple hops environments. When the communication distance $d$ between the transmitter and receiver exceeds a predefined constant, $d_0$, which is calculated using Eq. (3), the multiple hops channel model ($d^4$ energy-loss) is adopted. Conversely, the free-space model ($d^2$ energy-loss) is utilized if the distance is below $d_0$.

$$d_0 = \sqrt{\frac{E_{\text{fs}}}{E_{\text{mp}}}} \tag{3}$$

Based on the equations defined above, it is evident that the energy loss of nodes is influenced by two factors: the length of the transmitted data and the distance between the transmitting nodes. Assuming that all sensor nodes transmit data of equal length, energy consumption is primarily determined by the distance raised to the second or fourth power.

### 3.2 Latency

The network latency [4] is the duration necessary for data transmission from the sensor node to the sink. In our study, latency encompasses the duration UAVs spent traveling and collecting data from CHs. The network latency is defined as the maximum time taken by UAVs. To accomplish this, we employ $k$ UAVs to visit CHs and collect data, with only one UAV visiting each CH per round, assuming the UAVs travel between CHs at a constant speed $v$, and the velocity is consistent for all UAVs. Furthermore, we assume that each sensor node sends and receives the same length of data and when the UAV flies to CH, the CH has already received data from its cluster members. The time required for the $i$-th UAV to gather data from corresponding CHs is denoted by $T_i$, as shown in Eq. (4).

$$T_i = \frac{D_i}{v} \tag{4}$$

where $D_i$ is the flying distance of the $i$-th UAV. There are $k$ UAVs gathering data from CHs simultaneously after the sensor nodes are clustered. Since the sensor nodes are unevenly distributed, the CHs are also unevenly distributed. Therefore, the time required for a UAV to gather information from its member CHs varies. The maximum time spent by UAVs gathering data is defined as network latency, as shown in Eq. (5).

$$T = \max\{T_1, T_2, ...T_k\} \tag{5}$$

## 4 Methods and experimental setup

Initially, the sensor nodes are distributed randomly across the sensing area, and data from the sensor nodes is collected using multiple UAVs. In step 1, the sensing area is organized into many clusters by DHC considering the maximum sum of squared distances between member sensor nodes and cluster centroid in each cluster. DHC is integrated with the K-means++ algorithm, which is useful for increasing cluster compactness, and reducing intra-cluster energy consumption. The sensor nodes closest to the cluster centroid are then selected as the CHs in the first round. Next, during the remaining rounds, CHs are dynamically chosen for each cluster, taking into account the candidate sensor nodes' residual energy and the distance to the centroid node in the cluster, which contributes to the balanced energy consumption of sensor nodes. In step 2, the data collection tasks of the UAVs are assigned based on the selected CHs of all clusters using the modified IPGA method for solving MTSP. In Sects. 4.1 and 4.2, a clustering method, DHC, based on K-means++ integrated with the MTSP method is proposed to effectively extend the lifetime of sensor nodes and decrease the data collection latency. Section 4.3 describes how to solve MTSP in detail, and the basis for selecting the optimal number of UAVs is introduced in Sect. 4.4.

### 4.1 DHC based on K-means++

The hierarchical algorithm can organize the data objects into a tree-like structure known as a hierarchy. In this paper, hierarchical clustering is executed through divisive clustering. Divisive procedures start with a whole cluster and then form a sequence by successively splitting clusters.

As previously stated, sensor nodes are distributed unevenly in the sensing area, consequently, we employ K-means++ instead of conventional methods for their clustering. K-means++ is a more enhanced version of the K-means clustering algorithm with the purpose of clustering centers that minimize the intra-class variance [38]. Specifically, it can minimize the squared distance between CH and its member sensor nodes. K-means++ modifies the centroid selection step in K-means by relating the centroid to the chosen centroid. This modification reduces randomness and results in stable performance.

K-means++ divides the sensor area into many Voronoi cells with the goal of grouping $m$ data points $\{x_1, x_2, ..., x_m\}$ into $n$ sets $\{S_1, S_2, ..., S_n\}$ [39]. These sets are designed to minimize the overall mean squared point-to-centroid distance within each cluster, also referred to as the mean squared error. The optimization objective of K-means++ can be expressed as

$$\{S_1, S_2, ..., S_n\} = \arg\min \frac{1}{m} \sum_{i=1}^{n} \sum_{x \in S_i} \|x - \mu_i\|^2 \tag{6}$$

where $x$ represents the location information, and $\mu_i$ denotes the average position of the data points in cluster $S_i$.

Figure 2 illustrates the process of the DHC algorithm. Initially, we take all the nodes as a single cluster. The K-means++ algorithm is then applied to partition this cluster into two distinct clusters, referred to as Cluster 1 and Cluster 2. Next, we calculate the sum of the squared distances between member nodes and the centroid of their cluster. The cluster with the highest sum of squared distances (Cluster 2) is selected as the target cluster for further refinement. Subsequently, the target cluster is re-clustered into two clusters, which we designate as Cluster 2 and Cluster 3. The sum of the squared distances is recalculated for these new clusters, and the cluster with the highest value is chosen as the target cluster (Cluster 1). This iterative process continues until the specified cutoff condition $d_{th}$ is met. Keeping the sum of squared distances within clusters below $d_{th}$ helps maintain a relative balance between node density and the number of nodes in each cluster. In areas with higher node density, the cluster can accommodate a larger number of nodes, while in regions with lower density, the number of nodes is reduced accordingly. According to Eq. (1), the energy consumption of the node is related to the square of the distance between the transmitting nodes, and appropriately setting the threshold can effectively balance energy consumption across clusters. Overall, the detailed algorithm of DHC is summarized in Algorithm 1.

### 4.2 Multi-Depot MTSP

Through the DHC operation, the sensor nodes are organized into different clusters, and CHs are selected. The sensor nodes in WSANs transmit their data to their respective CHs. Subsequently, the UAVs can be utilized to gather data by visiting the CHs as data collectors. Since the assignment should be evenly allocated among UAVs and network

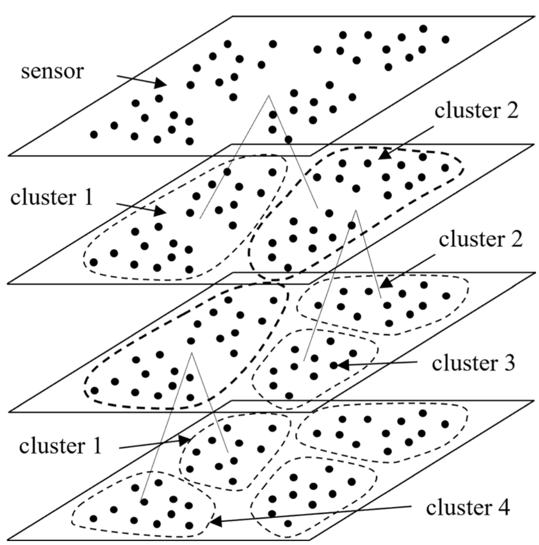


**Fig. 2** Illustration of divisive hierarchical clustering

gathering latency should be minimized as much as possible, we can consider this issue a problem involving MTSP.

Multi-Depot MTSP is a variation of the well-known TSP in which multiple depots are used instead of a single depot or starting point. In MTSP, a set of salesmen (or vehicles) start from different depots and visit a set of customers to minimize the total travel distance [24]. The multiple-depots MTSP is shown in Fig. 3.

