IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 9, 2020, accepted April 18, 2020, date of publication April 23, 2020, date of current version May 11, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2989763*

# DEDC: Joint Density-Aware and Energy-Limited Path Construction for Data Collection Using Mobile Sink in WSNs

**WEIMIN WEN[1], CUIJUAN SHANG[2], CHIH-YUNG CHANG[2], (Member, IEEE), AND DIPTENDU SINHA ROY[3]**

[1]School of Computer and Information Engineering, Chuzhou University, Chuzhou 239000, China
[2]Department of Computer Science and Information Engineering, Tamkang University, New Taipei City 25137, Taiwan
[3]Department of Computer Science and Engineering, National Institute of Technology Meghalaya, Shillong 793003, India

Corresponding author: Chih-Yung Chang (cychang@mail.tku.edu.tw)

This work was supported in part by the Engineering Technology Research Center of Intelligent Perception and Health Endowment Anhui Education Department, Natural Science Foundation of Anhui Higher Education under Grant KJ2019ZD44, Grant KJ2019A0648, and Grant KJ2018B01, in part by the Science and Technology Major Project of Anhui Province under Grant 201903a06020026, and in part by the Natural Science Foundation of Anhui Provincial under Grant 1908085MF211.

**ABSTRACT** Data collection is one of the most important issues in Wireless Sensor Networks (WSNs). Many data collection algorithms have been proposed for collecting data using a mobile sink. However, the appropriateness and the number of the selected anchors, which significantly impact the network lifetime of the given WSNs, still can be improved. This paper proposes a Joint Density-Aware and Energy-Limited Path Construction algorithm for Data Collection, called DEDC, aiming to select as more as possible appropriate anchors under the path length constraint for prolonging the network lifetime. Initially, the proposed DEDC determines the grid size according to the path length constraint, partitions the monitoring region into several grids and identifies the grids to be balance or unbalance grids. Based on the partitioned grids, the proposed DEDC constructs a regular path and then further adjusts the path segments for these unbalanced grids. The regular path construction and path adjustment aim to construct a path passing through as more as possible anchors for balancing the forwarding loads and prolonging the network lifetime. Performance evaluations reveal that the proposed DEDC outperforms existing data collection mechanisms in terms of energy consumption, network lifetime, and SD energy consumptions.

**INDEX TERMS** Wireless sensor networks, mobile sink, energy consumption, data collection, anchor.

## I. INTRODUCTION

Wireless sensor networks (WSNs) consist of a large number of sensors deployed in a given region. The WSNs have been used in many applications, including traffic monitoring [1], [2], smart city [3], healthcare [4], [5], trajectory tracking [6] as well as environmental data collection [7]–[9]. Each sensor is composed of sensing and communication components. The sensing component is responsible for environmental monitoring or event detection while the communication component receives data from neighbors and transmits its reading or received data to the node closer to the sink node. Since sensors are usually battery-powered, how to collect data from sensors to reduce energy consumption is the primary challenge of the WSNs.

Many studies have presented data collection schemes in recent years. These studies are typically divided into two classes: no-data-forwarding and data-forwarding. In the class of no-data-forwarding, studies [10]–[12] adopted the mobile sink to visit all sensor nodes and directly collect data from the visited sensors. Since there is no data forwarding, the energy consumptions of all sensor nodes can be balanced. However, the length of the constructed path is too long, leading to the problems of energy exhaustion of mobile sink or buffer overflow of the static sensor.

Some other studies [13]–[18] fallen in the data-forwarding class selected a set of sensors, called anchor nodes, and used the mobile sink to visit the anchor nodes. All the other sensors transmitted their data to the anchors and then the anchors

The associate editor coordinating the review of this manuscript and approving it for publication was Arun Prakash.

directly forwarded their data to the mobile sink. All the sensor nodes were partitioned into several clusters each of which was organized as a sub-tree rooted by one anchor. In each sub-tree, each node transmitted its sensing data to the root along with the topology of the sub-tree. Then the proposed mechanisms constructed a path passing through the roots of all sub-trees. After that, the mobile sink moved along the path and collected data from the roots. However, the data anchor nodes selected by these studies might not be appreciated. This might reduce the lifetime of wireless sensor networks.

Given a mobile sink and a set of sensors, this paper proposes a data collection algorithm, called DEDC, which selects the anchors as more as possible by considering the balanced and unbalanced deployments. Initially, the monitoring region is partitioned into several equal-sized grids based on the constrained path length. The constrained path length is divided into two parts, the regular path length, and the irregular path length. Then a regular path that passes through each grid is constructed. All sensors that are located within the communication range of the constructed regular path will be considered as anchors. This helps maximize the number of anchors and minimize the number of hops from each sensor to the anchors in each grid. To overcome the unbalanced deployment where some holes might have existed, the regular path is further adjusted such that the irregular path can be constructed for those unbalanced grids using the budget of irregular path length. The following highlights the contributions and key features of this paper.

(1) ***Finding as more as possible anchors.*** The proposed DEDC utilizes the budget of regular path length to construct a path passing through all grids. All sensor nodes which are fallen in the communication range of the constructed path will be treated as the anchors. Therefore, the proposed DEDC finds as more as possible anchors, aiming to reduce the number of hops from each sensor to the corresponding anchor, prolonging the network lifetime of the given WSNs.
(2) ***Dynamically adjusting the path for unbalance deployment.*** The proposed DEDC reserves a certain ratio of path length as the budget for adjusting path. For the unbalanced grids, the regular path that passes through the grids might not find the appropriate anchors. For the unbalance grids, the anchor will be selected based on the benefit of energy conservation. Then the regular path can be dynamically adjusted based on the newly selected anchors such that the energy consumption for forwarding packets from each sensor to the selected anchor can be minimized.
(3) ***Prolonging the network lifetime***. The network lifetime of the given WSNs can be prolonged. This occurs because of two major reasons. First, the constructed regular path finds more anchors for balancing the forwarding loads of each anchor. This significantly helps prolong the network lifetime. The other reason is that the irregular path can further conserve the energy consumption of sensors in the unbalanced grids. As a result, the mobile sink which moves along the constructed path and collects data from anchors can prolong the network lifetime.

The remaining parts of this paper are organized as follows. Section II reviews the related works and compares them with our work. Section III presents the network environment, assumptions, notations and problem formulation of this paper. Section IV presents the design of the proposed DEDC. Section V presents the performance improvement of the proposed mechanism against the existing studies. Finally, Section VI presents the conclusions.

## II. RELATED WORKS

In literature, many studies adopted mobile sink for collecting data from sensors. These studies can be further classified into two categories: no-data-forwarding and data-forwarding. The following briefly reviews these related works.

### A. NO-DATA-FORWARDING

In this category, most studies adopted a mobile sink to visit all sensor nodes and collect data from them. In [10], the sensing and communication modules were mounted on animals or vehicles that moved without control to collect data from static sensors. Since the path was not well planned, the data collection was usually time-consuming, which raised the problems that the collected data were not fresh or the overflowed buffer occurred in static sensors.

Reference [11] proposed a heuristic algorithm that handled the path construction and speed control for the mobile sink to visit and collect data from each sensor. The speed control aimed to guarantee that the sensor data could be fully transmitted to the mobile sink while the path construction aimed to minimize the data delivery latency. However, it was time-consuming for visiting each sensor, especially for a large scale sensor network.

Reference [12] proposed an approach to solving the path selection problem. The proposed approach established a path of the mobile sink such that it can collect all the data before the buffer of each sensor overflows. The mobile sink collected data at a constant speed. They showed that the problem was NP-hard and formulated it as an Integer Linear Programming (ILP) problem. However, for a large number of static sensors, the approach resulted in high computational complexity.

### B. DATA-FORWARDING

Since visiting each sensor becomes impractical when there are a large number of sensor nodes, some other studies [13]–[18] fallen in the data-forwarding category allocated the mobile sink to visit a subset of the sensor nodes, called anchors. All the other sensors only needed to transmit their data to the corresponding anchor.

Reference [13] proposed a data aggregation mechanism which allowed the mobile sink to collect data from WSNs where the path is not previously determined. The mobile sink stopped at a location in the network and broadcasted an aggregate query which was flooded with a limited number of hops and then the data were aggregated and collected from sensor nodes. The proposed mechanism ensured the uniform energy consumption among all nodes, aiming at prolonging the network lifetime. However, it did not take into account the unbalanced deployment where some area might have few sensors or even no sensor.

Tunca *et al.* [14] proposed a ring routing approach that aimed to minimize the flooding overhead when the mobile sink moved and collected data from all sensor nodes. The proposed approach established a virtual ring structure that allowed the current sink position to be easily delivered to the ring. The regular nodes were easy to acquire the sink position from the ring with minimal overhead whenever needed. In addition, the ring nodes can switch roles with regular nodes by executing the proposed mechanism, thus mitigating the hotspot problem. Though each node can timely maintain the route from itself to the mobile sink and transmit its data to the mobile sink, the packet forwarding from each node to the mobile sink still raised the energy consumption problem which reduced the network lifetime of the WSNs.

Zhu *et al.* [15] proposed a greedy scanning data collection strategy (GSDCS). The GSDCS divided the whole monitoring region into many grids. Each grid cell was labeled with a row-column number (RCN) and direction number (DN). The mobile sink collected data from one grid to another. According to the amount of data received from each direction, the mobile sink moved along the direction with more sensory data. However, the GSDCS did not consider the unbalanced deployment where some grids might have few or even no sensors. The mobile sink consumed its energy to traverse these grids but only collected a few data. As a result, it was not cost efficient if the path passes through the sparse grids.

