

Research Article

Event-Sensitive Network: A Construction Algorithm of Agricultural Sensor Network Driven by Environmental Change

Shipu Xu ^{1,2} Yong Liu ^{1,2} WenWen Hu ^{1,2} Yingjing Wu ^{1,2} Sijia Liu ³
Yunsheng Wang ^{1,2} and Chang Liu⁴

¹Institute of Agricultural Information Science and Technology, Shanghai Academy of Agricultural Sciences, Shanghai 201403, China

²Shanghai Engineering and Technological Research Center for Digital Agriculture, Shanghai 201403, China

³Department of Railway Transportation, Shanghai Institute of Technology, Shanghai 201418, China

⁴School of Information Engineering, Nanchang Hangkong University, Nanchang, Jiangxi 330038, China

Correspondence should be addressed to Yunsheng Wang; wys188@163.com

Received 2 November 2020; Revised 24 December 2020; Accepted 4 January 2021; Published 13 January 2021

Academic Editor: Yi-Zhang Jiang

Copyright © 2021 Shipu Xu 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.

In a wireless sensor network, the sensor nodes transmit the acquired information to the server through the data transmission link. On the serverside, the data are processed, fused, and expressed to serve the user. Sensor deployment is a key factor related to the stability and security of wireless networks. This article uses environmental changes to drive related technologies to deploy wireless sensors. In this article, environmental change-driven means that through certain deployment cost model assumptions and problem descriptions, network deployment is artificially divided into two stages: initial deployment and redeployment. In the deployment phase, by referring to the idea of virtual force, a new sensor deployment algorithm is proposed in the redeployment phase, which can well solve the stability- and security-related issues encountered in agricultural wireless sensor networks. In this algorithm, the moving distance of the mobile receiver and the average coverage of the network are calculated based on the virtual force, the direction, and the number of adjacent clusters. Finally, the algorithm model was simulated in MATLAB, and the feasibility of the algorithm was verified by analyzing the event coverage and the moving distance of nodes. The final simulation results show that the algorithm proposed in this paper can achieve better performance than existing algorithms in terms of average coverage and moving distance.

1. Introduction

In wireless sensor networks, the spatiotemporal randomness of events in the monitoring area leads to a significant increase in the probability of missing event processing opportunities. This is a huge challenge for sensor network applications. Two major environmental monitoring applications in the agricultural Internet of Things, water environment monitoring and greenhouse environmental monitoring, are very sensitive to appeal issues. The former targets the randomly-occurring pollution sources, while the latter finds that the temperature and humidity in the indoor greenhouses and the sudden changes in pests and diseases become the main target [1]; the timeliness of abnormal event

monitoring is extremely high so that the best opportunity to handle the corresponding anomaly. The above two applications are typical representatives of current agricultural Internet monitoring applications, which drive the exploration of new wear-sensitive networks. Therefore, this paper takes the three-dimensional wireless sensor network as an example and introduces the connection between the virtual force idea and the cube square into the network. Finally, a new event-sensitive sensor network construction strategy is proposed.

In recent years, the construction strategies of agricultural sensor networks can be divided into two categories: the first one is based on the network connectivity construction strategy, which is aimed at the connection

critical conditions of the target full coverage and maintenance node-set connectivity relationship, when the connectivity conditions cannot be satisfied, an optimized deployment strategy for maintaining connectivity is proposed [2, 3]. A typical representative of this strategy is the deterministic search mechanism and cost-based deployment algorithm proposed by Liu. Dario Pompili proposed an underwater-based three-dimensional sensor network bottle bottom network format algorithm. This strategy fully exploits the advantages of network reconnection and ensures the continuity of network applications when network sensors or nodes are partially lost [3]. However, under this strategy, the network still belongs to the fixed network, and thus the location of the sensor or the node does not change with the change of the external environment, which leads to an increase in the probability of network delay and packet loss rate in such network applications. The second is a construction strategy based on the conventional virtual force algorithm, which is based on the virtual clustering dynamic deployment strategy of nodes and divides the dynamic deployment of traditional virtual power-based nodes into two stages: intercluster deployment and intracluster deployment. The central node is balanced by force, reduces intercluster interference during deployment, and improves node coverage. The typical representative of this strategy is the Dynamic Virtual Force-Based Algorithm (VFE) proposed by Luo and Cai for data aggregation, which can adapt to different scene changes [4, 5]. The energy-saving self-deployment scheme proposed by Han et al. utilizes the attractive force generated by the centroid of the local Voronoi polygon of the sensor and the repulsive force often used in the self-deployment scheme using the potential field [6, 7]. These networks using virtual power ideas have significant improvements in latency and packet loss due to dynamic changes in sensor or node locations. However, due to the dynamic changes in sensor and node locations, the network will perform poorly in terms of connectivity.

Taking into account the advantages of the above two network construction strategies are the original intention of the event-sensitive sensor network construction. The construction of this paper is divided into two parts: (1) constructing a network with massive connectivity. (2) The node redeployment based on virtual force [8, 9].

The specific steps are divided as follows:

- (1) Uniformly dividing the detection space to construct the storage subarea of the node
- (2) Randomly distributing the node position within the storage subarea
- (3) Transmitting the location of the new node to the adjacent node
- (4) Receiving the node modified by the adjacent node information will judge its own overlap
- (5) Correct the overlapping node position based on the virtual force idea and transmit it to the adjacent node
- (6) Repeat (4) and (5) until all node overlap problems are solved

Through these steps of this strategy, we have realized the problem of 3D sensor network node deployment for non-uniform coverage requirements. For two-dimensional sensor networks, three-dimensional sensor networks are more complex and more practical in agriculture. In this strategy, a virtual force-based event-sensitive agricultural sensor network construction mechanism is proposed. This mechanism divides network deployment into two main phases: initial deployment and redeployment. The initial deployment phase is mainly to deploy the network reconnecting idea before the sensor or node stops working. In this stage, the random depth adjustment method is used to make the nodes distribute as evenly as possible to capture more events. The redeployment phase mainly uses the virtual force idea to redeploy after the sensor or node stops working. In this stage, the node uses the virtual force method to move according to the coverage requirement according to the detected event information. In order to reduce the scale of network redeployment, improve the response speed of the network, and ensure the connectivity of the network, this chapter adopts the idea of clustering control and limits the range of node movement to each cluster, avoiding the large-scale movement of nodes. In addition, for the sports event, the corresponding network reconstruction mechanism is also proposed. The main contributions of this paper are summarized as follows:

- (1) Raise the initial deployment method of the network based on random depth adjustment, so that the network nodes are evenly distributed and the number of network event captures increases.
- (2) The network node virtual force redeployment based on event detection is raised. The method enables the network node to perform real-time mobile optimization.
- (3) Raise a clustering control strategy, which is aimed at reducing the corresponding deployment scale, improving the response speed of the network, and ensuring network connectivity through the above two network deployment methods. And avoid the wide range of movement of the node.

2. Related Works

According to the determination of the deployment environment, WSN's sensor deployment strategy is divided into two types: deterministic deployment and nondeterministic deployment. Deterministic methods include both conventional methods and planning methods. This type of approach uses statically deployed sensors to achieve full coverage. This method can also improve the sensing efficiency of coverage with a smaller number of sensors [7]. However, the deterministic strategy of WSN placement (peer-to-peer) can be used for small-scale deployments [10]. Random and grid-based are one of the strategies for nondeterministic processing. In nondeterministic deployments, the sensors are usually scattered, resulting in a random distribution of the sensors, although their density can be controlled to some extent [11, 12]. In applications where access to the area of

interest (RoI) is not available, such as in areas where the environment is dynamically changing, random deployment may sometimes be the only viable option. Logically speaking, random deployment will cause the related performance of WSN to decrease.

