

Method for the Optimal Sensor Deployment of WSNs in 3D Terrain Based on the DPSOVF Algorithm

YANZHI DU 

School of Optical Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China

e-mail: duyanzhi1@163.com

ABSTRACT Maximizing coverage and maintaining connectivity are two major objectives in designing and deploying wireless sensor networks (WSNs). In this article, a novel approach is proposed to obtain better sensor deployment in three-dimensional (3D) terrain in terms of coverage and connectivity. The proposed approach is based on a combination of the distributed particle swarm optimization (DPSO) algorithm and a proposed 3D virtual force (VF) algorithm. The communication limit (CL) of the sensor nodes (SNs) is taken into consideration. A heuristic algorithm that is suitable for a limited communication environment is proposed to cluster the SNs. To effectively guide the SN deployment optimization and to speed up convergence, the addition of a VF term in the velocity update equation of each particle is adopted. To improve the efficiency of increase of coverage and connectivity of the WSN, the selected redundant sensor nodes (RSNs) move toward certain selected areas. In addition, measures are taken to guarantee population diversity in the early stage and the convergence speed of the proposed algorithm in a later stage. Finally, to verify the effectiveness of the proposed algorithm, some comparative experiments are performed. The experimental results show that the proposed algorithm performs well against other algorithms in deploying sensors in 3D terrain when considering CL.

INDEX TERMS Sensor deployment, coverage, connectivity, communication limit, DPSO, VF.


I. INTRODUCTION

Due to their important roles in transportation, infrastructure and monitoring [1], [2], WSNs are widely used and attract much attention. Among the characteristics of WSNs, such as coverage, connectivity, cost and lifetime, coverage and connectivity are the most important characteristics that directly affect the performance of the whole network. Maximizing network coverage is necessary for a WSN to be able to sense information at every location in the region of interest (ROI). Data gathered from sensors are usually sent to a base station (BS). The BS is often connected to a network backbone. Maintaining connectivity is crucial for a BS to continuously gather data from SNs. In addition, sensors are usually battery powered. It is inconvenient to replace batteries in practical applications [3]. Therefore, the energy of the sensors is limited, and the sensor will die if its initial energy is exhausted. To extend the network lifetime, reducing energy

consumption in the WSN during sensor deployment is also an important issue.

A. RELATED WORK

In recent years, many studies have been performed on SN deployment in WSNs [4]–[6]. Generally, SN deployment methods can be divided into random strategies and deterministic strategies. In random strategies, massive SNs are randomly deployed, and they remain static after the initial deployment [7]. Random deployment may cause sensors to be centralized or blocked by terrain features. Some areas of the ROI may have holes in coverage [8], [9] or serious coverage overlaps, which significantly decrease the probability of detection and sensing in the environment. Mobile sensor nodes (MSNs) were used to provide self-deployment for WSNs in [10], [11]. WSNs in practice are always hybrid, and consist of both MSNs and static sensor nodes (SSNs) to reduce cost and energy consumption [12]. This was intuitively inspired by the fact that MSNs should be moved

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

from an initial uneven distribution to an even distribution to increase coverage [13]–[15].

In deterministic strategies, sensors are deployed according to a predefined constraint to maximize the coverage. In [16], each sensor was placed in the middle of a Delaunay triangulation (DT) or in the middle of Voronoi polygons of the sensor coordinates. In [17], the target region was divided into subgrids. SNs resided on the vertex of each grid. The blind nodes were determined by comparing their received signal strength indicator (RSSI) values to deploy the minimum number of sensors. A divide-and-conquer algorithm based on a DT method was proposed in [18]. Although it could improve network efficiency, the method failed to guarantee an optimal solution. For deterministic strategies, the fatal disadvantage is that the limit in the number of sensors will lead to holes in coverage. In addition, the decision space of the deployment problem increases exponentially with the expansion of the terrain.

