Energy-Efficient Scheduling in UAV-Assisted Hierarchical Wireless Sensor Networks

Rui Lai, Boyuan Zhang, Guanghong Gong, Haitao Yuan, *Senior Member, IEEE*, Jinhong Yang, Jia Zhang, *Senior Member, IEEE*, and MengChu Zhou, *Fellow, IEEE*

***Abstract*—In emerging applications of the Internet of Things, wireless sensor networks (WSNs) are often utilized to gather, track, and monitor data in remote areas with limited communication infrastructure. Since the majority of WSNs employ sensors powered by batteries, maintaining energy efficiency, and conservation is crucial for ensuring their sustained operations over time. This work designs an Energy-efficient unmanned aerial vehicle (UAV)-assisted hierarchical architecture of WSNs (EUW). EUW supports fast transmission of data collected from WSNs to a cloud server. Based on this architecture, this work first formulates a joint optimization problem for cluster head selection, time slot allocation, and UAV path planning to minimize the weighted sum of energy consumption of WSNs and that of a UAV. Then, a hybrid meta-heuristic algorithm named knowledge transfer-based particle swarm optimization (KTPSO) is designed, which utilizes previous optimization results to increase the convergence speed and find better results. Finally, numerical analysis and evaluation results are shown to demonstrate the efficiency of KTPSO and the proposed UAV-assisted architecture of hierarchical WSNs.**

***Index Terms*—Cluster head (CH) selection, hybrid meta-heuristic algorithms, time slot allocation (TSA), unmanned aerial vehicle (UAV) path planning, wireless sensor networks (WSNs).**

I. Introduction

IN RECENT years, wireless sensor networks (WSNs) have gained extensive utilization in monitoring environmental parameters, including temperature, humidity, air quality, and water quality in remote areas, forests, and other natural reserves [1], [2], [3], [4]. Due to the use of batteries for power supply in WSNs, using energy as efficiently as possible is crucial. As a result, sensor nodes (SNs) in WSNs are typically organized into distinct clusters, comprising cluster heads (CHs), and their corresponding cluster members (CMs). This clustering approach aims to facilitate energy-efficient data transmission and avoid energy holes in WSNs [5]. CHs gather and process data from CMs in the same cluster and send data to their nearest base station (BS). CMs in each cluster only need to transmit data to their closest CHs. By leveraging this clustering mechanism, the lifetime of WSNs can be greatly increased [6]. Many clustering protocols in WSNs have been proposed in recent years, such as LEACH [7], PEGASIS [8], and PSO-UFC [14], which are proven to improve the lifetime of WSNs greatly. When using cluster routing to transmit data, each CH needs to determine and maintain the scheduling mechanism of time division multiple access (TDMA). In most cases, the distribution of nodes is random, resulting in overlapping coverage among clusters. This leads to intercluster interference caused by uncoordinated TDMA scheduling among clusters, resulting in data collisions [15]. Data collision can make data received by destination SNs and data sent by source SNs invalid, and the retransmission of packets will consume more energy. Thus, how to reduce the intercluster interference has to be considered in the protocol design [16].

Most existing researches focus on the data transmission mechanism among SNs and their nearby BSs [17], [18], [19]. However, in many remote areas, ensuring sufficient coverage of BSs can be challenging. Recently, a novel network architecture known as space–air–ground integrated networks (SAGINs) has been proposed [20], [21], [22] to provide a feasible Internet access method for network users in remote areas [23]. Similar to SAGINs, low-Earth-orbit (LEO) satellites can be utilized to provide connections for WSNs to a cloud server [24]. Considering limited bandwidth in LEO networks and high power consumption for an SN to transmit data to the satellite, unmanned aerial vehicle (UAV) is deployed in the target area to serve as a mobile sink (MS) [25]. It can fly above WSNs, collect data from SNs, preprocess data, and transmit processed data to LEO satellites. As a UAV has high mobility, CHs can transmit data to UAV at a close distance with low energy consumption instead of consuming a large amount of energy to transmit data to LEO satellites directly [26], [27]. Thus, the UAV planning in data-gathering scenarios is of paramount importance [28]. Unlike ground BSs with convenient energy supplies, the UAV cannot receive an effective energy supply in the air. Besides, the position of the UAV affects SNs' energy consumption of data

Manuscript received 24 December 2023; revised 6 February 2024; accepted 20 February 2024. Date of publication 26 February 2024; date of current version 23 May 2024. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62173013 and Grant 62073005; in part by the Beijing Natural Science Foundation under Grant 4232049; and in part by the Fundamental Research Funds for the Central Universities under Grant YWF-23-03-QB-015. *(Corresponding author: Haitao Yuan.)*

Rui Lai, Boyuan Zhang, Guanghong Gong, and Haitao Yuan are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100000, China (e-mail: ggh@buaa.edu.cn; yuan@buaa.edu.cn).

Jinhong Yang is with CSSC Systems Engineering Research Institute, Beijing 100036, China (e-mail: yangjinhong.66@163.com).

Jia Zhang is with the Department of Computer Science, Southern Methodist University, Dallas, TX 75206 USA (e-mail: jiazhang@smu.edu).

MengChu Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/JIOT.2024.3369722, provided by the authors.

Digital Object Identifier 10.1109/JIOT.2024.3369722

transmission [29]. Therefore, further research is needed on UAV path planning in UAV-assisted data collection due to the above-mentioned issues.

Up to now, there have been limited studies focusing on joint optimization of UAV paths and clustering strategies in WSNs with an MS.

Besides, interference among sensors can greatly limit communication efficiency and big data transmission, which has also rarely been considered before. This work adopts the UAV and LEO satellites to assist WSNs in monitoring the environment in a remote area where there are few BSs. The main contributions of this work are given as follows.

1) This work proposes an energy-efficient UAV-assisted hierarchical architecture of WSNs (EUW) to monitor the environment in remote areas, which consists of terrestrial WSNs, a UAV, LEO satellites, and a cloud server. Experimental results show that EUW outperforms other peers in the network average energy consumption, standard deviation of network energy consumption, network lifetime (NL), and UAV's average energy consumption (UAEC).
2) This work mathematically models the communication speed and energy consumption under interference in UAV-assisted WSNs for the first time. Then, a joint optimization approach is proposed to improve energy efficiency and balance the network load. Specifically, this work jointly optimizes the SN clustering strategy, time slot allocation (TSA), and the flying path of the UAV with constraints of data transmission rate and UAV's motion. The problem is formulated as a mixed-integer nonlinear program known as an NP-hard problem.
3) To efficiently address the problem, a knowledge transfer-based particle swarm optimization (KTPSO) algorithm is proposed. It adopts knowledge transfer-based initialization to utilize previous high-quality solutions. Compared with other knowledge transfer methods such as deep learning-based knowledge transfer algorithms, KTPSO transfers knowledge with the minimal computational complexity. Besides, results are not sensitive to the amount of knowledge data and only one set of data is needed to guide it to converge through knowledge transfer quickly. In addition, the algorithm incorporates strategies in the genetic algorithm (GA) during particle exploration to avoid falling into local optima. Experimental results demonstrate that KTPSO outperforms other typical and widely adopted meta-heuristic algorithms.

The remainder of this work is organized as follows. Section II reviews the existing research. Section III describes the model of our proposed UAV-assisted hierarchical WSNs and introduces the problem formulation. Section IV introduces the implementation details of KTPSO. Section V evaluates the performance of our proposed method and its peers. Section VI concludes the work.

II. Related Works

Recently, WSNs have been well applied in different industrial areas. With the proposed clustering protocols, the performance of WSNs has been greatly improved. However, traditional WSNs with fixed sinks still have disadvantages, such as high energy consumption, limited lifetime, and significant data transmission delay. To solve these problems, researchers have turned their attention to WSNs with MSs and have proposed many protocols. These protocols can be roughly categorized based on whether the SNs are clustered or not.

A. MS-Based WSNs Without Clustering

The protocols without clustering have no clustering process and the MS follows a predefined trajectory, accessing specific points to collect data from SNs within the scope, i.e., rendezvous points (RPs). Taleb [30] used a single MS to collect information, and plan the path based on the minimum connected dominating set (MCDS). By constructing the MCDS, the number of intermediate nodes can be reduced, and the performance of WSNs can be improved. However, it is not suitable for large-scale WSNs because of its computational overhead. Donta et al. [31] build the path of MS based on the extended ant colony optimization (ACO) algorithm. This work proposes the RP reselection mechanism and virtual RPs to balance the energy and improve the NL. Wu et al. [32] optimized RPs and trajectory simultaneously in their studies. The algorithm they proposed constructs a data-forwarding tree, and optimizes collection points and an MS path based on a heuristic algorithm. Although the methods in [31] and [32] perform well in prolonging NL, their transmission latency is high as nodes need to communicate with each other through multihop. Khedr et al. [33] proposed a framework of WSNs with multiple MSs and a fuzzy C-means algorithm is adopted to select RPs. Using multiple MSs can improve the data collection speed and reduce the transmission energy consumption of nodes, but it can lead to an increase of overall energy consumption.

These protocols are usually relatively simple to implement yet have the drawbacks of high data transmission latency and high energy consumption. Most nodes transmit data to the MS through multihop, which limits the overall performance of the network. Our work adopts a clustering-based WSN framework, which has lower latency and energy consumption and is more suitable for large-scale WSNs.

B. MS-Based WSNs With Clustering

The protocols with clustering are different from the above schemes. In each cycle, data is gathered by CHs from all CMs and transmitted to MSs. Therefore, this type of protocol can help reduce the delay and balance the load among different nodes. Gao et al. [34] adopted virtual grids to partition the target area and select the SNs with the highest weight in each subarea as CHs. Then, they combine ACO and PSO to generate the path of MS. The complexity of clustering using virtual grids is relatively low, but it cannot adapt to the scenarios with complex node distribution. Roy et al. [35] established a model of WSNs with one MS and optimize sojourn points for the MS. Then, they use ACO to generate the path of the MS. It allows WSNs to balance between energy conservation and data transfer latency. However, this work does not take into account the influence of different clustering schemes on time delay.