During the deployment process, the classification can be divided into network connectivity-based deployment and enhanced virtual force deployment based on the deployment principle. Regarding the problem of deployment and reconstruction in three-dimensional sensor networks, literature [13–15] studied the coverage and connectivity of three-dimensional sensor networks under regular deterministic deployment methods. I hope to achieve the complete coverage and connectivity of the area with the fewest nodes. Xuxun Liu proposed a deterministic search mechanism and a cost-based deployment algorithm in which nodes are assigned to different groups connected by the nearest path and achieve a significant reduction in path length and deployment cost [3]. Dario Pompili proposed an underwater-based three-dimensional sensor network bottom network format algorithm, which has low complexity and low consumption, but at the same time has a high degree of dependence on target coverage [5]. Liu proposed a new deployment method, ACO-Greedy, to solve this problem. This approach is based on ant colony optimization and greedy migration mechanisms, which can quickly complete full coverage and significantly reduce deployment costs [16]. In addition, ACO-Greedy can dynamically adjust the follow-up/communication radius, alleviate energy leaks, and extend the network life cycle. Wan et al. proposed a new metric for routing (expected no-distance transmission time) and further adapts the distance-free transmission to a low duty cycle sensor network [17]. Lee and Shin proposed that the sensor signal is modeled as a Gaussian-based signal point group, and the signal points in the overlapping area between two different sensors are classified into each sensor group by the e-SVM (support vector machine) method. In addition, trilateration techniques are used to estimate the position of moving objects [18]. Zhu et al. proposed an NSLP model that integrates optimal heterogeneous sensor deployment and operational strategies for dynamic O-D demand estimation under budget constraints. The deployment strategy includes the number of links and node sensors and their installation locations. The operational strategy includes the start time and duration of the sensor, which has not been resolved in previous studies [19]. An algorithm was developed to solve the proposed model. Sun et al. proposed a new topology type called uniquely determined topology. In each extension, the edges in the d -dimensional space can be reduced from $d + 1$ to d , which is strict and more suitable for beacon deployment [20]. Another aspect of this paper is to improve the positioning accuracy of deployed beacons.

The above algorithms use conventional ideas, and their method is to deploy and reconfigure the network through the idea of network connectivity. This method is a relatively stable algorithm, which is more suitable in less complex networks. But in a three-dimensional environment, it is not applicable and a new algorithm is needed to solve this problem. In recent years, there have been some researches on

virtual force thinking combined with sensor deployment and reconstruction. Literature [21, 22] designed a virtual force-based three-dimensional spatial redeployment algorithm for randomly deployed sensor networks. The basic idea of the method is to virtualize the sensor nodes into charged particles, and there is a virtual repulsion or gravity between the nodes and the nodes. The node moves to the force balance position according to Newton's law of motion (or the motion rule set by the author) under the action of the external force. In this way, the nodes are spread from the high-density area to the low-density area to form a uniform distribution, and the coverage and connectivity of the network are greatly improved. Different from the above article, the need for uniform coverage of monitoring areas, the literature [23] studied the deployment of underwater sensor nodes under event-driven nonuniform coverage requirements. The events here refer to various dynamic and static targets of interest to users. The article uses artificial fish school thought and node congestion degree control to simulate the fish foraging behavior to make the nodes autonomously close and cover the event, and finally get the ideal deployment effect. However, this method has the disadvantages of long deployment time, no moving nodes, large moving distance, and too many nodes participating in the network redeployment. Frequent movements of large-scale nodes like this lead to drastic changes in network topology and routing, which are very disadvantageous for underwater sensor networks with limited communication, computing, mobility, and energy resources. Besides, the deployment method only focuses on the coverage performance of the event and does not consider the connectivity of the network. Ghowdhury and De proposed an energy-efficient self-deployment scheme that takes advantage of the attractiveness of the centroid of the sensor's local VORONOI polygon and the repulsive force often used in self-deployment scenarios using the potential field [7]. In contrast to existing self-deployment solutions, we also provide the design and implementation of a simulator for analyzing the performance of the proposed method. The simulation results show that our solution can achieve higher coverage and achieve less sensor movement in a shorter time than the self-deployment scheme using the traditional potential field. Hamid et al. proposed an efficient deployment algorithm to increase the coverage of wireless mobile sensor networks [24]. The proposed strategy iteratively calculates new candidate locations for each sensor based on existing coverage holes. For sensors with the same sensing range, these holes are obtained using a VORONOI diagram, and for the case of sensors with different sensing ranges, these holes are obtained using a multiplicative weighted VORONOI (MW-VORONOI) map. Each sensor is driven by a virtual force that is applied to the vertices and boundaries of its VORONOI unit. These forces are obtained in such a way that when the sensor is repositioned, the coverage area of the respective unit increases. Wang et al. proposed a dynamic virtual force-based algorithm (VFE) for data aggregation to adapt to different scene changes. Inspired by the cost domain and virtual power concepts, VFE builds dynamic routing without structural overhead, making data aggregation more efficient [4]. Hamid et al. proposed a fast

hybrid position/virtual force controller to create a virtual force field between the obstacle and the robot [24]. The virtual force is a control parameter that is set to maintain the expected distance between the obstacle and the robot with unknown nonlinearity and parameter uncertainty. Secondly, to reduce the computational burden of parameter learning and enhance the dynamic mapping of network capabilities, the Elman neural network is introduced. The output signal from the hybrid position/virtual force controller is fed back to the Elman neural network. Due to the uncertainty of robot dynamics and obstacle position information, Elman neural network is also used to compensate for uncertainty and improve system stability. Su et al. proposed a data aggregation mechanism with mobile receivers in a wireless sensor network based on virtual force [25, 26]. Firstly, the hierarchical mixture of genetic algorithm and particle swarm optimization algorithm in distributed clustering is introduced, and the whole region is divided into multiple grids. Second, virtual force theory is used to calculate the boundary, the virtual repulsive force of obstacles and blank areas, and the virtual attraction of sensor nodes without being traversed. By combining all of the virtual forces according to the direction vector, the dwell time of the mobile receiver and the coordinates of the next rendezvous point can be calculated based on the magnitude of the force, the direction, and the number of adjacent clusters. Then, get the best motion trajectory of the sink node to achieve high efficiency and energy saving. Therefore, a data aggregation tree can be dynamically built for data collection. Finally, the mobile sink node moves along the rendezvous point to aggregate data from sensor nodes within the communication range based on the selected movement path.

Virtual power is also well used in other areas. For example, Aznar et al. proposed the Virtual Atomic Force Microscopy (VAFM) Algorithm, a Brownian dynamics simulation of a coarse-grained virus model designed to mimic the standard settings of atomic force microscopy (AFM) nanoindentation experiments. Despite the high granularity, these simulations provide valuable information that was not available in the experiment. VAFM will analyze how mechanical response and viral disruption depend on different parameters controlling the effective interaction between capsid structural units, rather than focusing on specific viruses [27]. Wei et al. proposed a new WSN anchor node mobility strategy based on the improved virtual force model. The number of neighbor nodes and the distance between neighbor nodes and anchor nodes are introduced as their own dense weight attributes. Unknown node strength is used as a weight to improve the traditional virtual force model. The number of messages received by the new node is used to calculate the virtual force from the unknown node to the anchor node. Then, according to the virtual force, select the direction and move the anchor node [28].