Yang *et al.* [16] proposed a heuristic algorithm for constructing a path for data collection. It firstly found the root of the tree and then constructed a tree for minimizing the hop distance from any node to the root. Then it selected the sensor that has the high residual energies and a large number of packets required to be forwarded as an anchor. After that, it applied the nearest neighbor algorithm to find a movement path with minimum length to connect the root and the selected anchor. The anchor selection process will be repeatedly performed until the length of the constructed path was larger than the given path length. However, the heuristic algorithm did not take into account the distance between the current anchor and the next anchor. As a result, it might construct an inefficient path for the mobile sink. The path construction can be more efficient to include more anchors for better balancing the forwarding loads.

Salarian *et al.* [17] proposed a weighted rendezvous planning (WRP). The WRP initially constructed a tree and assigned a weight to each sensor node according to the number of forwarding packets and the number of hops from that sensor to the tree root. The proposed approach selected the sensor with the highest weight as the anchor. Then the path that passed through all anchors was constructed. Each anchor then constructed subtree and played the role of the root. Each sensor should join one subtree whose root was closest to the sensor. Then the weight of each sensor should be recalculated such that the new anchor can be further selected. The anchor selection process was repeatedly performed until the length of the constructed path was larger than the predefined path length. The proposed WRP has the advantage that the selected anchor can reduce the energy consumption for the packet forwarding from each sensor to the base station. However, the WRP did not take into account the path length from the location of the current anchor to the next anchor. The path construction can be more efficient and more anchors can be found to better balance the forwarding loads.

In [18], an Energy-Aware Path Construction algorithm (EAPC) was proposed. The EAPC firstly constructed a minimum spanning tree rooted at the base station. Then it calculated the benefit of each sensor node and selected a subset of sensor nodes with maximal benefits as anchors. Different from the WRP [17], the EAPC selected the next anchor by considering the distance between the current anchor and the next one. As a result, the path with length constraint can be better utilized for passing through more anchors, balancing the forwarding workload and hence improving the network lifetime.

All of the algorithms mentioned above emphasized the improvement of the energy unbalanced issue or aimed to cope with the data fresh problem. However, most of them selected anchor nodes one by one using a greedy algorithm, without globally constructing a regular path and locally adjusting the regular path to an irregular one. Different from the previous studies, the proposed DEDC initially constructs a regular path under the budget of regular path length constraint, aiming at finding more anchors nodes to balance the forwarding loads. Since the path is regularly constructed, it passes over the monitoring region. All the sensors that fall in the communication range of the path can play the anchor role, increasing the number of possible anchors. This is different from the previous studies since the anchor nodes were selected one by one. Then the proposed DEDC locally adjusts the path under the constraint of the irregular path constraint, aiming at finding the appropriate anchors. As a result, the proposed DEDC outperforms existing studies [17], [18] in terms of network lifetime, energy consumption, and SD energy consumptions.

Table 1 summaries the comparisons between the proposed mechanism and the related studies. The '*Considering Balanced Deployments*' presents whether or not the proposed mechanism considered the sensor deployment issues, including the density and the balancing degree. The '*Selecting Anchor*' is a metric to observe whether or not the proposed mechanism selected anchors to avoid the problems including long path length, data non-fresh or buffer overflow. The '*Anchor Selection Policy*' can be one of the three policies including no anchor, denoted by '×', point-based and path based. The point-based policy represents that the anchor is selected one by one in each round while the path-based policy represents that the anchors are determined by the constructed path. That is, all anchors fallen in the communication range

**TABLE 1. Comparisons of the proposed DEDC and existing mechanisms.**

| Studies | Considering Balanced Deployments | Selecting Anchor | Anchor Selection Policy | Adjusting Path | Data Forwarding |
|---|---|---|---|---|---|
| [10] | × | × | × | × | × |
| [11] | × | × | × | × | × |
| [12] | × | × | × | × | × |
| [13] | × | × | Point-Based | × | ○ |
| [14] | × | × | Point-Based | ○ | ○ |
| [15] | × | ○ | Point-Based | × | ○ |
| [16] | × | ○ | Point-Based | × | ○ |
| [17] | × | ○ | Point-Based | × | ○ |
| [18] | × | ○ | Point-Based | × | ○ |
| The proposed *DEDC* | ○ | ○ | Path-Based | ○ | ○ |

of the constructed path will play the role anchor for balancing the forwarding loads. In case the sensors are normally distributed, the path-based anchor selection policy can obviously find more anchors, as compared with the point-based anchor selection policy. The 'Adjusting Path' indicates whether or not the path is firstly constructed and then further improved aiming to find more anchors or construct a more efficient path. The last matric is 'Data Forwarding' which indicates whether or not the proposed mechanism adopted the data forwarding policy. In comparison with the related studies, the proposed DEDC selects anchors by applying the path-based policy and thus constructs the regular path for finding more anchors. The proposed DEDC also handles the unbalanced deployment for finding better anchors and adjusts the regular path such that all sensors can forward their data to the anchors with a shorter hop-distance, as compared with existing studies [17] and [18].

## III. NETWORK ENVIRONMENT AND PROBLEM FORMULATION

This section presents the network environment and the assumptions of the given WSNs. The problem formulation is subsequently presented.

### A. NETWORK ENVIRONMENT

This paper assumes that a set of $n$ static sensors $S = \{s_1, s_2 \ldots, s_n\}$ are randomly deployed in a regular area $M$. A mobile sink moves along the path $P$ aiming to collect data from all sensors with a constraint of maximal length $\mathbb{L}$. The constraint of path length for mobile sink reflects the requirements of data fresh carried by the mobile sink and limited energy of the mobile sink. It is also assumed that the mobile sink is aware of its location and the locations of all static sensors. Let $h(s_j, P)$ denote the minimum number of hops from $s_j$ to $P$. Let $A = \{a_j | a_j \in S \text{ and } h(a_j, P) = 0\}$ denote the set of anchors which is a subset of $S$ and satisfy $h\left(a_j, P\right) = 0$. Each anchor node can directly transmit data to the mobile sink when the mobile sink moves along path $P$ and falls in its communication range. Let $s_0$ represent the base station. The mobile sink will leave the base station $s_0$, move and collect data from anchors along the path and then go back to the base station.

### B. PROBLEM FORMULATION

This paper aims to construct a path along which the mobile sink moves and collects data from sensors while the lifetime of the given WSN can be maximized. The lifetime of a WSN is defined by the time length starting from the time point that all sensors are working for monitoring region $M$ to the time point that the first sensor is failure due to energy exhaustion. The power consumption of a sensor mainly includes sensing, transmitting and receiving data. Assume that the size of each packet is $k$-bit. Let $d_{ij}$ be the distance between $s_i$ and $s_j$.The energy consumption for the sender $s_i$ to transmit one packet ($k$ bits) to its parent $s_j$ is measured by:

$$E_t = k\varepsilon_1 + kd_{i,j}^2\varepsilon_2 \tag{1}$$

where $\varepsilon_1$ is the transmitting circuit of energy consumption per bit, $\varepsilon_2$ is the amplifier circuit of energy consumption per bit.

The energy consumption of each node for receiving one packet ($k$ bits) can be measured by (2):

$$E_r = k\beta \tag{2}$$

where $\beta$ is the receiving circuit energy consumption for a bit. Let $E_s$ be the energy cost for executing sensing operation. Let $N(s_j)$ denote the set of nodes rooted by the sensor $s_j$. The total energy cost of the sensor $s_j$ in a round is expressed as:

$$E_j = (N(s_j) + 1) \times E_t + (N(s_j) \times E_r + E_s \tag{3}$$

The network lifetime highly depends on anchor lifetime because that each anchor consumes more energy than any other sensor in its subtree. This paper aims to minimize the energy consumption of the bottlenecked anchor that consumes the most energy among all the anchors. Equation (4) reflects this goal.

***Objective:***

$$Min\ Max_{a_i \in A} E_i \tag{4}$$

**TABLE 2.** Main notations used in this paper.

| Symbol | Description |
|---|---|
| $h(s_j, P)$ | The minimum number of hops from $s_j$ to $P$ |
| $E_t$ | Energy consumption for transmitting one packet |
| $E_r$ | Energy consumption for receiving one packet |
| $\varrho$ | The data generation rate of each sensor node |
| $b$ | The buffer size of each sensor |
| $\mathbb{L}$ | The total path length |
| $\varphi$ | The scale parameter |
| $l_g$ | The edge length of each square grid |
| $d(C_x, C_y)$ | The minimal distance of sensor nodes $s_i$ and $s_j$ |
| $\zeta_{trh}$ | The distance threshold |
| $NP(y)$ | The number of packets received by the sensor node $s_y$ |
| $H(y, \tilde{a}_q)$ | The number of hops from $s_y$ to the root $\tilde{a}_q$ of tree containing sensor node $s_y$ |
| $\rho_y$ | The benefit index obtained by selecting sensor node $s_y$ as the anchor |
| $s^{best}$ | The best sensor node to be selected as the anchor in grid $\tilde{g}_{i,j}$ |
| $\eta_i^{best}$ | The best small grid to be passed by the data collection path in each row $R_i$ |
| $p_{x,y}$ | The path segment passing though $g_{x,y}$ |

The following presents the constraints which should be satisfied when achieving the objective (4). Let $\ell_p$ denote the shortest Hamiltonian path passing through each anchor node. An important constraint for the path length is the limited energy of the mobile sink. The energy required for receiving packets from $k$ anchors is:

$$\sum_{a_j \in A, j=1}^{k} (N(a_j) + 1) \times E_r$$

Let $E_{moving}$ denote the energy consumption of mobile sink for moving one unit distance. Let $E_{full}$ denote the original battery energy. The minimal energy capacity required for mobile sink moving for length $L_P$ is:

$$E_{full} = E_{moving} \times L_P + \sum\nolimits_{a_j \in A, j=1}^{k} (N(a_j) + 1) \times E_r \tag{5}$$

$$L_P = (E_{full} - \sum\nolimits_{a_j \in A, j=1}^{k} (N(a_j) + 1) \times E_r)/E_{moving} \tag{6}$$

The following constraint gives the lower bound of path length $\mathbb{L}$.