Chowdary and Kuppili [36] proposed a new CH selection method and a cluster formation method. The path of the MS in this article is also optimized based on ACO. It can greatly reduce network energy consumption, but the transmission delay is not discussed in their work. Except for ACO, several studies also attempt to adopt other optimization algorithms to generate the MS path. Al-Kaseem et al. [37] proposed a protocol of WSNs with multiple MSs. They design a new clustering method and a location determination algorithm. A multiobjective evolutionary algorithm is adopted to optimize MS trajectories. Biabani et al. [38] divided nodes into different tree-like clusters based on the Voronoi diagram. Then, to achieve efficient path planning, nodes are prescreened based on buffer limitation, and a local search algorithm is adopted to generate the shortest tour path. Compared with ACO-based path planning algorithms, the above two algorithms have also been proven to be competitive. However, if optimization algorithms are only used to optimize the MS path, the performance improvement is relatively limited. Therefore, some studies also apply meta-heuristic optimization algorithms to generate clusters. For example, Amutha et al. [39] adopted the butterfly optimization algorithm to optimize the CH selection, and use ACO to perform the energy-saving routing. In their work, it is demonstrated that distinct algorithms have unique advantages, enabling enhanced optimization effects on different problems. Compared with this study, the node clusters and MS path are generated simultaneously through one optimization process in our work, which has higher solving efficiency.

In contrast to the aforementioned clustering-based protocols, this work mathematically models the communication speed and the energy consumption under interference. Besides, it allocates the communication time slots among SNs to minimize interference. Besides, the above protocols require the MS to ultimately send data to the nearby BS while our work considers the scenario in a remote area without BSs. In our work, MS only needs to upload data through LEO satellites.

C. UAV-Assisted Data Collection in WSNs

Most protocols proposed in the above studies use low-speed cars for data acquisition. However, using cars as MSs has some problems, such as unbalanced energy consumption, data delay, and limited coverage area. Recently, with the advancement of UAV technologies, UAV-assisted WSNs have begun to appear. UAV can serve as an MS, which is not limited by terrain and has good adaptability. In addition, the UAV can achieve a large range of communication coverage, and the energy consumption of WSNs is more balanced. the Chen et al. [40] adopted virtual grids to divide the target area and select SNs with the highest energy as CHs in each subarea. Additionally, they propose a direct future prediction model to generate the UAV trajectory. This method can maximize the total information value collected by UAV and reduce the charging frequency of UAV. Nevertheless, this work does not discuss how to extend the NL. Alfattani et al. [41] presented a multi-UAV data collection framework. They adopt the K-means clustering to generate clusters and GA to minimize the total flight time. This work takes two scenarios into account where the UAV hovers over the visited CHs and does within the range of CHs. It does not involve the energy consumption problem of WSNs either. In [42], a UAV-assisted WSNs framework with balanced energy consumption is proposed. A balanced cluster generation (BCG) algorithm is adopted and the trajectory of UAV is optimized by an improved multipopulation GA (MPGA). The goal of optimizing the trajectory is to reduce the length of the UAV flight path, while in our work, we directly optimize the UAV's energy consumption. Saxena et al. [43] adopted a random election (RE) method based on nodes' residual energy to select CHs and a minimum spanning tree (MST)-based path planning method to generate the UAV trajectory. The computational cost of this method is low but the energy consumption of each part is not discussed in this work. To minimize the overall energy consumption of UAV and WSNs, a reinforcement learning (RL)-based trajectory planning algorithm is proposed in [44]. A-star algorithm is adopted to select nodes as CHs. This method has advantages in reducing overall energy consumption, especially for large-scale WSNs. However, the balance of energy consumption is not considered in this work. Wang et al. [45] formulated the UAV energy consumption minimization problem subject to the energy limitation of SNs. The successive-convex-approximation method is adopted to solve this minimization problem. This method can give a scheduling scheme that is closer to the global optimum, yet it is not suitable for solving large-scale WSN problems.

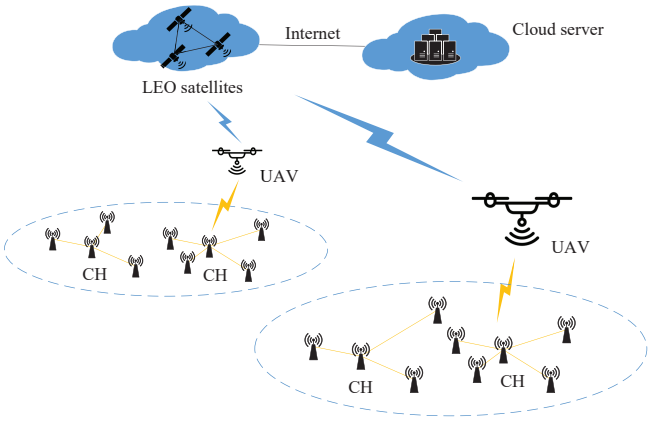

Fig. 1. Architecture of UAV-assisted hierarchical WSNs. A UAV is responsible for collecting sensor data in one region and data from different regions is aggregated to the cloud server *via* LEO satellites.

The above studies usually assume that the UAV flies at a constant speed and only receives data at the hovering points. This assumption ignores the acceleration and deceleration process of the UAV, and only receiving data at hovering points means inefficient time utilization. Therefore, we assume that the UAV does not need to hover and can receive data within a threshold range in this work, thereby further reducing the UAV energy consumption.

III. Problem Formulation

A. System Model

As shown in Fig. 1, an architecture of UAV-assisted hierarchical WSNs is considered. It is deployed in remote areas

TABLE I
MAIN NOTATIONS

Notation	Meaning
$\overrightarrow{\omega_i}$	Position of each node
S	Number of SNs
H	Number of clusters
x_n^i	Whether SN i is waked up at time slot n
$\Re$	Aggregation rate of data
$\mathcal{T}$	Data collection period
N	Number of time slots
h_k	Cluster head k
δ	Length of a time slot
B	Channel bandwidth
P_i	Transmission power of SN i
σ^2	Additive white Gaussian noise power
β_0	Channel power gain at one meter
υ	Value of path loss exponent
v	Speed of UAV
Ω	Size of data generated by one SN in time $\mathcal{T}$
a_{n}	Centripetal acceleration of UAV
$\mathcal{L}$	Length of a continuous UAV path
r	Turning radius in the Dubins path
$\Gamma_{h_k,U(n)}$	Channel power gain between h_k and UAV
E_{r}	Residual energy of SNs
E_i^{T}	Data transmission energy consumption of node i
$E_{h_k}^{\mathrm{R}}$	Data receiving energy consumption of CH h_k
s_k	Number of CMs in the cluster k
η	Number of UAV path points
E_{e}	Amount of energy dissipated in circuit of electronics
E_{U}	UAV's energy consumption
P_{p}	Propulsion power of UAV
E_{W}	Energy consumption of WSNs
F	Objective function
ρ	Compensation factor
k_{a}	Amplifier coefficient
$\overrightarrow{\phi_\theta}$	Position of path point θ of UAV
κ	Large positive constant
$\mho$	Penalty of all constraints
$\mathbb{H}$	Set of CHs
$\mathbb{M}_k$	Set of CMs in the kth cluster

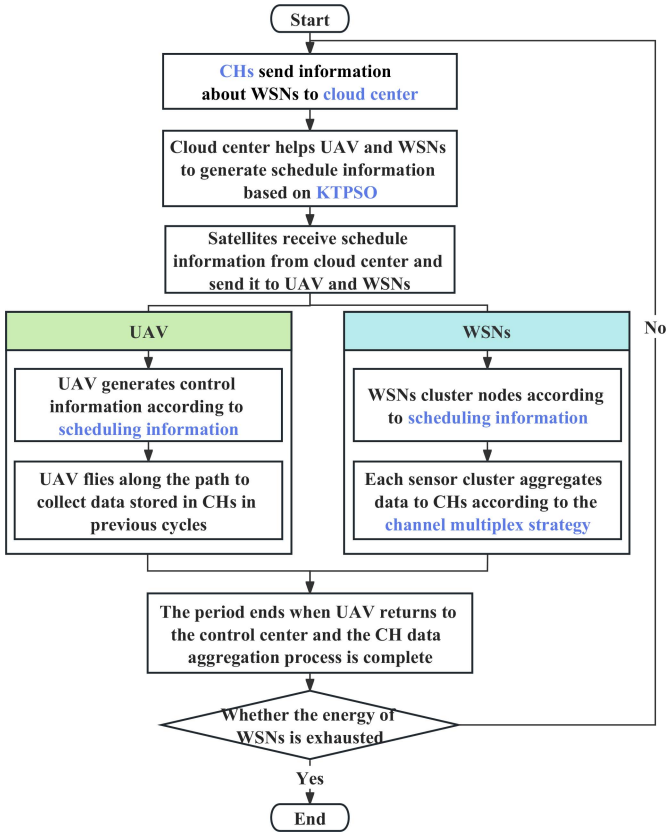

Fig. 2. Operation flowchart of UAV-assisted hierarchical WSNs. The entire process starts with sending status information to the cloud center from CHs. The cloud center calculates scheduling information and transmits it to the UAV and WSNs. The UAV sends scheduling information to its control module to start executing scheduled tasks, while WSNs cluster nodes based on scheduling information and generate time scheduling tables for each node. Data transmission of WSNs is based on TDMA (intracluster) and CDMA (intercluster). When the UAV returns to the control center, a new round of scheduling is initiated.

(such as large reservoirs and lakes) that lack the communication infrastructure. For one region, it includes three parts: 1) the ground part consisting of WSNs; 2) the aerial part consisting of a UAV; and 3) the space part consisting of LEO satellites. It is assumed that three different frequency bands are adopted in ground-to-ground communication, ground-to-air communication, and air-to-space communication, respectively, which do not affect each other.