**Algorithm 1** Divisive Hierarchical Clustering

**Input:**
The position of sensor nodes $\{x_1, x_2, \ldots, x_m\}$, the threshold of the maximum sum of squared distance between sensor nodes in each cluster: $d_{th}$;
**Output:**
Clusters $\{c_1, c_2, \ldots c_n\}$ with uniform energy consumption;
1: Initialize the number of clusters $n = 1$;
2: Calculate the sum of squared distances between member nodes and cluster centers in the $i$-th cluster: $\hat{d}_i, i \in [1, n]$;
3: Calculate the maximum value of $\hat{d}_i$ in the $max$-th cluster: $\hat{d_{max}}$;
4: **while** $\hat{d_{max}} > d_{th}$ **do**
5:     Divide the $max$-th cluster into 2 clusters based on K-means++;
6:     Update the number of clusters: $n = n + 1$;
7:     Calculate the $\hat{d}_i$ of each cluster;
8:     Calculate the maximum value of $\hat{d}_i$ in the $max$-th cluster: $\hat{d_{max}}$.
9: **end whilereturn** The optimal clustering result.

In WSANs, multiple UAVs assist in collecting data from CHs and the MTSP has the same properties. Hence, we can use the MTSP algorithm to solve the problem of UAV task allocation. The goal of minimizing the travel distance of all single traveling salesman is compatible with our gathering latency model.

### 4.3 IPGA for the MTSP

The genetic algorithm (GA) is a stochastic search technique introduced by Professor Holland inspired by the laws of biological evolution [40]. The traditional GA algorithm includes encoding the solution as individuals, generating the initial generation, and then generating offspring by mutation, crossover and recombination, natural selection, and other operators to achieve the optimal solution. In this study, we choose the two-part chromosome encoding, which is based on the breakpoint sets [41], to solve the MTSP. An example of an encoded solution in our research is shown in Fig. 4.

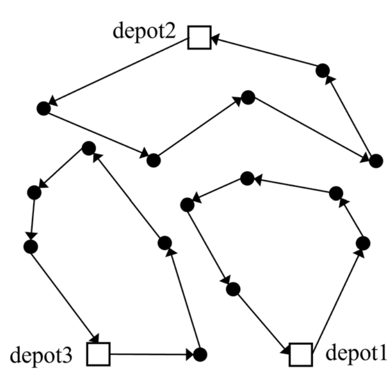


**Fig. 3** Multiple-depots multiple traveling salesman problem

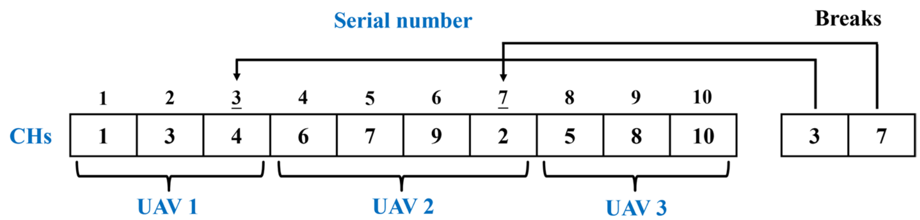


**Fig. 4** Example of an encoded solution representation for 10 CHs MTSP with 3 UAVs

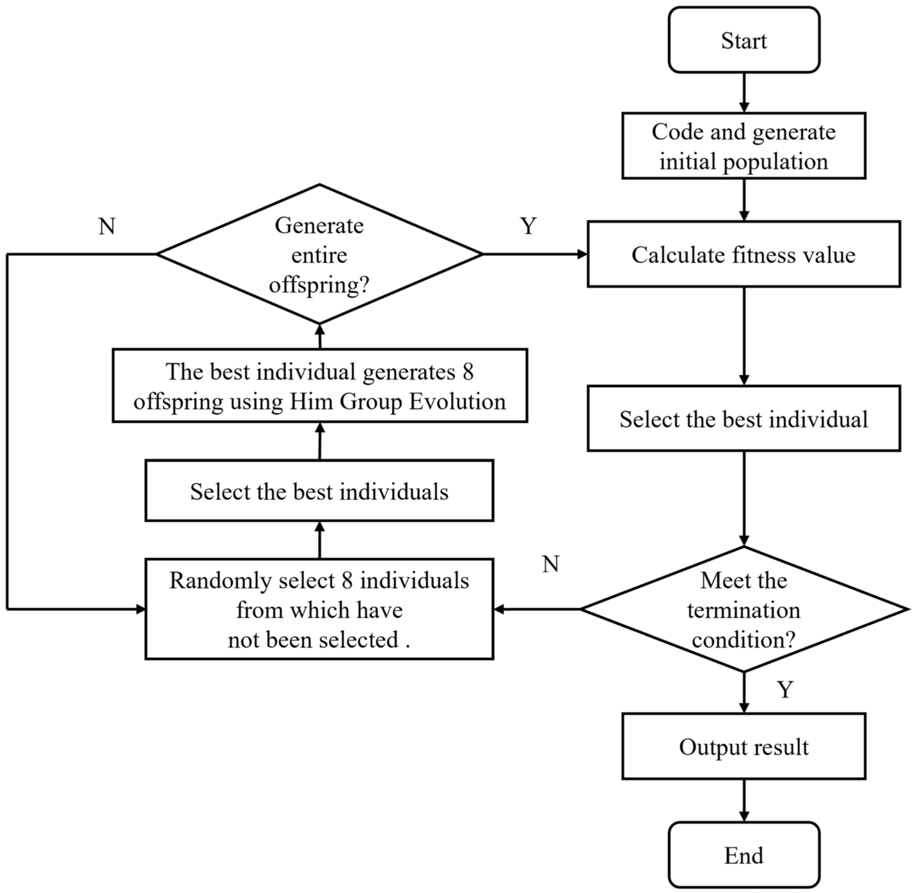


**Fig. 5** Flowchart of modified IPGA processing

Based on [42], we propose a modified IPGA algorithm in which cycle shifting an individual to the left or right is considered the same operation.

The flowchart of the modified IPGA is shown in Fig. 5. Algorithm 2 summarizes the specific algorithm implementation steps. We employ Him Group Evolution to generate offspring. The best individual is selected and subjected to several mutations to produce new individuals that are expected to be relatively high-quality solutions. This approach facilitates the IPGA in finding the optimal solution more effectively, thereby enhancing the efficiency and convergence ability of the modified IGPA. In details, the evolutionary processes of Him Group Evolution are outlined as follows (Fig. 6):

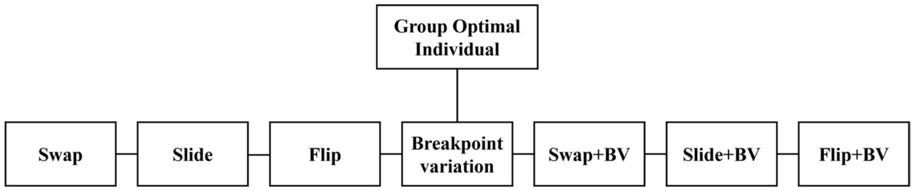


**Fig. 6** Him group evolution

$$\text{Swap: } [x_1x_2x_3x_4x_5x_6] \xrightarrow{x_1 \text{ and } x_4} [x_4x_2x_3x_1x_5x_6]$$
$$\text{Slide 1 unit: } [x_1x_2x_3x_4x_5x_6] \xrightarrow{x_1 \text{ and } x_3} [x_3x_1x_2x_4x_5x_6]$$
$$\text{Flip: } [x_1x_2x_3x_4x_5x_6] \xrightarrow{x_1 \text{ and } x_4} [x_4x_3x_2x_1x_5x_6]$$
$$\text{Breakpoint variation: } [x_1x_2x_3|x_4x_5x_6] \xrightarrow{x_3 \text{ to } x_4} [x_1x_2x_3x_4|x_5x_6]$$