### 1) PATH LENGTH CONSTRAINT

$$\mathbb{L} \geq min\left(\ell_p, L_P\right) \tag{7}$$

The anchor should receive all the data packets generated by the sensors in its sub-tree. Let $\varrho$ be the data generation rate of each sensor node. Let $b$ denote the buffer size of each sensor. Let $T$ denote the time length of each round which is defined by the time required for round trip from the base station along path $P$. The number of all packets for anchor $a_j$ in each round would be $N(a_j) \cdot \varrho \cdot T$. To prevent the buffer overflow, $T$ should satisfy the following constraint.

### 2) BUFFER CONSTRAINT

$$(N(a_j) + 1) \cdot \varrho \cdot T \leq b \tag{8}$$

Let $B(s_j)$ denote the set of neighboring sensors of $s_j$. Constraint (9) ensures that any non-anchor sensor $s_j$ can find a forward $s_k$ to relay its packets to the anchor node.

### 3) DATA RELAY CONSTRAINT

$$h\left(s_k, p_i\right) = h\left(s_j, p_i\right) - 1, \quad \exists s_k \in B(s_j) \tag{9}$$

Let $\lambda_i$ denote the data buffered in the anchor $a_j$. Let $\nu$ denote the data transmission rate $T_j = \{t_{j,1}, t_{j,2}, \ldots t_{j,q}\}$ denote the contacting period that mobile sink passes through the communication range of $a_j$. Let Boolean variable $\rho_{j,k}$ denote whether or not anchor node $a_j$ can transmit data to mobile sink during the $t_{j,k} \in T_j$. That is,

$$\rho_{j,k} = \begin{cases} 1, & s_j\ can\ transmit\ \ data\ to\ mobile\ sink\ during \\ & the\ t_{j,k} \in T_j \\ 0, & otherwise \end{cases}$$

To ensure the mobile sink can receive all data buffered in each anchor node, the following *Data transportation constraint* (10) should be satisfied.

### 4) DATA TRANSPORTATION CONSTRAINT

$$\sum_{k=1}^{q} \rho_{j,k} \times t_{j,k} \times \nu \geq \lambda_i \tag{10}$$

## IV. THE PROPOSED DEDC ALGORITHM

The proposed DEDC algorithm is composed of two phases: Initial Path Construction phase and Path Adjustment phase. The Initial Path Construction phase aims to partition the rectangle region into multiple square grids and construct an initial path along which the mobile sink can move. Then the Path Adjustment phase aims to balance the forwarding load of each anchor node by adjusting the constructed path. The following presents the details of the two phases.

### A. INITIAL PATH CONSTRUCTION PHASE

The initial path construction phase mainly consists of two tasks. The first task aims to partition the rectangle area $M$ into several equal-sized grids and construct a path for the mobile sink. The mobile sink will partition the rectangle area $M$ into a set of equal-size grids while the second task aims to construct an initial path. Assume the size of the monitoring region is $\mathcal{L} \times \mathcal{W}$, where the length and width of $M$ are $\mathcal{L}$ and $\mathcal{W}$, respectively. In the first task, the region $M$ is partitioned into a set of $m = q_1 \times q_2$ square grids $\mathbb{G} = \{g_{1,1}, g_{1,2}, \ldots, g_{q_1,q_2}\}$, where $q_2$ is even. In the second task, a snake-like path is constructed starting from grid $g_{1,1}$, passing through each grid once and finally going back to $g_{1,1}$. The mobile sink will move along the path and collect data . Fig.1 gives an example where the blue line denotes the constructed path while the red nodes denote the anchor nodes.

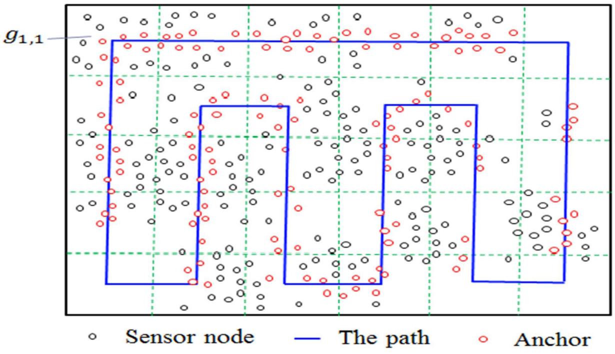


**FIGURE 1.** **The initial path.**

Recall that the path length is limited by $\mathbb{L}$. The total length $\mathbb{L}$ will be divided into two parts, $\mathbb{L}_{init}$ and $\mathbb{L}_{adjust}$, where $\mathbb{L}_{init}$ denotes the length of the initial path and $\mathbb{L}_{adjust}$ denotes the length for supporting path adjustment. Let $\varphi$ denote the scale parameter. It is obvious that we have

$$\mathbb{L} = \mathbb{L}_{init} + \mathbb{L}_{adjust} \tag{11}$$

$$\mathbb{L}_{adjust} = (1 - \varphi)\,\mathbb{L} \tag{12}$$

Since parameter $\varphi$ can impact the identification result of the network lifetime, its value will be discussed in the simulation section.

Next, we will discuss the grid size which highly depends on the limited length of the initial path. Assume the length of the square grid is $l_g$. Let $P_{init}$ denote the initial path. Let $P_{init}^{vert}$ and $P_{init}^{hori}$ denote the vertical and horizontal parts of $P_{init}$, respectively. As shown in Fig.1, there are $\mathcal{W}/l_g$ columns. Ignoring the first and last columns, the length of each vertical blue line passing through each column is $\mathcal{L}-2l_g$. The vertical blue lines passing through the first and last columns have equal length, which has additional $l_g$ length than the other vertical blue lines. Therefore, the length of $P_{init}^{vert}$ can be calculated by

$$\frac{\mathcal{W}}{l_g}\left(\mathcal{L} - 2l_g\right) + 2l_g.$$

The length of $P_{init}^{hori}$ is $2\left(\mathcal{W}-l_g\right)$. As a result, the value of $\mathbb{L}_{init}$ can be represented by

$$\mathbb{L}_{init} = \frac{\mathcal{W}}{l_g}\left(\mathcal{L} - 2l_g\right) + 2l_g + 2\left(\mathcal{W}-l_g\right) \tag{13}$$

Then the grid length can be obtained by the calculation

$$l_g = \frac{\mathcal{W}\mathcal{L}}{\mathbb{L}_{init}} \tag{14}$$

### *B. PATH ADJUSTMENT PHASE*

The energy consumption of the mobile sink in each grid depends on the segment length of the path belonging to that grid. Since the total path length of the mobile sink is limited, the path length should be treated as a limited resource, which should be allocated to each grid efficiently. As shown in Fig. 1, there are few sensors or even no sensor deployed in grids $g_{5,1}$,$g_{1,6}$ and $g_{5,6}$. That is, the mobile sink consumes its energy to traverse these grids can only collect few data. Therefore, it does not cost efficient if the same segment lengths allocated to the grids with sparse sensors. How to allocate the proper length of the path segment to each grid is an important issue that should be further investigated. To develop policies for allocating the segment length to each grid, the following defines balance and unbalance grids, which will be applied by different segment allocation policies.

A grid $g_{i,j}$ is said to be balanced if the set of sensor nodes $\{s_1, s_2, \ldots s_{n_i}\} \in g_{i,j}$ are uniformly distributed. More specifically, consider a grid to be partitioned into $\delta * \delta$ smaller equal-sized subgrids. In case that several consecutive subgrids do not contain any sensor, the grid $g_{i,j}$ is unbalanced. Otherwise, grid $g_{i,j}$ is said to be balanced. The following illustrates how to distinguish a grid to be balanced or unbalanced.

#### 1) IDENTIFICATION OF BALANCED AND UNBALANCED GRID TASK

This task aims to identify whether or not a given grid is a balanced grid. The unbalanced grid is characterized by several subgrids without containing any sensor. This implies that there exists a large enough space that consists of these empty subgrids. To check this property, the agglomerative Hierarchical Clustering (*HC*) algorithm will be applied to the sensor nodes in each grid $g_{i,j} \in G$, aiming to distinguish the grid to be balanced or unbalanced. Let $g_b$ denote the balanced grid and $g_{\tilde{b}}$ denote the unbalanced grid. Given a grid $g_{i,j}$, this task will perform the following steps.

The following presents the *HC* algorithm applied to the set of sensors in a grid. Initially, each sensor in the grid $g_{i,j}$ will form an independent cluster. In each round, a merging procedure will be applied to merge the nearest two clusters into one. All sensors in the grid $g_{i,j}$ will be merged round by round until they have been merged into one cluster and finally a hierarchical tree is formed. When applying the merging procedure, the distance between the two clusters should be measured. Consider two clusters $C_x$ and $C_y$. Let $d(C_x, C_y)$ denote the minimal distance of sensor nodes $s_i$ and $s_j$, where sensor $s_i$ in $C_x$ and $s_j$ in $C_y$. That is,

$$d\left(C_x, C_y\right) = \min_{\forall s_i \in C_x, \forall s_j \in C_y} d\left(s_i, s_j\right) \tag{15}$$

Two clusters $C_x$ and $C_y$ that have minimal distance among all possible pairs will be merged into a new cluster $C_{h=(x,y)}$ in each round. Let $C_x^{best}$ and $C_y^{best}$ denote the two clusters which satisfy Exp. (16).

$$(C_x^{best}, C_y^{best}) = \underset{C_x, C_y \in g_{i,j}}{arg\ min}\ d\left(C_x, C_y\right) \tag{16}$$

Then clusters $C_x^{best}$ and $C_y^{best}$ will be merged into a larger one, say $C_{h=(x,y)}$. The two clusters will be treated as two nodes in the tree and the merged result will be treated as their parent node in the tree. When the merging procedure is applied round by round, a hierarchical tree $C_{h=(x,y)}$ is also created accordingly.