The operation flowchart of UAV-assisted hierarchical WSNs. The process starts with sending status information to the cloud center from CHs. The cloud center calculates scheduling information and transmits it to the UAV and WSNs. The UAV sends scheduling information to its control module to start executing scheduled tasks, while WSNs cluster nodes based on scheduling information and generate time scheduling tables for each node. Data transmission of WSNs is based on TDMA (intracluster) and code-division-multiple-access (CDMA) (intercluster). Table I lists main notations throughout this article. When the UAV returns to the control center, a new round of scheduling is initiated. WSNs contain S SNs, which form a set of nodes, and the position of each node can be expressed as $\vec{w}_i \in \mathbb{R}^{3\times 1}$ $(1 \le i \le S)$. WSNs adopt a hierarchical routing structure commonly used currently, and they include CHs and CMs. The entire data collection process is shown in Fig. 2. Each member node monitors the environmental data and promptly transmits it to its associated CH. At the same time, CH aggregates data with an aggregation rate $\Re$, and sends the aggregated data to UAV. Additionally, CH transmits the cluster status to UAV at a data collection period $\mathcal{T}$. The UAV sends data to the cloud center *via* LEO satellites. After the end of each data collection period, CH election, TSA, and UAV path planning are completed by KTPSO in the cloud center. Then, the message is returned to the UAV and each SN. After determining CHs, other nodes sort CHs by distance. They join the cluster with the closest CH. For computation convenience, we divide the data collection period length $\mathcal{T}$ into N time slots, whose length is denoted as δ.

We denote a set of CHs as $\mathbb{H} = \{h_1, h_2, \ldots, h_H\}$ where H denotes the number of clusters. Cluster k contains s_k members, which belong to a set $\mathbb{M}_k = \{m_1^k, m_2^k, \ldots, m_{s_k}^k\}$ and they communicate with the CH with the TDMA protocol. An indicator variable is denoted as $x_n^i \in \{0, 1\}$. If SN i is waked up for data transmission at time slot n, $x_n^i = 1$; otherwise, $x_n^i = 0$. Each cluster adopts CDMA to reduce interference of nodes. Following [29] and [46], channel capacity between CM

$i, i \in \mathbb{M}_k$ and CH h_k is given as:

$$R_{i,h_k}(n) = Bx_n^i \log_2\left(1 + \frac{P_i \Gamma_{i,h_k}}{\sigma^2 + \sum_j x_n^j P_j \Gamma_{j,h_k}}\right) \quad (1)$$

where B denotes the channel bandwidth, P_i denotes the transmit power of SN i, σ^2 denotes the additive white Gaussian noise power, and j ($j \notin \mathbb{M}_k$) denotes the index of a CM that belongs to other clusters. The channel power gain between CM i and its CH is given as follows:

$$\Gamma_{i,h_k} = \frac{\beta_0}{\|\vec{w}_i - \vec{w}_{h_k}\|^v} \quad (2)$$

where β_0 represents the channel power gain at one meter and v denotes the value of the path loss exponent.

The size of data generated by one SN in time $\mathcal{T}$ is denoted by Ω. To ensure the data is uploaded in time, we have

$$\sum_{n=1}^{N} R_{i,h_k}(n)\delta \geq \Omega \quad \forall i \in \mathbb{M}_k \quad (3)$$

where $\delta = (\mathcal{T}/N)$.

It is assumed that the speed and height of the UAV are constant and UAV's initial position is the same as its final position. Given a sufficiently large N, the distance between an SN and UAV remains approximately constant for a time slot δ. Thus, the UAV path is discretized as $\{\vec{q}_n, 0 \leq n \leq N\}$. This work optimizes η path points of UAV. We connect those path points and generate a continuous UAV path with the Dubins generation of a UAV path [47]. Then, the speed of UAV is

$$v = \frac{\mathcal{L}}{\mathcal{T}} \quad (4)$$

where $\mathcal{L}$ denotes the length of the continuous UAV path.

Thus, when the UAV flies in the straight segment, $a_{n=}0$; otherwise, $a_{n=}(v^2/r)$, and r denotes the turning radius in the Dubins path. Mobility constraints of the UAV are given as follows:

$$\check{v} \leq v \leq \hat{v} \quad (5)$$

$$0 \leq a \leq \hat{a}. \quad (6)$$

Besides, the CH h_k communicates with UAV by using an access scheme of nonorthogonal multiple access. Then, similar with [46], the channel capacity between the CH h_k and UAV is given as follows:

$$R_{h_k,U}(n) = Bx_n^{h_k} \log_2\left(1 + \frac{P_{h_k}\Gamma_{h_k,U}(n)}{\sigma^2 + \sum_{\tilde{k}\neq k} x_n^{h_{\tilde{k}}} P_{h_{\tilde{k}}} \Gamma_{h_{\tilde{k}},U}(n)}\right) \quad (7)$$

where $\Gamma_{h_k,U}(n)$ is the channel power gain between CH h_k and UAV, and its value is given as follows:

$$\Gamma_{h_k,U}(n) = \frac{\beta_0}{\|\vec{w}_{h_k} - \vec{w}_U^n\|^v} \quad (8)$$

where $\vec{w}_U^n$ denotes the UAV's position in the nth time slot. To ensure the timely uploading of data, for the kth cluster with s_k CMs, we have

$$\sum_{n=1}^{N} R_{h_k,U}(n)\delta \geq (s_k + 1)\Re\Omega \quad (9)$$

where $\Re$ is the rate of data aggression.

Besides, for prolonging the lifetime of WSNs, it is assumed CHs' residual energy must be larger than the median of all SNs' residual energy, which can be expressed as follows:

$$\text{median}(E_r) \leq E_r(i) \quad \forall i \in \mathbb{H}. \quad (10)$$

B. Energy Consumption Model

1) Energy Consumption of SNs: According to [5], for SN i ($1 \leq i \leq S$) in the kth cluster, the total energy consumed when transmitting data over time $\mathcal{T}$ is

$$E_i^T = \sum_{n=1}^{N} x_n^i \delta k_a P_i + y_i(s_k + 1)\Omega\Re E_e + (1 - y_i)\Omega E_e \quad (11)$$

where y_i is an indicator variable, and when SN i is CH, $y_i = 1$; otherwise, $y_i = 0$. k_a is an amplifier coefficient. E_e denotes the energy consumed for activating the circuits of the transceiver.

According to [5], for any CH h_k, the total energy consumed when receiving data over time $\mathcal{T}$ is

$$E_{h_k}^R = s_k \Omega E_e. \quad (12)$$

2) Energy Consumption of UAV: The UAV energy E_U can be divided into the propulsion energy and the communication energy. As the propulsion energy consumption is much larger than the communication energy consumption, the total energy consumption can be simply expressed as follows:

$$E_U = \sum_{n=1}^{N} \delta P_p \quad (13)$$

where the propulsion power P_p is calculated based on [48].

C. Problem Formulation

For prolonging the lifetime of WSNs and saving UAV's energy, we define the objective function of the optimization problem as a weighted sum of energy consumption of WSNs and the UAV over a data collection period. F denotes the objective function, which is calculated as follows:

$$F = \rho E_W + E_U \quad (14)$$

where ρ denotes a compensation factor. As E^R of SNs is not affected by the optimization variables, we set E_W to be the same as the sum of E^T of each node. Thus, the final optimization problem is formulated as follows:

$$\textbf{(P1)}: \min_{P,X,Y,\Phi} F \quad (15)$$

s.t.

$$\sum_{n=1}^{N} R_{i,h_k}(n)\delta \geq \Omega \quad \forall k \in \{z \in \mathbb{Z} | 1 \leq z \leq H\} \quad \forall i \in \mathbb{M}_k \quad (16)$$

$$\sum_{n=1}^{N} R_{h_k,U}(n)\delta \geq (s_k + 1)\Re\Omega \quad \forall k \in \{z \in \mathbb{Z} | 1 \leq z \leq H\} \quad (17)$$

$$\check{v} \leq v \leq \hat{v} \quad (18)$$

$$0 \leq a \leq \hat{a} \quad (19)$$

$$\text{median}(E_r) \leq E_r(i) \quad \forall i \in \mathbb{H} \quad (20)$$

where $P \triangleq \{P_i\}$, $X \triangleq \{x_n^i\}$, and $Y \triangleq \{y_i\}$, $1 \le n \le N$, $1 \le i \le S$. Besides, $\Phi \triangleq \{\vec{\phi}_\theta\}$, and $\vec{\phi}_{\theta \in} \mathbb{R}^{3\times 1}$, $1 \le \theta \le \eta$. $\vec{\phi}_\theta$ denotes the position of a UAV path point θ. Then, we transform **P1** into an unconstrained optimization problem *P2* as follows:

$$\textbf{(P2)}: \min_{P,X,Y,\Phi} \{\widetilde{F} = F + \kappa \mho\} \tag{21}$$

where κ denotes a large positive constant, $\mho$ denotes the penalty of all constraints, and $\kappa\mho$ denotes the total penalty. $\mho$ can be calculated as follows:

$$\mho = \sum_{\tilde{n}=1}^{\mathcal{N}} \max\{0, \Delta_{\tilde{n}}\} \tag{22}$$

where $\mathcal{N}$ denotes the total number of constraints. Each constraint in (16)–(20) needs to be transformed into an inequality one $\Delta_{\tilde{n}} \le 0$, $1 \le \tilde{n} \le \mathcal{N}$. For example, (18) can be transformed into two inequalities, i.e., $\check{v} - v \le 0$ and $v - \hat{v} \le 0$.

IV. Knowledge Transfer-Based Particle Swarm Optimization

For the optimization problem of UAV-assisted WSNs, its state (mainly the residual energy of SNs) is constantly changing. The entire framework runs in each round, and it needs to run the optimization algorithm once in a round to provide new results. Meanwhile, there is a strong similarity in the state between the current round and the previous one. As described in Section III, we assume the UAV always initiates a new round of flight from a fixed position and only SNs' residual energy changes in different rounds. Similar to [9] and [10], our proposed algorithm can effectively utilize this similarity and adopt the optimization results from the previous round to guide the optimization of the current round. KTPSO consists of three main steps: 1) knowledge transfer-based initialization; 2) exemplar update; and 3) particle update.