In recent years, heuristic algorithms have been used to perform sensor deployment. The PSO algorithm [19], [20] is one of the most popular algorithms and is frequently utilized to solve the sensor deployment problem of WSNs [5], [21]–[23]. A parallel PSO (PPSO) was proposed in [24], which divided the ROI and the SNs into several parts. It was especially suitable when there were a large number of SNs to be deployed. The dimension of the searching space was partitioned to save time. The discrete PSO algorithm was used to handle sensor deployment in a nonconvex region [25]. Because PSOs have the disadvantage of being premature and easily falling into local optima, to deploy the sensors effectively, the combination of the PSO algorithm and VF algorithm was used to overcome these shortcomings [6], [26].

Most of the literature referred to above concerns sensor deployment problems in 2D terrain. However, they are not suitable for 3D terrain, which is more realistic. The problem of sensor deployment on 3D terrain was proven to be an NP-hard problem [27], and solving this problem is more challenging than solving the problem of sensor deployment on 2D terrain. Some studies have addressed that address the problem of sensor deployment on 3D terrain [28]–[32]. Wavelet transform (WT) and genetic algorithms (GAs) were used to perform sensor deployment [28]. WT and the cat swarm optimization (CSO) algorithm were used to improve the coverage of a WSN [29].

B. CONTRIBUTIONS

In this article, we propose a novel approach based on the combination of the DPSO algorithm and a proposed 3D VF algorithm to dynamically deploy the SNs of a WSN in 3D terrain. Compared with the existing methods, the contributions of this study can be summarized as follows: (1) the proposed sensor deployment algorithm takes the CL of the SNs into consideration, which has not yet been seriously considered; (2) we present a probability sensing model that is suitable for SNs in 3D terrain; (3) we propose a 3D VF algorithm; (4) the proposed sensor deployment algorithm can improve

the coverage and guarantee the connectivity of the WSN at the same time; (5) we introduce a relocation manner for selected RSNs to help more efficiently eliminate coverage holes and enhance connectivity of the WSN; (6) we take energy consumption into consideration and apply measures to extend the WSN lifetime; (7) we consider the population diversity and convergence speed of the proposed approach; (8) the proposed algorithm needs no central unit.

The rest of this article is organized as follows: Section II reviews the PSO considering CL. Section III analyzes some existing methods of sensor deployment. The proposed approach is described in Section IV. The probability sensing model is proposed in Subsection IV-A. The 3D VF algorithm is proposed in Subsection IV-B. The proposed sensor deployment algorithm is explained in detail in Subsection IV-C. Section V presents the experiments and results analysis. Finally, Section VI concludes this article.

II. PSO CONSIDERING CL

The DPSO algorithm was proposed to apply the PSO algorithm in a real physical environment [33]. The update manner of the velocities and positions of particles is similar to that of the PSO algorithm. In the DPSO algorithm, the velocity and position of the i th particle (par_i) are updated according to:

$$v_{ij}(t+1) = \omega \cdot v_{ij}(t) + c_1 \cdot r_1 \cdot (x_{ij}^L(t) - x_{ij}(t)) + c_2 \cdot r_2 \cdot (x^G(t) - x_{ij}(t)) \quad (1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (2)$$

where ω is the inertial weight, which can be used to balance the exploration and exploitation search abilities of PSO [34]; i denotes the particle's index; $j \in (1, \dots, d)$ represents the dimension; t is the iteration number; $x_{ij}^L(t)$ is the history best position experienced by the i th particle at t ; $x^G(t)$ is the global best position of all particles at t ; c_1 is the self-cognitive factor which reflects the effect of the history best position on velocity; c_2 is the social-cognitive factor which reflects the effect of the global best position on velocity, proper selections of c_1 and c_2 are of considerable help in the performance of the DPSO algorithm; r_1 and r_2 represent random numbers which are generated within the range of $[0, 1]$; $v_i = [v_{i1}, \dots, v_{id}]$ and $x_i = [x_{i1}, \dots, x_{id}]$ are the velocity and position vectors of the i th particle, respectively. The history best position of the i th particle is updated based on:

$$x_{ij}^L(t+1) = \begin{cases} x_{ij}^L(t), & \text{if } f(x_{ij}^L(t)) \geq f(x_{ij}(t+1)) \\ x_{ij}(t+1), & \text{otherwise} \end{cases} \quad (3)$$

where f denotes the objective function. The global best position is updated based on:

$$x^G(t) = \operatorname{argmax}\{f(x_i^L(t))\} \quad \text{for } i = 1, \dots, N_p \quad (4)$$

where N_p is the number of particles.

