Research Article

Iterative Sensor Clustering and Mobile Sink Trajectory Optimization for Wireless Sensor Network with Nonuniform Density

Joohan Park , Soohyeong Kim , Jiseung Youn , Seyoung Ahn , and Sunghyun Cho

Department of Computer Science and Engineering, Hanyang University, Republic of Korea

Correspondence should be addressed to Sunghyun Cho; chopro@hanyang.ac.kr

Received 3 July 2020; Revised 20 July 2020; Accepted 8 October 2020; Published 20 October 2020

Academic Editor: Ki-Il Kim

Copyright © 2020 Joohan Park et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Sensor clustering and trajectory optimization are a hot topic for last decade to improve energy efficiency of wireless sensor network (WSN). Most of existing studies assume that the sensor is uniformly deployed or all regions in the WSN coverage have the same level of interest. However, even in the same WSN, areas with high probability of disaster will have to form a “hotspot” with more sensors densely placed in order to be sensitive to environmental changes. The energy hole can be serious if sensor clustering and trajectory optimization are formulated without considering the hotspot. Therefore, we need to devise a sensor clustering and trajectory optimization algorithm considering the hotspots of WSN. In this paper, we propose an iterative algorithm to minimize the amount of energy consumed by components of WSN named ISCTO. The ISCTO algorithm consists of two phases. The first phase is a sensor clustering phase used to find the suitable number of clusters and cluster headers by considering the density of sensor and residual battery of sensors. The second phase is a trajectory optimization phase used to formulate suitable trajectory of multiple mobile sinks to minimize the amount of energy consumed by mobile sinks. The ISCTO algorithm performs two phases repeatedly until the amount of energy consumed by the WSN is not reduced. In addition, we show the performance of the proposed algorithm in terms of the total amount of energy consumed by sensors and mobile sinks.

1. Introduction

With the advent of the Internet of Things (IoT), the wireless sensor network (WSN) has attracted much attention as a key enabler of IoT technology. The WSN is used in a wide range of fields such as home IoT [1], smart agriculture [2], and smart city [3]. The WSN is a network consisting of a large number of battery-powered sensors deployed in a large area. To maximize the performance of the WSN, two challenges must be considered. One is the lifetime of the WSN. Since the sensor is powered by a battery, the lifetime of the sensor cannot last indefinitely. Therefore, in order to increase the lifetime of the sensor, numerous approaches such as energy-efficient routing path construction [4], duty-cycling management [5], and radio frequency energy harvesting [6] have been conducted [7]. The other problem is a coverage hole. The WSN is constructed in a very large Area of Interest (AoI) such as battlefields, radioactive power plants, and

farms [8]. It is very important to place the sensor so that the coverage of the WSN is not empty in the AoI. The WSN needs to monitor the changes in the environment such as temperature, humidity, air pressure, and radiation occurring in the AoI. If there is a coverage hole in the AoI, there is a possibility that the change in the environment may not be correctly recognized. WSNs with important targets that should not be missed, such as severe radioactive leaks, can be more damaging if environmental changes are not properly monitored. Therefore, many researches have been conducted on sensor deployment to eliminate the coverage hole within a very large AoI [9].

However, it may not be appropriate to consider the entire coverage of a very wide WSN with the same level of interest. Where there is a relatively high probability of a disaster, such as fires, or radioactive spills, the area must be sensitive to environmental changes compared to those that are not. This means that sensors in areas with a high probability of disaster

have to sense more frequently than other sensors. Sensors that sense frequently consume energy faster than other sensors. This causes an imbalance in energy consumption between sensors. More sensors need to be deployed to increase density to balance energy consumption between sensors across all coverage and thoroughly investigate environmental changes in the AoI [10].

However, an energy hole issue may occur where the sensor density is high [11]. Sensors transmit data to the sink through relaying between sensors to process the generated data [12]. Due to the geographical characteristics, a node close to the sink frequently communicates with a lot of data. This causes the sensor closer to the sink to discharge the battery faster than the far sensor, and this problem is called an energy hole issue. If the sensor density is high, the sensors located geographically close to each other generate a lot of data. A lot of data is relayed using a common intermediate sensor to reach the sink. As a result, the intermediate sensor which was in charge of relaying high-density sensors had to discharge the battery quickly.

The energy hole can be relaxed by using clustering and mobile sink [13]. It is possible to mitigate energy holes by sending data directly to the near cluster header by creating a large number of fine clusters. In addition, the energy hole can be mitigated by moving the sink near the cluster header to directly receive the collected data.

However, most sensor cluster and mobile sink trajectory planning studies do not take into account the possibility of “hotspot” in which sensors are densely located in areas of high interest. As mentioned above, a large number of fine clusters must be created in order to mitigate the energy hole in the hotspot. If the cluster header is selected without considering the density of each sensor, there is a probability that a sufficient number of fine clusters are not generated in the hotspot. If only a small number of clusters are formed in the hotspot, the number of member nodes responsible for the cluster header increases. This increases the energy consumption generated when the cluster header receives data sent by the member node and increases the probability that an energy hole will occur. Likewise, forming a trajectory of mobile sinks without considering hotspot can cause an imbalance in energy consumption between mobile sinks. The mobile sink in charge of the hotspot needs to visit a larger number of cluster headers than other mobile sinks, and energy consumption is also increased. Therefore, it is necessary to devise a method for forming clusters and trajectories of mobile sinks according to the hotspot.

Therefore, this paper proposes iterative sensor clustering and mobile sink trajectory optimization (ISCTO) method considering the hotspot of the WSN. The algorithm uses a density function using a k -nearest neighbor which can measure sensor density. We formulate the total energy consumption function of the components of the WSN. The energy consumption function is selected as an objective function of the optimization problem. To solve this problem, this algorithm separates the original optimization problem into clustering and trajectory optimization problems. In the clustering problem, the number of clusters and the cluster header and member nodes is configured to decrease the amount of

energy consumed by the sensors. In the trajectory optimization, the locations of the cluster headers presented as the output in the clustering step are used to form a trajectory that minimizes the amount of energy consumed by all mobile sinks. The algorithm is repeated until energy efficiency no longer increases.

The main contributions of this paper are as follows:

- (i) We propose a clustering method to increase the energy efficiency of the sensor considering the hotspot of the WSN. The clustering proceeds to select suitable cluster headers considering the density and residual battery of sensors
- (ii) We propose a trajectory optimization algorithm to minimize the amount of energy consumed by all mobile sinks. In the proposed trajectory optimization method, we propose a sector boundary control algorithm to minimize the amount of energy consumed by all mobile sinks by decreasing the total trajectory length of mobile sinks
- (iii) We present the performance of the proposed method through simulation. We show visually the process of sensor clustering and trajectory change. We also analyze the running time of our algorithm

The contents of this paper are as follows. In Section 2, we introduce related studies. We present the system model in Section 3. We also present the mathematical problem to be solved in Section 3. In Section 4, we introduce the proposed ISCTO algorithm to maximize the energy efficiency of the WSN. We show the performance of our proposed algorithm by simulation in Section 5. We conclude this paper in Section 6.

2. Related Works

In this section, we introduce some studies related to energy hole, sensor clustering, and mobile sink trajectory formulation in the WSN.