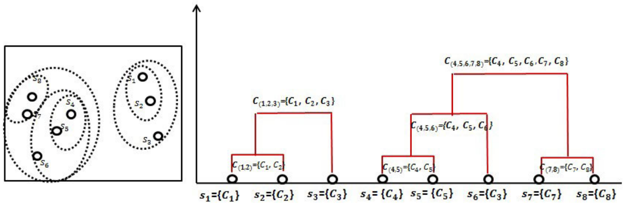


**FIGURE 2.** The merging of clusters and the construction of the hierarchical tree.

When the merging procedure is terminated, the hierarchical tree has been created. The next important step is to decide whether or not the grid $g_{i,j}$ is balanced. Let $\zeta_{trh}$ denote the distance threshold. When the minimal distance between any pair of clusters is larger than $\zeta_{trh}$, the merging procedure is terminated. That is, the merging procedure will be terminated if condition Exp. (17 ) is satisfied.

$$d\left(C_x^{best}, C_y^{best}\right) < \zeta_{trh} \tag{17}$$

Then, a tree corresponding to the last merged pair, denoted by $T_{h=(k,l)}$, will be constructed by merging clusters $C_x^{best} = C_k$ and $C_y^{best} = C_l$. After the merging procedure has been finished, we will check the number of unmerged clusters. In case that all clusters have been merged into one, the grid is balanced. Otherwise, the grid is unbalanced. For instance, if there are two clusters existed in grid $g_{i,j}$, the grid $g_{i,j}$ is unbalanced. This is because the two clusters have a big distance more than $\zeta_{trh}$. This also indicates that there must exist several empty subgrids between the two clusters. Since parameter $\zeta_{trh}$ can impact the identification result of grid $g_{i,j}$ is balanced or unbalanced, its value will be discussed in the simulation section.

Fig. 2 depicts an example of a merging process for those sensors in the grid $g_{i,j}$. As shown in Fig. 2, each sensor $s_j \in S_i$ is initially considered as a single cluster $C_j$ which will be the leaf node of the constructed tree. For example, the sensor node $s_1$ is considered as a cluster $C_1$. In the first round, assume that distance $d\,(C_7, C_8)$ has the smallest value, as compared with all distances of the other pairs. Also, the distance $d\,(C_7, C_8)$ is less than the threshold $\zeta_{trh}$. Therefore, clusters $C_7$ and $C_8$ will be firstly merged into a cluster $C_{h=(7,8)}$, or $C_{(7,8)}$ in short. Continuously, in the second round, clusters $C_4$ and $C_5$ will be merged into a cluster $C_{h=(4,5)}$, or $C_{(4,5)}$, and then the tree $T_{h=(4,5)}$ is constructed. The merging procedure will be repeatedly executing until the distance $d\left(C_{1,2,3}, C_{4,5,6,7,8}\right)$ greater than the threshold $\zeta_{trh}$. Finally, the grid $g_{i,j}$ is marked with unbalanced since there exist two unmerged clusters.

### 2) CALCULATING LENGTH OF EVERY GRID

In the last task, all grids have been identified as balanced or unbalanced grids. This task will allocate the adjustable length of the path to the balanced and unbalanced grids according to the density of sensors in each grid. Let $l^b$ denote the additional path length allocated to each balanced grid and $\eta$ be the total number of balanced grids. The total path length reserved for unbalanced grids is $\mathbb{L}_{adjust} - \eta * l^b$. Let $l_i^{\tilde b}$ denote the additional path length allocated to each unbalanced grid. Let $\tilde G$ denote the set of all unbalanced grids and $\tilde n_{i,j}$ denote the number of sensor nodes in each unbalanced grid $\tilde g_{i,j}$. According to the ratio of sensor density in each grid, the additional path length $l_i^{\tilde b}$ reserved for grid $\tilde g_{i,j}$ can be calculated by

$$l_{i,j}^{\tilde b} = \left(\mathbb{L}_{adjust} - \eta * l^b\right) * \frac{\tilde n_{i,j}}{\sum_{\tilde g_{i,j}\in \tilde G} \tilde n_{i,j}} \tag{18}$$

Thus, the total path length of each unbalanced grid can be calculated by

$$L_i = l_g + l_{i,j}^{\tilde b} \tag{19}$$

The length of path adjustment $l^b$ can be discussed in the simulation section for a balanced grid.

### 3) PATH ADJUSTMENT

This phase is comprised of two tasks, namely Path Adjustment of Unbalanced Grid and Path Adjustment of Balanced Grid. The following will present the details of the two tasks.

***Task 1: Path Adjustment of Unbalanced Grid***

The goal of this task is to select $\tilde k_{i,j}$ anchors from the sensors in the grid $\tilde g_{i,j}$ and construct a path $\tilde \pi_{i,j}$ which visits the $\tilde k_{i,j}$ anchors $\tilde a_1, \ldots, \tilde a_{\tilde k_{i,j}}$ for data collection. The path $\tilde \pi_{i,j}$ starts at the last connection anchor of the last grid and ends at the first connection anchor of the next grid.

Initially, the shortest path tree $\tilde T_{i,j}$ will be constructed in each unbalanced grid $\tilde g_{i,j}$. The goal of this task is to select $\tilde k_{i,j}$ anchors from $\tilde T_{i,j}$ and then reorganize the tree $\tilde T_{i,j}$ as $\tilde k_{i,j}$ subtrees $T_{i,j,q}$, for $1 \leq q \leq \tilde k_{i,j}$. Then the path $\tilde \pi_{i,j}$ can be constructed by passing through all $\tilde k_{i,j}$ anchors $\tilde a_1, \ldots, \tilde a_{\tilde k_{i,j}}$. All nodes rooted by anchor $\tilde a_q$ in each subtree $\tilde T_{i,j,q}$ will forward their sensing information to root $\tilde a_q$. Then root $\tilde a_q$ will forward all sensing data collected from the tree members along with its sensing data to the mobile sink.

Let $\tilde A_{i,j}$ denote the set of anchors selected to play the role of the root in the subtree $\tilde T_{i,j}$. That is,

$$\tilde A_{i,j} = \{\tilde a_q | 1 \leq q \leq \tilde k_{i,j}\}$$

Let $\tilde S_{i,j}$ denote the set of all sensors in $\tilde g_{i,j}$ and $\tilde X_{i,j} = \tilde S_{i,j} \backslash \tilde A_{i,j}$ denote the set of sensors that are not selected to play the role of anchors. Initially, $\tilde A_{i,j} = \emptyset$. This task finds one appropriate sensor node $s^{best}$ from set $\tilde X_{i,j}$ at a time. Then the selected $s^{best}$ will join the set $\tilde A_{i,j}$ and be removed from $\tilde X_{i,j}$. After that, the subtree will be restructured. Each node in $\tilde A_{i,j}$ will play the role of anchor. Each anchor is responsible for collecting data from its tree members and then directly sending them to the mobile sink since the mobile sink will visit each selected anchor in set $\tilde A_{i,j}$. In this way, the energy consumptions for those data forwarding from each anchor to the root of the tree $\tilde T_{i,j}$ can be further saved.

The following illustrates the method which selects the best anchor node $s^{best}$ from set $\tilde{X}_{i,j}$. Let $H(y, \tilde{T}_{i,j,q})$ denote the number of hops from the sensor $s_y$ to the root of its subtree $\tilde{T}_{i,j,q}$. Let $S\left(y, \tilde{T}_{i,j,q}\right)$ denote the number of sensors rooted by $s_y$. Assume that each sensor creates one packet in each round. Let $NP\left(y\right)$ denote the number of packets received by the sensor node $s_y$, including its reading. The value of $NP\left(y\right)$ is be calculated by

$$NP\left(y\right) = S\left(y, \tilde{T}_{i,j,q}\right) + 1 \tag{20}$$

Let $H(y, \tilde{a}_q)$ denote the number of hops from $s_y$ to the root contains sensor node $s_y$. If the sensor $s_y$ is selected as the root of some subtree $\tilde{T}_{i,j,q}$, the number of packets saved by selecting $s_y$ as the anchor is

$$H\left(y, \tilde{a}_q\right) \times NP\left(y\right) \tag{21}$$

Let $dis(\tilde{A}_{i,j}, s_y)$ denote the minimal distance from the sensor node $s_y$ to $\tilde{a}_q \in \tilde{A}_{i,j}$. It is noticed that $\tilde{A}_{i,j} = \emptyset$ when this task initially executes for finding $\tilde{a}_1$. In this case, the $dis(\tilde{A}_{i,j}, s_y)$ will return the distance between $s_y$ and the last visiting anchor in the neighboring grid. Let $\rho_y$ denote the benefit index obtained by selecting a sensor node $s_y$ as the anchor. The value of the benefit index can be calculated by

$$\rho_y = \frac{H(y, \tilde{a}_q) \times NP\left(y\right)}{dist\left(\tilde{A}_{i,j}, s_y\right)} \tag{22}$$

$$s^{best} = \underset{s_y \in \tilde{X}_{i,j}}{arg\ max}\ \rho_y \tag{23}$$

The value of the benefit index $\rho_y$ can be measured by the number of packets saved for transmission, divided by the cost of tour distance from some $s_j \in \tilde{X}_{i,j}$ to $s_y$. Based on Equ. (23), the $s^{best}$ can be obtained. After selecting the best sensor node as the anchor in grid $\tilde{g}_{i,j}$, a new Hamiltonian route will be constructed by adding the new anchor $s^{best}$ to the existing route $\tilde{\pi}_{i,j}$ for the mobile sink, where $\tilde{\pi}_{i,j}$ connects all $(u+1)$ anchors in the set $\tilde{A}_{i,j} = \tilde{A}_{i,j} \cup \{s^{best}\}$. After that, the length of new route will be checked whether or not it is smaller than the length upper bound $L_i = l_g + l^b_{i,j}$.