In a WSN, SNs communicate with each other within a certain reception distance. This distance is called the communication range (CR) of an SN. Due to the energy limitation,

the CR of an SN is limited. Generally, the value of the CR is no more than 300 meters. The CL considered in this article is that data cannot be transmitted between SNs that are outside of the CR.

Considering the CL, the value of the global best position (x^G) in the update equation of a particle may actually be a local known best position (x^K). Losing information from the particles that are outside of the CR of the other particles leads to poor search cooperation among particles.

The main difference between the DPSO algorithm and PSO algorithm is that it is assumed that all particles can communicate with each other in the PSO algorithm, while this cannot be guaranteed in the DPSO algorithm due to the CL of SNs in a real physical environment. Thus, the global best position cannot be precisely known by all particles in the DPSO algorithm, which greatly affects the updates of the particles.

An extension of the DPSO algorithm considering a CL was studied in [35]. All particles transmitted their newly found global best position to a server one after another as time went on. A communication term was added to the velocity update equation of each particle. For the i th particle, the velocity update equation can be written as:

$$v_i(t+1) = \left(1 - \min\left(1, \left\lfloor \frac{t-t_c}{t_m + \theta_i} \right\rfloor\right)\right) r_1 \cdot (x_i^L(t) - x_i(t)) + \left(1 - \min\left(1, \left\lfloor \frac{t-t_c}{t_m + \theta_i} \right\rfloor\right)\right) r_2 \cdot (x^G(t) - x_i(t)) + \min\left(1, \left\lfloor \frac{t-t_c}{t_m + \theta_i} \right\rfloor\right) r_3 \cdot (x_{c_i}(t) - x_i(t)) + \omega v_i(t) \quad (5)$$

$$\theta_i = \frac{R_{id} \cdot t_m}{N_p} \quad (6)$$

where ω is the inertial weight; t_m is the maximal time step before communication with the server was restored; $R_{id} \in [1, N_p]$ was the index assigned to a particle and N_p was the number of particles; t_p was the current time step; t_c was the last time step when successful communication with the server occurred; r_1 , r_2 and r_3 were random numbers within the range of $[0, 1]$; v_i and x_i were the velocity and position, respectively, of the i th particle; x_{c_i} denoted the best position of communication; x_i^L was the history best position of the i th particle and x^G was the global best position.

In the method, each particle only attempted to communicate with the server when it was its turn to do so. When a particle attempted to communicate, other particles moved based on the DPSO algorithm referred to in [33]. Experiments have shown that this communication-aware PSO (CPSO) algorithm can noticeably improve the performance when communication is restricted. However, the disadvantage is that communications with the server cannot always be guaranteed.

In this article, we propose a novel approach in which the DPSO algorithm is used to effectively manage the challenge of the self-deployment of sensors considering a CL.

III. METHODS OF SENSOR DEPLOYMENT

In this section, we review some methods of sensor deployment. In [36], the artificial bee colony (ABC) algorithm was used in the dynamic deployment of SNs to increase coverage. In [28], the author proposed a guided WT-based sensor deployment strategy (WTDS) for 3D terrain, in which the sensor movements were carried out in the mutation phase of a GA. The algorithm aimed to maximize the quality of coverage (QOC) of a WSN by deploying a limited number of sensors on a 3D surface. When compared with the DT method, the results revealed that it was a more powerful and successful method for sensor deployment in 3D terrain. In [29], the author proposed a deterministic sensor deployment method based on a WT and the CSO algorithm. Compared with the DT and GA based methods, the performance of this algorithm was proven to be better for sensor deployment in 3D terrain. However, [28] and [29] only aimed to maximize the QOC of a WSN and did not consider connectivity. A Lagrangian heuristic approach was proposed in [37] to solve a multiobjective sensor deployment problem. Their model could be used to concurrently decide the locations, activities, scheduling and data routings of sensors. However, it was questionable whether this model could be adopted for WSNs with a large number of sensors in 3D terrain.