**Algorithm 2** The process of modified operation of IPGA

**Input:**
Individuals set $P$, Individuals size $P_s$, Maximum iterations $g_{max}$, fitness function $f$;
**Output:**
The best individual $P_{best}$, best fitness $f_{best}$;

Initialize $P$;
**for** $g = 1$ to $g_{max}$ **do**
  Find the best individual $P_{best}$ and its fitness best $f_{best}$ of the contemporary individuals;
  **for** $i = 1 : 8 : P_s$ **do**
    **for** $j = 1$ to $8$ **do**
      Calculate the fitness of the $(i + j)$-th individual $f_{ij}$;
      Find the best individual $P_{ibest}$ and its best fitness $f_{ibest}$;
      Generate new 8 offsprings set $P_{inew}$ according to $P_{ibest}$ utilizing Him Group Evolution;
    **end for**
    **if** $f_{ibest} < f_{best}$ **then**
      Update the best individual $P_{best}$ and its fitness $f_{best}$;
    **end if**
    $i = i + 1$
  **end for**
  $g = g + 1$
**end forreturn** The best individual $P_{best}$, the best fitness $f_{best}$.

The objective function of MTSP is usually to minimize the sum of the lengths of all travelers' paths, or to minimize the longest of all travelers' paths. In our research, we consider these two factors together in desiging our objective function, to minimize both the sum and the longest length of all UAVs' paths. The sum of the lengths of $k$ UAVs' paths $D_t$ can be represented as Eq. (7), and the longest of $k$ UAVs' paths $D_m$ can be represented as Eq. (8).

$$D_t = \sum_{i=1}^{k} \left( \sum_{j=1}^{n_i - 1} x_{j,j+1} + x_{n_i,1} \right) \tag{7}$$

$$D_m = \max\left(\sum_{j=1}^{n_i-1} x_{j,j+1} + x_{n_i,1}\right) \quad (8)$$

where $x_{j,j+1}$ represents the distance between CH $j$ and $j+1$, while $n_i$ denotes the total number of CHs the $i$-th UAV needs to visit.

To consider both the total distance and the degree of equilibrium, we must first unify the dimensions. We discover that $D_t$ is of the same dimension as $k \times D_m$. Therefore, we define the objective function as Eq. (9).

$$f(x) = \alpha \times D_t + (1-\alpha) \times k \times D_m \quad (9)$$

where $\alpha$ is the weighting coefficient that regulates the weights of $D_t$ and $D_m$. We can achieve the optimal solution by minimizing the objective function. Algorithm 3 summarizes the approach employing MTSP to optimize the trajectory of UAVs.

**Algorithm 3** Task allocation and UAVs path planning

**Input:**
The position of CHs $\{ch_1, ch_2, ..., ch_n\}$, the number of UAVs $k$;
**Output:**
Groups of CHs $\{CH_1, CH_2, ..., CH_k\}$, the optimal trajectory of UAVs;
Code and generate initial population;
Evaluate the fitness value of each individual using Eq. (9);
Record the best individual of the population;
**while** the termination condition is not met **do**
  Select the optimal individual from every 8 individuals of contemporary population;
  The optimal individual generates 8 offspring as the next generation;
  One offspring is itself and the other 7 offspring are generated using Him Group Evolution;
  Generate all new offsprings;
  Evaluate the fitness value of each individual using Eq. (9);
  Update the best individual of the contemporary population;
**end whilereturn** The optimal grouping solution and the trajectory

## 4.4 Selecting the number of collectors

After implementing the DHC operation, clusters are generated and CHs are determined. Then we can assign data collection tasks and plan the path of UAVs. Collecting data with more UAVs reduces latency but raises device costs. David L. Davies and Donald W. Bouldin introduced Davies–Bouldin index (DBI) to evaluate the results of clustering algorithms [43]. The DBI is based on the ratio of the intra-cluster distance to the inter-cluster distance; it is defined as Eq. (10).

$$DB(k) = \frac{1}{k}\sum_{i=1}^{k} \max_{j\neq i}\{D_{i,j}\} \quad (10)$$

where $D_{i,j}$ represents the ratio of the intra-cluster to inter-cluster distance between the $i$-th and $j$-th groups. In Eq. (11):

$$D_{i,j} = \frac{\bar{d}_i + \bar{d}_j}{d_{i,j}} \tag{11}$$

$\bar{d}_i$ denote the mean distance from each member in the $i$-th group to its centroid. Likewise, $\bar{d}_j$ denotes the mean distance from each member in the $j$-th group to its centroid. Additionally, $d_{i,j}$ signifies the Euclidean distance between the $i$-th and $j$-th groups. The maximum value of $D_{i,j}$ represents the worst intra-cluster to inter-cluster ratio of the $i$-th group. The ideal clustering solution is characterized by the minimum DBI value.

Task allocation assigns CHs to different groups for multiple UAVs. The group formation process is similar to that of the clustering operator. Therefore, simulation is conducted about different numbers of UAVs to assist in collecting data, and the DBI is calculated for different groups of CHs. Then, we employ the fitness function Eq. (12) to determine the optimal number of UAVs.

$$N^* = arg\ minDB(k) \tag{12}$$

## 5 Results and discussion

In this section, we analyze the performance of our proposed DHC-MTSP protocol and compare it with several established algorithms, such as the VGDRA [18], mobile data collectors—traveling salesman problem—low energy adaptive clustering hierarchy K-means (MDC-TSP-LEACH-K) [19], and splitting merging based automatic scheduling scheme (SMASS) [20]. We utilize MATLAB 2021b on a Windows 11 platform, powered by an Intel i5-12500 H CPU. To ensure consistency, different numbers of sensor nodes 100, 200, 300, and 400, respectively, are randomly deployed within a 200 m $\times$ 200 m area. The transmission radius of each sensor node is 100 ms. Table 2 describes the other relevant parameters in the WSANs system.

For the DHC operation, the threshold of the maximum value Path planning in each cluster is set as $20 \times m$ for $m = 100, 200, 300, 400$, respectively. Furthermore, in fairness to the comparison of the above methods, VGDRA, MDC-TSP-LEACH-K, and SMASS employ the same parameters as DHC-MTSP for simulation. General parameters include various factors such as the dimensions of the target area, the quantity of deployed sensor nodes, the communication range of sensor nodes, and the velocity of UAVs. The

**Table 2** Simulation parameters

| Settings | Values |
|---|---|
| Simulation area | 200 m $\times$ 200 m |
| Number of nodes ($m$) | 100, 200, 300, 400 |
| Transmission radius ($s_r$) | 100 m |
| Initial power ($E_0$) | 0.1 J |
| Data length | 4000 bit |
| $E_{\text{elec}}$ | 50 nJ/bit |
| $E_{\text{fs}}$ | 10 pJ/bit/m$^2$ |
| $E_{\text{mp}}$ | 0.0013 pJ/bit/m$^4$ |

comparative experiments have a uniform initialization state, including identical initial locations for all sensor nodes, and the state of UAVs. Furthermore, specific parameters pertinent to VGDRA, MDC-TSP-LEACH-K, and SMASS are assigned values recommended by the relevant literature. For the MTSP operation, the individual size is set to 200, and the number of iterations is set to 5000. The DHC-MTSP result presented here is an average of the results of 100 simulations due to the randomness of the centroid position selected by the K-means++ algorithm.