A. Knowledge Transfer-Based Initialization

For each particle $\tilde{i}$ in set $\mathbb{P}$, the dimension is denoted as D. We randomly initialize its velocity $V_{\tilde{i}}$ ($V_{\tilde{i}} = [\tilde{v}_{\tilde{i},1}, \tilde{v}_{\tilde{i},2}, \ldots, \tilde{v}_{\tilde{i},D}]$), its position $\Psi_{\tilde{i}}$($\Psi_{\tilde{i}} = [\psi_{\tilde{i},1}, \psi_{\tilde{i},2}, \ldots, \psi_{\tilde{i},D}]$), and its locally best position $L_{\tilde{i}}$ ($L_{\tilde{i}} = [l_{\tilde{i},1}, l_{\tilde{i},2}, .., l_{\tilde{i},D}] = \Psi_{\tilde{i}}$, $1 \le \tilde{i} \le |\mathbb{P}|$). The representation of a randomly generated particle position $\Psi_{\tilde{i}}$ is shown in Fig. 3.

The average energy consumed by one node in WSNs after the previous round of data transmission is $\overline{\Delta E}$ and the node's average residual energy is $\overline{E_r}$. Let $(\overline{E_r}/\overline{\Delta E})$ denote the temperature τ. According to simulated annealing (SA)'s Metropolis acceptance rule, it means that the larger the gap between the two rounds of data collection, the harder it is for the solutions of the previous round to be accepted in the new round. Fig. 4 shows the structure of each particle in KTPSO. In KTPSO, each particle has an exemplar. For the exemplar of particle $\tilde{i}$, generate an alternate random value $\chi_{\tilde{i},d}$, and $\chi_{\tilde{i},d} = \text{rand}(\check{b}_d, \hat{b}_d)$ for each dimension. $\check{b}_d$ and $\hat{b}_d$ represent lower and upper bounds, respectively. Here, $\Upsilon_{\tilde{i}} = [\chi_{\tilde{i},1}, \chi_{\tilde{i},2}, \ldots, \chi_{\tilde{i},D}]$. Then, the value of the exemplar's dimension d can be initialized based on SA's Metropolis acceptance rule, i.e.,

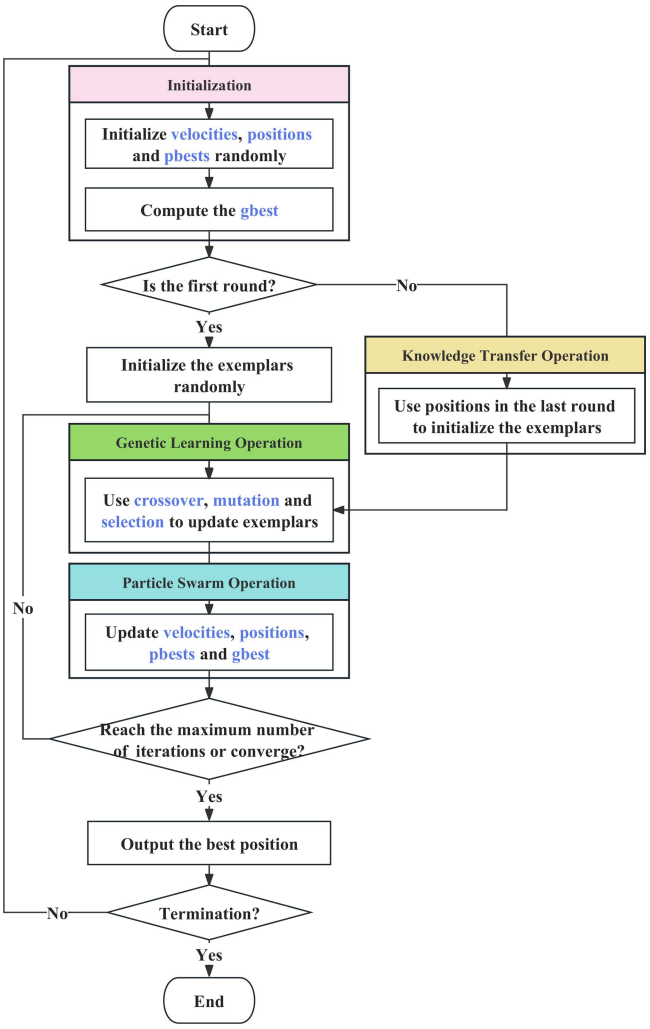

Fig. 3. Flowchart for optimizing periodic repetitive optimization problems with KTPSO.

$$e_{\tilde{i},d}^{\epsilon} = \begin{cases} \psi_{\tilde{i},d}^{\epsilon-1}, & \widetilde{F}\left(\Psi_{\tilde{i}}^{\epsilon-1}\right) < \widetilde{F}\left(\Upsilon_{\tilde{i}}\right) \\ \psi_{\tilde{i},d}^{\epsilon-1}, & \widetilde{F}\left(\Psi_{\tilde{i}}^{\epsilon-1}\right) \ge \widetilde{F}\left(\Upsilon_{\tilde{i}}\right) \text{ and } e^{\frac{-\left(\widetilde{F}(\Psi_{\tilde{i}}^{\epsilon-1}) - \widetilde{F}(\Upsilon_{\tilde{i}})\right)}{\lambda\tau}} > \xi \\ \chi_{\tilde{i},d}, & \widetilde{F}\left(\Psi_{\tilde{i}}^{\epsilon-1}\right) \ge \widetilde{F}\left(\Upsilon_{\tilde{i}}\right) \text{ and } e^{\frac{-\left(\widetilde{F}(\Psi_{\tilde{i}}^{\epsilon-1}) - \widetilde{F}(\Upsilon_{\tilde{i}})\right)}{\lambda\tau}} \le \xi \end{cases} \tag{23}$$

where ϵ is the index of current round, and $\Psi_{\tilde{i}}^{\epsilon-1}$($\Psi_{\tilde{i}}^{\epsilon-1} = [\psi_{\tilde{i},1}^{\epsilon-1}, \psi_{\tilde{i},2}^{\epsilon-1}, \ldots, \psi_{\tilde{i},D}^{\epsilon-1}]$) represents the final position of particle $\tilde{i}$ in the previous round. ξ is a randomly generated value, and $\xi \in (0, 1)$. Finally, the globally best position G ($G = [g_1, g_2, \ldots, g_D]$) of the swarm is updated, which can be calculated as follows:

$$G = \arg\min_{\Psi} \widetilde{F}. \tag{24}$$

B. Exemplar Update

Similar to [11], [12], and [13], to prevent particles from falling into local optima, operations in GA are used to update the positions of exemplars. This step includes three components. The first one is crossover. For each particle $\tilde{i}$, its offspring $O_{\tilde{i}}$ ($O_{\tilde{i}} = (o_{\tilde{i},1}, o_{\tilde{i},2}, \ldots, o_{\tilde{i},D})$) is generated as follows:

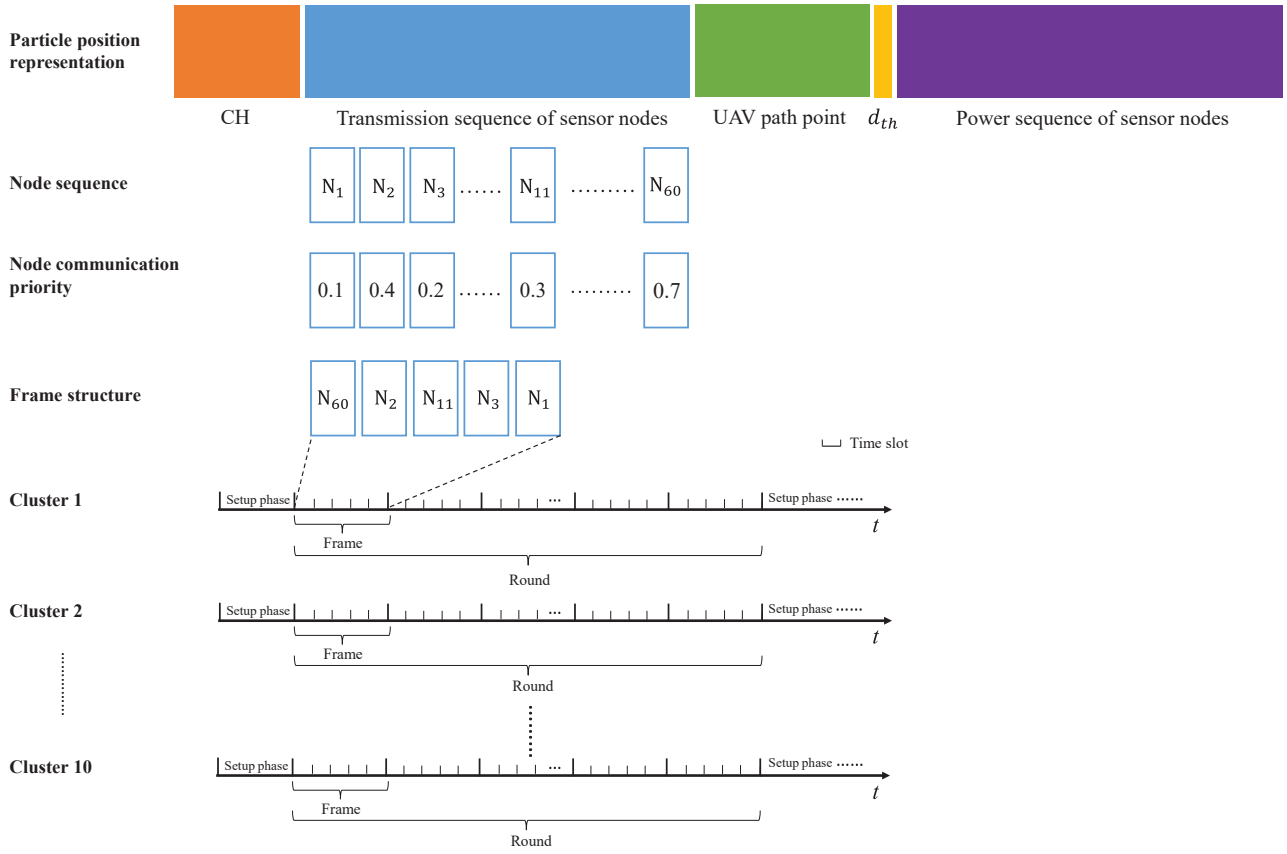

Fig. 4. Structure of particles in KTPSO. A scenario with 60 SNs and 10 clusters is considered. The first part includes CHs' indexes. The second part represents the node communication priority. In Fig. 4, N_1, N_2, N_3, N_{11}, and N_{60} belong to the same cluster, and their order in one frame is determined by the node communication priority. The third part represents the UAV path points. Each path point includes x value, y value, and UAV deflection angle. The fourth part represents the communication range between the UAV and CHs. The final part represents the transmission power of each node.

$$o_{\tilde{i},d} = \begin{cases} \tilde{r}_{\tilde{i},d} \cdot l_{\tilde{i},d} + \left(1 - \tilde{r}_{\tilde{i},d}\right) \cdot g_d \ \widetilde{F}\left(\Psi_{\tilde{i}}\right) < \widetilde{F}\left(\Psi_{\tilde{j}}\right) \\ \psi_{\tilde{j},d}, \qquad \text{otherwise} \end{cases} \tag{25}$$

where $\tilde{r}_{\tilde{i},d}$ ($\tilde{r}_{\tilde{i},d} \in [0, 1]$) is a random number and the particle $\tilde{j}$ is also randomly chosen from the particle set.