2.1. Energy Hole. There are numerous studies to mitigate energy hole issues in the WSN [14]. The authors of [15] propose an algorithm to balance the consumed energy among the whole network to avoid the energy hole issue. To balance a load distribution and energy consumption, the algorithm finds an optimal communication distance and uses a transmission strategy based on the amount of consumed energy. In [16], a transmission distance adjustment algorithm is proposed to minimize and balance the energy consumption of the whole sensors with a static sink. A short-trip forwarding strategy of an ant colony optimization and reference transmission distance is adopted to minimize and balance the energy consumption. The authors of [17] propose a routing path reconstructing algorithm when a dead node occurs in a network. In this paper, two protocols are introduced named on-hole children reconnection and on-hole alert. The authors of [18] analyze theoretically the lifetime of a WSN from initialization to the entire dead by estimating

the energy consumption and traffic load of the sensors. In addition, the authors analyze the boundary of energy hole issues that happened. The authors of [19] propose a deployment model of a WSN consisting of heterogeneous and homogeneous sensors. The authors also give some guide to deploy the sensors by considering the energy hole issue to maximize the lifetime of the WSN on the several network topology scenarios. The authors of [20] propose two algorithms to find an optimal sojourn duration and find an optimal sojourn location of a mobile sink in a cluster. The optimal sojourn duration is determined by balancing the amount of consumed energy between cluster heads. By this algorithm, the energy hole can be alleviated. The second algorithm to find an optimal sojourn duration operates to balance the amount of consumed energy between member sensor nodes in a cluster to alleviate the coverage hole issue. In [14], a data gathering algorithm named EPEGASIS is proposed to alleviate the energy hole issue with the mobile sink. In the algorithm, a sensor calculates an optimal communication distance and then chooses a relay node from its neighboring sensors. In addition, the algorithm has a mechanism to protect the sensors from power outage by unbalanced energy consumption.

Unfortunately, the previous works may not operate efficiently in the WSN where the sensors are not deployed uniformly. When the sensors are deployed nonuniformly, the probability of the energy hole issue occurring depends on the density of the sensors. Therefore, this paper considers the hotspot location to mitigate energy hole issues to formulate energy-efficient sensor cluster and mobile sink trajectory.

2.2. Sensor Clustering. There are numerous studies to formulate an efficient sensor cluster in the WSN with a static sink [21] or mobile sink [24–22]. In [22], the wedge-based clustering and cluster merging algorithm is proposed in the static sink and the static sensor nodes of a WSN. In the algorithm, the cluster is merged when the residual energy of sensors in a wedge falls below a threshold. In addition, the authors propose a cluster header election mechanism to minimize the amount of needed energy to forward a packet to the sink. In [23], the cluster header selection mechanism is proposed in case of public safety network. To balance the load between sensors, energy and sensor density-based clustering algorithm is proposed in [21]. This paper proposes an algorithm to formulate clusters by dividing the whole network into equally sized subclusters. In addition, this paper proposed a multihop routing path-making protocol based on the clusters to maximize the lifetime of sensors.

The authors of [24] propose a clustering algorithm to maximize the lifetime of the sensors in mobile sink-based WSN. The algorithm uses the adaptive immune algorithm to make up a trajectory of the mobile sink. In addition, the algorithm determines the optimal number of cluster heads to minimize the amount of consumed energy by communication and control packet overhead. The authors of [25] propose an energy-efficient routing algorithm by considering the mobility of the sink node. The algorithm is based on the clustering. Therefore, the algorithm contains sectorizing

of network and cluster head selection mechanism by considering the weight value of member sensor nodes. After cluster header selection, the member nodes construct a routing path by calculating the amount of consumed energy. To reduce the burden of sensors caused by location changes of the mobile sink, the author of [26] proposes a routing algorithm with a virtual multiring-shaped infrastructure. By this infrastructure, the burden can be reduced caused by advertising the new location of the mobile sinks.

However, most sensor clustering studies consider the situation where the sensors are relatively uniformly deployed. If sensor clustering is formulated without considering the hotspot where a large number of sensors are concentrated in the WSN, energy holes may occur. Therefore, this paper proposes an algorithm to find the appropriate cluster number and cluster header considering the density of sensors.

2.3. Mobile Sink Trajectory. There are numerous studies to formulate an energy-efficient mobile sink trajectory in the WSN [27–34]. In [27], an obstacle in the communication link between a sensor and a mobile sink is considered. This paper proposes a trajectory formulation mechanism by considering the obstacles in a cluster-based WSN. In [28], rechargeable sensor network is considered. The data gathering problem in this paper is formulated as a network utility maximization problem to maximize the amount of collected data by the mobile sink with a constraint about fairness. The authors of [29] propose a duty-cycling management and unmanned aerial vehicle (UAV) trajectory formulation algorithm to minimize the energy consumption of the whole sensor nodes in a WSN. In this paper, the problem is formulated as a mixed-integer nonconvex programming and the suboptimal solution is obtained by the successive convex optimization method. The artificial intelligence approach is applied in [30]. This paper proposes a data gathering mechanism in the WSN with obstacles. The mechanism has two rounds; the first round is clustering round by using ant colony optimization. A genetic algorithm is applied in the second round of the mechanism to construct an effective trajectory of the mobile sink. In [31], the UAV's trajectory formulation algorithm is proposed to maximize the lifetime of the network. In this algorithm, collection points similar with rendezvous point are selected before constructing the trajectory. After the collection points are determined, the mobile sink's trajectory is constructed by visiting all the collection points. In [32], a rendezvous-based data gathering protocol is proposed. This protocol consists of two steps. The first step is choosing the appropriate rendezvous point in the WSN. The second step is an energy-efficient mobile sink path planning step. The goal of the path planning step is making sure that as many sensors as possible are near the trajectory. The authors of [33] propose two trajectory planning algorithms based on rendezvous point. In this paper, two assumptions of rendezvous point papers are relaxed. One is the speed of data generated is the same between the sensors. The other is the sensors have an infinite buffer size. The authors of [34] propose a distributed UAV's trajectory determination algorithm to maximize the lifetime of a WSN based on game

theory and enhanced ant colony optimization technology. By game theory, the mobile sink finds the best rendezvous point and the optimal trajectory is calculated by enhancing ant colony optimization.

However, most of the trajectory planning studies formulate the trajectory in the case where sensors uniformly deployed. If the sensor is placed nonuniformly, the cluster headers might be closely placed in a specific region. To avoid this situation, it is necessary to dynamically change the sector visited by each mobile sink. Therefore, this paper proposes an algorithm that dynamically changes the sector based on the location of the cluster header.

3. System Model

In this section, we describe the system model considered in this paper. We also provide the mathematical problem to be optimized considering the total amount of consumed energy in the target WSN. We present basic information about the kinds of nodes consisting of the WSN. In addition, we describe two kinds of wireless propagation models used in this paper. We also define many mathematical models about cluster, trajectory, and energy. Finally, we formulate a mathematical problem. Table 1 describes the notation used in this paper and the corresponding meaning.

3.1. System Model. We consider a single WSN consisting of N sensor nodes (SN) and M mobile sinks as the system model. Figure 1 is a picture depicting the system model considered in this paper. We define a set, N , as the set of SNs: $\{1, 2, \dots, n, \dots, N\} \in \mathcal{N}$. We also define a set, M , as the set of mobile sinks: $\{1, 2, \dots, m, \dots, M\} \in \mathcal{M}$. $\psi_n = (x_n, y_n, h_n)$ is defined as a location profile of the n th SN. It is assumed that all sensors are placed on the ground; that is, the height value, h_n , of all sensors is set as zero. $\psi_m(t) = (x_m^t, y_m^t, h_m)$ is defined as a location profile of the m th mobile sink in time t . We assume that the altitude does not need to be changed while the mobile sinks are running. We define the location profile $(0, 0, 0)$ as the center of the WSN. The domain of the WSN is defined as a circle with the center at $(0, 0, 0)$ and radius R . By using the location profile of the SNs, we can calculate the distance between the n th SN and n' th SN, $d_{n,n'}$, as follows:

$$d_{n,n'} = \sqrt{(x_n - x_{n'})^2 + (y_n - y_{n'})^2}. \quad (1)$$

In a similar way, we can calculate the distance between the m th mobile sink and n th SN in time t , $d_{m,n}(t)$, as follows:

$$d_{m,n}(t) = \sqrt{(x_m^t - x_n)^2 + (y_m^t - y_n)^2 + (h_m)^2}. \quad (2)$$

As mentioned above, it is necessary to grasp the density of each sensor for the efficient clustering and the trajectory optimization since we are considering hotspot SN deployment. We define the density of a SN by using the k -nearest neighbor function. $\mathcal{N}_n$ is defined as a set of the x number

TABLE 1: Notation and description.