### 5.1 The optimal number of UAVs

As depicted in Fig. 7, the network employs 4 UAVs instead of other numbers of UAVs when the minimum value of DBI occurs. To select the best *k* value, DBI is calculated as the ratio of the sum of intra-cluster distances to the inter-cluster distances. The smaller the intra-cluster distances and the bigger the inter-cluster distances are, the better the group members' compactness. Thus, the path connecting group members becomes shorter. In addition, network latency can be reduced. After setting the cutoff threshold for the divisive operation in the DHC process, the number of clusters remains approximately the same, even though the number of nodes varies in four scenarios. As a result, we get the same minimum value of DBI across the four scenarios.

### 5.2 Analysis of the weighting coefficient

In our study, we define network latency as the maximum time taken by the UAVs. A shorter maximum flight distance among the UAVs corresponds to lower latency. We can achieve the optimal solution by minimizing the fitness function (9). The weighting coefficient $\alpha$ plays a crucial role in this function. To select appropriate parameter values, we tested four different values for the simulation.

As shown in Fig. 8, we utilize four UAVs to visit the CH for data gathering. The figure illustrates the maximum flight distance (MAX Dis) of the UAVs and demonstrates

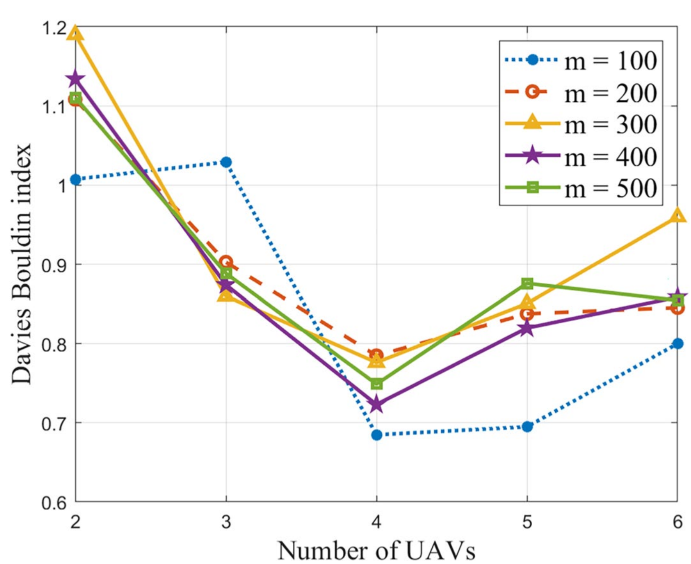


**Fig. 7** Comparison of Davies–Bouldin index of different number of UAVs

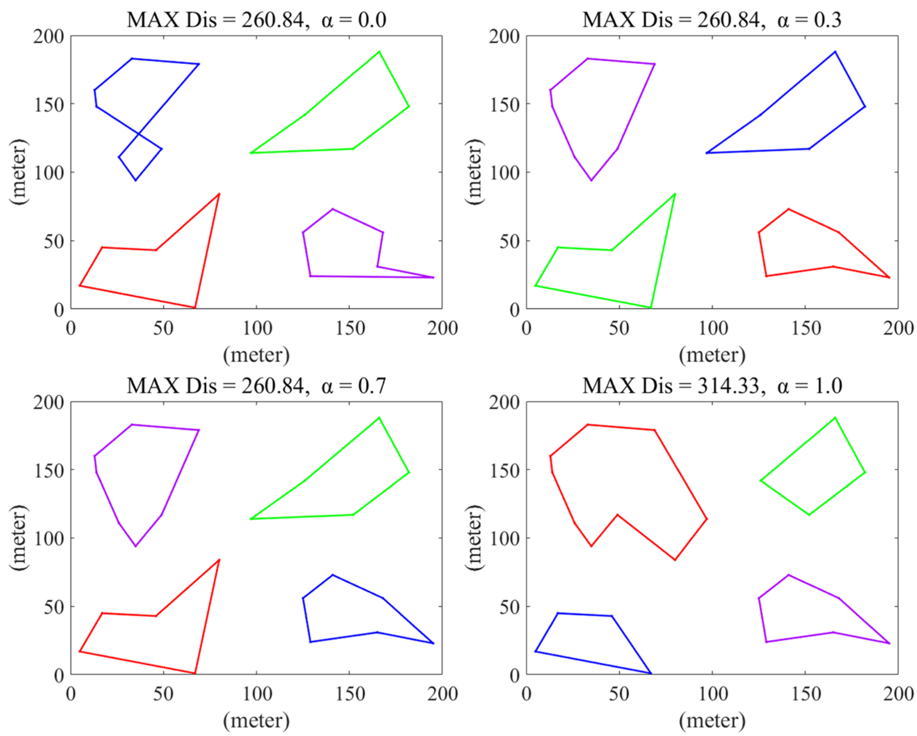


**Fig. 8** The influence of the weighting coefficient $\alpha$ on UAV flight paths

the effect of different values of $\alpha$ on their trajectories. When $\alpha$ is set to 1, the optimization function focuses solely on minimizing the total flight distance of the UAVs. This approach can result in uneven flight distances among the UAV group, leading to an increased maximum flight distance and, consequently, prolonged data collection times. Conversely, when $\alpha$ is set to 0, the optimization emphasizes reducing the maximum flight distance of individual UAVs without considering the flight paths of other UAVs. This can result in overlapping trajectories, which ultimately wastes UAV resources. When considering both the total distance and the maximum flight distance, setting the parameter to values of 0.3 or 0.7 yields optimal results.

### 5.3 Effectiveness of the proposed DHC-MTSP method

This research conducts simulations to validate the efficacy of the proposed DHC-MTSP approach in achieving energy consumption balance between clusters. The sensor nodes are clustered using a self-organizing DHC method, which takes into account the maximum sum of squared distances between sensor nodes and the cluster centroid in each cluster. The validity of the clustering results can be assessed by examining the energy consumption proportion between clusters. The energy consumption primarily includes the aggregation of data by CHs within their respective members and the transmission of data from common sensor nodes to their corresponding CHs. In general, a more balanced energy consumption among sensor nodes results in roughly equal energy consumption between clusters, hence, extending the network's lifetime. To demonstrate the efficacy of the proposed DHC-MTSP approach comprehensively, simulations are conducted across four scenarios characterized by varying numbers of sensor nodes, specifically 100, 200, 300, and 400.

Figure 9 shows each cluster's energy consumption proportion based on different sensor node numbers. The proportion of energy consumption is defined as the energy consumption ratio to the sum of the initial energy of all sensor nodes in one cluster when the network's lifetime is over. The target area is clustered into many clusters using the K-means++ algorithm according to the maximum sum of squared distance between sensor nodes and the cluster centroid. As shown in Fig. 9, the DHC method effectively achieves energy consumption balance between clusters. Moreover, only a few clusters have a slightly lower proportion of energy consumption. This is because of the random selection of CHs by K-means++ in the DHC operation. Some CHs have fewer nodes around them, so these clusters have fewer member nodes consuming less energy. Nevertheless, the vast majority of clusters' energy consumption is higher than 90 percent indicating an approximate balance of energy consumption between these clusters.