The second one is the mutation, which is the same as the one in the canonical GA. It is expressed as follows:

$$o_{\tilde{i},d} = \text{rand}\left(\check{b}_d, \hat{b}_d\right), \ \text{if } \tilde{r}_{\tilde{i},d} < \zeta \tag{26}$$

where ζ is the mutation probability.

The last one is selection. The fitness function value of the offspring $O_{\tilde{i}}$ generated above is compared with that of the exemplar $E_{\tilde{i}}$ in the last generation. If the former is less than the latter, the offspring is selected as the new exemplar.

C. Particle Update

$\hat{g}$ denotes the maximum number of generations, and $\omega_{\tilde{g}}$ denotes the value of inertia weight in generation $\tilde{g}$ ($1 \le \tilde{g} \le \hat{g}$). $\omega_{\tilde{g}}$ decreases linearly from the maximum value $\hat{\omega}$ to the minimum value $\check{\omega}$, i.e.,

$$\omega_{\tilde{g}} = \hat{\omega} - \frac{\tilde{g}\left(\hat{\omega} - \check{\omega}\right)}{\hat{g}}. \tag{27}$$

Then, the velocity and position of particle i are updated as follows:

$$\tilde{v}_{\tilde{i},d}^{\tilde{g}+1} = \omega_{\tilde{g}} \cdot \tilde{v}_{\tilde{i},d}^{\tilde{g}} + c \cdot \tilde{r}_{\tilde{i},d}\left(e_{\tilde{i},d}^{\tilde{g}} - \psi_{\tilde{i},d}^{\tilde{g}}\right) \tag{28}$$

$$\psi_{\tilde{i},d}^{\tilde{g}+1} = \psi_{\tilde{i},d}^{\tilde{g}} + \tilde{v}_{\tilde{i},d}^{\tilde{g}+1} \tag{29}$$

where c is an accelerate coefficient, which reflects the impact of the difference between $e_{\tilde{i},d}^{\tilde{g}}$ and $\psi_{\tilde{i},d}^{\tilde{g}}$.

In this way, each particle can get close to its exemplar and may change the exemplar when it finds a better position. KTPSO utilizes the previous knowledge to accelerate its optimization process and uses operations in GA to avoid falling into local optima.

The novelty of KTPSO first lies in that it adopts knowledge transfer-based initialization to utilize previous high-quality solutions. For the optimization problem proposed, the state variable, i.e., sensor's residual energy, is slowly changing, and the closer two states are in the time dimension, the higher their similarity. Therefore, extracting knowledge from the optimization results of the previous round can effectively avoid the occurrence of negative knowledge transfer. In addition, the algorithm incorporates GA's strategies during particle exploration. The pbests and the gbest are replaced with exemplars generated by GA operations in the particle update stage. Through this approach, the local exploration ability of each particle can be improved. Meanwhile, it also ensures good distribution of particles in the initial stage of

Algorithm 1 Knowledge Transfer-Based PSO

1: **for** $\tilde{i}$=1 to $|\mathbb{P}|$ **do**
2: Initialize $V_{\tilde{i}}$, $\Psi_{\tilde{i}}$, $L_{\tilde{i}}$;
3: Apply (23) to generate $E_{\tilde{i}}$;
4: **end for**
5: Find the best position G;
6: **while** $\tilde{g} \leq \hat{g}$ **do**
7: **for** $\tilde{i}$=1 to $|\mathbb{P}|$ **do**
8: Apply (25) and (26) to generate $O_{\tilde{i}}$;
9: **if** $\widetilde{F}(O_{\tilde{i}}) < \widetilde{F}(E_{\tilde{i}})$ **then**
10: $E_{\tilde{i}} = O_{\tilde{i}}$;
11: **end if**
12: Update $V_{\tilde{i}}$ with (28);
13: Update $\Psi_{\tilde{i}}$ with (29);
14: Update $L_{\tilde{i}}$;
15: **end for**
16: Update G;
17: $\tilde{g} = \tilde{g} + 1$;
18: **end while**
19: **return** G

optimization, which is crucial to adopt knowledge transfer for jumping out of local optima.

Algorithm 1 shows pseudo codes of KTPSO. Lines 1–4 initialize velocities V, positions Ψ, and locally best positions L, and generate exemplars E. Line 5 finds the best position and set it to G. Line 6 shows that the *while* loop stops when the number of generations ($\tilde{g}$) is larger than $\hat{g}$. Line 8 applies (25) and (26) to generate an offspring of particle $\tilde{i}$. Lines 9–11 update the exemplar. Lines 12–14 update velocities, positions, and the locally best positions, respectively. Line 19 returns the G, which is the optimization result.

The complexity analysis of KTPSO is given as follows. We assume the dimension of search space is D, and the maximum number of fitness evaluations allowed in KTPSO is $\hat{F}$. The time complexity is associated with that of initialization (T_I) PSO operations (T_P) and GA operations (T_G). Here, T_P is composed of the evaluation (T_e) and the updated operation (T_U), which includes the updated velocity (T_v) and the updated position (T_p). Besides, T_G involves mutation (T_m) and crossover (T_c). Moreover, the number of the evaluation process and the update process for both a particle and an exemplar is ($\hat{F}/2$). The complexity of KTPSO can be calculated as follows:

$$\begin{aligned} T(D) &= T_I + \frac{\hat{F}(T_P + T_G)}{2} \\ &= T_I + \frac{\hat{F}\left[\left(T_e + T_v + T_p\right) + \left(T_e + T_m + T_c\right)\right]}{2} \\ &= D + 3D\hat{F}. \end{aligned} \tag{30}$$

According to the above-mentioned analysis, the time complexity of KTPSO is $O[D\hat{F}]$.

V. Numerical Experiments and Analysis

This section first determines the parameters used in the experiments. Then, we evaluate the performance of our proposed framework and its other peers. Finally, we analyze the effectiveness of knowledge transfer-based initialization and TSA, respectively.

TABLE II
Setting of Simulation Parameters

Parameter	Value	Parameter	Value
N	500	δ	0.2 s
k_a	18	D	50 KBytes
B	1 MHZ	β_0	10^6
σ^2	-110 dbm	$\Re$	50%
$\check{v}$	0 m/s	$\hat{v}$	30 m/s
ρ	100	$\mathcal{N}$	7

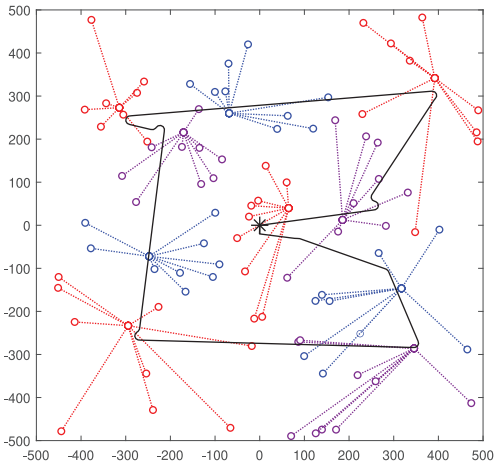

Fig. 5. Final result of clustering and UAV's path of 100 SNs. Circles represent SNs, the star represents the initial position of the UAV, the dashed lines represent connections between CHs and their CMs, and the solid line represents the UAV's trajectory.

A. Parameter Setting

We consider six scenarios of UAV-assisted hierarchical WSNs with 50 to 100 SNs, and randomly generate their coordinates within a square area. To ensure consistent node density in different scenarios, the length of the square area is set as $\sqrt{(S/100)}$ km. Here, $\eta = H = 0.1$ s. The UAV collects data from CHs and serves as a communication relay. The value of path loss exponent ν is determined based on [14]. The total energy of an SN is 1500 J. Other simulation parameters are given in Table II.

Besides, the parameters of KTPSO are as follows. $\zeta = 0.05$, $\hat{\omega} = 0.7$, $\check{\omega} = 0.35$, $c = 1.49618$, $\lambda = 10^4$, and $|\mathbb{P}| = 100$. Fig. 5 shows the final result of clustering and UAV's path of 100 SNs.