If it is the case, the proposed algorithm will add the anchor point $s^{best}$ to set $\tilde{A}_{i,j}$ and remove the $s^{best}$ from set $\tilde{X}_{i,j}$. The edge that connects $s^{best}$ and its parent will be removed accordingly. Otherwise, the selection operation will be terminated. Following this procedure, the set of anchors $\tilde{A}_{i,j}$= $\{\tilde{a}_1, \tilde{a}_2, \tilde{a}_3, \ldots \tilde{a}_{\tilde{k}_{i,j}}\}$ would be automatically obtained.

The following gives an example of executing the path adjustment of unbalanced grid task. Fig. 3(a) gives an unbalanced grid $\tilde{g}_{i,j}$. The path adjustment of unbalanced grid task constructs a shortest-path tree as shown in Fig 3(b). Assume the bond length of traversal is $L_i = 100m$. Initially, the path adjustment of unbalanced grid task selects $\tilde{a}_1 = s_{15}$ as the first anchor since it is closest to the anchor in the neighboring grid. Then the task estimates $\rho_y$ of each sensor node $s_y$ in tree $SPT$ and then selects $\tilde{a}_2 = s_{14}$ which is the best sensor node as shown in Fig. 3(d). Because that the length $L_{\tilde{A}_{i,j}}$ of the tour set $\tilde{A}_{i,j} = \{s_{15}, s_{14}\}$ is still smaller than $L_i = 100m$, the path adjustment of unbalanced grid task then adds $s_{14}$ into anchor set $\tilde{A}_{i,j}$ and removes $s_{14}$ and edge $(s_{15}, s_{14})$ from the shortest-path tree $SPT$ as shown in Fig. 3(d). The red line shown in Fig. 3(d) denotes the route that connects all selected anchors. The $SPT$ trees will be reconstructed by considering $s_{15}$ and $s_{14}$ as the tree roots. The repetitions of the execution of the path adjustment of unbalanced grid task will continuously select one best sensor node to play the anchor role until the tour length is larger than the upper bound length $L_i$. Finally, the path adjustment of unbalanced grid task selects $s_{14}$, $s_6$ and $s_{19}$ to serve as anchors, as shown in Figs. 3(d), 3(e) and 3(f), respectively. Thus, the final tour path of $\tilde{A}_{i,j} = \{\tilde{a}_1 = s_{15}, \tilde{a}_2 = s_{14}, \tilde{a}_3 = s_6, \tilde{a}_4 = s_{19}\}$.

**Algorithm** ***Path Adjustment of Unbalanced Grid Task***

***Input:*** $L_i$, $\tilde{S}_{i,j}$=$\{\tilde{s}_{i,1}, \tilde{s}_{i,2}, \tilde{s}_{i,3}, \ldots \tilde{s}_{i,n_i}\}$, $\tilde{X}_{i,j} = \tilde{S}_{i,j} \backslash \tilde{A}_{i,j}$, $\tilde{A}_{i,j} = \emptyset$, $L_{\tilde{A}_{i,j}} = 0$;

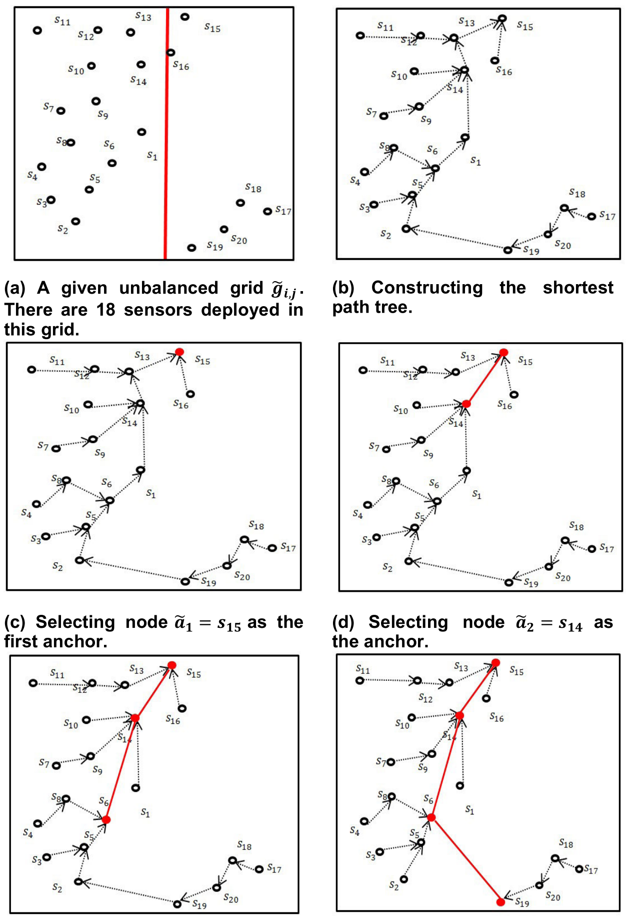
***Output:*** The adjusted path $\tilde{\pi}_{i,j}$ for the unbalanced grid $\tilde{g}_{i,j}$

```
Construct the shortest path tree;
$\tilde{A}_{i,j} = \{a_0 = root\}$;
While $(L_i \geq L_{\tilde{A}_{i,j}})${
    Evaluate $NP\ (y)$ according to Equ. (20), for each $s_y \in \tilde{S}_{i,j}$;
    Evaluate $H\left(y, \tilde{a}_q\right) \times NP\ (y)$, for each $s_y \in \tilde{S}_{i,j}$;
   Evaluate $\rho_y$ according to Equ.(22);
   $s^{best} = arg\ max_{s_y \in \tilde{X}_{i,j}} \rho_y$;
   If( $L_i \geq L_{\tilde{A}_{i,j}}$) {
      $\tilde{A}_{i,j} = \tilde{A}_{i,j} \cup \left\{s^{best}\right\}$;
      $\tilde{X}_{i,j} = \tilde{X}_{i,j} - \left\{s^{best}\right\}$;
      Reconstruct $SPT\ \tilde{T}_{i,j}$;
      Adding the new anchor $s^{best}$ to $\tilde{\pi}_{i,j}$}
    else
      Exit; }
For $(i = 0, i \leq k, i++)${
Find the nearest anchor $\tilde{a}_q$ connected the last grid $\tilde{g}_{i,j-1}$;}
Return $\tilde{\pi}_{i,j}$;
```

***Task 2: Path Adjustment of Balanced Grid***

In the last task, the path has been constructed for each unbalanced grid. This task aims to slightly adjust the path for each balanced grid. Assume that grid $g_{x,y}$ is a balanced grid and has been divided into a set of $f \times d$ small grids $\Phi = \{\eta_{1,1}, \eta_{1,2}, \ldots, \eta_{d,f}\}$. Let $\Re$= $\{R_1, \ldots, R_d\}$ where $R_i$ consists of the small grids in the $i$-th row. That is, $R_i = \{\eta_{i,1}, \eta_{i,2}, \ldots, \eta_{i,f}\}$.

Let $n_{i,j}$ denote the number of sensors in a small grid $\eta_{i,j}$. The following presents the path adjustment operations for the balanced grid $g_{x,y}$. The adjustment operations are executed for one row in each round. Initially, the first row $R_1$ is considered. The basic idea behind this task is to find the best small grid, denoted by $\eta_i^{best}$, to be passed by the data collection path in each row $R_i$. More specifically, sensors in $\eta_i^{best}$ should play



(a) A given unbalanced grid $\tilde{g}_{i,j}$. There are 18 sensors deployed in this grid.

(b) Constructing the shortest path tree.

(c) Selecting node $\tilde{a}_1 = s_{15}$ as the first anchor.

(d) Selecting node $\tilde{a}_2 = s_{14}$ as the anchor.

(e) Selecting $\tilde{a}_3 = s_6$ as the third anchor.

(f) Selecting node $\tilde{a}_4 = s_{19}$ as the fourth anchor.

**FIGURE 3. Example of path adjustment for the unbalanced grid.**

the role of anchors such that all sensors in the other grids of row $R_i$ can forward data to the sensors in $\eta_i^{best}$. To balance the energy consumption of each anchor, the $\eta_i^{best}$ should satisfy the following Exp. (24).

$$\eta_i^{best} = \min_{\eta_{i,j}\in R_i, 1\le j\le f} \frac{\sum_{\eta_{a,b}\in(R_i\setminus\eta_{i,j})} n_{a,b}}{n_{i,j}} \tag{24}$$

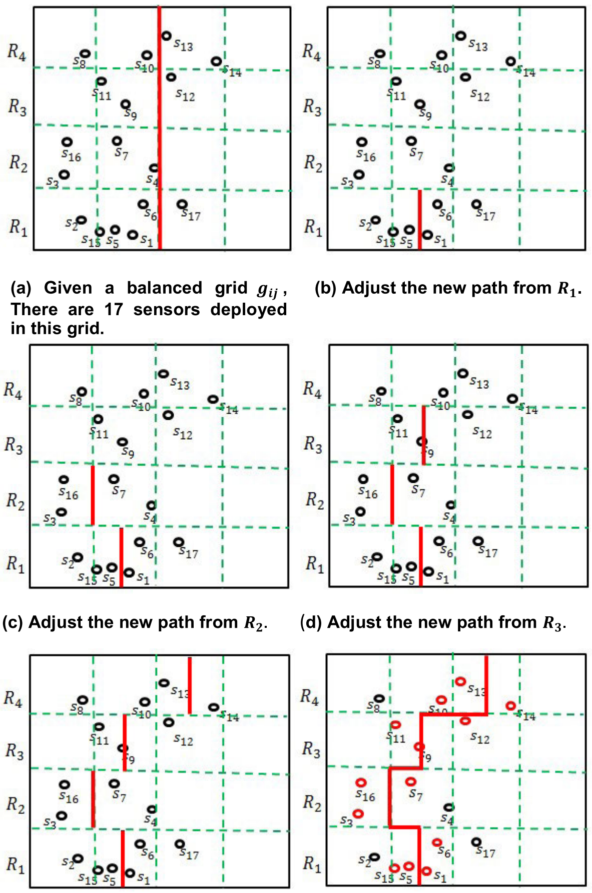
Let $A^{best}$ denote the set of all best grids $\eta_i^{best}$ in rows $R_i$, $1 \le i \le d$. That is

$$A^{best} = \left\{\eta_i^{best} \mid 1 \le i \le d\right\} \tag{25}$$

The path passing through the grid $g_{x,y}$ will be constructed by connecting all best grids $\eta_i^{best} \in A^{best}$. Let $p_{x,y}$ be the path segment passes though $g_{x,y}$. Let $\varsigma_i$ denote the vertical centerline of the grid $\eta_i^{best}$. The path $p_{x,y}$ can be constructed by operation:

$$p_{x,y} = \oplus_{1\le i\le d}(\eta_i^{best}) \tag{26}$$

where $\oplus$ denotes the connection operation performed on all segments $\eta_i^{best}$, for $1 \le i \le d$.