Notation	Description
N	The number of sensors
M	The number of mobile sinks
$\mathcal{N}$	The set of sensors
$\mathcal{M}$	The set of mobile sinks
R	Radius of area of the WSN
ψ_n	Location profile of sensor n
$\psi_m(t)$	Location profile of mobile sink m in time t
$d_{n,n'}$	Distance between sensor n and sensor n'
$d_{m,n}(t)$	Distance between mobile sink m and sensor n in time t
$\mathcal{N}_n$	The set of nearest sensor from sensor n
$\phi_{n,n'}$	Binary variable to express the near sensor of n
ξ_n	Density value
$PI_{m,n}^{AG}(t)$	Ground-to-air propagation model from sensor n to mobile sink m in time t
$PI_{n,n'}^{GG}(t)$	Ground-to-ground propagation model from sensor n to sensor n'
L_{FS}	Free-space propagation model
η	Environmental factor of the propagation model
f	Center frequency
P^{TX}	Transmission power of sensor
$P^{\min}$	Minimum received signal strength to decode
K	The number of clusters
$\mathcal{K}$	The set of clusters
$\delta_{k,n}$	Binary variable to express the cluster header of cluster k
$\mathcal{N}^{CH}$	The set of cluster headers
$\gamma^{k,n}$	Binary variable to express the member node of cluster k
D_k	Summation of distance between the cluster header and all member nodes of cluster k
$\lambda_{n,n'}^m$	Binary variable to express the path of mobile sink m from cluster header n to cluster header n_0
L_m	Total trajectory length of mobile sink m
e^{move}	Unit energy consumed per unit distance
E_m^{move}	The amount of consuming energy of the m th mobile sink
E^{AG}	The amount of consuming energy of the cluster header
$E_{n,n'}^{GG}$	The amount of consuming energy of sensor node n' when its cluster header is n
α	Scaling factor
$\mathbb{L}_D$	A list of density value of sensors in an ascending order
$\mathbb{L}_E$	A list of residual battery of sensors in a descending order
$\gamma_{\mathbb{L}_D}(n)$	Rank of sensor n in $\mathbb{L}_D$
$\gamma_{\mathbb{L}_E}(n)$	Rank of sensor n in $\mathbb{L}_E$

FIGURE 1: System model.

SNs closest to the n th SN. Using this set, we define a binary variable $\phi_{n,n'}$ as follows:

$$\phi_{n,n'} = \begin{cases} 1, & \text{if } n' \in \mathcal{N}_n, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

We define the density values of the n th SN, $\mathcal{X}_n$, as follows using $\phi_{n,n'}$ and Euclidean distance:

$$\mathcal{X}_n = \sum_{n' \in \mathcal{N}\{n\}} \phi_{n,n'} \cdot d_{n,n'}. \quad (4)$$

3.2. Wireless Channel Model. We assume that there are two types of communication in the considered system model. One is ground-to-ground propagation, where the member node SN sends data to its cluster header. The other is ground-to-air propagation, where the cluster header sends the collected data to the mobile sink. We use the ground-to-air propagation model already presented in [35, 36]. If the n th SN transmits the collected data to the m th mobile sink in time t , the considered ground-to-air propagation model, $\text{PL}_{m,n}^{\text{AG}}(t)$, is as follows:

$$\text{PL}_{m,n}^{\text{AG}}(t) = L_{\text{FS}} + 20 \log(d_{m,n}(t)) + \eta, \quad (5)$$

where η is an additional attenuation factor determined by the environment. L_{FS} is a free-space path loss model. This model with center frequency f is defined as follows:

$$L_{\text{FS}} = 20 \log(f) + 20 \log\left(\frac{4\pi}{c}\right), \quad (6)$$

where c is also environmental variable.

We also use the ground-to-ground propagation model already presented in [36, 37]. When the n th SN transmits a data to the n' th SN, the propagation model, $\text{PL}_{n,n'}^{\text{GG}}$, is as follows:

$$\text{PL}_{n,n'}^{\text{GG}}(t) = -55.9 + 38 \log(d_{n,n'}) + \left(24.5 + \frac{1.5f}{925}\right) \log(f). \quad (7)$$

The SN is the only transmitter of the two kinds of communication; we only define P^{TX} as transmission power of the SN. In addition, we define P^{min} as the minimum received signal strength for a successful reception. Overall, we can define a condition for successful signal reception as follows:

$$\frac{P^{\text{TX}}}{10^{PL/10}} \geq P^{\text{min}}. \quad (8)$$

3.3. Cluster and Trajectory Model. We assume that there are the K numbers of clusters in the considered system model. Let us suppose that K is a set of the clusters $\{1, 2, \dots, k, \dots, K\} \in \mathcal{K}$. We define a binary variable, $\delta_{k,n}$, to express the cluster header of the k th cluster as follows:

$$\delta_{k,n} = \begin{cases} 1, & \text{if } n \text{ is the cluster header of cluster } k, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

We also define a set of cluster headers of all cluster headers, $\mathcal{N}^{\text{CH}}$. In a similar way, we define a binary variable, $\gamma_{k,n}$, to express the member nodes of the k th cluster as follows:

$$\gamma_{k,n} = \begin{cases} 1, & \text{if } n \text{ is the member node of cluster } k, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

We can obtain the sum of the distance between the cluster header and the corresponding member SNs of the k th cluster, D_k , as follows:

$$D_k = \sum_{n \in \mathcal{N}^{\text{CH}}} \sum_{n' \in \mathcal{N}^{\text{CH}} \setminus \{n\}} \delta_{k,n} \cdot \gamma_{k,n'} \cdot d_{n,n'}. \quad (11)$$

In order to obtain the trajectory length for each mobile sink, we first need to define whether there is a trajectory path between cluster headers. If the m th mobile sink has a path from cluster header n to another cluster header n' , $\lambda_{n,n'}^m$ is defined as one. Otherwise, it is defined as zero. Using $\lambda_{n,n'}^m$, we can get the total trajectory length, L_m , of the m th mobile sink as follows:

$$L_m = \sum_{n \in \mathcal{N}^{\text{CH}}} \sum_{n' \in \mathcal{N}^{\text{CH}} \setminus \{n\}} \lambda_{n,n'}^m \cdot d_{n,n'}. \quad (12)$$

3.4. Energy Model. In the proposed system model, energy consumption occurs in three cases. One is the energy consumption that the mobile sink generates as it moves along a given trajectory path. When the trajectory length of the m th mobile sink is given through (12), the amount of consuming energy of the m th mobile sink by moving, $\mathbb{E}_m^{\text{move}}$, is defined as follows:

$$\mathbb{E}_m^{\text{move}} = L_m \cdot e^{\text{move}}. \quad (13)$$

e^{move} represents the unit energy consumed per unit distance. The other two energy consumption cases are caused by the SN. One of them appears when the cluster header sends the collected data to the mobile sink. We assume that the mobile sink comes to the exact location of the cluster header in order to receive data from the cluster header. Therefore, the transmission distance when the cluster header transmits the collected data to the mobile sink is unconditionally h_m . To increase the energy efficiency of the WSN, we assume that the cluster header sets the transmission power as the minimum strength for the mobile sink to successfully receive data. By these assumptions, the amount of energy consumed by the cluster header, $\mathbb{E}^{\text{AG}}$, is defined as follows:

$$\mathbb{E}^{\text{AG}} = P^{\min} \cdot 10^{PL^{\text{AG}}/10}. \quad (14)$$

One remaining energy consumption case is when the member node transmits data to the cluster header. We assume that all member nodes send data directly to its cluster header without relaying to mitigate energy hole issues. The same as (14), it is assumed that the member node sets the minimum transmission power at which the cluster header can successfully receive a signal. By these assumptions, the amount of energy consumed by member node n , $\mathbb{E}_{n,n'}^{\text{GG}}$, is as follows:

$$E_{n,n'}^{\text{GG}} = P^{\min} \cdot 10^{PL_{n,n'}^{\text{GG}}/10}. \quad (15)$$

By utilizing (13)–(15), we can formulate a mathematical expression of the total amount of consumed energy in the WSN as follows:

$$\mathbb{E}_{\text{tot}} = \sum_{m \in \mathcal{M}} \mathbb{E}_m^{\text{move}} + K \times \mathbb{E}^{\text{AG}} + \alpha \cdot \sum_{k \in \mathcal{K}} \sum_{n, n' \in \mathcal{N}} \gamma_{k,n} \cdot \gamma_{k,n'} \cdot \mathbb{E}_{n,n'}^{\text{GG}}. \quad (16)$$

α is a scaling factor to fairly consider the energy consumption according to the battery capacity of the SN and mobile sink. Obviously, the absolute energy consumption of mobile sinks is incomparably large for the consumption of SNs. However, mobile sinks have a large battery capacity compared to the SN, and charging is easy due to mobility. On the other hand, the SN has a very small battery and it is very difficult to charge or replace the battery. Therefore, it can be considered that the amount of battery consumed by communication has a large burden to the SN. Therefore, α is introduced to compensate for the absolute difference in energy consumption between the mobile sink and the SN.