B. Framework Performance Evaluation

In this section, EUW is compared with its three peers, which are shown in Table III. For BCG-MPGA, there are five populations in the MPGA stage and each population has 100 individuals. The crossover probability is set to a random number in [0.7, 0.9] and the mutation probability is also a random number in [0.001, 0.05]. For Ptr-A*, the training batch size is set to 512. The learning rate is initialized to 0.0001 and it decreases by 4% every 5000 steps. The specific details of the three algorithms can be referred to the original text in Table III. For UAV-assisted WSNs, the data collection

TABLE III
ALGORITHMS COMPARED IN THE EXPERIMENT

Method	Type	UAV's data collection mode
BCG-MPGA [42]	Heuristic-based	Flying
RE-MST [43]	Graph-based	Hovering
Ptr-A* [44]	RL-based	Hovering

TABLE IV
FEATURES OF DATA COLLECTION MODES

Data collection mode	Hovering at data collection point (CP)	Continuing data collection after passing through CP
Hovering	✓	✗
Flying	✗	✓

modes of UAV can be mainly divided into hovering mode and flying mode [49]. The features of the two modes are shown in Table IV. Most researches focus on planning methods of hovering mode, which adopts a simple UAV kinematic model, i.e., the UAV flies at a constant speed throughout the entire journey without considering the acceleration and deceleration processes between hovering and moving. This assumption can cause significant errors in the energy consumption calculation of the UAV in the experimental scenario. To facilitate the comparison, we adopt the tauj algorithm [50] to generate a 4-D trajectory for the UAV on the basis of their hovering mode algorithms.

This section chooses the network's average energy consumption (NAEC), standard deviation of the network's energy consumption (SDNEC), NL, and UAEC as our performance metrics. We yield each metric by running 10 individual experiments. The details are as given follows.

1) Network's Average Energy Consumption: NAEC is defined as the average energy consumption of all nodes, which can be calculated as follows:

$$\text{NAEC} = \sum_{\tilde{k}=1}^{K} \frac{\sum_{i=1}^{S}\left(E_{i,\tilde{k}}^{T} + E_{i,\tilde{k}}^{R}\right)}{K} \tag{31}$$

where K denotes the NL, which is defined as the number of rounds until half of the energy of the nodes is exhausted.

The results of NAEC are shown in Fig. 6. It is shown that EUW can improve the performance of NAEC. Compared with BCG-MPGA, RE-MST, and Ptr-A*, the improvement percentages of EUW's NAEC are 21%–27%, 11%–17%, and 10%–15%, respectively. Our proposed framework both directly models and optimizes the network energy consumption, thus consuming less network energy compared with BCG-MPGA and RE-MST. Compared with Ptr-A*, our EUW is more refined, reflecting the impact of internode interference on energy consumption, and achieving the lowest NAEC.

2) Standard Deviation of Network's Energy Consumption: SDNEC can be computed as follows:

$$\text{SDNEC} = \sqrt{\frac{\sum_{i=1}^{S}\left(\sum_{\tilde{k}=1}^{K}(E_{i,\tilde{k}}^{T} + E_{i,\tilde{k}}^{R}) - \mu_e\right)^2}{S}} \tag{32}$$

where μ_e denotes the average energy consumption of nodes, which can be calculated as follows:

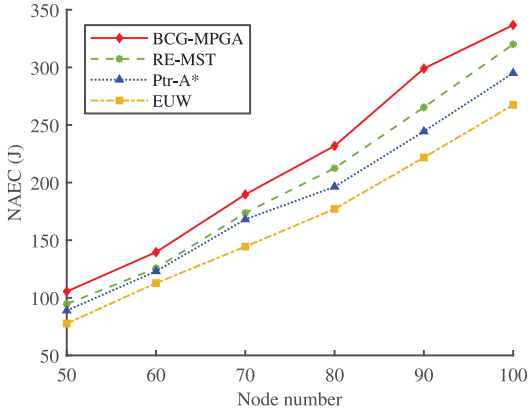

Fig. 6. NAEC under scenarios with different numbers of nodes.

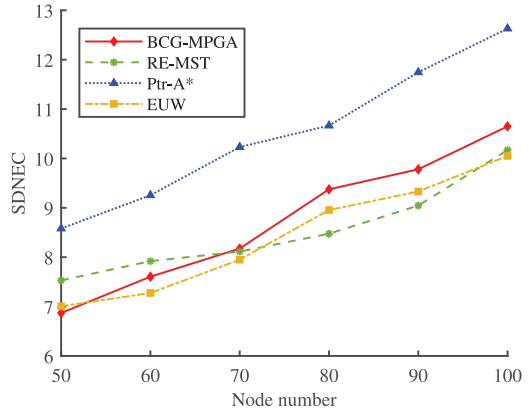

Fig. 7. SDNEC under the scenarios with different numbers of nodes.

$$\mu_e = \sum_{\tilde{k}=1}^{K} \frac{\sum_{i=1}^{S}\left(E_{i,\tilde{k}}^{T} + E_{i,\tilde{k}}^{R}\right)}{S}. \tag{33}$$

In Fig. 7, SDNEC of EUW, BCG-MPGA, and RE-MST is almost equal, and these three protocols have better load balance in almost all scenarios. Compared with Ptr-A*, the decrement of SDNEC is 16%–22%. All three methods except Ptr-A* consider energy balance in the network, and therefore, their SDNEC is significantly smaller than that of Ptr-A*.

3) Network Lifetime: As shown in Fig. 8, EUW has an advantage in varying NL of WSNs. It prolongs NL by 23%–34% compared with BCG-MPGA, 12%–19% compared to RE-MST, and 11%–16% compared to Ptr-A*. For WSNs, lower network energy consumption and a more balanced distribution of energy among nodes contribute to the extended NL. Because our proposed framework performs well in the first two metrics, it can greatly prolong the NL, which is consistent with the results shown in Fig. 8.

4) UAV's Average Energy Consumption: UAEC can be computed as follows:

$$\text{UAEC} = \sum_{\tilde{k}=1}^{K} \frac{E_U^{\tilde{k}}}{K} \tag{34}$$

where $E_U^{\tilde{k}}$ denotes the energy consumption of UAV in the $\tilde{k}$th round. In Fig. 9, the UAEC of the proposed

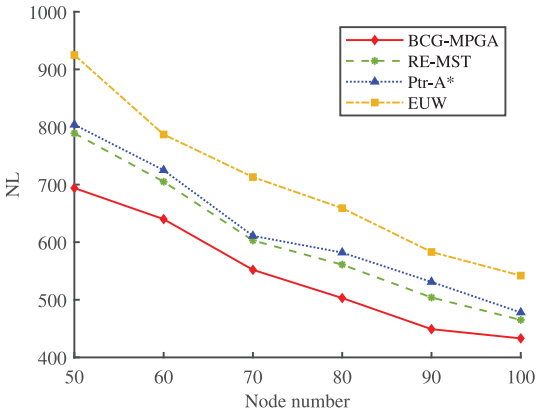

Fig. 8. NL under the scenarios with different numbers of nodes.

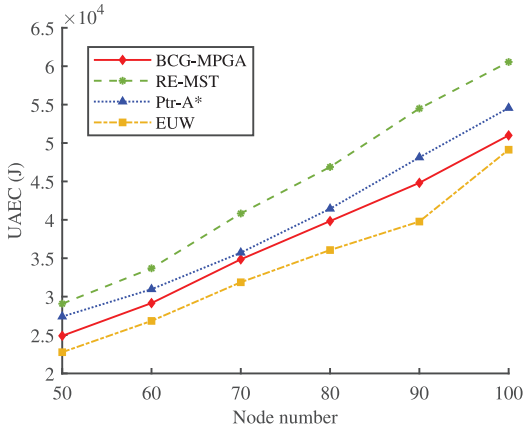

Fig. 9. UAEC under the scenarios with different numbers of nodes.

EUW is optimized by around 3%–11% compared with the BCG-MPGA. Similarly, EUW's improvements over RE-MST and Ptr-A* are 18%–28% and 10%–18%, respectively. Here, the UAV needs to collect data from all CHs in one period and return to the starting position. When the UAV flies along the same path, the maximum flight speed required in the hovering mode is faster than in the flying mode. Meanwhile, the UAV also consumes extra energy during hovering. Due to the above two reasons, the UAV has lower energy consumption when adopting the flying mode, which can be observed from the results in Fig. 9. Besides, it can also be observed from the results that optimizing the UAV's energy consumption directly performs better than optimizing the UAV's path length in UAEC.

C. Analysis of the Effectiveness of Knowledge Transfer

To further analyze the effectiveness of knowledge transfer, a random initialization is adopted to replace knowledge transfer-based initialization, and this benchmark algorithm is named as KTPSO-RI. In addition, we also choose PSO and another effective knowledge transfer-based method proposed in [51] named CPSO-NNIT as benchmark algorithms. All the benchmark algorithms adopt the same particle position representation. The population sizes of PSO, KTPSO-RI, and CPSO-NNIT are all 100. The inertia weight of PSO is set to 0.7 [52]. For KTPSO-RI, the lower and upper bounds of inertia weight are set to 0.35 and 0.7, and for CPSO-NNIT, they are set to 0.3 and 0.6, respectively [51]. The acceleration coefficient is set to 2.0, 1.49618, and 1.7 in PSO, KTPSO-RI, and CPSO-NNIT, respectively. To visually demonstrate the performance differences of these optimization algorithms, we keep the $\widetilde{F}$ values of 20 sets of experiments (each algorithm runs for ten rounds), and the results are given in Table V.