(a) Given a balanced grid $g_{ij}$, There are 17 sensors deployed in this grid.

(b) Adjust the new path from $R_1$.

(c) Adjust the new path from $R_2$.

(d) Adjust the new path from $R_3$.

(e) Adjust the new path from $R_4$.

(f) Connect the path for a balanced grid $g_{ij}$.

**FIGURE 4. Example of path adjustment in a balanced grid.**

In summary, the mobile sink applies the path adjustment of unbalanced grid task to construct the path passing through unbalanced grids but applies the path adjustment of balanced grid task to construct the path passing through balanced grids.

The following gives an example for illustrating the path adjustment of balanced grid task. Fig. 4(a) gives the initial path of a balanced grid $g_{i,j}$. The balanced grid $g_{i,j}$ is divided into sixteen small grids $\{\eta_1, \eta_2 \ldots, \eta_{16}\}$. We have $\Re = \{R_1, R_2, R_3, R_4\}$. The path adjustment of balanced grid task identifies the best small grid $\eta_1^{best} = \eta_2$ for $R_1$ as shown in Fig. 4(b). The repetitions of executing the path adjustment of balanced grid task will continuously identify the best small grids $\eta_2^{best} = \eta_5$, $\eta_3^{best} = \eta_{10}$ and $\eta_4^{best} = \eta_{15}$ as shown in Figs. 4(c), 4(d) and 4(e), respectively. Therefore, we have $A^{best} = \{\eta_2, \eta_5, \eta_{10,}, \eta_{15,}\}$. Finally, the path adjustment of balanced grid task constructs the new path as shown in Fig.4(f) where sensors $s_1, s_5, s_{15}, s_6, s_3, s_{16}, s_7, s_9, s_{11}, s_{10}, s_{12}, s_{13}$ and $s_{14}$ serve as the anchor nodes.

## V. PERFORMANCE EVALUATIONS

This section investigates the performance improvements of the proposed algorithm *DEDC* against the existing algorithms EAPC [18] and WRP [17], in terms of the energy

**TABLE 3. Simulation parameter settings.**

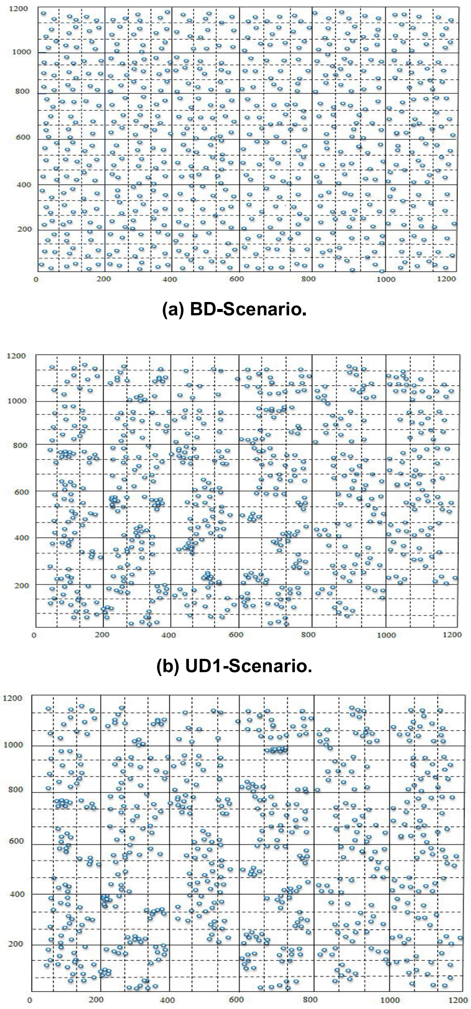
| Parameters | Values |
|---|---|
| Node deployment | Random |
| Region size | 1200*1200 |
| The number of sensor nodes | 200-700 |
| The sensing range of each sensor node | 20 m |
| Mobile sink speed | 3 m/s |
| Sensor node transmission range | 30m-40m |
| Consumed energy in the transmitter circuit | 0.18J |
| Consumed energy in the receiver circuit | 0.1J |
| The initial energy of each sensor node | 120J |

consumption, network lifetime, and *SD* of energy consumptions. In WRP, each sensor was assigned a weight, which was calculated by multiplying the number of packets that it forwarded by its hop distance to the closest anchor. The highest weighted sensor node was treated as an anchor. Then a tour was constructed by starting from the base station, passing through all anchor nodes and finally returning to the base station. Since the tour length was limited, the visited path length must be equal to or less than the maximum tour length.

Similar to the WRP, the EAPC selected anchors according to the weight assigned to each sensor. One main difference between EAPC and WRP was that EAPC considered the path length from the current anchor to the next anchor during the anchor selection process. Then, the traveling path of the mobile sink can be constructed using convex polygons. The MATLAB simulator is used as the simulation tool. Three scenarios are considered in the experiments.

The parameter values set in the simulation environment are illustrated. A set of sensors, ranging from 200 to 700 nodes, are randomly deployed in a given region with the size 1200 m *1200 m. The initial energy of each sensor node is 120 J. The sensing range of each sensor node is set at 20 m, while the communication range of each sensor varies ranging from 30m to 40m. Assume that each sensor knows the information of the mobile sink, including the movement trajectory and arrival time of the mobile sink. The parameter values of the experiments are summarized in Table 3. Each sensor node generates one data packet in each round. During each round, the mobile sink starts from the base station, collects data in each grid and then goes back to the base station.

To further investigate the performance of the proposed algorithms, three scenarios are considered in the experiments. In the first scenario, called the balanced deployment scenario (BD-Scenario), sensors are randomly deployed over every square grid. The other two scenarios are uneven deployment scenarios, called UD1-Scenario and UD2-Scenario. In the UD1-Scenario, the numbers of big, middle and small holes are almost identical. In the UD2-Scenario, the ratio of the numbers of big, middle and small holes is 1:2:5. Fig. 5 shows examples of the three scenarios.



(a) BD-Scenario.

(b) UD1-Scenario.

(c) UD2-Scenario.

**FIGURE 5. Three scenarios considered in the experiments.**

As shown in Fig. 5, the given region is initially partitioned into 36 square grids. All sensor nodes can communicate with each other in each square grid. Fig. 5(a) depicts a deployment snapshot of 600 sensor nodes in the BD-Scenario where all sensors are randomly deployed in each grid. On the contrary, in the UD1-Scenario as shown in Fig. 5(b), there are few sensors or even no sensors deployed in some grids. Fig. 5(c) depicts a snapshot of the UD2-Scenario.

Figs. 6 and 7 show the constructed data collection path by applying DEDC in the UD1-scenario where the path lengths are 9000m and 5500m, respectively. As shown in Fig. 6, a large number of anchor nodes marked with red color are selected. These anchors help forward data from other sensor nodes to the mobile sink, balancing the forwarding loads of the anchor nodes. This occurs because the path length of the mobile sink in the UD1-scenario is long enough. However, as shown in Fig. 7, when the path length is constrained with 5500m, there is no extra path length to be adjusted. As a result, fewer sensor nodes are selected as the anchors which are marked with red color, as compared with Fig. 6.