Figure 10 illustrates the flying time variations of each UAV with varying numbers of sensor nodes based on the proposed DHC-MTSP. The latency can be used as an evaluation parameter to evaluate the proposed DHC-MTSP method, and the efficiency of the proposed approach in achieving latency balance among UAVs is also demonstrated. After implementing the DHC operation, we utilize the MTSP algorithm to schedule UAVs, which can synchronously assign data collection tasks, and optimize the path of each UAV. Several UAVs concurrently perform data-gathering tasks following their optimized flight paths. Network latency is defined as the maximum time

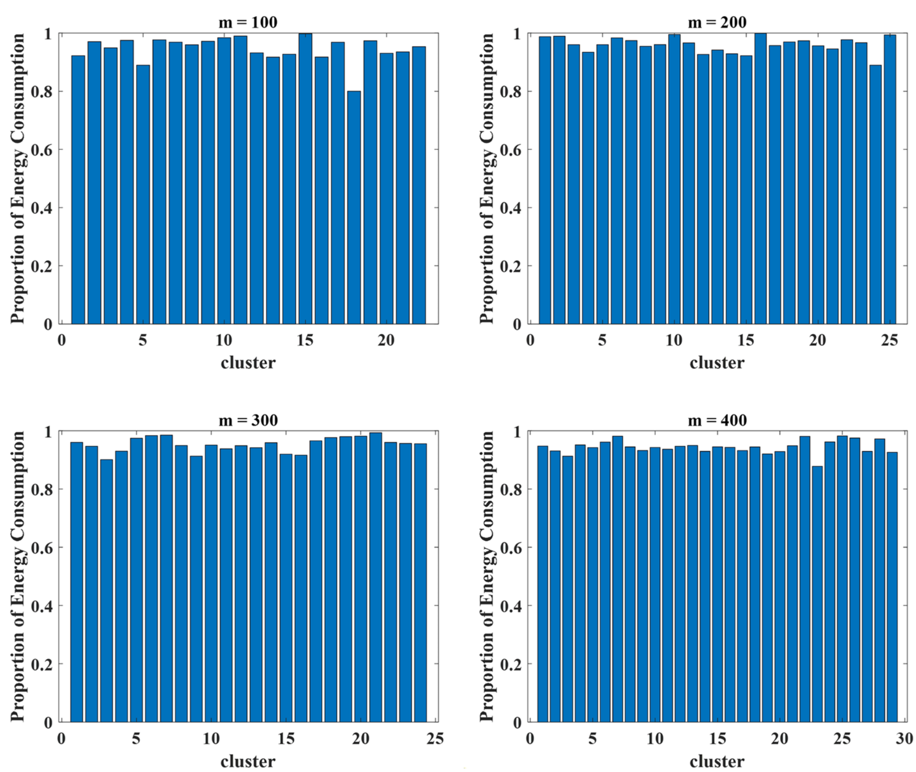


**Fig. 9** Energy utilization factor of each cluster on different numbers of sensor nodes

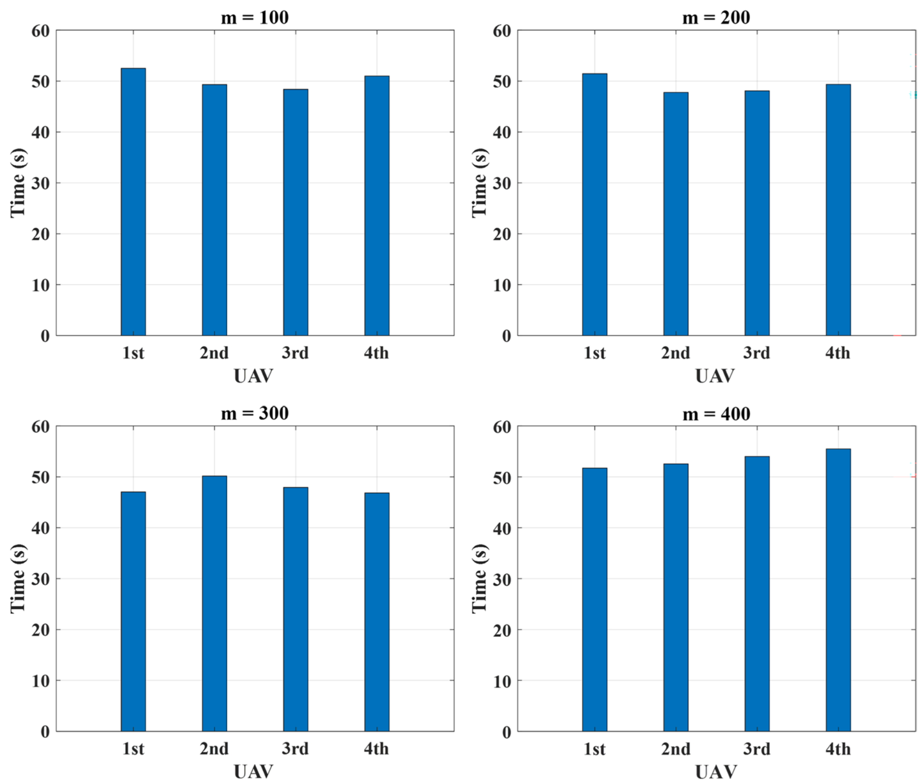


**Fig. 10** The latency of each UAV on different numbers of sensor nodes

UAVs consume in each round. In the actual application environment, it is required not only to reduce the maximum time consumed by UAVs in each round but also to ensure that the time spent by each UAV is as short as possible. Finally, by definition of latency, our goal is to minimize the maximum flying distances of all UAVs at each round. The latency is shorter when the flying distance is more balanced for each UAV. MTSP can achieve this goal by allocating data-gathering tasks and planning the path properly and effectively. According to the result of Fig. 7, the number of UAVs is set to 4.

In practical applications, when the number of sensors is constant, increasing the number of clusters results in fewer nodes per cluster. Specifically, as we group nodes into more clusters by DHC, the number of nodes per cluster decreases, accordingly shortening data transmission distances between the relay nodes and the CH, or between nodes and the CH. This configuration enhances energy efficiency and extends the overall network lifespan. However, a larger number of clusters also means that the mobile collectors should visit more CHs to gather data. Consequently, his can lead to increased network delay due to the longer paths required for data transmission. Conversely, if we reduce the number of clusters, each cluster will include more nodes. This can result in higher energy consumption for member nodes within clusters, especially when using multi-hop forwarding or transmitting data over long distances to mobile collectors. While this may shorten the network's lifetime, it significantly reduces the number of CHs that the mobile collectors need to visit, thereby decreasing network latency.

### 5.4 Discussions

To validate the validity of the proposed DHC-MTSP method, various existing methods (VGDRA, MDC-TSP-LEACH-K, and SMASS) are compared on the lifetime and latency of the network. Lifetime refers to the time extending from network initialization until the first sensor node energy runs out, measured in rounds. The latency of a multi-mobile collectors' network refers to the duration required for the longest time taken by the UAV of the UAV tour, which assists in collecting data from CHs in one round.