$\mathbb{E}_m^{\text{move}}$ and $\mathbb{E}^{\text{AG}} + \mathbb{E}^{\text{GG}}$ have a trade-off relationship. In order to reduce the energy consumption of the SN, the size of the cluster must be made finer to reduce the consumption caused by the communication between the member node and the cluster header. However, as the number of cluster headers increases, the number of points to be visited by the mobile sink increases. This causes an increase of L_m . On the other hand, in order to reduce the energy consumption of the mobile sink, minimizing the L_m by reducing the number of clusters as much as possible is advantageous. However, it

increases the energy consumption of SNs. Therefore, we need to investigate this trade-off relationship and find the optimal cluster number and trajectory.

3.5. Problem Formulation. We intend to determine the number of clusters to minimize the amount of energy consumed by WSN components. In addition, we also intend to present the energy-efficient trajectory of each mobile sink. We also intend to determine the number of member nodes of all clusters to alleviate the energy hole problem of the cluster header. Therefore, the following mathematical optimization problem is presented by using (16):

$$\min_{K, \Delta, \Gamma, \Lambda} \mathbb{E}_{\text{tot}} \quad (17a)$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}} \gamma_{k,n} = 1, \quad \forall n \in \mathcal{N}, \quad (17b)$$

$$\sum_{n \in \mathcal{N}} \delta_{k,n} = 1, \quad \forall n \in \mathcal{N}, \quad (17c)$$

$$\sum_{m \in \mathcal{M}} \lambda_{n,n'}^m \leq 1, \quad \forall n \in \mathcal{N}^{\text{CH}}, \quad (17d)$$

$$\sum_{n' \in \mathcal{N}^{\text{CH}} \setminus \{n\}} \lambda_{n,n'}^m = 1, \quad \forall n \in \mathcal{N}^{\text{CH}}, \quad (17e)$$

$$\sum_{n' \in \mathcal{N}^{\text{CH}} \setminus \{n\}} \lambda_{n,n'}^m = 1, \quad \forall n \in \mathcal{N}^{\text{CH}}. \quad (17f)$$

Δ , Γ , and Λ are defined as the cluster header, member node profile, and trajectory path profile, respectively. Δ and Γ is a matrix of $K \times N$. Λ is a matrix of $M \times K^2$. Equation (17b) is a constraint that means that all SNs can belong to only one cluster. Equation (17c) is a constraint that every cluster can only have at most one cluster header. Constraint (17d) means that the path between the n th cluster header and the n' th cluster header can only be moved by one mobile sink. Constraints (17e) and (17f) allow the trajectory not to be scattered into multiple branches but to have only one path.

4. Proposed Method

This section provides a detailed description of the ISCTO algorithm to solve the presented problem efficiently. We reformulate the problem to solve (17a) easily. The reformed problem is divided into sensor clustering problem and trajectory optimization problem. To solve the sensor clustering problem, we define the cluster header election algorithm considering the density function defined in (4) and the amount of residual energy of the battery of the sensors. In the trajectory optimization process, we propose a trajectory path planning algorithm to minimize the amount of consumed energy by mobile sinks. Finally, we introduce an algorithm that combines the two proposed algorithms to reduce energy consumption by an iterative way.

4.1. Problem Decomposition. There is an obstacle that the domain of problem (17a) presented earlier is very wide to solve the problem directly. In order to determine the number

of cluster headers, there are a total of N cases ranging from a minimum of 1 to a maximum of N . If there are k clusters, we need to examine N^k cases to find the appropriate k cluster headers among all sensors. Finally, $(k-1)^k$ paths should be considered for trajectory path planning. Consequently, the domain of (17a) is total $N^{k+1} \cdot (k-1)^k$. Therefore, we need to decompose the problem into two problems in order to solve the existing problems quickly and efficiently. One is a problem considering the energy consumption of the SN. The other is a problem of considering the energy consumption of the mobile sink. The decomposed problem about minimizing the amount of consumed energy by all SNs is as follows:

$$\min_{K, \Delta, \Gamma} K \times \mathbb{E}^{\text{AG}} + \sum_{k \in \mathcal{K}} \sum_{n, n' \in \mathcal{N}} \gamma_{k,n} \cdot \gamma_{k,n'} \cdot \mathbb{E}_{n,n'}^{\text{GG}} \quad (18a)$$

$$\text{s.t.} \quad \sum_{k \in \mathcal{K}} \gamma_{k,n} = 1, \quad \forall n \in \mathcal{N}, \quad (18b)$$

$$\sum_{n \in \mathcal{N}} \delta_{k,n} = 1, \quad \forall n \in \mathcal{N}. \quad (18c)$$

In a similar way, the decomposed problem of mobile sink energy consumption is as follows:

$$\min_{\Lambda} \sum_{m \in \mathcal{M}} \mathbb{E}_m^{\text{move}} \quad (19a)$$

$$\text{s.t.} \quad \sum_{m \in \mathcal{M}} \lambda_{n,n'}^m \leq 1, \quad \forall n \in \mathcal{N}^{\text{CH}}, \quad (19b)$$

$$\sum_{n' \in \mathcal{N}^{\text{CH}} \setminus \{n\}} \lambda_{n,n'}^m = 1, \quad \forall n \in \mathcal{N}^{\text{CH}}, \quad (19c)$$

$$\sum_{n' \in \mathcal{N}^{\text{CH}} \setminus \{n\}} \lambda_{n',n}^m = 1, \quad \forall n \in \mathcal{N}^{\text{CH}}. \quad (19d)$$

Despite going through the decomposition process, the domain is still very large to solve both problems directly. In particular, (19a) is a famous NP-hard problem that can be reduced to a traveling salesman problem [38]. Therefore, we first propose an algorithm to select the appropriate cluster header SN while gradually increasing the K value from the minimum value. Also, we present the energy-efficient trajectory path planning algorithm of mobile sinks using the cluster header SN given as the result of the aforementioned algorithm.

4.2. Sensor Clustering. As mentioned earlier, we need to devise an algorithm to solve problem (18a) easily because the domain is still very large to solve directly. We propose an algorithm to minimize the energy consumption of all SNs by forming sensor clusters and finding a suitable cluster header. Algorithm 1 shows the pseudocode of the sensor clustering algorithm.