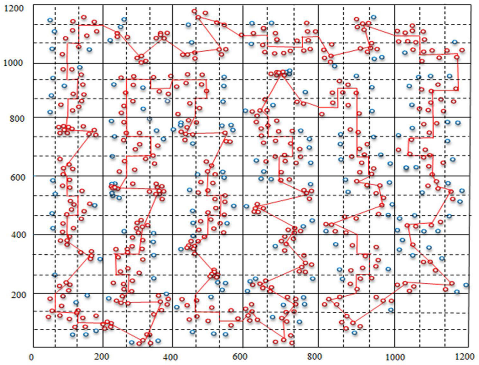


**FIGURE 6.** The result by applying DEDC using UD1-scenario. The path length is constrained by 9000m.

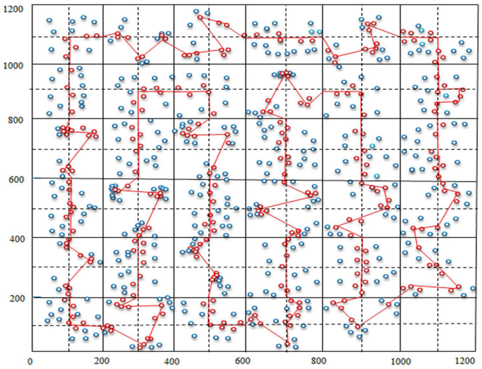
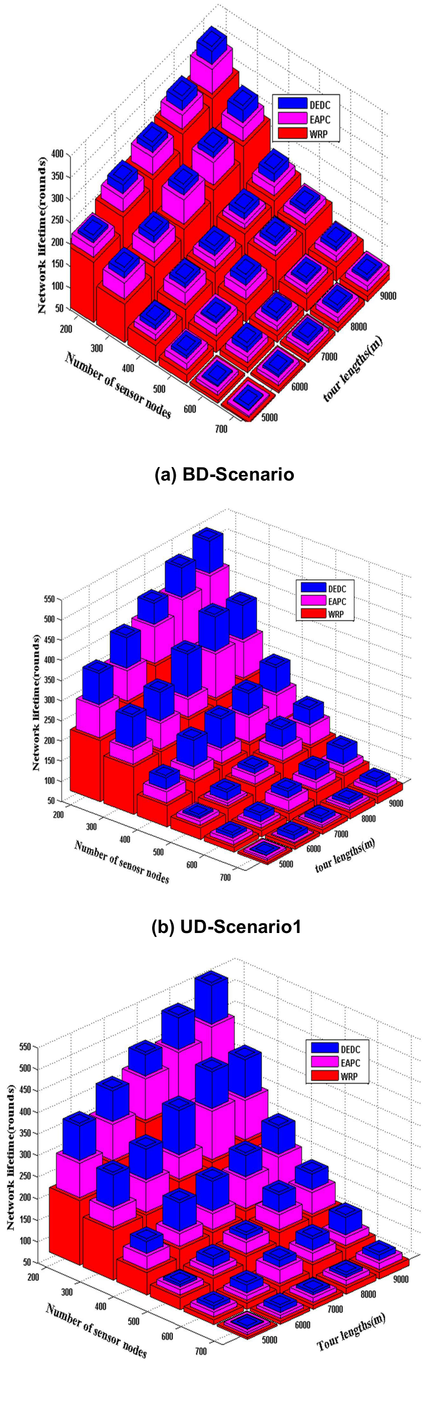


**FIGURE 7.** The result by applying DEDC using UD1-scenario. The path length is constrained by 5500m.

Fig. 8 compares the network lifetimes of the three algorithms by varying the number of sensor nodes ranging from 200 to 700 and tour lengths ranging from 5000m to 9000m for three scenarios. The network lifetime is measured from the starting time point of network operations to the time point that the first sensor runs out its energy. The three algorithms have a similar trend that the network lifetime increases with the tour length. This occurs because the longer path allows more sensors to be selected as anchors. These anchors help forward data from other sensors to the mobile sink, balancing the forwarding loads of anchors. The WRP selects anchors and constructs a tour that passes through all anchors. However, it does not take into account the path length from the current anchor to the next anchor. Therefore, the path is not efficient, as compared with the other two algorithms.

The proposed DEDC outperforms existing EAPC in terms of network lifetime. This occurs because DEDC constructs a path passing through every grid regularly. This helps find more anchors and balance the forwarding workloads of anchors in case the sensor deployment is balanced distribution. To handle the unbalanced distribution, DEDC further remains a certain path length, selects the best set of the anchor node $s^{best}$ and constructs the path passing through unbalanced grids. Meanwhile, the mobile sink applies the DEDC algorithm can dynamically adjust the regular path. As a result, the proposed DEDC achieves higher resource utilization of the limited path length and finds more anchors, as compared with the other two mechanisms, as shown in Fig. 8.



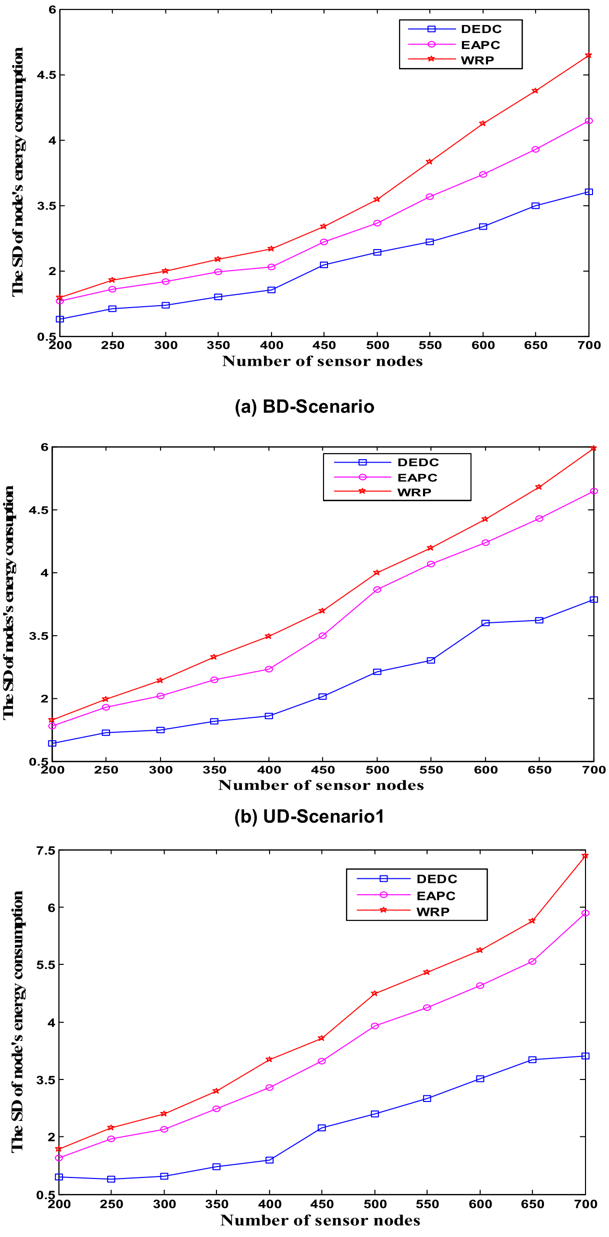
(a) BD-Scenario

(b) UD-Scenario1

(c) UD-Scenario2

**FIGURE 8.** The comparisons of network lifetimes of three algorithms.

Fig. 9 further compares the three algorithms in terms of standard deviation (SD) which is used to measure the balance degree of energy consumption of each sensor node. The *SD*

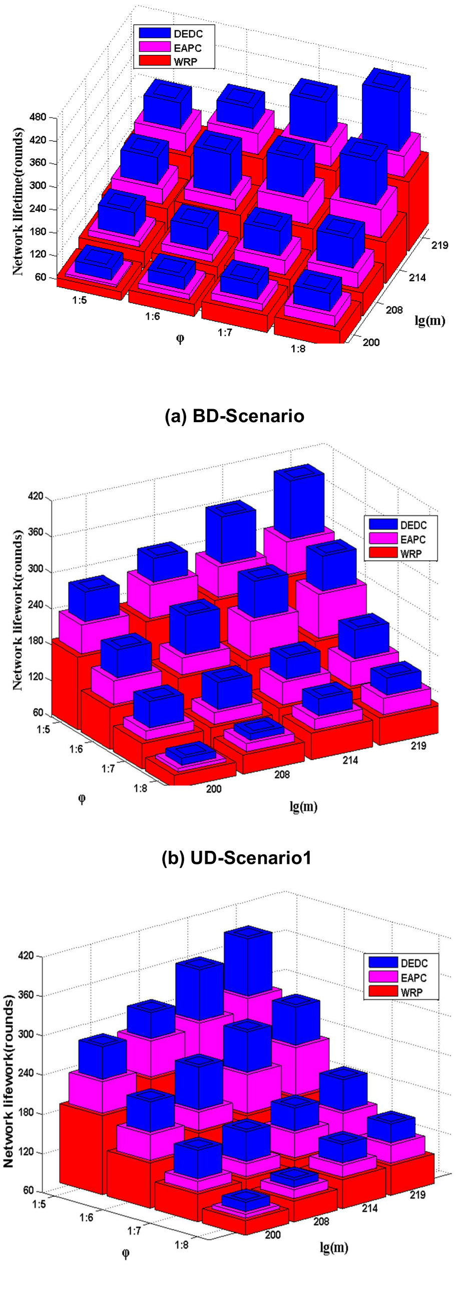


(a) BD-Scenario

(b) UD-Scenario1

(c) UD-Scenario2

**FIGURE 9. The comparisons of *SD* energy consumptions of three algorithms.**



(a) BD-Scenario

(b) UD-Scenario1

(c) UD-Scenario2

**FIGURE 10. The comparisons of the network lifetimes of three algorithms.**

is defined by Exp. (27).

$$SD = \sqrt{\frac{\sum_{1}^{n} (E_i - \mu)^2}{n}} \tag{27}$$

where $E_i$ denotes energy consumption of node $s_i$ and $\mu$ denotes the average energy consumption of all nodes. A small *SD* value indicates that the energy consumptions of all nodes are balanced, which helps prolong the network lifetime of the WSN. In this experiment, the number of sensor nodes varies ranging from 200 to 700. In general, three algorithms have a common trend that the *SD* value is decreased with the number of sensor nodes. In comparison, WRP yields the largest *SD* value because it finds the least number of anchors. The EAPC has a better performance than WRP because it considers the path length from the current anchor to the next anchor. The proposed DEDC considers balanced and unbalanced grids and adopts different path construction policies to efficiently utilize the limited path length. Consequently, the proposed DEDC mechanism finds more anchors and achieves the lowest *SD* value, as shown in Fig. 9.