VGDRA is a virtual grid structure method. Based on the number of nodes, VGDRA partitions the target area into multiple subareas of the same size. The mobile collector moves around the sensing area, and the CH of each subarea dynamically adapts its routing based on the collector's real-time position. The MDC-TSP-LEACH-K method also utilizes the grid sizes clustering on the WSN area in its first step. It divides the whole target area into logical grids of a predefined size determined by the nominal radio range. The second step of MDC-TSP-LEACH-K applies the K-means algorithm on the nodes in each grid to obtain centroid coordinates. Then CHs are selected according to their distance from the centroid. The target area is divided into several subareas based on the maximum distance between sensor nodes and the maximum number of sensor nodes in each subarea, using the SMASS technique. Then, the CHs are selected in every round, considering the distance between sensor nodes and their residual energy for each subarea. Finally, SMASS uses the GA combined ant colony optimization (ACO) to allocate tasks and optimize the path of collectors.

The simulations are conducted under four scenarios, and different numbers of sensor nodes are deployed in each scenario: 100, 200, 300, and 400. For a fair comparison, SMASS and DHC-MTSP employ 4 mobile collectors in these simulations. In the methods relying on multiple mobile collectors, the number of collectors is set to 4, and mobile collectors all gather data directly from the CHs.

Figures 11 and 12 show the comparison between the proposed DHC-MTSP method and other methods on the lifetime and latency of the network, respectively. As illustrated in Fig. 11, VGDAR has the shortest lifetime. As the optimal number of subareas

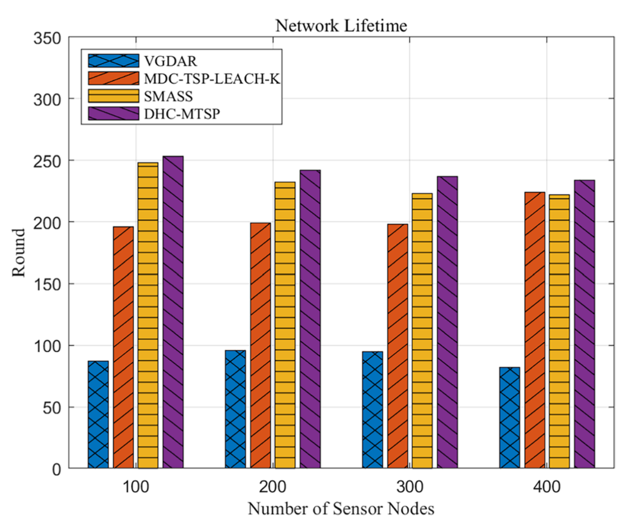


**Fig. 11** Comparison of the network lifetime of different methods with different number of sensor nodes

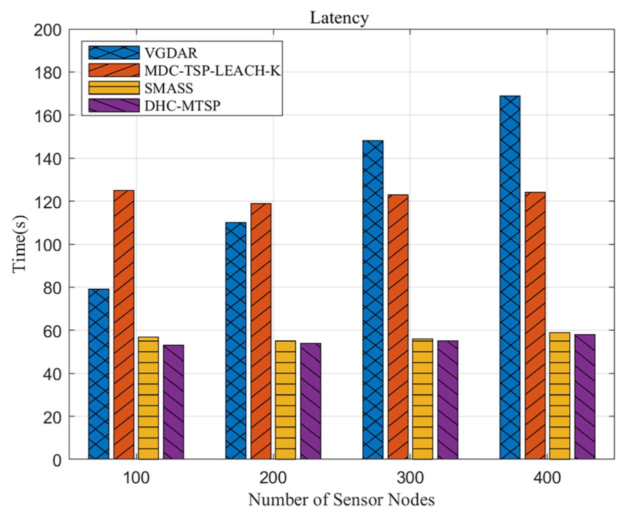


**Fig. 12** Comparison of latency of different methods with different number of sensor nodes

of VGDAR is determined, the number of clusters is determined. Each cluster has a large number of node members, and the distance between cluster members is not considered, which consumes more energy. Moreover, the CHs rotation of VGDAR depends on the residual energy level of nodes and their distance to the mid-point of the subarea. In addition, a certain residual energy threshold determines when to re-elect the CH and influences the network lifetime. MDC-TSP-LEACH-K has the same performance on lifetime when the number of nodes increases from 100 to 300 and shows a slightly longer lifetime than SMASS when $m = 400$. The latency of MDC-TSP-LEACH-K is almost the same in four scenarios. It has a longer latency than that of VGDAR when $m = 100$ and $m = 200$, but VGDAR outperforms it when $m = 300$, $m = 400$. The reason for this is that the division of the whole WSN area in MDC-TSP-LEACH-K is determined by the nominal radio radius ($s_r$). Furthermore, in the first two scenarios, the number of nodes is relatively small, and the distribution is uneven, making it challenging to achieve balanced energy consumption of CHs and even allocation of tasks to collectors. However, in the last scenario when $m = 400$, the increase in the number of sensor nodes results in the increase in distribution density of sensor nodes, and the MDC-TSP-LEACH-K protocol uses K-means and grid clustering algorithm, which decreases energy consumption in the CH election phase, then MDC-TSP-LEACH-K and SMASS have the same performance.

Except for path planning of mobile sinks, the time complexity of VGDAR is $O(m \times n)$, where $n$ represents the number of clusters. MDC-TSP-LEACH-K has a complexity of $O(m \times rd)$, for which $rd$ is the lifetime of the WSN. Both SMASS and the proposed algorithm, DHC-MTSP, have a time complexity of $O(m \times n \times rd)$. Table 3 presents the average simulation time of each algorithm for one round after executing it 100 times. In the initial round, the sensor nodes closest to the cluster centroid are chosen as the CHs. Subsequently, CHs are dynamically selected for each cluster in the following rounds. The simulation time per round is computed to assess the computational cost of various algorithms. VGDAR incurs minimal computational cost since the algorithm primarily involves planning a path for a mobile sink. With an increase in the number of nodes, the number of clusters also rises, resulting in

**Table 3** The results of simulation time

| Algorithms | Simulation time (s) | | | |
|---|---|---|---|---|
| | **100** | **200** | **300** | **400** |
| VGDAR | 3.00 | 3.18 | 3.26 | 3.26 |
| MDC-TSP-LEACH-K | 3.10 | 3.17 | 3.17 | 3.18 |
| SMASS | 3.44 | 3.42 | 3.41 | 3.42 |
| DHC-MTSP | 3.37 | 3.46 | 3.42 | 3.43 |

a proportional escalation of computational cost for path planning. Similarly, MDC-TSP-LEACH-K utilizes a UAV for data collection, but the number of clusters remains constant and depends on the transmission radius ($s_r$). Consequently, the computational cost remains relatively consistent across different scenarios. Conversely, SMASS and DHC-MTSP demonstrate the highest computational cost due to the simultaneous computation of paths for multiple UAVs in these algorithms. Although DHC-MDSP is slightly more complex than MDC-TSP-LEACH-K and VGDAR, the higher computational expense can be justified by its improved network lifetime performance, as evidenced in the simulation.