In [38], the author proposed a sensor deployment strategy in which the PSO algorithm and bacterial foraging algorithm (BFA) were used for image segmentation. The study showed that bioinspired algorithms could perform multilevel image segmentation faster than an exhaustive search for optimal thresholds. In [32], the PSO algorithm and adaptive PSO (APSO) algorithm were used to cover 3D terrain with a limited number of sensors.

Some researchers have tried to solve the problem of sensor deployment based on the VF algorithm [25], [39]–[41]. A scheme based on the potential field method to optimize the coverage of sensor networks was presented in [41]. In [39], the author presented a VF-based algorithm that was executed by cluster heads (CHs). The CHs calculated the desired destination for each sensor through a sum of four types of forces: the repulsive forces (RFs) of obstacles, the attractive forces (AFs) of areas with low density, and the RFs and AFs of neighbor nodes. In [42], to enhance coverage, a VF-based algorithm was proposed to perform sensor deployment. The total number of sensors was fixed. However, it was assumed that after the initial random deployment, all SNs were able to communicate with a CH. In practice, communications could not always be guaranteed due to CLs. In addition, it was assumed that the sensors could realize a one-time movement to a desired location, which was also not realistic because the step length of the one-time movement was usually restricted. Moreover, the method was not suitable for 3D terrain.

The performance of the VF algorithm may be affected in hybrid WSNs. The forces exerted by the SSNs hinder the movements of MSNs [11], [26]. Different from the VF algorithm, the PSO algorithm searches the optimal results globally. Stationary SNs cannot affect the implementation

of the PSO algorithm. However, a PSO has disadvantages such as low convergence speed and a tendency to easily fall into local optima. A PSO algorithm directed by a virtual force (PSOVF) was proposed [6], in which a virtual force is adopted into the update of the velocities of the particles to increase the speed of regional convergence in the PSO algorithm. Simulation results have proven that the algorithm is effective and performs better than the single VF algorithm and single PSO algorithm. However, in the PSO algorithm, a particle is used to represent a complete solution vector, and the search space will be enlarged exponentially as the dimensionality of the solution vector increases. Similar to the PSOVF algorithm, in the improved PSO that is directed by the virtual force (IDPSOVF) algorithm, the PSO algorithm was used to implement a global search for the optimal deployment solution. The VF algorithm was used to direct the update of particles towards better positions. Instead of adopting one swarm to find the optimal n-dimensional vector, the vector is split into its components so that each swarm attempts to optimize a single component of the solution vector. The simulation results have shown that the IDPSOVF algorithm is more efficient than the PSOVF algorithm in terms of effective coverage area and computation time [6]. However, it is worth noting that this comparison was performed when the CL was not considered. In addition, the implementation of the PSOVF algorithm and the IDPSOVF algorithm in 3D terrain was missing. Moreover, these methods required a supernode to act as the processing center to implement the algorithm and did not consider the connectivity.

In this article, to improve the coverage and connectivity of a WSN in 3D terrain after the initial random deployment and to overcome the disadvantages of the PSO algorithm and VF algorithm, we propose a novel approach that combines the DPSO algorithm with a proposed 3D VF algorithm. In addition, we take the CL and energy consumption of SNs into consideration.

IV. THE PROPOSED APPROACH

We consider the sensor deployment of a hybrid WSN consisting of m ($1, \dots, m$ are the indexes) SSNs, n ($m+1, \dots, m+n$ are the indexes) MSNs and a BS in 3D terrain. The position of the i th SN (denoted as s_i) is represented by a 3D vector, $x_i = \{x_{i1}, x_{i2}, x_{i3}\}$ for $i = 1, 2, \dots, N$. N is the total number of SNs. Thus, the WSN can be described as a coordinate sequence of the positions of all SNs:

$$x = \{x_1, x_2, \dots, x_N\} = \{x_{11}, x_{12}, x_{13}, \dots, x_{N1}, x_{N2}, x_{N3}\}$$

Since the coverage and connectivity of a WSN rely largely on the positions of the sensors, the sensor deployment problem can be seen as a constrained optimization problem of the positions of SNs:

$$\begin{cases} \max f(x) \\ \text{s.t. } x \in \text{ARs} \end{cases} \quad (7)$$

The constraint is that the positions of the SNs must belong to regions where sensors can be deployed in practice. We call

such regions accessible regions (ARs). f denotes the fitness function used. In this article, we assume that all sensors know their positions precisely. In addition, all sensors are initially deployed in a random manner in a 3D terrain.