TABLE V
OPTIMIZATION RESULTS OF DIFFERENT ALGORITHMS

Node number	Algorithm	$\widetilde{F}$	
		Mean	Std
50	PSO	4.36E+04	1.78E+04
	KTPSO-RI	4.08E+04	1.50E+04
	CPSO-NNIT	3.92E+04	6.17E+03
	KTPSO	**3.34E+04**	**2.41E+03**
60	PSO	6.18E+04	1.86E+04
	KTPSO-RI	5.21E+04	1.93E+04
	CPSO-NNIT	4.61E+04	6.22E+03
	KTPSO	**3.88E+04**	**2.99E+03**
70	PSO	7.06E+04	1.92E+04
	KTPSO-RI	6.72E+04	2.07E+04
	CPSO-NNIT	5.51E+04	7.07E+03
	KTPSO	**4.74E+04**	**3.65E+03**
80	PSO	9.25E+04	2.25E+04
	KTPSO-RI	8.48E+04	2.32E+04
	CPSO-NNIT	7.11E+04	8.18E+03
	KTPSO	**5.81E+04**	**5.63E+03**
90	PSO	1.23E+05	2.93E+04
	KTPSO-RI	1.14E+05	2.53E+04
	CPSO-NNIT	8.89E+04	1.13E+04
	KTPSO	**7.25E+04**	**8.22E+03**
100	PSO	2.36E+05	3.78E+04
	KTPSO-RI	1.97E+05	3.24E+04
	CPSO-NNIT	1.50E+05	1.73E+04
	KTPSO	**9.71E+04**	**1.19E+04**

In Table V, when the number of SNs increases, the means and their variances of four algorithms all increase rapidly. Their growth rates are faster than that of node number. It is because when the number of SNs increases, the problem dimension also increases dramatically, which makes meta-heuristic algorithms easier to be trapped in local optima. Through comparison, it can be found that KTPSO yields the least increase in the mean and variance compared with other algorithms as the problem dimension increases. The first reason is that KTPSO combines the advantages of GA and PSO. Specifically, KTPSO adopts the particle update method of PSO to accelerate the convergence speed and adopts crossover, mutation, and selection in GA to help itself jump out of local optima. The second reason is that it can utilize the previous optimization results to guide the search direction of particles. The results show that compared with CPSO-NNIT, our KTPSO is more efficient in solving the optimization problem of UAV-assisted WSNs.

D. Analysis of Effectiveness of Time Slot Allocation

To directly reflect the performance increment when adopting TSA, we remove the TSA part from our proposed framework and calculate its final NAEC and NL. Fig. 10 shows the results of NAEC, and it can be observed that optimizing the TSA can significantly decrease NAEC. What is more, when the number

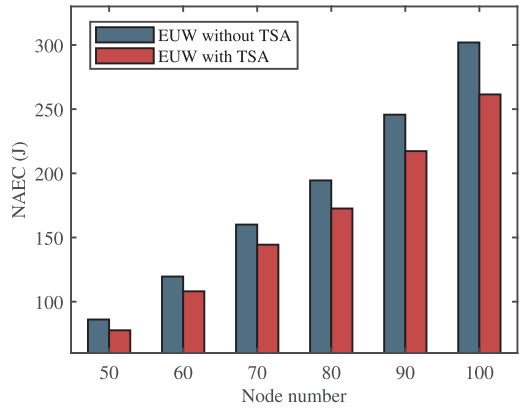

Fig. 10. NAEC with two different conditions.

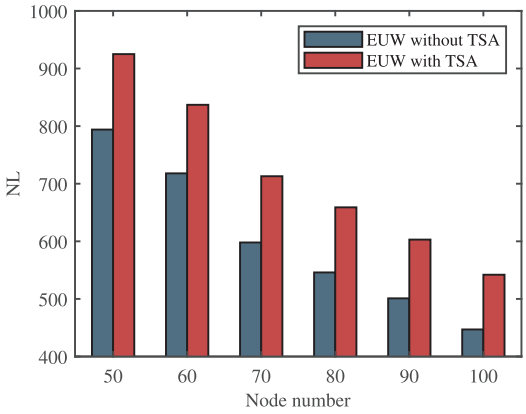

Fig. 11. NL with two different conditions.

of nodes increases, the decrease in NAEC is greater. From the scenario with 50 nodes to that with 100 nodes, the reduction rates of NAEC are 9.35%, 9.67%, 10.14%, 11.25%, 11.61%, and 13.40%, respectively. As we maintain the consistent node density in the scenarios with different node numbers, the number of nodes becomes the most important factor affecting the magnitude of interference. A larger number of nodes leads to higher significant intercluster interference. Therefore, optimizing TSA can better reduce NAEC in scenarios with many nodes. Fig. 11 shows the results of NL. From the scenario with 50 nodes to that with 100 nodes, the growth rates of NL are 16.50%, 18.22%, 19.23%, 20.59%, 20.75%, and 21.25%, respectively. We find that regardless of the number of nodes, adopting TSA does not result in a significant improvement in the SDNEC and the reduction rates of SDNEC remain below 1%. Therefore, SDNEC is not kept in detail here. For EUW without TSA, it is shown that as NAEC increases and SDNEC remains nearly unchanged, NL decreases accordingly, which is the same as the experiment results.

VI. Conclusion

WSNs have witnessed widespread applications for monitoring environmental parameters in remote areas, forests, and other natural reserves. However, as most WSNs rely on battery-powered sensors, energy efficiency, and conservation are critical to ensuring long-term operations. To address these challenges, this work proposes a framework of UAV-assisted hierarchical WSNs that can be allocated in a remote area without BSs. A joint optimization problem for the CH selection, the TSA, and the UAV path planning is mathematically formulated. To tackle it, a meta-heuristic method called KTPSO is proposed, which utilizes previous optimization results to increase the convergence speed and find better results. Experimental results demonstrate that the performance of our proposed framework is better than that of the other three state-of-the-art frameworks in four different metrics. Finally, we also demonstrate the effectiveness of knowledge transfer and TSA.

In our future research, we will extend it from the following three directions. First, we plan to consider a more complex scenario including multiple UAVs gathering data from WSNs. Second, to make the scheduling scheme more flexible, we also consider replacing the current single-objective optimization algorithm with a multiobjective one. Third, to utilize knowledge and learning experience more efficiently, other methods such as deep learning will be considered to assist optimization methods. For example, autoencoders can be adopted to reduce dimension and extract knowledge from past optimization results.

References

[1] M. A. U. Rehman, R. Ullah, B.-S. Kim, B. Nour, and S. Mastorakis, "CCIC-WSN: An architecture for single-channel cluster-based information-centric wireless sensor networks," *IEEE Internet Things J.*, vol. 8, no. 9, pp. 7661–7675, May 2021.

[2] R. Manjula, T. Koduru, and R. Datta, "Protecting source location privacy in IoT-enabled wireless sensor networks: The case of multiple assets," *IEEE Internet Things J.*, vol. 9, no. 13, pp. 10807–10820, Jul. 2022.

[3] R. M. A. Haseeb-Ur-Rehman et al., "Sensor cloud frameworks: State-of-the-art, taxonomy, and research issues," *IEEE Sensors J.*, vol. 21, no. 20, pp. 22347–22370, Oct. 2021.

[4] X. Fu, G. Fortino, W. Li, P. Pace, and Y. Yang, "WSNs-assisted opportunistic network for low-latency message forwarding in sparse settings," *Future Gener. Comput. Syst.*, vol. 91, no. 3, pp. 223–237, Feb. 2019.

[5] B. M. Sahoo, H. M. Pandey, and T. Amgoth, "GAPSO-H: A hybrid approach towards optimizing the cluster based routing in wireless sensor network," *Swarm Evol. Comput.*, vol. 60, Feb. 2021, Art. no. 100772.

[6] R. Prajapat, R. N. Yadav, and R. Misra, "Energy-efficient k-hop clustering in cognitive radio sensor network for Internet of Things," *IEEE Internet Things J.*, vol. 8, no. 17, pp. 13593–13607, Sep. 2021.

[7] S. R. Yeduri, N. S. Chilamkurthy, O. J. Pandey, and L. R. Cenkeramaddi, "Energy and throughput management in delay-constrained small-world UAV-IoT network," *IEEE Internet Things J.*, vol. 10, no. 9, pp. 7922–7935, May 2023.

[8] K. Wang, C.-M. Yu, M.-L. Ku, L.-C. Wang, and W.-K. Jia, "Joint shortest chain and fair transmission design for energy-balanced PEGASIS in WSNs," *IEEE Internet Things J.*, vol. 10, no. 8, pp. 6803–6817, Apr. 2023.

[9] J. Bi, Z. Wang, H. Yuan, J. Qiao, J. Zhang, and M. Zhou, "Self-adaptive teaching-learning-based optimizer with improved RBF and sparse autoencoder for complex optimization problems," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 7966–7972.

[10] J. Bi et al., "Multi-swarm genetic gray wolf optimizer with embedded autoencoders for high-dimensional expensive problems," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2023, pp. 7265–7271.

[11] J. Bi, H. Yuan, K. Zhang, and M. Zhou, "Energy-minimized partial computation offloading for delay-sensitive applications in heterogeneous edge networks," *IEEE Trans. Emerg. Topics Comput.*, vol. 10, no. 4, pp. 1941–1954, Oct.–Dec. 2022.

[12] H. Yuan, J. Bi, and M. Zhou, "Geography-aware task scheduling for profit maximization in distributed green data centers," *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 1864–1874, Jul.–Sep. 2022.