Before we explain the procedures of the algorithm, we need to figure out which SN is suitable to be a cluster header. We consider the density value defined by (4) and the amount of residual energy of all SNs in selecting a suitable cluster

```

Input :  $\Delta, \Gamma, K$ 
Output:  $\Delta, \Gamma$ 
1 Make  $\mathbb{L}_D, \mathbb{L}_E$ 
2 for  $n \in \mathcal{N}$  do
3   Obtain  $r_{\mathbb{L}_D}(n), r_{\mathbb{L}_E}(n)$ 
4 end
5 if  $K = M$  then
6   Find  $M$  number of  $n$  with lowest  $r_{\mathbb{L}_D}(n), r_{\mathbb{L}_E}(n)$ 
7   for  $n$  with lowest  $r_{\mathbb{L}_D}(n), r_{\mathbb{L}_E}(n)$ , and  $k \in \mathcal{K}$  do
8      $\delta_{k,n} = 1$ 
9   end
10 end
11 if  $K \neq M$  then
12   Find  $k$  with most number of member nodes
13   Find  $n$  with lowest  $r_{\mathbb{L}_D}(n), r_{\mathbb{L}_E}(n)$  in  $k$ 
14    $\delta_{k,n} = 1$ 
15 end
16 for  $n \in \mathcal{N}$  do
17   Update  $\gamma$  with shortest  $d_{n,n'}, n' \in \mathcal{N}_{CH}$ 
18 end
19 return  $\Delta, \Gamma$ 

```

ALGORITHM 1. Density-battery coupled sensor cluster algorithm.

header. It can be said that the SN is suitable for becoming a cluster header in the following situations. In terms of density, it is better that a SN which can minimize D_k becomes the cluster header to reduce the amount of energy that member nodes consume by transmission. In terms of residual energy, a SN with larger residual energy is good for a cluster header so that data transmitted by a large number of member nodes can be successfully received without discharge. We define $\mathbb{L}_D$ as a list in which all SNs in N are sorted in an ascending order by density value, $\mathcal{X}_n$. In a similar way, we define $\mathbb{L}_E$ as a list of all SNs in N sorted in a descending order by residual battery. We define $\gamma_{\mathbb{L}_D}(n)$ and $\gamma_{\mathbb{L}_E}(n)$ as a rank using the order of a specific SN n in the two lists of a specific SN, L_D and L_E . We get the sum of the two ranks, $\gamma_{\mathbb{L}_D}(n) + \gamma_{\mathbb{L}_E}(n)$ of all SNs. A SN n with the smallest sum of ranks is a suitable SN for cluster header. The SN is selected as a cluster header.

The proposed algorithm works in the following order. If there is no cluster header already placed, that is, $K = 0$, the number of cluster headers of the same number as the mobile sink, M , is selected in the order of the lowest sum of the two ranks defined above. The reason for selecting the first M cluster header is to form the trajectory sector of each mobile sink, which is explained in detail in Section 4.3. If there are previously formed cluster headers, that is, $k \geq M$, an additional cluster header is selected. Unlike the initial M cluster header selection, the newly selected cluster header is selected of a SN with the lowest sum of ranks in the cluster with the most member nodes among the existing clusters. The reason of this procedure is to mitigate the energy hole issue. The header of a cluster with the most member nodes increases the battery consumption due to data reception compared to other headers. Therefore, the more the member nodes, the faster the header battery is

discharged, so it is necessary to disassemble the cluster. If cluster headers are selected through the above-mentioned process, all remaining SNs form a cluster by belonging to the shortest $d_{n,n'}$ cluster header.

4.3. Trajectory Optimization. The trajectory optimization given by (19a) is a famous NP-hard problem, and we need to devise an algorithm for trajectory optimization. Fortunately, the ant colony optimization can be a good solution to find the minimum length of the trajectory. We form the mobile sink trajectory using the ant colony optimization. The remaining task is to determine the cluster headers that each mobile sink should visit. To solve this problem, a sectorization-based technique is proposed to determine the coverage of each mobile sink in the entire network. It is assumed that each mobile sink visits all cluster headers in its own sector. Also, we propose an algorithm that adjusts the sector to find the optimal trajectory when a new cluster header is added by the algorithm mentioned in Section 4.2.

Algorithm 2 shows the pseudocode of the trajectory optimization algorithm. If the first M cluster header is selected by the sensor clustering algorithm, one mobile sink is matched for each cluster header. At this time, the first sector of each mobile sink should be formed based on the location of the cluster headers. We make line segments so that both ends are the center of the WSN and the location of each cluster header n . In addition, we make line segments that bisect the angle formed by neighboring segments by the length of R starting from the center of the WSN. The arc formed by the segments of the length of R becomes the initial sector in charge of each mobile sink. s_m is the sector of mobile sink m , and θ_m is the central angle of arc-shape sector s_m .

When a new cluster header appears by the clustering algorithm, it is inevitably included in an area of one sector among previously set sectors. In this case, we define two cases named C1 and C2. First, C1 is a case in which the mobile sink in charge of the sector including the added cluster header visits the newly added cluster header. In this case, the trajectory length of mobile sinks in charge of other sectors does not change. L_m^{C1} is defined as the trajectory length of mobile sink m in case C1. L^{C1} is defined as the sum of the trajectory lengths of all mobile sinks calculated in case C1:

$$L^{C1} = \sum_{m \in \mathcal{M}} L_m^{C1}. \quad (20)$$

C2 is defined as adjusting the sector boundary. At this time, the way to adjust the sector is as follows. A sector with a new cluster header is defined as a source sector, and a sector with a small central angle of arc among two neighboring sectors is defined as a destination sector. The boundary adjustment between sectors is a method of moving one cluster header from the source sector to the destination sector and adjusting the boundary of the sector accordingly. The cluster header to be moved to the destination sector is defined as the cluster header having the shortest Euclidean distance from the boundary of two neighboring sectors. After the sector boundary adjustment is completed, the trajectory length of all mobile sinks is calculated. Mobile sinks in charge of other

```

Input :  $\Delta, \Lambda, K$ 
Output:  $\Lambda$ 
1 if  $K = M$  then
2   for  $n \in \mathcal{N}_{CH}$  and  $m \in \mathcal{M}$  do
3      $\lambda_{0,n}^m = 1$  and  $\lambda_{n,0}^m = 1$ 
4     Forming  $s_m$ 
5     Calculating  $\theta_m$ 
6      $L_m = 2 \times \sqrt{x_n^2 + y_n^2}$ 
7   end
8 end
9 if  $n \in \mathcal{N}_{CH}$  has no  $\lambda^m = 1 \forall m \in \mathcal{M}$  then
10   Find  $s_m$  included  $\psi_n$ 
11   Update  $\Lambda^{C1}$ 
12   for  $m \in \mathcal{M}$  do
13     Calculate  $L_m^{C1}$ 
14   end
15  $L^{C1} = \sum_{m \in \mathcal{M}} L_m^{C1}$ 
16    $s_{src} = s_m$ 
17    $s_{dest} = m'$  which min  $\{\theta_{m-1}, \theta_{m+1}\}$ 
18   Find  $n' \in s_{src}$  which has shortest Euclidean distance
19    $s_{src}$  removes  $n'$ 
20    $s_{dest}$  includes  $n'$ 
21   Update  $\Lambda^{C2}$ 
22   for  $m \in \mathcal{M}$  do
23     Calculate  $L_m^{C2}$ 
24   end
25    $L^{C2} = \sum_{m \in \mathcal{M}} L_m^{C2}$ 
26   if  $L^{C1} \geq L^{C2}$  then
27      $\Lambda = \Lambda^{C2}$ 
28   end
29   if  $L^{C1} < L^{C2}$  then
30      $\Lambda = \Lambda^{C1}$ 
31   end
32 end
33 return  $\Lambda$ 

```

ALGORITHM 2. Arc-shape sector boundary adjustment.

sectors except for the source sector and destination sector do not have a change in trajectory length. L_m^{C2} is defined as the trajectory length of mobile sink m in case C2. The total trajectory length at this case is defined as

$$L^{C2} = \sum_{m \in \mathcal{M}} L_m^{C2}. \quad (21)$$

The values of L^{C1} and L^{C2} are compared to determine the final trajectory. In this case, since the case with a smaller value reduces the amount of energy consumed by the mobile sink, the case is presented as the output of the algorithm. The sector adjustment method is explained in Figure 2. Number 1 is when the first M cluster headers appeared. In case 2, the first sector is created based on the location of the first M cluster headers. In case 3, a new cluster header has appeared. 4-1 is the process of calculating L^{C1} , and 4-2 is the process of calculating L^{C2} . Finally, number 5 is the result of calculating the final trajectory.