Through the above literature analysis, in the agricultural wireless sensor network, the idea of using virtual force concerning the deployment of sensors is a vital scientific problem worth studying. The main content is to use the concept of virtual force to create an event-sensitive type.

3. Detailed Design of Deployment Algorithm

3.1. Model Hypothesis and Problem Description. Unless otherwise stated, this paper is based on the following assumptions:

- (1) Deploy n sensor nodes $S = \{s_1, s_2, \dots, s_n\}$ into the cube region to be monitored $V = L_x \times L_y \times L_z$ ($L_x = L_y = L_z$), and the number of nodes is insufficient to cover all regions. That is, $(4\pi n R_s^3 / 3V) < 1$. R_s is the maximum value of the sensing distance of all sensor nodes.
- (2) The sensor node adopts an omnidirectional communication model, and all nodes have the same communication distance.
- (3) The sensing range of the sensor node is a spherical area centered on the node and the sensing distance is a radius, and the sensing distance of each node may be different.
- (4) There are m uncertain events in V , and the set of events is recorded as $E = \{e_1, e_2, \dots, e_m\}$.
- (5) The network adopts a clustered topology, and the nodes are divided into several clusters. The cluster head node of each cluster has an aggregation function, which fuses the data in the cluster and then forwards it to the fixed base station. The member nodes in the cluster send the collected monitoring information to the cluster head node by one-hop or multihop forwarding, and they usually do not directly communicate with the fixed base station.
- (6) The node can be positioned by a positioning device or a positioning algorithm.
- (7) With the aid of robots and other auxiliary means, the node can move accurately within the area and stay stable somewhere for a long time.

The sensor node adopts a Boolean sensing model, that is, the node can only cover events within its perceived radius. Then, the event e_i at $(a_i, b_i, \text{ and } c_i)$ in the three-dimensional space is covered by the probability function of the sensor node s_i located at $(x_i, y_i, \text{ and } z_i)$ as follows:

$$f(e_i, s_i) = \begin{cases} 1, & \sqrt{(x_i - a_i)^2 + (y_i - b_i)^2 + (z_i - c_i)^2} \leq r_{si}, \\ 0, & \sqrt{(x_i - a_i)^2 + (y_i - b_i)^2 + (z_i - c_i)^2} \geq r_{si}. \end{cases} \quad (1)$$

The r_{si} is the perceived radius of the node s_i . The coverage of the event e_i in the network is defined as

$$D(e_i) = \sum_{S_j \in S} f(e_i, s_j). \quad (2)$$

The deployment problem in this paper can be described as follows: under the premise of ensuring network connectivity, the event can reduce the change of the network topology and shorten the moving distance of the node while obtaining the desired coverage.

3.2. *Initial Deployment.* In the initial deployment phase of the network, in order to be able to detect all events in the monitoring area, seamless coverage of the monitoring area is required. Since the sensor node's sensing range is spherical, the sphere does not have the characteristics of independently filling the three-dimensional space and cannot form a seamless and nonoverlapping connection. Therefore, it is generally considered to adopt other regular polyhedrons with independent filling three-dimensional properties, and the nodes are, respectively, deployed at the center of these ordered stacked polyhedrons to achieve complete coverage of the three-dimensional space. Under the condition that the monitoring area is completely covered and the adjacent nodes can communicate the literature [15] states. The r_c is the node communication radius, and r_s is the node perceived radius.

- (1) When $0 < (r_c/r_s) \leq 0.99$, the number of nodes required for the positive triangular prism is the least
- (2) When $0.99 < (r_c/r_s) \leq 1.211$, the cube requires the fewest number of nodes
- (3) When $1.211 < (r_c/r_s) \leq 1.414$, the number of nodes required for the regular hexagonal prism is the least

- (4) When $1.414 < (r_c/r_s) \leq 1.587$, the number of nodes required for the regular hexagonal prism and the rhombohedra dodecahedron is the same and the least
- (5) When $(r_c/r_s) \geq 1.587$, the number of nodes required for truncated octahedron is the least

Using the above conclusions, the optimal position of the node can be conveniently determined. However, this conclusion only applies to homogeneous sensor networks and does not apply to sensor networks with heterogeneous perception capabilities in this chapter. Moreover, for some exceptional cases, such as sparseness of sensor networks in underwater and space, there are usually fewer nodes deployed and often cannot be fully covered. Therefore, this chapter proposes a distributed method based on random depth adjustment to complete the initial deployment of the network. The steps are as follows:

- (1) The top surface $L_x \times L_y$ of the three-dimensional area is evenly divided into n small grids of equal area, and the nodes are placed to the center points of the respective grids. When $L_x = L_y$, the plane is square rasterized, then the coordinates of the placement position of the node s_i are as follows:

$$(x_i, y_i, z_i) = \left(\frac{lx}{2\sqrt{n}} + \frac{lx}{\sqrt{n}} \times \text{mod}(i-1, \sqrt{n}), \frac{ly}{2\sqrt{n}} + \frac{ly}{\sqrt{n}} \times \left\lfloor \frac{i-1}{\sqrt{n}} \right\rfloor, 0 \right). \quad (3)$$

The mod (\cdot) function is a reminder function.

- (2) The node randomly selects its own depth in the region (relative to the top surface) $L_z \times \text{rand}(i)$, rand represents the random factor between (0, 1), and then

sends its new location to the communication neighbor. The position coordinates of the node s_i are as follows:

$$(x_i, y_i, z_i) = \left(\frac{lx}{2\sqrt{n}} + \frac{lx}{\sqrt{n}} \times \text{mod}(i-1, \sqrt{n}), \frac{ly}{2\sqrt{n}} + \frac{ly}{\sqrt{n}} \times \left\lfloor \frac{i-1}{\sqrt{n}} \right\rfloor, L_z \times \text{rand}(i) \right), \quad (4)$$

$$d(i, j) < r_{si} + r_{sj}. \quad (5)$$

- (3) In the initial deployment, in order to detect as many events as possible, the nodes should be spread out to reduce the overlap of the sensing range between nodes. To this end, after receiving the location information of the neighboring nodes, each node calculates its own distance from all communication neighboring nodes and finds a node that overlaps with its own sensing range according to formula (5). If there is an overlapping node, the node will randomly move a distance away from the overlapping node after transmitting a distant message to inform the neighboring node and rebroadcast its own location information. To avoid simultaneous movement of two adjacent nodes, the node waits for a

random delay $\tau_i = T_d \times \text{rand}$ before leaving. If no distant message is received from the overlapping node within this time delay, the node will send a message away from the message and then leave. Otherwise, the node is stationary.

In the formula (5), the $d(i, j)$ represents the Euclidean distance between nodes i and j .

In the worst case (that is, all nodes can communicate with one-hop), the complexity of sending a message by a single node in the above initial deployment method is $O(1)$, and the complexity of receiving a message is $O(n)$. The complexity is only $O(n)$, and the complexity of accepting messages is only $O(n^2)$.

3.3. Network Clustering. In [29, 30], the authors proposed a clustering method based on a two-dimensional virtual mesh model. The basic idea is to divide the plane to be monitored into a number of two-dimensional square grids of the same size. The nodes in each grid form a cluster, and one node is selected as the cluster head. The cluster head can communicate with adjacent clusters. The article also pointed out that if each grid has a cluster head, then the connectivity and coverage performance of the network can be guaranteed.