Based on the performance depicted in Figs. 11 and 12, both SMASS and the proposed DHC-MTSP approach can effectively extend the network lifetime and shorten the latency. This is because their CHs do not consume extra energy for forwarding data from other clusters. The CHs only perform tasks data aggregation within their clusters. The number of nodes and the farthest distance between them in each subarea is restricted to a smaller range in SMASS by using the sensing area splitting-merging scheme to divide the WSAN area. This can save the energy consumption in each subarea to a certain extent. However, the proposed DHC-MTSP method shows the longest network lifetime among the four methods. DHC-MTSP clusters the sensor nodes using the DHC operation based on K-means++, and the divisive criterion is according to the sum of squared distances between member nodes and their cluster centroid. The cluster with the maximum sum of squared distances between sensor nodes from all current clusters was selected as the target area, and the target area was then divided into two clusters using the K-means++ algorithm, which can guarantee the compactness of the clusters and balance the energy consumption between clusters. The compactness of the clusters can also reduce energy consumption in data transmission between CH and its member nodes. Therefore, the reduction and balance of energy consumption in clusters contribute to prolonging the network lifetime. Furthermore, the MSTP strategy optimizes the allocation of data collection tasks and the route of multiple UAVs, thereby reducing network latency. Collectors in SMASS gathering balanced data along optimized paths based on GA and ACO have the same effect. In terms of computational cost, DHC-MTSP incurs a negligible increase in computational overhead in exchange for extending the overall network lifetime and reducing system latency due to the utilization of multiple UAVs. This trade-off is highly beneficial and justifiable.

In conclusion, the proposed DHC-MTSP method demonstrates superior performance in enhancing energy saving and reducing latency by effective balance of sensor node energy consumption and the implementation of automated UAV scheduling.

## 6 Conclusion

In this paper, we propose a DHC-MTSP method aimed at enhancing energy savings and optimizing task allocation for UAVs during data collection, which is essential for addressing the well-known sink hole problem in WSAN. Our research demonstrates that the iterative application of the K-means++ algorithm for sensor clustering leads to more effective node groupings, which facilitate a balanced reduction in energy consumption and further prolong the network's lifetime. Additionally, by employing multiple UAVs for data collection and utilizing an improved IPGA algorithm for path planning, we significantly reduce the time required for data collection. This optimization is particularly beneficial in delay-sensitive real-time applications, allowing the network to respond swiftly to changes and make timely decisions.

The iteration of K-means++ within a hierarchical framework not only expands its application area, but also sheds light on broadening the field of application for other clustering methods. These applications can achieve various effective clustering outcomes by adjusting the iteration cutoff condition. Specifically, in this study, the DHC method iteratively divides the target cluster into two sub-clusters using K-means++ until the cutoff condition is met. In future research, we plan to optimize the cutoff condition for this iterative division process, in an attempt to additionally improve the clustering performance of the proposed method.

**Abbreviations**

| | |
|---|---|
| WSAN | Wireless sensor actuator network |
| UAV | Unmanned aerial vehicles |
| CH | Cluster head |
| DHC-MTSP | Divisive hierarchical clustering and multiple traveling salesman problem |
| IPGA | Improved partheno genetic algorithm |
| DBI | Davies–Bouldin index |
| LEACH | Low energy adaptive clustering hierarchy |
| HEED | Hybrid energy efficient distributed |
| VGDRA | Virtual grid-based dynamic routes adjustment |
| SMASS | Splitting-merging-based automatic scheduling scheme |
| MS | Mobile sink |
| TSP | Traveling salesman problem |
| MDC | Mobile data collectors |
| MTSP | Multiple traveling salesman problem |
| RL | Reinforcement learning |
| TDMA | Time division multiple access |
| GA | Genetic algorithm |
| ACO | Ant colony optimization |

**Author contributions**

XZ performed the algorithm design and performance evaluations, and was a major contributor in writing the manuscript. Y-LW contributed to the conception of the study, as well as providing substantial revisions to the manuscript. HB analyzed the proposed method and interpreted the simulation results regarding the energy savings and delay reduction. All authors read and approved the final manuscript.

**Availiability of data materials**

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

## Declarations

**Competing of interests**

The authors declare that they have no conflict of interests.

Received: 3 September 2024 Accepted: 16 December 2024
Published online: 03 January 2025