[13] J. Bi, H. Yuan, J. Zhang, and M. Zhou, "Green energy forecast-based Bi-objective scheduling of tasks across distributed clouds," *IEEE Trans. Sustain. Comput.*, vol. 7, no. 3, pp. 619–630, Jul.–Sep. 2022.
[14] T. Kaur and D. Kumar, "Particle swarm optimization-based unequal and fault tolerant clustering protocol for wireless sensor networks," *IEEE Sensors J.*, vol. 18, no. 11, pp. 4614–4622, Jun. 2018.
[15] S. Gong, X. Liu, K. Zheng, W. Lu, and Y.-H. Zhu, "TDMA scheduling schemes targeting high channel utilization for energy-harvesting wireless sensor networks," *IET Commun.*, vol. 15, no. 16, pp. 2097–2110, Oct. 2021.
[16] B. Li, Y. Zou, J. Zhu, and W. Cao, "Impact of hardware impairment and co-channel interference on security-reliability trade-off for wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 11, pp. 7011–7025, Nov. 2021.
[17] I. B. Prasad, S. S. Yadav, and V. Pal, "HLBC: A hierarchical layer-balanced clustering scheme for energy efficient wireless sensor networks," *IEEE Sensors J.*, vol. 21, no. 22, pp. 26149–26160, Nov. 2021.
[18] V. Pal, G. Singh, and R. Yadav, "Balanced cluster size solution to extend lifetime of wireless sensor networks," *IEEE Internet Things J.*, vol. 2, no. 5, pp. 399–401, Oct. 2015.
[19] S. Verma, S. Zeadally, S. Kaur, and A. K. Sharma, "Intelligent and secure clustering in wireless sensor network (WSN)-based intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 13473–13481, Aug. 2022.
[20] N. Kato et al., "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 140–147, Aug. 2019.
[21] J. Ye, S. Dang, B. Shihada, and M.-S. Alouini, "Space-air-ground integrated networks: Outage performance analysis," *IEEE Trans. Wireless Commun.*, vol. 19, no. 12, pp. 7897–7912, Dec. 2020.
[22] P. Zhang, Y. Li, N. Kumar, N. Chen, C.-H. Hsu, and A. Barnawi, "Distributed deep reinforcement learning assisted resource allocation algorithm for space-air-ground integrated networks," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 3, pp. 3348–3358, Sep. 2023.
[23] Q. Tang, Z. Fei, B. Li, and Z. Han, "Computation offloading in LEO satellite networks with hybrid cloud and edge computing," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9164–9176, Jun. 2021.
[24] Y. Xia, M. Zhou, X. Luo, Q. Zhu, J. Li, and Y. Huang, "Stochastic modeling and quality evaluation of infrastructure-as-a-service clouds," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 162–170, Jan. 2015.
[25] S. Zhang, H. Zhang, Q. He, K. Bian, and L. Song, "Joint trajectory and power optimization for UAV relay networks," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 161–164, Jan. 2018.
[26] F. Xiong et al., "Energy-saving data aggregation for multi-UAV system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9002–9016, Aug. 2020.
[27] C. Lin, G. Han, X. Qi, J. Du, T. Xu, and M. Martínez-García, "Energy-optimal data collection for unmanned aerial vehicle-aided industrial wireless sensor network-based agricultural monitoring system: A clustering compressed sampling approach," *IEEE Trans. Ind. Informat.*, vol. 17, no. 6, pp. 4411–4420, Jun. 2021.
[28] J. Baek, S. I. Han, and Y. Han, "Energy-efficient UAV routing for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1741–1750, Feb. 2020.
[29] C. Kai, H. Li, L. Xu, Y. Li, and T. Jiang, "Energy-efficient device-to-device communications for green smart cities," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1542–1551, Apr. 2018.
[30] A. A. Taleb, "Using minimum connected dominating set for mobile sink path planning in wireless sensor networks," *Int. J. Commun. Netw. Inf. Secur.*, vol. 15, no. 1, pp. 1–8, Apr. 2023.
[31] P. K. Donta, T. Amgoth, and C. S. R. Annavarapu, "An extended ACO-based mobile sink path determination in wireless sensor networks," *J. Ambient Intell. Humanized Comput.*, vol. 12, pp. 8991–9006, Oct. 2020.
[32] X. Wu, Z. Chen, Y. Zhong, H. Zhu, and P. Zhang, "End-to-end data collection strategy using mobile sink in wireless sensor networks," *Int. J. Distrib. Sens. Netw.*, vol. 18, no. 3, Mar. 2022, Art. no. 15501329221077932.
[33] A. M. Khedr, Z. Al Aghbari, and B. E. Khalifa, "Fuzzy-based multi-layered clustering and ACO-based multiple mobile sinks path planning for optimal coverage in WSNs," *IEEE Sensors J.*, vol. 22, no. 7, pp. 7277–7287, Apr. 2022.
[34] Y. Gao, J. Wang, W. Wu, A. K. Sangaiah, and S.-J. Lim, "A hybrid method for mobile agent moving trajectory scheduling using ACO and PSO in WSNs," *Sensors*, vol. 19, no. 3, p. 575, Jan. 2019.
[35] S. Roy, N. Mazumdar, and R. Pamula, "An optimal mobile sink sojourn location discovery approach for the energy-constrained and delay-sensitive wireless sensor network," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 8, pp. 10837–10864, Jan. 2021.
[36] K. M. Chowdary and V. Kuppili, "Enhanced clustering and intelligent mobile sink path construction for an efficient data gathering in wireless sensor networks," *Arabian J. Sci. Eng.*, vol. 46, no. 9, pp. 8329–8344, Mar. 2021.
[37] B. R. Al-Kaseem, Z. K. Taha, S. W. Abdulmajeed, and H. S. Al-Raweshidy, "Optimized energy–efficient path planning strategy in WSN with multiple mobile sinks," *IEEE Access*, vol. 9, pp. 82833–82847, 2021.
[38] M. Biabani, N. Yazdani, and H. Fotouhi, "EE-MSWSN: Energy-efficient mobile sink scheduling in wireless sensor networks," *IEEE Internet Things J.*, vol. 9, no. 19, pp. 18360–18377, Oct. 2022.
[39] J. Amutha, S. Sharma, and 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.*, vol. 203, Oct. 2022, Art. no. 117334.
[40] J. Chen et al., "Efficient data collection in large-scale UAV-aided wireless sensor networks," in *Proc. 11th Int. Conf. Wireless Commun. Signal Process.*, 2019, pp. 1–5.
[41] S. Alfattani, W. Jaafar, H. Yanikomeroglu, and A. Yongacoglu, "Multi-UAV data collection framework for wireless sensor networks," in *Proc. IEEE Glob. Commun. Conf.*, 2019, pp. 1–6.
[42] Y. Chen and S. Shen, "A UAV-based data collection approach for wireless sensor network," in *Proc. 7th Int. Conf. Inf. Sci. Control Eng.*, Changsha, China, 2020, pp. 2298–2302.
[43] K. Saxena, N. Gupta, J. Gupta, D. K. Sharma, and K. Dev, "Trajectory optimization for the UAV assisted data collection in wireless sensor networks," *Wireless Netw.*, vol. 28, no. 4, pp. 1785–1796, Mar. 2022.
[44] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and J. Henry, "UAV trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9540–9554, Sep. 2021.
[45] 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 Things J.*, vol. 9, no. 13, pp. 11214–11226, Jul. 2022.
[46] C. Zhan and R. Huang, "Energy minimization for data collection in wireless sensor networks with UAV," in *Proc. IEEE Glob. Commun. Conf.*, 2019, pp. 1–6.
[47] W. Wu, J. Xu, and Y. Sun, "Integrate assignment of multiple heterogeneous unmanned aerial vehicles performing dynamic disaster inspection and validation task with Dubins path," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 4, pp. 4018–4032, Aug. 2023.
[48] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, pp. 2329–2345, Apr. 2019.
[49] Z. Wei et al., "UAV-assisted data collection for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 15460–15483, Sep. 2022.
[50] Z. Zhang and X. Yang, "Bio-inspired motion planning for reaching movement of a manipulator based on intrinsic tau jerk guidance," *Adv. Manuf.*, vol. 7, no. 1, pp. 315–325, Aug. 2019.
[51] X.-F. Liu et al., "Neural network-based information transfer for dynamic optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1557–1570, May 2020.
[52] P. C. S. Rao, P. K. Jana, and H. Banka, "A particle swarm optimization based energy efficient cluster head selection algorithm for wireless sensor networks," *Wireless Netw.*, vol. 23, no. 7, pp. 2005–2020, 2017.

Rui Lai received the B.E. and M.E. degrees in mechanical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2019 and 2021, respectively. He is currently pursuing the Ph.D. degree with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China.

His research interests include evolutionary optimization and machine learning.

Boyuan Zhang received the B.E. degree in automation from Beihang University, Beijing, China, in 2022, where he is currently pursuing the master's degree with the School of Automation Science and Electrical Engineering.

His research interests include optimization algorithms, wireless sensor networks, energy management, machine learning, and deep learning.

Guanghong Gong received the B.S., M.S., and Ph.D. degrees from Beihang University, Beijing, China, in 1990, 1993, and 1997, respectively.

She is currently a Professor with the Key Laboratory of Advanced Simulation Aeronautical Technology, Beihang University. She is also a Managing Director of the China Simulation Society. Her research interests are virtual reality, artificial intelligence, and distributed interactive simulation.

Prof. Gong has won the Science and Technology Progress Award of the Ministry of Education and the National Defense Science and Technology Progress Award for many times.

Haitao Yuan (Senior Member, IEEE) received the Ph.D. degree in computer engineering from the New Jersey Institute of Technology (NJIT), Newark, NJ, USA, in 2020.

He is currently an Associate Professor with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China. His research interests include cloud computing, edge computing, data centers, big data, machine learning, deep learning, and optimization algorithms.

Dr. Yuan received the Chinese Government Award for Outstanding Self-Financed Students Abroad, the 2021 Hashimoto Prize from NJIT, and the Best Paper Award in the 17th ICNSC. He serves as an Associate Editor for *Expert Systems With Applications*.

Jinhong Yang received the Ph.D. degree in computer application technology from Harbin Engineering University, Harbin, Heilongjiang, China, in 2017.

She is currently a Senior Engineer with CSSC Systems Engineering Research Institute, Beijing, China. Her research interests include machine learning, data mining, knowledge reasoning, deep learning, and intelligent optimization.

Dr. Yang serves as a Reviewer for *Expert Systems With Applications* and *International Journal of Machine Learning and Cybernetics*.

Jia Zhang (Senior Member, IEEE) received the Ph.D. degree in computer science from the University of Illinois at Chicago, Chicago, IL, USA, in 2000.

She is currently the Cruse C. and Marjorie F. Calahan Centennial Chair in Engineering, a Professor with the Department of Computer Science, Lyle School of Engineering, Southern Methodist University. Her research interests emphasize the application of machine learning and information retrieval methods to tackle data science infrastructure problems, with a recent focus on scientific workflows, provenance mining, software discovery, knowledge graph, and interdisciplinary applications of all of these interests in Earth science.

MengChu Zhou (Fellow, IEEE) received the Ph.D. degree from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

Then, joined the New Jersey Institute of Technology, where he is currently a Distinguished Professor. He has over 900 publications, including 12 books, more than 600 journal papers (more than 450 in IEEE transactions), 28 patents, and 29 book chapters. His interests are in Petri nets, automation, Internet of Things, and big data.

Dr. Zhou is a Fellow of IFAC, AAAS, CAA, and NAI.