The above idea of clustering control is introduced into the three-dimensional sensor network, and the area to be monitored is divided into a plurality of cubes of size $l \times l \times l$. We select the node with the most energy remaining in each cube as the cluster head and control the remaining nodes in the cube. And assign each cluster (cube) a unique ID number. If a node is on the common boundary of multiple cubes, it is placed in any of these cubes. In order to balance the energy load between nodes and extend the network life cycle, the cluster head can be rotated periodically by the nodes in the cluster. Record the communication distance of the node as t . As shown in the example in Figure 1, when $r_c \geq \sqrt{6}l$, it can be ensured that in the worst case, each cluster head can communicate with the cluster heads of the adjacent six clusters; when $r_c \geq \sqrt{3}l$, it can ensure that one-hop communication can be realized between nodes in the cluster. Obviously, to ensure connectivity between clusters and clusters, there should be at least one node in each cube, and at the same time, the following formula should be satisfied:

$$l \leq \frac{r_c}{\sqrt{6}}. \quad (6)$$

In order to confirm whether there is at least one node in each cube after the clustering ends, each cluster head sends a confirmation message to the surrounding six clusters. If there is a confirmation message of a cluster, it is considered that there is no cube in the cube. Node, the cluster head will assign a member node within the cluster to move to the cube and become its cluster head. If there are too many cluster heads missing in the network, i.e., $n < ((Lx = Ly = Lz)/l)^3$, then it must be solved by increasing the number of initial deployment nodes n .

In order to ensure that the network can remain connected without being affected after some sensor nodes fail, this chapter also expects the network to have heavy connectivity. The so-called reconnectedness means that there are at least two mutually exclusive connected paths between any two nodes in the network. Removing any node network from a network with heavy connectivity is still connected.

3.4. Redeployment Based on Virtual Force. With the idea of virtual force, it is assumed here that the sensor node can be subjected to forces applied by two different types of objects in the monitored area: event force and boundary force. Each sensor node moves the corresponding distance according to the magnitude and direction of the resultant force until the force balance point is reached. The two forces are specifically analyzed.

3.4.1. Event Force. When the event is not detected by any sensor node, it does not exert a force on the node. Once detected, gravity or repulsion can be generated on the surrounding nodes. Gravity is applied to the nodes in order to gather more nodes and increase the coverage of the events. The repulsion is applied to avoid the node being too dense and causing waste of resources.

In order to reduce the impact of network redeployment on routing and topology, we limit the redeployment of nodes to each cluster, that is, an event only exerts a force on nodes in a cluster.

Specifically, after detecting an event, the network first determines which cube region the event is located with and the ID number of the cluster to which it belongs, and then broadcasts an event message notifying the nodes in the cluster.

As shown in Table 1, the event message is mainly composed of four parts. Only the node whose cluster ID number matches will retain this message and move accordingly.

In order to ensure the monitoring of the event, it is generally assumed that the event to be monitored needs to have a certain degree of coverage expectation \bar{D} . When the coverage of the event e_j is insufficient, that is, $D(e_j) < \bar{D}$, the event e_j , the force on the node s_i is expressed as gravity, as in the following formula:

$$F_{e_j}(s_i) = \begin{cases} \left(\frac{k_e}{d(e_j, s_i)}, a_{se} \right), & d(e_j, s_i) > r_s, \\ 0, & d(e_j, s_i) \leq r_s. \end{cases} \quad (7)$$

When the coverage of the event e_j is too high, that is, $D(e_j) < \bar{D}$, the force of the event e_j on the node s_i is expressed as a repulsive force, as shown in the following formula:

$$F_{e_j}(s_i) = \begin{cases} 0, & d(e_j, s_i) > r_s, \\ \left(\frac{k_e}{d(e_j, s_i)}, a_{se} \right), & d(e_j, s_i) \leq r_s. \end{cases} \quad (8)$$

In formulae (7) and (8), $d(e_j, s_i)$ represents the distance between e_j and s_i ; k_e and k'_e , respectively, represent the gravitational coefficient and repulsion coefficient of the event-to-node; a_{se} and a_{es} represent, respectively, the direction of the force is indicated by the node pointing event and the unit vector pointed to by the event, indicating the direction of the force.

3.4.2. Boundary Force. After the network is clustered, there may be nodes moving to a place other than the small cube where the cluster belongs. In order to ensure the reconnectedness of the network, the boundary force is introduced to constrain the range of motion of the node. The single-step movable distance of the node is recorded as Δd , which indicates the maximum distance that the node can move

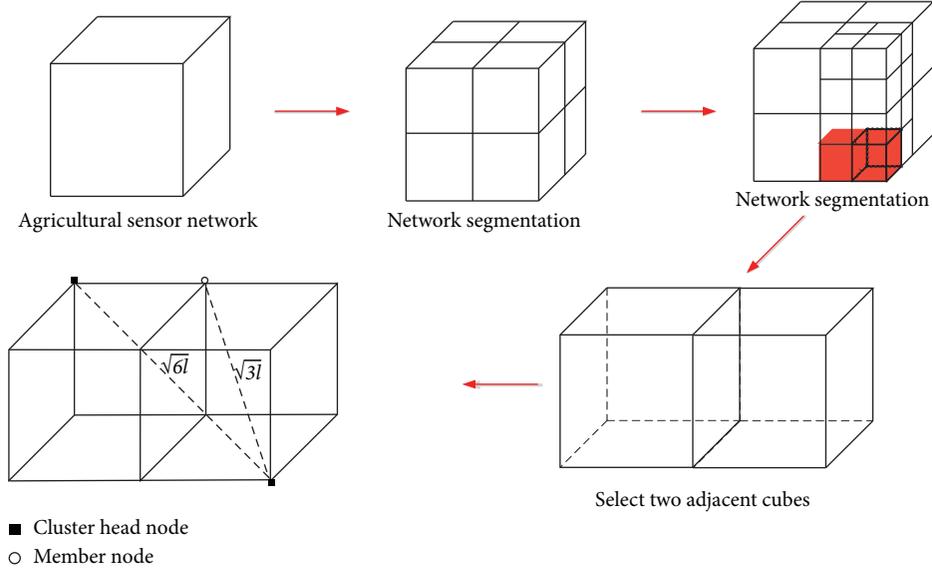


FIGURE 1: Three-dimensional mesh model.

TABLE 1: Event model components.

Time ID	Cluster-ID	Event location	Event coverage
---------	------------	----------------	----------------

during a move. To prevent the node from moving outside the cube under the influence of other effects, it is assumed that when the node is smaller than Δd from the cube, it will receive the repulsion applied to it by the boundary. And the closer to the boundary, the greater the repulsive force should be. The repulsive force of the cube boundary to the node S_i can be expressed as

$$F_b(s_i) = \begin{cases} 0, & d(b, s_i) \geq \Delta d, \\ \left(\frac{k_b}{d(b, s_i)}, a_{bi} \right), & d(b, s_i) < \Delta d. \end{cases} \quad (9)$$

In formula (9), $d(b, s_i)$ represents the vertical distance of s_i to the boundary; k_b represents the boundary repulsion coefficient; a_{bi} is the unit vector of the node pointed to by the boundary, indicating the direction of the repulsive force.