For the convenience of evaluation, the ROI is divided into a number of cubic grids. The granularity (l) of division is determined by the required accuracy of evaluation. The smaller the value of granularity, the higher the evaluation accuracy. However, a granularity that is too small requires too much computation time. A strategy is introduced to balance the evaluation speed and accuracy. The basic idea is to decrease the value of granularity when the fitness value has no obvious improvement over several successive steps (δ steps) when executing the proposed algorithm.

A. PROBABILITY SENSING MODEL

Compared with the traditional binary sensing model, a probabilistic sensing model allows a more realistic modeling of sensor coverage. In the proposed probability sensing model, the uncertainty of sensor detection is defined as r_e ($r_e < r_s$). r_s is the sensing radius of a sensor. Generally, if a space point lies in the range of $(r_s - r_e)$, the point can certainly be sensed. If a point lies outside of $(r_s + r_e)$, the point cannot be sensed at all. In addition, if a point lies within $(r_s - r_e)$ and $(r_s + r_e)$, then the detection of this point can be represented by a probability whose value is $e^{-\alpha \cdot \text{dist}^\beta}$, which decreases as the distance between the point and the SN increases.

Thus, the sensing probability to a space point $p(x_{p1}, x_{p2}, x_{p3})$ for SN s_i is defined as:

$$c_p(s_i) = \begin{cases} 1 & d(s_i, p) < r_s - r_e \\ e^{-\alpha \cdot \text{dist}^\beta} & r_s - r_e \leq d(s_i, p) < r_s + r_e \\ 0 & d(s_i, p) > r_s + r_e \end{cases} \quad (8)$$

$$\text{dist} = \frac{d(s_i, p) - (r_s - r_e)}{2} \quad (9)$$

where $d(s_i, p)$ is the Euclidean distance between SN s_i and point $p(x_{p1}, x_{p2}, x_{p3})$; α and β are related to the characteristics of the sensors and environmental terrain. For ground-based sensors, their binary detection regions can be modeled as cones. Thus, for ground-based SN s_i , the sensing probability to a space point $p(x_{p1}, x_{p2}, x_{p3})$ is defined as:

$$c_p(s_i) = \begin{cases} 1 & \text{if } d(s_i, p) < r_t \text{ and } r_e \leq \Delta x < r'_h \\ e^{-\alpha \cdot \text{dist}^\beta} & \text{if } r_t \leq d(s_i, p) < r'_h \cap 0 \leq \Delta x < r_h \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

$$r_t = \sqrt{\left(1 + \frac{r_{cs}^2}{h^2}\right)(x_{p3} - x_{i3})^2 - r_e} \quad (11)$$

$$r'_t = \sqrt{\left(1 + \frac{r_{cs}^2}{h^2}\right)(x_{p3} - x_{i3})^2 + r_e} \quad (12)$$

$$\Delta x = |x_{p3} - x_{i3}| \quad (13)$$

$$r_h = h + r_e \quad (14)$$

$$r'_h = h - r_e \quad (15)$$

where the apex of the cone is SN s_i ; r_{cs} is the radius of the cross section and h is the height of the cone.

The coverage probability of a point that is covered by a set of SNs is:

$$c_p(s_{ov}) = 1 - \prod_{s_i \in S_{ov}} (1 - c_p(s_i)) \quad (16)$$

where S_{ov} denotes the set of SNs that cover the point.

B. THE PROPOSED 3D VF ALGORITHM

A VF algorithm that is suitable for 3D terrain is proposed in this section. In this 3D VF algorithm, three kinds of forces are considered. The total force on SN s_i is the sum of these VFs:

$$\vec{F}_i = \sum_{j=1, j \neq i}^N \vec{F}_{ij} + \sum_{m=1}^{N_o} \vec{F}_{iO_m} + \sum_{m'=1}^{N_a} \vec{F}_{iA_{m'}} \quad (17)$$

where N_o and N_a are the total number of obstacles and preference areas (PAs), respectively.