FIGURE 2: Procedure of modifying the territories of each sector.

FIGURE 3: Special case of sector adjusting.

As a special case, there is a case where all cluster headers of the source sector are delivered to the destination sector as a result of the sector boundary adjustment. This special case is shown in Figure 3. In other words, it means the case where the mobile sink in charge of the source sector does not take off. This occurs when all the cluster headers of the source sector are included in the area that goes to the destination sector. Assuming that the trajectory length at this case is calculated as the output of the algorithm. If the next new cluster header appears, C1 and C2 are defined as follows. C1 is defined as the situation in which trajectory is designed without sector adjustment as before. C2 is defined as a case in which a cluster header is newly allocated to a mobile sink that has not taken off in the previous step. The sector boundary adjustment and trajectory are calculated according to this case instead of setting a neighboring sector as a destination sector.

4.4. ISCTO: Iterative Sensor Clustering and Trajectory Optimization. Using the algorithms proposed in Sections 4.2 and 4.3, we propose a sensor clustering and trajectory optimization algorithm to minimize the energy consumption, which is named ISCTO. As mentioned earlier, there is a trade-off relationship between the amount of energy consumption by the SN and mobile sink due to the different trends according to the value of K . Therefore, we find that trade-off point through an iterative approach.

Algorithm 3 shows the pseudocode showing the operation procedure of ISCTO. We define the initial Δ , Γ and Λ as follows:

$$\begin{aligned} \Delta(0) &= \{0 | \forall \delta_{k,n}, k \in \mathcal{K}, n \in \mathcal{N}\}, \\ \Gamma(0) &= \{0 | \forall \gamma_{k,n}, k \in \mathcal{K}, n \in \mathcal{N}\}, \\ \Lambda(0) &= \{0 | \forall \lambda_{n,n}^m, k \in \mathcal{K}, n \in \mathcal{N}^{\text{CH}}\}. \end{aligned} \quad (22)$$

```

Input :  $\mathcal{N}, \mathcal{M}, \Psi, \mathcal{K} = \phi$ 
Output:  $K, \Delta, \Gamma, \Lambda$ 
1 Initialize  $\Delta, \Gamma, \Lambda$  as  $\Delta(0), \Gamma(0), \Lambda(0)$ 
2  $K = M$ 
3  $\Delta(1), \Gamma(1) = \text{Algorithm1}(\Delta(0), \Gamma(0), K)$ 
4  $\Lambda(1) = \text{Algorithm2}(\Delta(1), \Lambda(0), K)$ 
5  $\mathbb{E}_{\text{tot}}^{\text{old}} = \text{calculate } \mathbb{E}_{\text{tot}}(K, \Delta(1), \Gamma(1), \Lambda(1))$ 
6  $t = 2$ 
7 while  $\mathbb{E}_{\text{tot}}^{\text{new}} \leq \mathbb{E}_{\text{tot}}^{\text{old}}$  do
8    $K = K + 1$ 
9    $\Delta(t), \Gamma(t) = \text{Algorithm1}(\Delta(t-1), \Gamma(t-1), K)$ 
10   $\Lambda(t) = \text{Algorithm2}(\Delta(t), \Lambda(t-1), K)$ 
11   $\mathbb{E}_{\text{tot}}^{\text{new}} = \text{calculate } \mathbb{E}_{\text{tot}}(K, \Delta(t), \Gamma(t), \Lambda(t))$ 
12  if  $\mathbb{E}_{\text{tot}}^{\text{new}} = \mathbb{E}_{\text{tot}}^{\text{old}}$  then
13     $\mathbb{E}_{\text{tot}}^{\text{old}} = \mathbb{E}_{\text{tot}}^{\text{new}}$ 
14     $\mathbb{E}_{\text{tot}}^{\text{new}} = 0$ 
15     $t = t + 1$ 
16  end
17 end
18 return  $K - 1, \Delta(t-1), \Gamma(t-1), \Lambda(t-1)$ 

```

ALGORITHM 3. ISCTO.

TABLE 2: System parameters.

Notation	Value
N	50-400
M	3, 5
R	500 (m)
h_m	50, 100 (m)
x	10
f	1 (GHz)
c	0.6
η	1
p^{TX}	10 (dBm)
p^{min}	-70 (dBm)
e^{move}	10^{-5}
α	54

FIGURE 4: Clustering example of random deployment.

First, M initial cluster headers must be set to form M sectors. If initial M cluster headers are given, M initial sectors are formed based on the location of the cluster headers. Based on the two algorithms, the energy consumption $\mathbb{E}_{\text{tot}}$ is calculated using the given sensor cluster and trajectory. At this time, $\mathbb{E}_{\text{tot}}$ is defined as the first $\mathbb{E}_{\text{tot}}^{\text{old}}$. If $\mathbb{E}_{\text{tot}}^{\text{old}}$ is defined, we have to process the sensor cluster algorithm again, select one new cluster header, and perform trajectory optimization by considering the newly added cluster header. The next step is to calculate the energy consumption $\mathbb{E}_{\text{tot}}^{\text{new}}$ using the newly obtained sensor cluster and trajectory. Then, the values of $\mathbb{E}_{\text{tot}}^{\text{old}}$ and $\mathbb{E}_{\text{tot}}^{\text{new}}$ are compared. If $\mathbb{E}_{\text{tot}}^{\text{old}}$ is higher, it means that the new cluster header and trajectory represent better energy efficiency. Therefore, $\mathbb{E}_{\text{tot}}^{\text{old}}$ is overwritten with $\mathbb{E}_{\text{tot}}^{\text{new}}$. After that, the algorithm goes back to the sensor cluster algorithm and add a new cluster. If $\mathbb{E}_{\text{tot}}^{\text{old}}$ has a value lower than $\mathbb{E}_{\text{tot}}^{\text{new}}$, this algorithm is ended. The sensor cluster and trajectory from the last step are derived as the output of the entire algorithm.

The ISCTO algorithm forms clusters independently of the mobility of mobile sinks to improve energy efficiency in hotspots where the SN is densely located. As the defined density value in (4) is smaller, the probability of being selected as a cluster header increases, so a relatively large number of clusters are formed in the hotspot. It is possible to mitigate the energy hole issue of the cluster header by inducing that cluster headers in charge of hotspot and nonhotspot have

similar member nodes. In order to improve the energy efficiency of the mobile sink, the ISCTO algorithm divides the entire network into M sectors to form the trajectory of the mobile sink. The mobility of the mobile sink is dependent on the location of the cluster headers located in its own sector.

5. Performance Evaluation

In this section, we present the performance of the proposed algorithm through simulation. The simulation environment performed is described. We consider two uneven sensor deployment scenarios. One is the fully random deployment of sensors. The other is a hotspot scenario where the majority of sensors are placed in a specific area. We present each performance in each sensor node deployment scenario. Finally, we present a graph analyzing the running time of the proposed algorithm.

5.1. Simulation Environment. We create a simulation to demonstrate the performance of the ISCTO. Table 2 shows the system parameters used in the created simulation. System parameters are determined based on [39, 40]. The values of all graphs are the average values of 10,000 times of experiments with the changing deployment of sensors. We considered two SN deployment scenarios. One is a scenario in which SNs are completely randomly deployed in the entire

FIGURE 5: Trajectory example of random deployment.

area of the WSN and is called a “fully randomly deployment” scenario. The other is a scenario where many SNs are placed in a specific area of the entire area of the WSN, which is called a “hotspot deployment” scenario. For the “hotspot” scenario, we experiment by placing 50% of the SN in the 1/4 area of the WSN.