3.4.3. Motion Rules. It can be seen from the above analysis that the resultant force of the node s_i is the vector sum of the event in the monitoring area and the repulsive force of the small cube boundary between them, which can be expressed as

$$F(s_i) = \sum_{j=1}^m F_{ei}(s_i) + F_b(s_i), \quad s_i \in S. \quad (10)$$

The direction of motion of the node s_i is the direction of the virtual external force as above. Different from the traditional virtual force method, the force magnitude is directly related to the motion distance. We specify that the node is fine-tuned each time with a small fixed step size $\Delta d (\Delta d \ll l)$. Adjacent nodes exchange their respective position information. In the time synchronization mode, the

force analysis is re-executed after waiting for the time, and finally, the method is adjusted to the equilibrium position by multiple fine adjustment methods.

The following describes how to set four important parameters in the motion model: the gravitational coefficient k_e of the event-to-node, the repulsion coefficient of the event-to-node k'_e , the boundary repulsion coefficient k_b , and the fixed step Δd of the node movement. On the one hand, because the ultimate goal of node redeployment is to improve network coverage performance and maximize event coverage, the attraction of the event to the node should be considered first, so the value of k_e should be set to the largest of the three force coefficients. On the other hand, in order to ensure that the motion of the node is always located in the area V to be monitored, $k_b \geq k'_e$. When considering the value of Δd , if it is set too large, the single moving distance of the node will be too large, causing the node to reciprocate back and forth. Setting it too small will increase the number of node movements and increase the communication energy consumption. Therefore, the value of Δd will need to be determined through trial and error. When the number of nodes satisfies $(4\pi R_s^3/3V) < 1$, the parameter is preferably 10% to 30% of the sensor's perceived distance.

By adopting the above motion rules, on the one hand, the motion range of the node can be limited to the original cube, and the original cluster structure of the network is not changed, thereby maintaining the connectivity of the network. On the other hand, the calculation process of the node is greatly simplified by replacing the complex equation of motion with a simple fixed step size. When the network is nearing stability, the force threshold f_{th} is set in order to prevent the node from oscillating near the equilibrium position and cannot be stationary. When the resultant force is less than f_{th} , it is considered that the node has reached the state of stress balance, and the motion can be stopped. The single motion distance $Once_d(s_i)$ of the node s_i can be expressed in the following formula:

$$\text{Once_}d(s_i) = \begin{cases} 0, & F(s_i) > f_{\text{th}}, \\ \Delta d, & F(s_i) \leq f_{\text{th}}. \end{cases} \quad (11)$$

Obviously, the smaller the f_{th} is, the better the final coverage is, but this will increase the deployment time of the network and reduce the convergence speed of the network. It is usually set to a value that is small relative to the resultant virtual force.

3.5. Algorithm Description. Next, based on the above three-dimensional sensor network deployment method, a complete algorithm is given. When all nodes are placed on the top surface according to formula (11), the algorithm starts running. First, in the initial deployment process, each node performs random depth adjustment, sends the respective location information to the neighboring communication node, and then waits for a period of time T to acquire the location information of the neighboring node. After the T time, each node judges whether it overlaps with the sensing range of other nodes. If there is an overlap, the random delay τ_i , then move backward away from the overlapping node for a certain distance and reissue the position information; otherwise, it does not move. Subsequently, the initial deployment phase ends, the neighboring nodes exchange energy information and select the cluster head nodes of the respective clusters. After the clustering ends, the node enters the redeployment phase, and each node moves to the final position by multiple adjustments according to formula (10). In order to ensure that the algorithm can be terminated, a maximum value C_{max} is set for the number of motions of the node. When the number of motions of the node exceeds the value, the algorithm ends. In order to make the node as close as possible or even reach equilibrium, it is necessary to set the parameter C_{max} reasonably. This chapter will set it to 10. Of course, it can also be adjusted according to the specific situation. The pseudocode for this algorithm is shown in Algorithm 1.

In the algorithm, the indication function $\text{Overlap}(s_i)$ is true to indicate that s_i overlaps with other nodes, and the number $\text{Clustering}(s_i)$ implements clustering on the cube where the node s_i is located. The variables Location_{s_i} , Energy_{s_i} , and Event_{s_i} represent the location, energy, and detected events of s_i , respectively.

4. The Mechanism of 3D Sensor Network Reconstruction

In actual situations, the events monitored by the sensor network may be relatively static or dynamic. For example, in the application of a seabed mineral resources survey, the event is static; in the application of marine water quality monitoring, the event to be monitored, that is, the pollutant may drift with the movement of seawater. For example, the monitoring of space junk, space junk is also free to fly in the air. In the case of dynamic changes, the sensor network and the events to be monitored are likely to be continually changing. However, it is not realistic to let the sensor node update the event information all the time and perform new

force analysis. Therefore, it is necessary to reanalyze the stress of the node according to a certain period. We divide the network runtime into a number of equally spaced time periods, called rounds. Each round consists of a decision phase and an operational phase. In the decision-making stage, a new evaluation of the event monitoring situation is carried out, and the node reconfiguration scheduling is implemented. In the operational phase, there are nodes that work according to the scheduling results of the decision phase until the end of the round. In addition, in order to save network energy, a member node in a cluster that does not have a monitoring task in a certain round can enter a sleep state until the start of the next round. However, in order to ensure network connectivity, the cluster head node must remain active at all times. Figure 2 is a flow chart of the reconstruction mechanism.

5. Simulation Analyses

In this section, the above deployment methods are simulated and verified, and the performance of the virtual force-based redeployment algorithm in this chapter is compared with the FSSD algorithm in [23] under the same simulation conditions. The simulations in this chapter are based on the MATLAB2010b platform and run under a computer environment of 2.9 GHz CPU and 512 M memory. In the virtual force-based redeployment algorithm, parameters $k_e = 15$, $k'_e = 10$, $k_b = 10$, $f_{\text{th}} = 0.2$, and $C_{\text{max}} = 10$ are set. In the FSSD algorithm, the maximum number of foraging for a single node is set to 3, and the maximum number of moves is 30. In order to better reflect the statistical laws, all results are the average of multiple simulation results.

5.1. Simulation Example. In the simulation of Figure 3, the volume of the monitoring area is $20 \text{ m} \times 20 \text{ m} \times 20 \text{ m}$. In order to clearly observe the deployment effect, the parameters are set as follows: a total of 4 sensor nodes are deployed, the communication distance of the node is $r_c = 20\sqrt{3}$, the sensing radius is $r_s = 6 \text{ m}$, the node single moving step is $\Delta d = 2 \text{ m}$, the size of the cluster $l = 20$, that is, the network has only one cluster, and the network can be regarded as one cluster on a large-scale network. Figure 4(a) shows the initial deployment of the network using the initial deployment algorithm of this chapter. Four sensor nodes are randomly distributed at different depth values: Figure 4(b) shows that there are 30 events randomly distributed in the monitoring area. When the coverage expectation $\bar{D} = 2$, the node is based on the redeployment of the virtual force. In the figure, the grid sphere represents the three-dimensional sensing range of the node, and the black circle represents the event. At this time, only two events in the area have a coverage of 0, 1 event has a coverage of 1, and the remaining 27 events have a coverage of not less than 2, that is, 90% of the events satisfy the coverage expectation.