**References**


1. L. Chettri, R. Bera, A comprehensive survey on internet of things (IoT) toward 5G wireless systems. IEEE Internet Things J. **7**, 16–32 (2020). https://doi.org/10.1109/JIOT.2019.2948888
2. B. Zhu, E. Bedeer, H.H. Nguyen, R. Barton, J. Henry, Improved soft k-means clustering algorithm for balancing energy consumption in wireless sensor networks. IEEE Internet Things J. **8**, 4868–4881 (2021). https://doi.org/10.1109/JIOT.2020.3031272
3. H. Mohapatra, B.K. Mohanta, M.R. Nikoo, M. Daneshmand, A.H. Gandomi, MCDM-based routing for IoT-enabled smart water distribution network. IEEE Internet Things J. **10**(5), 4271–4280 (2023)
4. M. Zhao, Y. Yang, C. Wang, Mobile data gathering with load balanced clustering and dual data uploading in wireless sensor networks. IEEE Trans. Mob. Comput. **14**, 770–785 (2015). https://doi.org/10.1109/TMC.2014.2338315
5. H. Mohapatra, A.K. Rath, Fault tolerance in WSN through PE-LEACH protocol. IET Wirel. Sens. Syst. **9**, 358–365 (2019)
6. A. Lipare, D.R. Edla, V. Kuppili, Energy efficient load balancing approach for avoiding energy hole problem in WSN using grey wolf optimizer with novel fitness function. Appl. Soft Comput. **84**, 105706 (2019). https://doi.org/10.1016/j.asoc.2019.105706
7. I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks. IEEE Commun. Mag. **40**, 102–114 (2002). https://doi.org/10.1109/MCOM.2002.1024422
8. K. Xu, H. Hassanein, G. Takahara, Q. Wang, Relay node deployment strategies in heterogeneous wireless sensor networks. IEEE Trans. Mob. Comput. **9**, 145–159 (2010). https://doi.org/10.1109/TMC.2009.105
9. N. Sharmin, A. Karmaker, W.L. Lambert, M.S. Alam, M.S.A. Shawkat, Minimizing the energy hole problem in wireless sensor networks: a wedge merging approach. Sensors (Basel, Switzerland) (2020). https://doi.org/10.3390/s20010277
10. E. Lee, S. Park, F. Yu, S.-H. Kim, Data gathering mechanism with local sink in geographic routing for wireless sensor networks. IEEE Trans. Consum. Electron. **56**, 1433–1441 (2010). https://doi.org/10.1109/TCE.2010.5606280
11. D. Jea, A. Somasundara, M. Srivastava, *Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks* (Springer, Berlin, 2005)
12. O.J. Pandey, R.M. Hegde, Low-latency and energy-balanced data transmission over cognitive small world WSN. IEEE Trans. Veh. Technol. **67**, 7719–7733 (2018). https://doi.org/10.1109/TVT.2018.2839562
13. I. Chatzigiannakis, A. Kinalis, S. Nikoletseas, Efficient data propagation strategies in wireless sensor networks using a single mobile sink. Comput. Commun. **31**, 896–914 (2008). https://doi.org/10.1016/j.comcom.2007.12.011
14. D. Niyato, P. Wang, Optimization of the mobile router and traffic sources in vehicular delay-tolerant network. IEEE Trans. Veh. Technol. **58**, 5095–5104 (2009). https://doi.org/10.1109/TVT.2009.2025379
15. R. Anwit, A. Tomar, P.K. Jana, Tour planning for multiple mobile sinks in wireless sensor networks: a shark smell optimization approach. Appl. Soft Comput. (2020). https://doi.org/10.1016/j.asoc.2020.106802
16. J.-S. Lee, T.-Y. Kao, An improved three-layer low-energy adaptive clustering hierarchy for wireless sensor networks. IEEE Internet Things J. **3**, 951–958 (2016). https://doi.org/10.1109/JIOT.2016.2530682
17. R. Gantassi, B. Bengouissem, J. Othmen, Routing protocol LEACH-K using K-means algorithm in wireless sensor network, in *Web, Artificial Intelligent and Network Applications* (2020), pp. 299–309. https://doi.org/10.1007/978-3-030-44038-1_27
18. A.W. Khan, A.H. Abdullah, M.A. Razzaque, J.I. Bangash, VGDRA: a virtual grid-based dynamic routes adjustment scheme for mobile sink-based wireless sensor networks. IEEE Sens. J. **15**, 526–534 (2015). https://doi.org/10.1109/JSEN.2014.2347137
19. R. Gantassi, Z. Masood, Y. Choi, Enhancing QoS and residual energy by using of grid-size clustering, K-means, and TSP algorithms with MDC in LEACH protocol. IEEE Access **10**, 58199–58211 (2022). https://doi.org/10.1109/ACCESS.2022.3178434
20. Y. Han, L. Zhang, H. Byun, B. Yang, Splitting-mergingbased automatic scheduling scheme for balancing energy consumption and task allocation in WSANs. Appl. Soft Comput. (2022). https://doi.org/10.1016/j.asoc.2022.108642
21. R. Zhang, J. Pan, D. Xie, F. Wang, NDCMC: a hybrid data collection approach for large-scale WSNs using mobile element and hierarchical clustering. IEEE Internet of Things J. **3**, 533–543 (2016). https://doi.org/10.1109/JIOT.2015.2490162
22. R. Gantassi, B.B. Gouissem, J.B. Othman, Combination of MDC and K-means to improve LEACH, in *IEEE International Conference on Communications* (2021), pp. 1–5. https://doi.org/10.1109/ICC42927.2021.9500909
23. A.S. Nandan, S. Singh, L.K. Awasthi, An efficient cluster head election based on optimized genetic algorithm for movable sinks in IoT enabled HWSNs. Appl. Soft Comput. (2021). https://doi.org/10.1016/j.asoc.2021.107318
24. H. Zhou, M. Song, W. Pedrycz, A comparative study of improved ga and pso in solving multiple traveling salesmen problem. Appl. Soft Comput. **64**, 564–580 (2018). https://doi.org/10.1016/j.asoc.2017.12.031
25. O.Y. Al-Jarrah, O. Alhussein, P.D. Yoo, S. Muhaidat, K. Taha, K. Kim, Data randomization and cluster-based partitioning for botnet intrusion detection. IEEE Trans. Cybern. **46**, 1796–1806 (2016). https://doi.org/10.1109/TCYB.2015.2490802
26. S. Yalçın, E. Erdem, Teo-mcrp: Thermal exchange optimizationbased clustering routing protocol with a mobile sink for wireless sensor networks. J. King Saud Univ. Comput. Inf. Sci. **34**, 5333–5348 (2022). https://doi.org/10.1016/j.jksuci.2022.01.007
27. D. Yook, T. Lee, Y. Cho, Fast sound source localization using two-level search space clustering. IEEE Trans. Cybern. **46**, 20–26 (2016). https://doi.org/10.1109/TCYB.2015.2391252
28. W. Heinzelman, A. Chandrakasan, H. Balakrishnan, An application specific protocol architecture for wireless microsensor networks. IEEE Trans. Wirel. Commun. **1**, 660–670 (2002). https://doi.org/10.1109/TWC.2002.804190
29. O. Younis, S. Fahmy, Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach. IEEE INFOCOM **1**, 640 (2004). https://doi.org/10.1109/INFCOM.2004.1354534
30. M. Biabani, N. Yazdani, H. Fotouhi, Ee-mswsn: energy-efficient mobile sink scheduling in wireless sensor networks. IEEE Internet Things J. **9**, 18360–18377 (2022). https://doi.org/10.1109/JIOT.2022.3160377

31. J. Amutha, S. Sharma, S.K. Sharma, An energy efficient cluster based hybrid optimization algorithm with static sink and mobile sink node for wireless sensor networks. Expert Syst. Appl. (2022). https://doi.org/10.1016/j.eswa.2022.117334
32. R. Shah, S. Roy, S. Jain, W. Brunette, Data MULEs: modeling a threetier architecture for sparse sensor networks, in *IEEE International Workshop on Sensor Network Protocols and Applications* (2003), pp. 30–41. https://doi.org/10.1109/SNPA.2003.1203354
33. C.-F. Cheng, C.-F. Yu, Data gathering in wireless sensor networks: a combine-TSP-reduce approach. IEEE Trans. Veh. Technol. **65**, 2309–2324 (2016). https://doi.org/10.1109/TVT.2015.2502625
34. J. Chen, F. Yan, S. Mao, F. Shen, W. Xia, Y. Wu, L. Shen, Efficient data collection in large-scale UAV-aided wireless sensor networks, in *International Conference on Wireless Communications and Signal Processing (WCSP)* (2019), pp. 1–5. https://doi.org/10.1109/WCSP.2019.8927929
35. B. Zhu, E. Bedeer, H.H. Nguyen, R. Barton, J. Henry, Uav trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning. IEEE Trans. Veh. Technol. **70**, 9540–9554 (2021). https://doi.org/10.1109/TVT.2021.3102161
36. Y. Wang, M. Chen, C. Pan, K. Wang, Y. Pan, Joint optimization of uav trajectory and sensor uploading powers for uav-assisted data collection in wireless sensor networks. IEEE Internet Things J. **9**, 11214–11226 (2022). https://doi.org/10.1109/JIOT.2021.3126329
37. O.O. Ogundile, A.S. Alfa, A survey on an energy-efficient and energy-balanced routing protocol for wireless sensor networks. Sensors (2017). https://doi.org/10.3390/s17051084
38. J. Mi, X. Wen, C. Sun, Z. Lu, W. Jing, Energy-efficient and low package loss clustering in UAV-assisted WSN using Kmeans++ and fuzzy logic, in *IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops)* (2019), pp. 210–215. https://doi.org/10.1109/ICCChinaW.2019.8849956
39. P.O. Olukanmi, B. Twala, K-means-sharp: modified centroid update for outlier-robust k-means clustering, in *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)* (2017), pp. 14–19
40. J. H. Holland, Genetic algorithms and classifier systems: foundations and future directions, in *International Conference on Genetic Algorithms* (1987)
41. A.E. Carter, C.T. Ragsdale, A new approach to solving the multiple traveling salesperson problem using genetic algorithms. Eur. J. Oper. Res. **175**, 246–257 (2006). https://doi.org/10.1016/j.ejor.2005.04.027
42. Z. Wang, X. Fang, H. Li, H. Jin, An improved partheno-genetic algorithm with reproduction mechanism for the multiple traveling salesperson problem. IEEE Access **8**, 102607–102615 (2020). https://doi.org/10.1109/ACCESS.2020.2998539
43. D.L. Davies, D.W. Bouldin, A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. **2**, 224–227 (1979). https://doi.org/10.1109/TPAMI.1979.4766909


## Publisher's Note