Fig. 10 compares the network lifetimes of the three algorithms in three scenarios. Two parameters, including scale parameter $\varphi$ and the length of the grid, are varied. Assume

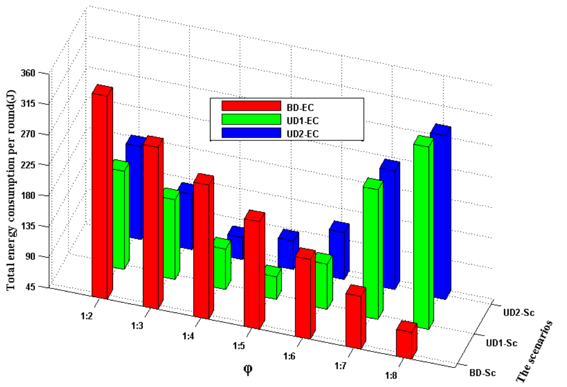


**FIGURE 11.** The comparisons of energy consumptions of three scenarios.

the path length is 9000m and the number of grid is 36. The length $l_g$ is calculated by $\frac{(9000-9000\times\varphi)}{36}$. The length $l_g$ of grid varies at the values 200m, 208m, 214m and 219m. Three algorithms have a common trend that the network lifetime is increased with the scale parameter $\varphi$ in the BD-Scenario but is decreased with the scale parameter $\varphi$ in UD-scenario1 and UD-scenario2. The main reason is that BD-scenario has a balanced deployment of sensors without containing hole which increases the movement overheads to the mobile sink. In UD-scenario1 and UD-scenario2, the unbalanced deployment results in many holes in grids. The mobile sink inefficiently consumes much energy to traverse the hole grids but can only collect few data. In comparison, the proposed DEDC outperforms the other two compared algorithms. The WRP does not take into account the path length from the current anchor to the next anchor, leading to large movement overhead of the mobile sink. Hence fewer anchors are selected due to the limited path length. The proposed DEDC outperforms existing EAPC in terms of network lifetime. This occurs because DEDC adjusts the initially constructed path length and allocates the path length to each grid efficiently.

Fig. 11 shows the energy consumption of three scenarios by varying scale parameter $\varphi$, ranging from 1:2 to 1:8. In BD-scenario, the energy consumption is decreased with the value of $\varphi$. This occurs because most grids are regularly deployed in BD-scenario. Hence the regular path is required to pass through all grids for finding more anchors. This also indicates that there is no need for an irregular path since there are almost no unbalanced grids. On the contrary, in UD-scenario1 and UD-scenario2, the energy consumptions are minimal when the $\varphi$ are 1:5 and 1:4, respectively. In UD-scenario, many grids are unbalanced. Therefore, a reservation for irregular path length is required since the regular path can only benefit to find more anchors in the regular grids. Since there are many unbalanced grids, it is required to reserve a certain ratio of path length, aiming to find appropriate anchors in the unbalanced grids.

## VI. CONCLUSIONS

Data collection is one of the most important issues in WSNs. The mobile sink can help reduce the energy consumption of sensor nodes because it visits some sensor nodes and collect data from them while guaranteeing the collected data to be fresh. This study proposed a data collection mechanism, called DEDC, aiming at prolonging the network lifetime. The proposed DEDC comprises Initial Path Construction and Path Adjustment Phases. The Initial Path Construction partitions the monitoring region into several equal-sized grids and constructs a regular path, aiming to pass through a maximal number of anchors for the balanced grids. The Path Adjustment Phase further identifies the unbalanced grids, finds appropriate anchors and finally adjusts the regular path for the unbalanced grids. Extensive performance evaluations show that the proposed DEDC outperforms existing schemes in terms of energy consumptions, network lifetime and SD of energy consumptions in both balance and unbalance deployment scenarios.

## REFERENCES

[1] R. Wang, F. Chang, and S. Ren, "TPLE: A reliable data delivery scheme for on-road WSN traffic monitoring," *Sensors*, vol. 17, no. 1, pp. 44–56, Dec. 2016.

[2] C.-Y. Chang, Y.-W. Kuo, P. Xu, and H. Chen, "Monitoring quality guaranteed barrier coverage mechanism for traffic counting in wireless sensor networks," *IEEE Access*, vol. 6, pp. 30778–30792, May 2018.

[3] C.-C. Kao, Y.-S. Lin, G.-D. Wu, and C.-J. Huang, "A comprehensive study on the Internet of underwater things: Applications, challenges, and channel models," *Sensors*, vol. 17, no. 7, pp. 1477–1497, Jun. 2017.

[4] N. K. Suryadevara and S. C. Mukhopadhyay, "Wireless sensor network based home monitoring system for wellness determination of elderly," *IEEE Sensors J.*, vol. 12, no. 6, pp. 1965–1972, Jun. 2012.

[5] N. Dey, A. S. Ashour, F. Shi, S. J. Fong, and R. S. Sherratt, "Developing residential wireless sensor networks for ECG healthcare monitoring," *IEEE Trans. Consum. Electron.*, vol. 63, no. 4, pp. 442–449, Nov. 2017.

[6] C. Shang, G. Chen, C. Ji, and C.-Y. Chang, "An efficient target tracking mechanism for guaranteeing user-defined tracking quality in WSNs," *IEEE Sensors J.*, vol. 15, no. 9, pp. 5258–5271, Sep. 2015.

[7] C.-Y. Chang, C.-C. Lin, C. Shang, I.-H. Chang, and D. S. Roy, "DBDC: A distributed bus-based data collection mechanism for maximizing throughput and lifetime in WSNs," *IEEE Access*, vol. 7, pp. 160506–160522, Oct. 2019.

[8] X. Liu, A. Liu, T. Wang, K. Ota, M. Dong, Y. Liu, and Z. Cai, "Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks," *J. Parallel Distrib. Comput.*, vol. 135, pp. 140–155, Jan. 2020.

[9] Y. Liu, Z. Zeng, X. Liu, X. Zhu, and M. Z. A. Bhuiyan, "A novel load balancing and low response delay framework for edge-cloud network based on SDN," *IEEE Internet Things J.*, early access, Nov. 6, 2019, doi: 10.1109/JIOT.2019.2951857.

[10] R. C. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: Modeling and analysis of a three-tier architecture for sparse sensor networks," *Ad Hoc Netw.*, vol. 1, nos. 2–3, pp. 215–233, Sep. 2003.

[11] R. Sugihara and R. K. Gupta, "Optimal speed control of mobile node for data collection in sensor networks," *IEEE Trans. Mobile Comput.*, vol. 9, no. 1, pp. 127–139, Jan. 2010.

[12] A. A. Somasundara, A. Ramamoorthy, and M. B. Srivastava, "Mobile element scheduling with dynamic deadlines," *IEEE Trans. Mobile Comput.*, vol. 6, no. 4, pp. 395–410, Apr. 2007.

[13] E. G. Prathima, H. Laxmikant, S. A. Naveen, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "DAMS: Data aggregation using mobile sink in wireless sensor networks," in *Proc. 5th Int. Conf. Commun. Broadband Netw. (ICCBN)*, Bali, Indonesia, 2017, pp. 6–11.

[14] C. Tunca, S. Isik, M. Y. Donmez, and C. Ersoy, "Ring routing: An energy-efficient routing protocol for wireless sensor networks with a mobile sink," *IEEE Trans. Mobile Comput.*, vol. 14, no. 9, pp. 1947–1960, Sep. 2015.

[15] C. Zhu, S. Wu, G. Han, L. Shu, and H. Wu, "A tree-cluster-based data-gathering algorithm for industrial WSNs with a mobile sink," *IEEE Access*, vol. 3, pp. 381–396, 2015.

[16] Y.-H. Yang, T. Lin, B.-H. Liu, S.-I. Chu, C.-Y. Lien, and V.-T. Pham, "An efficient mobile sink scheduling method for data collection in wireless sensor networks," in *Proc. Int. Conf. Syst. Sci. Eng. (ICSSE)*, Ho Chi Minh City, Vietnam, Jul. 2017, pp. 554–557.

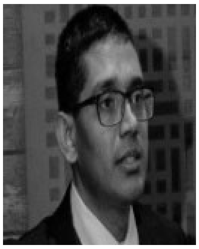
[17] H. Salarian, K.-W. Chin, and F. Naghdy, "An energy-efficient mobile-sink path selection strategy for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 63, no. 5, pp. 2407–2419, Jun. 2014.

[18] W. Wen, S. Zhao, C. Shang, and C.-Y. Chang, "EAPC: Energy-aware path construction for data collection using mobile sink in wireless sensor networks," *IEEE Sensors J.*, vol. 18, no. 2, pp. 890–901, Jan. 2018.

**WEIMIN WEN** received the M.S. degree from the Xi'an University of Science and Technology, in 2008. She is currently a Senior Laboratory Teacher with the School of Computer and Information Engineering, University of Chuzhou, Anhui, China. Her current research interests include wireless sensor networks and healthcare.

**CUIJUAN SHANG** received the B.S. degree in biomedical engineering from Zhengzhou University, Zhengzhou, China, in 2008, and the M.S. degree in biomedical engineering from Shandong University, Jinan, China, in 2011. She is currently pursuing the Ph.D. degree with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei, Taiwan. She is currently a Lecturer with the School of Computer and Information Engineering, Chuzhou University, Chuzhou, Anhui, China. Her current research interests include wireless sensor networks and healthcare.

**CHIH-YUNG CHANG** (Member, IEEE) received the Ph.D. degree in computer science and information engineering from National Central University, Taiwan, in 1995. He is currently a Full Professor with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei, Taiwan. His current research interests include the Internet of Things, wireless sensor networks, artificial intelligence, and deep learning. He has served as an Associate Guest Editor for several SCI-indexed journals, including the *International Journal of Ad Hoc and Ubiquitous Computing* (IJAHUC), from 2011 to 2019, the *International Journal of Distributed Sensor Networks* (IJDSN), from 2012 to 2014, *IET Communications*, in 2011, *Telecommunication Systems* (TS), in 2010, the *Journal of Information Science and Engineering* (JISE), in 2008, and the *Journal of Internet Technology* (JIT), from 2004 to 2008.



**DIPTENDU SINHA ROY** received the Ph.D. degree in engineering from the Birla Institute of Technology, Mesra, India, in 2010. In 2016, he joined the Department of Computer Science and Engineering, National Institute of Technology (NIT) Meghalaya, India, as an Associate Professor, where he has been serving as the Chair of the Department of Computer Science and Engineering, since January 2017. Prior to his stint at NIT Meghalaya, he had served the Department of Computer Science and Engineering, National Institute of Science and Technology, Berhampur, India. His current research interests include software reliability, distributed and cloud computing, and the IoT, specifically the applications of artificial intelligence and machine learning for smart integrated systems. He is a member of the IEEE Computer Society.