In the field deployment, there are five types of sensor nodes as shown in Figure 3. Monitoring indicators include soil temperature and humidity, EC value, PH value, and farmland weather information.

```

Algorithm 3D_Deployment( $s_i$ , Location_ $s_i$ ).
//Location_ $s_i$  is the node  $S$  in the initial deployment. Coordinates on the top surface
move to random depth using (formula (4))
broadcast Location_
wait for time  $T$ 
if isOverlap( $s_i$ ) == true then
    wait for time  $\tau_i$ 
    make an small arbitrary move
    broadcast Location_ $s_i$ 
endif
broadcast Energy_ $s_i$ 
execute Clustering ( $s_i$ ) using (formula (6))
broadcast Event_ $s_i$ 
Counter = 0
while counter  $\leq C_{max}$  do
    calculate total force  $F(s_i)$  using (formula (10))
    if  $F(s_i) \geq f_{th}$  then
        move to new position
        broadcast Location_ $s_i$ ,
        broadcast Event_ $s_i$ ,
    end if
    counter ++
    wait for time  $\nabla t$ 
end while
    
```

ALGORITHM 1: The 3D sensor deployment algorithm.

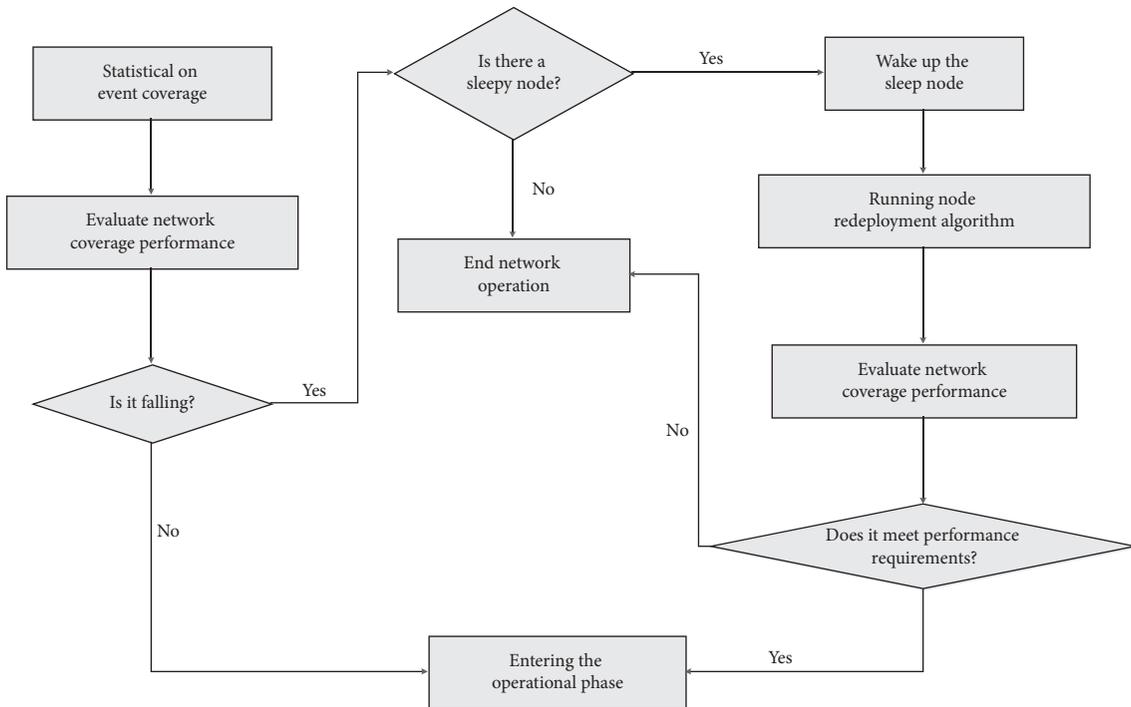


FIGURE 2: The flow chart of network reconfiguration mechanism.

The sensor deployment architecture is shown in Figure 5. In Figure 5, sensor nodes form a monitoring area, which is connected to the Internet through a gateway. The user accesses the server through the terminal device to obtain the relevant data of the sensor node.

5.2. Comparison of Event Coverage. In the simulation of Figure 4, the volume of the monitoring area is $100\text{ m} \times 100\text{ m} \times 100\text{ m}$, and there are 300 events randomly distributed, that is $r_c = 20\sqrt{6}\text{ m}$, $r_s = 6\text{ m}$, $\Delta d = 2\text{ m}$, and $l = 20\text{ m}$. Figure 4 shows the average coverage of events and



FIGURE 3: Sensor node type.

the nodes in the network obtained by the remake algorithm and the algorithms FSSD, EPPDG [31], and MMSM [32] in the same initial deployment scenario. The relationship among the numbers. The number of nodes is $n = 100, 200, 300, 400, 500, 600,$ and 700 .

It can be seen from the Figure 6 that as the number of nodes increases, the average coverage of the two algorithms increases, and the average coverage of the FSSD algorithm increases faster than the algorithm in this chapter. This is mainly because the redeployment of the algorithm in this

chapter is carried out in each cluster, and the range of free motion of the node is much smaller than that of the FSSD algorithm, thus affecting the network coverage performance. In addition, when the number of nodes reaches 700, the average coverage of the algorithm in this chapter remains near the coverage expectation 1, but the average coverage of the FSSD algorithm has far exceeded the coverage expectation, reaching about 1.4, which is 40% higher than the expected value. This is because the aggregation effect of events on the nodes in the FSSD algorithm is very obvious,

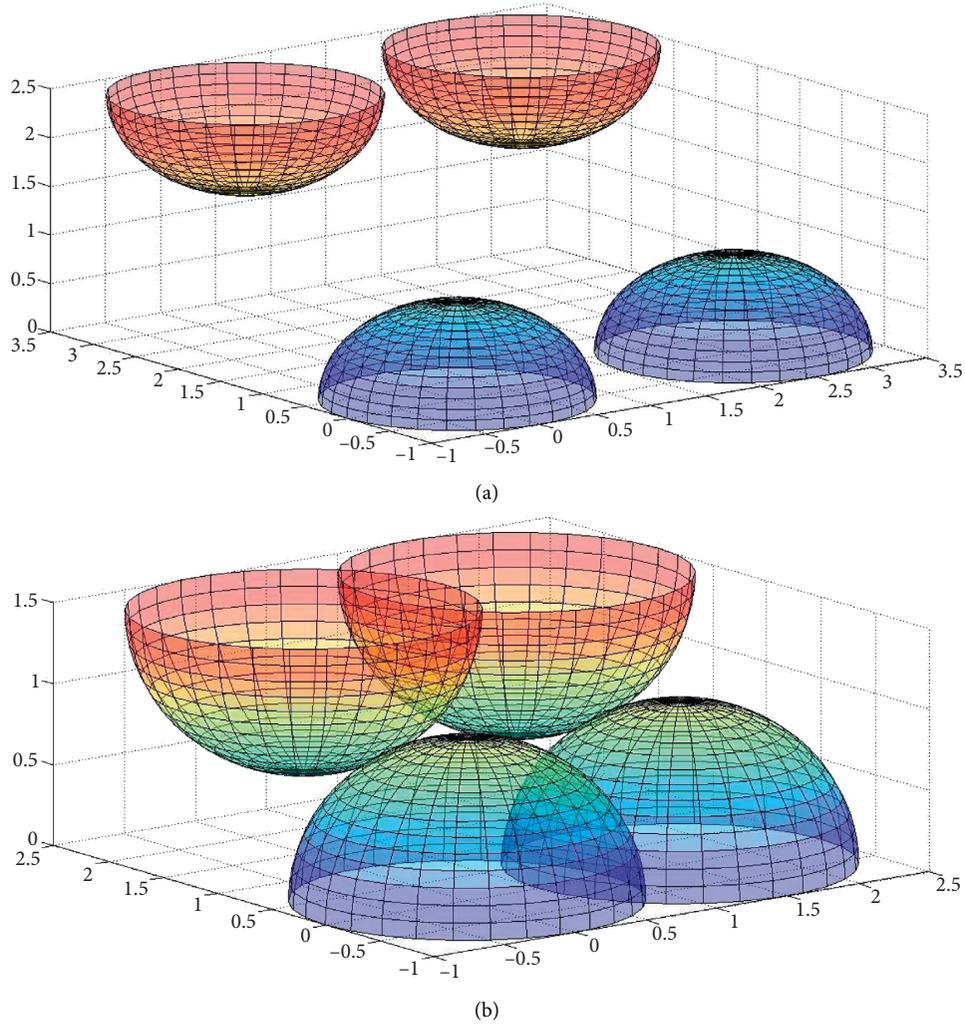


FIGURE 4: Simulation Example Renderings. (a) Initial deployment. (b) Redeployment.