5.2. Amount of Energy Consumption. Figure 4 shows the clustering transition for each iteration when $N = 200$ in a fully random deployment scenario. Figure 4(a) is the clustering result at the first iteration. The number of the first cluster is 3 because of the initial process of the sensor clustering algorithm when $M = 3$. Figure 4(b) is the result of the second iteration. In this figure, we can confirm that the new cluster head is separated from the green cluster head which had the most member nodes. Additionally, we can see that the cluster has been adjusted as a result. Figure 4(c) is the state at the end of sensor clustering; this example shows that the total of 6 clusters is formed. Figure 5 depicts the trajectory of each mobile sink formed in Figure 4. Each mobile sink trajectory is identified by the color of the arrow. Figure 5(a) is the situation where the first M cluster header is formed; each mobile sink is configured to handle one cluster header. Figure 5(b)

FIGURE 6: Performance of random deployment.

shows the updated trajectory after the new cluster header was created. In this example, the black arrow means the source sector, and the red arrow means the destination

FIGURE 7: Clustering example of hotspot deployment.

sector. In this situation, due to the fact that C2 returned a shorter trajectory, one cluster header was moved closest to the destination sector in the source sector. With repetition of the aforementioned process, we obtain the final trajectory like Figure 5(c).

Figure 6 shows the graph of the transition of $\mathbb{E}_{\text{tot}}$ according to the change of N in the random deployment scenario. The graph contains the results of experiments on four cases according to changes in the number of the mobile sink, M , and the altitude, h , of the mobile sink. It can be said that as the number of SNs increases, $\mathbb{E}_{\text{tot}}$ increases. The reason for $M=3$ showing a lower result compared to $M=5$ is that $\mathbb{E}^{\text{move}}$ increases with the increasing of M . It can be said that the altitude does not significantly affect the overall result. If the value of K is not extremely large, it can be said that the term $\mathbb{E}^{\text{AG}}$ does not play a large part in the result.

Figure 7. shows the cluster formation process for each iteration in a hotspot deployment scenario. In this example, it can be seen that the hotspot is formed at the upper left of the WSN. Since the density value, $\mathcal{X}$, is considered in the process of determining the cluster header, it can be seen that the sensor belonging to the hotspot was continuously determined as the cluster header in the beginning. In Figure 7(b), the first cluster header that does not belong to the hotspot is selected as a brown cluster. Figure 8 shows the trajectory formation process for each mobile sink in Figure 7. As it can be

seen in Figure 7, there is a probability which the number of the cluster headers in charge of mobile sinks can be unbalanced as many cluster headers are generated in the hotspot. In this example, it can be seen that the mobile sink, which follows the red arrow, is in charge of more than two or three cluster headers than other mobile sinks. However, the sector area between mobile sinks is adjusted to have a uniform central angle as much as possible, so that the total length of trajectory for each mobile sink is set to be balanced.

Figure 9 is the graph showing $\mathbb{E}_{\text{tot}}$ according to the change of N in the hotspot deployment scenario. The experiment also compared the results of the four cases according to the changes of M and h . Most of the trend of performance value change shows the same trends shown in the random deployment scenario, though it shows a higher result value compared to the random deployment scenario. This is because hotspot deployment scenarios tend to create more clusters than random deployments. As the number of clusters increases, the trajectory length of the mobile sink also increases, and this causes an increase in the result.

5.3. Distribution of Member Node. We present the distribution of cluster member nodes as a performance indicator associated with the energy hole issue. We investigate the distribution of member nodes by cluster after experiments in two SN deployment scenarios when $M=3$ and $M=5$.

FIGURE 8: Trajectory example of hotspot deployment.

FIGURE 9: Performance of hotspot deployment.

Figures 10 and 11 represent the CDF of the number of member nodes for each cluster when $N = 200$ and $N = 400$, respectively. As the graph is directed to the left, it means the

average number of member nodes in the cluster is less. Both graphs show the same trend regardless of N . Also, in the case of the same SN deployment scenario, it can be seen that the larger the M , the fewer the member nodes. The reason for this result is based on the beginning mechanism of proposing sensor clustering algorithm. The sensor clustering algorithm creates the first M cluster at the beginning. Therefore, the larger the M , the more clusters can be formed. As more clusters are formed, the number of member nodes will naturally decrease. Therefore, larger M tends to include fewer member nodes.

In the case of the same number of M , the trend of having fewer member nodes in the hotspot deployment than in the random sensor deployment can be seen. In the hotspot deployment scenario, most of the cluster headers are generated in the hotspot due to γ_{LD} .

Although most of SNs are in the hotspot, hotspot clusters are more finely divided as many cluster headers are generated. This results in creating clusters with fewer member nodes on average in hotspot deployments. In addition, as mentioned above, the tendency for more clusters to occur in a hotspot deployment scenario is also a reason for the cluster to have fewer member nodes.

FIGURE 10: CDF of member node on 200 SNs.

FIGURE 11: CDF of member node on 400 SNs.

On the other hand, the distribution of the number of member nodes in the two deployment scenarios crosses around $1/4 \cdot N$ points. The reasons for this result are as follows. In a random deployment scenario, sensors and cluster headers are relatively evenly distributed. There is low probability that numerous member nodes are included with bias in one cluster. However, in a hotspot deployment, due to γ_{LD} , only a small number of clusters are created in a region outside the hotspot which causes the number of member nodes gathered in one cluster.

5.4. Time Complexity Analysis. To show the running time of the proposed algorithm, we experiment the average number of iteration until the algorithm ended according to changes in N . The same as the above experiment, we investigate in two SN deployment scenarios when $M=3$ and $M=5$. Figure 12 is a graph presenting the number of iterations. In the case of the same M , it can be seen that iteration takes more in the hotspot than in the random deployment. As

FIGURE 12: Average iteration count.

shown in Figure 7, the reason is that the cluster header is continuously generated only at the hotspot. If the newly generated cluster header is not far from the previously generated cluster header, this means that the previously formed trajectory also does not need to be significantly changed. This means that even if a new cluster header is generated, the reduction amount of $\mathbb{E}^{\text{move}}$ does not occur significantly. Because of this, the proposed algorithm causes more iteration in the hotspot deployment environment.

Interestingly, depending on the change of the value of M , the trend of the iteration number changes differently. In the random deployment scenario, the larger the M , the lesser the iteration results. This is because when the M clusters are initially formed, the cluster headers are not biased to one place. Therefore, the benefit of EGG obtained by adding the cluster header is not significantly large. On the other hand, in the hotspot deployment scenario, when the initial M clusters are formed, cluster headers are intensively generated in the hotspot. The EGG generated when a new cluster header is created in the nonhotspot area is significantly large. Therefore, it runs more iterations.

In addition, we can infer that the average number of cluster headers generated by the ISCTO is $(M + \text{the number of iteration})$ through the graph.

6. Conclusion

We propose a sensor clustering and trajectory path planning algorithm named ISCTO for uneven SN deployed WSN with multiple mobile sinks. This algorithm iteratively performs two stages of sensor clustering and trajectory optimization to minimize the amount of energy consumed by WSN components. In the sensor clustering phase, two rankings are defined using density function and residual battery level.

Whenever a new cluster header is needed, the SN with the lowest sum of rankings is added as a new cluster header. In trajectory optimization, WSN is divided into several sectors to minimize the amount of energy that the mobile sink consumes. Whenever a new cluster header is created, consider the two cases defined and find the case that shows the shorter trajectory length. We show the performance of the ISCTO

algorithm through simulation. In the future, we will improve the proposed algorithm to support the one million devices per 1 square of kilometer situation, which is a requirement of 5G mMTC.

Data Availability

We create a simulation to demonstrate the performance of our proposed algorithm.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2018-0-00969, full duplex nonorthogonal multiple access (NOMA) optimization technologies using deep learning for 5G-based autonomous vehicular networks) and was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2018R1D1A1B07049043).