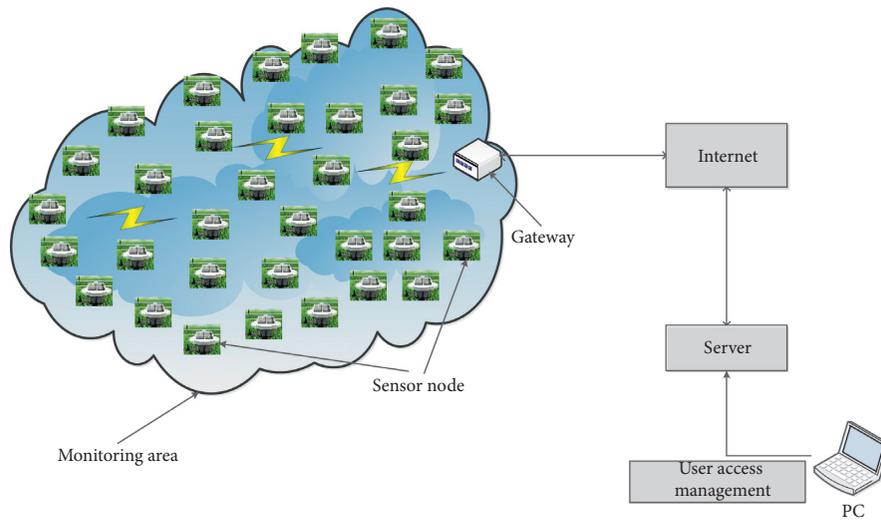


FIGURE 5: Sensor deployment architecture diagram.

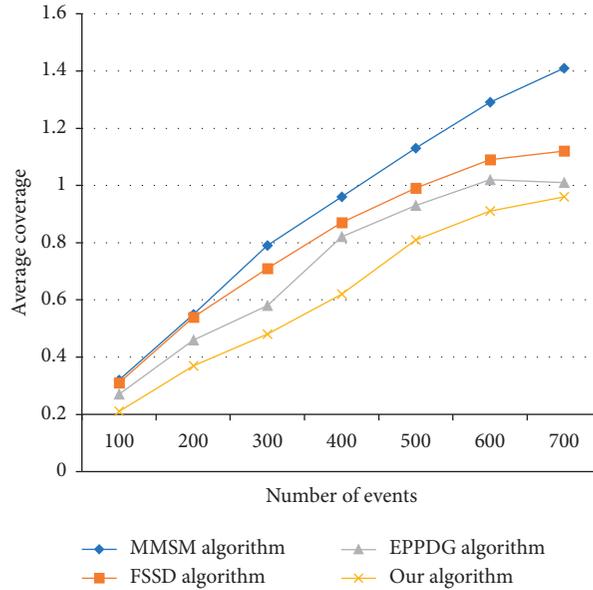


FIGURE 6: The relationship between average coverage and number of nodes.

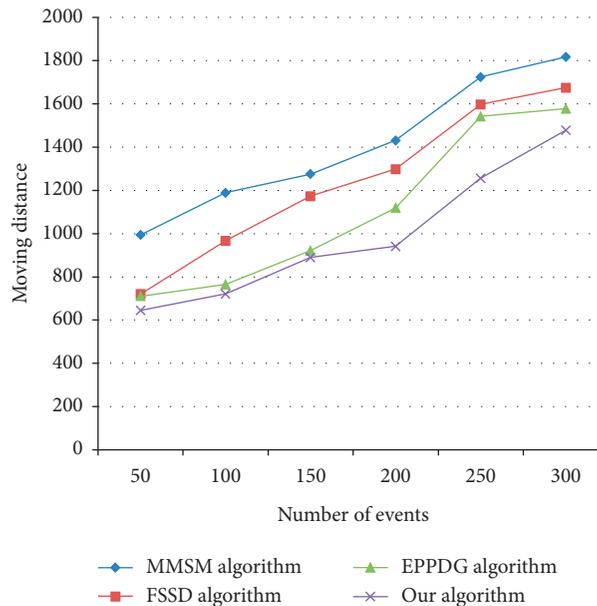


FIGURE 7: The relationship between the moving distance and the number of events.

but its congestion control strategy is loose, which eventually leads to a large number of redundant nodes and repeated coverage. In the algorithm of this chapter, if the coverage of the event exceeds, it is expected that the event will generate a repulsive force on the node to reduce coverage and save network resources. Therefore, in the case of a wide dispersion of events and insufficient number of nodes, the method of this paper can capture and cover more events.

5.3. Comparison of Node Moving Distance. In the simulation of Figure 6, the volume of the monitoring area is $100\text{ m} \times 100\text{ m} \times 100\text{ m}$, and 250 nodes are deployed, r_c

$= 20\sqrt{6}\text{ m}$, $r_s = 6\text{ m}$, $\Delta d = 2\text{ m}$, and $l = 20\text{ m}$. Figure 7 shows the relationship between the moving distance and the number of events obtained by the remake algorithm and FSSD, EPPDG [31], and MMSM [32] algorithms in the same initial deployment scenario when the event coverage is expected to be $\bar{D} = 2$. The number of events is $m = 50, 100, 150, 200, 250,$ and 300 . It can be seen that the moving distances of the two algorithms increase with the increase of the number of events, and the moving distance of the algorithm in this chapter is always smaller than the FSSD algorithm. This is mainly because, compared with the algorithm in this chapter, there is no information exchange between the nodes of the FSSD

algorithm, and the node moves in the process of finding the event (i.e., the foraging process). This is more obvious when the number of events is less: when the number of events is 300, the moving distance of the algorithm in this chapter is 10.1% less than other algorithms. When the number of events is 50, the moving distance of the algorithm in this chapter is less than other algorithms.

6. Conclusions

This paper introduces the virtual force into the construction mechanism of agricultural wireless sensor network and divides the network deployment into two stages: initial deployment and redeployment. Experiments show that the mechanism is suitable for agricultural wireless sensor networks driven by environmental changes. In this mechanism, the virtual force theory is used to calculate the boundary, the virtual repulsive force of the obstacle, and the blank area, and the virtual attraction between the sensor nodes are used to achieve the purpose of not being traversed. This mechanism calculates the moving distance of the mobile receiver and the average coverage of the network based on the magnitude of the virtual force, the direction, and the number of adjacent clusters. Based on the analysis of results, an event-sensitive network can be dynamically constructed. Simulation results show that the proposed mechanism can achieve better performance than the existing algorithms in terms of average coverage and moving distance. For future research, we plan to further improve the performance of the data acquisition algorithm by considering huge obstacle avoidance and delay constraints and finally form a set of plans to enhance VFDA.

Data Availability

The data used to support the findings of this research are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors thank generous support from the Tongji University, particularly the Department of Software Engineering. This work was supported in part by the Shanghai Academy of Agricultural Sciences for the Program of Excellent Research Team (2017[B-09]).

References

- [1] N. Saeed and A. Celik, "Underwater optical wireless communications, networking, and localization: a survey," *Ad Hoc Networks*, vol. 94, pp. 1–35, 2019.
- [2] J. Wang, Yu Gao, K. Wang et al., "An affinity propagation-based self-adaptive clustering method for wireless sensor networks," *Sensors*, vol. 19, no. 11, pp. 1–15, 2019.
- [3] X. Liu, "A deployment strategy for multiple types of requirements in wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2364–2376, 2015.
- [4] J. Wang, C. Ju, Yu Gao et al., "A PSO based energy efficient coverage control algorithm for wireless sensor networks," *Computers Materials & Continua*, vol. 56, no. 3, pp. 433–446, 2018.
- [5] Z. Yan, P. Goswami, A. Mukherjee, L. Yang, S. Routray, and G. Palai, "Low-energy PSO-based node positioning in optical wireless sensor networks," *Optik*, vol. 181, pp. 378–382, 2019.
- [6] J. Luo and J. Cai, "A dynamic virtual force-based data aggregation algorithm for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2015, pp. 1–7, 2015.
- [7] A. Ghowdhury and D. De, "FIS-RGSO: dynamic fuzzy inference system based reverse gloworm swarm optimization of energy and coverage in green mobile wireless sensor networks," *Computer Communication*, vol. 163, pp. 12–34, 2020.
- [8] 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.
- [9] R. Beigel, J. Wu, and H. Zheng, "Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs," *Computers, Materials & Continua*, vol. 62, no. 2, pp. 695–711, 2020.
- [10] LD2FA-PSO, "A novel learning dynamic deterministic finite automata with PSO algorithm for secured energy efficient routing in wireless sensor network," *Ad Hoc Networks*, vol. 97, pp. 1–14, 2020.
- [11] V. Bibin Christopher and J. Jasper, "Jellyfish dynamic routing protocol with mobile sink for location privacy and congestion avoidance in wireless sensor networks," *Journal of Systems Architecture*, vol. 7, pp. 145–157, 2020.
- [12] M. Kaur and M.J. Amit, "Data aggregation algorithms for wireless sensor network: a review," *Ad Hoc Networks*, vol. 100, pp. 1–20, 2020.
- [13] S. M. Mohamed, H. S. Hamza, and I. A. Saroit, "Coverage in mobile wireless sensor networks (M-WSN): a survey," *Computer Communications*, vol. 110, no. 110, pp. 133–150, 2017.
- [14] K. Nabar and G. Kadambi, "Affinity propagation-driven distributed clustering approach to tackle greedy heuristics in mobile ad-hoc networks," *Computers & Electrical Engineering*, vol. 71, no. 71, pp. 988–1011, 2018.
- [15] B. Cao, J. Zhao, Z. Lv, X. Liu, X. Kang, and S. Yang, "Deployment optimization for 3D industrial wireless sensor networks based on particle swarm optimizers with distributed parallelism," *Journal of Network and Computer Applications*, vol. 103, pp. 225–238, 2018.
- [16] X. Liu and D. He, "Ant colony optimization with greedy migration mechanism for node deployment in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 39, no. 39, pp. 310–318, 2014.
- [17] D. Wan, Z. Li, J. C. Liando, and Mo Li, "From rateless to distanceless: enabling sparse sensor network deployment in large areas," *IEEE Transactions on Networking*, vol. 24, no. 4, pp. 2498–2511, 2016.
- [18] J.H. Lee and B. S. Shin, "SensDeploy: efficient sensor deployment strategy for realtime localization," *Human-Centric Computing and Information Sciences*, vol. 7, pp. 1–12, 2017.
- [19] S. Zhu, Y. Guo, J. Chen, D. Li, and L. Cheng, "Integrating optimal heterogeneous sensor deployment and operation strategies for dynamic origin-destination demand estimation," *Sensors*, vol. 17, pp. 2–18, 2017.
- [20] D. Sun, V. C. M. Leung, Z. Qian, Y. Liu, B. Ge, and B. Ge, "Beacon deployment strategy for guaranteed localization in

- wireless sensor networks,” *Wireless Networks*, vol. 22, no. 6, pp. 1947–1959, 2016.
- [21] C. Liu, Z. Zhao, and W. Qu, “A distributed node deployment algorithm for underwater wireless sensor networks based on virtual forces,” *ISA Transactions*, vol. 91, pp. 262–280, 2019.
- [22] A. C. Santos, C. Duhamel, and L. Silva Belisário, “Heuristics for designing multi-sink clustered WSN topologies,” *Engineering Applications of Artificial Intelligence*, vol. 50, no. 50, pp. 20–31, 2016.
- [23] H. Ferreira Chame, d. S. Matheus Machado, and d. C. B. Silva Silva, “Neural network for black-box fusion of underwater robot localization under unmodeled noise,” *Robotics and Autonomous Systems*, vol. 110, pp. 57–72, 2018.
- [24] M. Hamid, K. Moezzi, G. A. Amir, S.-P. Kamran, and M. Vladimir, “Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 163–174, 2014.
- [25] W. Zheng, H. Wang, Z. Zhang, and X. Lu, “Hybrid position/virtual-force control for obstacle avoidance of wheeled robots using Elman neural network training technique,” *International Journal of Advanced Robotic System*, vol. 14, no. 3, pp. 1–14, 2017.
- [26] S. Su and S. Zhao, “A novel virtual force-based data aggregation mechanism with mobile sink in wireless sensor networks,” *Cluster Computing*, vol. 12, pp. 1–16, 2018.
- [27] M. Aznar, S. Roca-Bonet, and D. Reguera, “Viral nanomechanics with a virtual atomic force microscope,” *Journal of Physics: Condensed Matter*, vol. 30, no. 26, Article ID 264001, 2018.
- [28] L.-S. Wei, H.-B. Wang, and X.-C. Hu, “Research on the single anchor node moving strategy based on the weighted virtual force mode,” *Cluster Computing*, vol. 14, pp. 1–10, 2018.
- [29] A. B. Makar, K. E. McMartin, M. Palese, and T. R. Tephly, “Formate assay in body fluids: application in methanol poisoning,” *Biochemical Medicine*, vol. 13, no. 2, pp. 117–126, 1975.
- [30] M. Aleksandra and G. Liljana, “WSN Coverage & Connectivity Improvement Utilizing Sensors mobility,” in *Proceedings of the 11th European Wireless Conference*, pp. 117–126, Vienna, Austria, July 2011.
- [31] J.-P. Sheu, P. K. Sahoo, C.-H. Su, and W.-K. Hu, “Efficient path planning and data gathering protocols for the wireless sensor network,” *Computer Communications*, vol. 33, no. 3, pp. 398–408, 2010.
- [32] K. Lee, Y.-H. Kim, H.-J. Kim, and S. Han, “A myopic mobile sink migration strategy for maximizing lifetime of wireless sensor networks,” *Wireless Networks*, vol. 20, no. 2, pp. 303–318, 2013.