References

- [1] T. Kim, J. Noh, and S. Cho, "SCC: storage compression consensus for blockchain in lightweight IoT network," in *2019 IEEE 684 International Conference on Consumer Electronics (ICCE)*, pp. 1–4, Las Vegas, NV, 2019.
- [2] V. Bapat, P. Kale, V. Shinde, N. Deshpande, and A. Shaligram, "WSN application for crop protection to divert animal intrusions in the agricultural land," *Computers and Electronics in Agriculture*, vol. 133, pp. 88–96, 2017.
- [3] V. Garcia-Font, C. Garrigues, and H. Rifà-Pous, "A comparative study of anomaly detection techniques for smart city wireless sensor networks," *sensors*, vol. 16, no. 6, p. 868, 2016.
- [4] H. Mostafaei, "Energy-efficient algorithm for reliable routing of wireless sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 7, pp. 5567–5575, 2019.
- [5] Y. Liu, A. Liu, N. Zhang, X. Liu, M. Ma, and Y. Hu, "DDC: dynamic duty cycle for improving delay and energy efficiency in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 131, pp. 16–27, 2019.
- [6] T. Ruan, Z. J. Chew, and M. Zhu, "Energy-aware approaches for energy harvesting powered wireless sensor nodes," *IEEE Sensors Journal*, vol. 17, no. 7, pp. 2165–2173, 2017.
- [7] V. L. Quintero, C. Estevez, M. E. Orchard, and A. Perez, "Improvements of energy-efficient techniques in WSNs: a MAC-protocol approach," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1188–1208, 2019.
- [8] S. Abdollahzadeh and N. J. Navimipour, "Deployment strategies in the wireless sensor network: a comprehensive review," *Computer Communications*, vol. 91–92, pp. 1–16, 2016.
- [9] B. Wang, J. Zhu, L. T. Yang, and Y. Mo, "Sensor density for confident information coverage in randomly deployed sensor networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 5, pp. 3238–3250, 2016.
- [10] D. Saha and S. Bandyopadhyay, "Finding minimum node density for energy-efficient in-hop cooperative relaying in industrial WSNs," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, vol. 74no. 12, pp. 6633–6645, Jaipur, 2018.
- [11] H. Asharioun, H. Asadollahi, T.-C. Wan, and N. Gharai, "A survey on analytical modeling and mitigation techniques for the energy hole problem in corona-based wireless sensor network," *Wireless Personal Communications*, vol. 81, no. 1, pp. 161–187, 2015.
- [12] K. Cengiz and T. Dag, "Energy aware multi-hop routing protocol for WSNs," *IEEE Access*, vol. 6, pp. 2622–2633, 2018.
- [13] J. Wang, J. Cao, R. S. Sherratt, and J. H. Park, "An improved ant colony optimization-based approach with mobile sink for wireless sensor networks," *The Journal of Supercomputing*, vol. 74, no. 12, pp. 6633–6645, 2018.
- [14] J. Wang, Y. Gao, X. Yin, F. Li, and H. J. Kim, "An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 9472075, 9 pages, 2018.
- [15] N. Jan, N. Javaid, Q. Javaid et al., "A balanced energy-consuming and hole-alleviating algorithm for wireless sensor networks," *IEEE Access*, vol. 5, pp. 6134–6150, 2017.
- [16] X. Liu, "A novel transmission range adjustment strategy for energy hole avoiding in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 67, pp. 43–52, 2016.
- [17] R. E. Mohamed, A. I. Saleh, M. Abdelrazzak, and A. S. Samra, "Energy-efficient routing protocols for solving energy hole problem in wireless sensor networks," *Computer Networks*, vol. 114, pp. 51–66, 2017.
- [18] J. Ren, Y. Zhang, K. Zhang, A. Liu, J. Chen, and X. S. Shen, "Lifetime and energy hole evolution analysis in data-gathering wireless sensor networks," *IEEE transactions on industrial informatics*, vol. 12, no. 2, pp. 788–800, 2016.
- [19] H. S. Ramos, A. Boukerche, A. L. C. Oliveira, A. C. Frery, E. M. R. Oliveira, and A. A. F. Loureiro, "On the deployment of large-scale wireless sensor networks considering the energy hole problem," *Computer Networks*, vol. 110, pp. 154–167, 2016.
- [20] N. Gharai, K. Abu Bakar, S. Z. M. Hashim, and A. H. Pourasl, "Inter- and intra-cluster movement of mobile sink algorithms for cluster-based networks to enhance the network lifetime," *Ad Hoc Networks*, vol. 85, pp. 60–70, 2019.
- [21] K. A. Darabkh, S. M. Odetallah, Z. al-qudah, A. F. Khalifeh, and M. M. Shurman, "Energy-aware and density-based clustering and relaying protocol (EA-DB-CRP) for gathering data in wireless sensor networks," *Applied Soft Computing*, vol. 80, pp. 154–166, 2019.
- [22] N. Sharmin, A. Karmaker, W. L. Lambert, M. S. Alam, and M. S. T. S. A. Shawkat, "Minimizing the energy hole problem in wireless sensor networks: a wedge merging approach," *Sensors*, vol. 20, no. 1, p. 277, 2020.
- [23] A. R. Ansari and S. Cho, "CHESS-PC: cluster-head selection scheme with power control for public safety networks," *IEEE Access*, vol. 6, pp. 51640–51646, 2018.
- [24] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, and S. Sasaki, "Mobile sink-based adaptive immune energy-efficient clustering protocol for improving the lifetime and stability period of wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4576–4586, 2015.

- [25] J. Wang, Y. Gao, W. Liu, A. K. Sangaiah, and H.-J. Kim, "Energy efficient routing algorithm with mobile sink support for wireless sensor networks," *Sensors*, vol. 19, no. 7, p. 1494, 2019.
- [26] R. Yarinezhad, "Reducing delay and prolonging the lifetime of wireless sensor network using efficient routing protocol based on mobile sink and virtual infrastructure," *Ad Hoc Networks*, vol. 84, pp. 42–55, 2019.
- [27] G. Xie and F. Pan, "Cluster-based routing for the mobile sink in wireless sensor networks with obstacles," *IEEE Access*, vol. 4, pp. 2019–2028, 2016.
- [28] Y. Zhang, S. He, and J. Chen, "Near optimal data gathering in rechargeable sensor networks with a mobile sink," *IEEE Transactions on Mobile Computing*, vol. 16, no. 6, pp. 1718–1729, 2017.
- [29] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in UAV enabled wireless sensor network," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 328–331, 2018.
- [30] S. Najjar-Ghabel, L. Farzinvas, and S. N. Razavi, "Mobile sink-based data gathering in wireless sensor networks with obstacles using artificial intelligence algorithms," *Ad Hoc Networks*, vol. 106, p. 102243, 2020.
- [31] 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 Journal*, vol. 18, no. 2, pp. 890–901, 2018.
- [32] C. Konstantopoulos, N. Vathis, G. Pantziou, and D. Gavalas, "Employing mobile elements for delay-constrained data gathering in WSNs," *Computer Networks*, vol. 135, pp. 108–131, 2018.
- [33] Y. C. Wang and K. C. Chen, "Efficient path planning for a mobile sink to reliably gather data from sensors with diverse sensing rates and limited buffers," *IEEE Transactions on Mobile Computing*, vol. 18, no. 7, pp. 1527–1540, 2019.
- [34] P. V. P. Raj, A. M. Khedr, and Z. A. Aghbari, "Data gathering via mobile sink in WSNs using game theory and enhanced ant colony optimization," *Wireless Networks*, vol. 26, no. 4, pp. 2983–2998, 2020.
- [35] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," *IEEE Global Communications Conference*, pp. 2898–2904, 2014.
- [36] S. Zhang, H. Zhang, B. Di, and L. Song, "Cellular UAV-to-X communications: design and optimization for multi-UAV networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 2, pp. 1346–1359, 2019.
- [37] 3GPP TS 25 996, *Spatial channel model for multiple input multiple output (MIMO) simulations*, 3GPP, 2018.
- [38] M. J. Garey and D. S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness, 1st ed*, Siam Review, New York, NY, USA: Freeman, 1979.
- [39] 3GPP TR 22 289, *Enhancement for unmanned aerial vehicles; stage 1 (Rel. 17)*, 3GPP, 2019.
- [40] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, 2017.