The VF between SN s_i and SN s_j can be expressed in rectangular coordinates:

$$\vec{F}_{ij} = \begin{cases} (0, 0, 0) & \text{if } d(s_i, s_j) \geq r_c \\ B_1 & \text{if } r_c > d(s_i, s_j) > d_{th} \\ (0, 0, 0) & \text{if } d(s_i, s_j) = d_{th} \\ B_2 & \text{if } d(s_i, s_j) < d_{th} \end{cases} \quad (18)$$

$$B_1 = (r\gamma_1(\varphi\theta), r\gamma_2(\varphi\theta), r\gamma_3(\varphi)) \quad (19)$$

$$B_2 = (r'\gamma_1(\varphi'\theta'), r'\gamma_2(\varphi'\theta'), r'\gamma_3(\varphi')) \quad (20)$$

$$r = \omega_A(d(s_i, s_j) - d_{th}) \quad (21)$$

$$r' = \omega_R\left(\frac{1}{d(s_i, s_j)} - \frac{1}{d_{th}}\right) \quad (22)$$

$$\gamma_1(ab) = \sin a \cdot \cos b \quad (23)$$

$$\gamma_2(ab) = \sin a \cdot \sin b \quad (24)$$

$$\gamma_3(a) = \cos a \quad (25)$$

$$\varphi + \varphi' = \pi \quad (26)$$

$$\theta + \theta' = \pi \quad (27)$$

where r_c is the communication radius of a sensor; $d(s_i, s_j)$ is the Euclidean distance between SN s_i and SN s_j ; φ is the intersection angle between coordinate plane ZOX and the plane that goes through the Z-axis and the line segment from s_i to s_j ; θ is the intersection angle between the line segment from s_i to s_j and the Z-axis; ω_A and ω_R are weights; d_{th} is a threshold whose value is related to how close sensors are to each other; $\gamma_1(ab)$ and $\gamma_2(ab)$ are functions whose independent variables are a and b ; $\gamma_3(a)$ is a function with argument a .

The VF exerted on SN s_i by obstacle o_m is:

$$\vec{F}_{iO_m} = \begin{cases} B_3 & \text{if } r_1 < r_3 \\ (0, 0, 0) & \text{if } r_1 \geq r_3 \end{cases} \quad (28)$$

or

$$\vec{F}_{iO_m} = \begin{cases} B_3 & \text{if } r_1 < r_t'' + 2 \cdot r_e \cap 0 \leq \Delta x' < r_h \\ (0, 0, 0) & \text{otherwise} \end{cases} \quad (29)$$

$$B_3 = (r_2\gamma_1(\varphi_1\theta_1), r_2\gamma_2(\varphi_1\theta_1), r_2\gamma_3(\varphi_1)) \quad (30)$$

$$r_1 = d(s_i, o_m) - r_{o_m} \quad (31)$$

$$r_2 = \frac{\omega_{o_{ob}} p_{o_m}}{r_1} \quad (32)$$

$$r_3 = r_s + r_e \quad (33)$$

$$r_t'' = \sqrt{\left(1 + \frac{r_{cs}^2}{h^2}\right)(x_{O_m3} - x_{i3})^2 - r_e} \quad (34)$$

$$r_h = h + r_e \quad (35)$$

$$\Delta x' = |x_{O_m3} - x_{i3}| \quad (36)$$

where $d(s_i, o_m)$ denotes the Euclidean distance between SN s_i and obstacle o_m ; $\omega_{o_{ob}}$ is weight; r_{o_m} is the radius of obstacle o_m ; p_{o_m} is the importance level of obstacle o_m ; φ_1 is the intersection angle between coordinate plane ZOX and the plane that goes through the Z-axis and the line segment from s_i to obstacle o_m ; θ_1 is the intersection angle between the line segment from s_i to obstacle o_m and the Z-axis and x_{O_m3} is the coordinate of obstacle o_m on the Z-axis. (28) is suitable for general SNs and (29) is suitable for ground-based SNs.

The rectangular coordinate of the VF exerted by PA $A_{m'}$ on SN s_i is:

$$\vec{F}_{iA_{m'}} = \begin{cases} (0, 0, 0) & \text{if } \Delta d \geq r_c \\ B_4 & \text{if } r_3 \leq \Delta d < r_c \\ B_5 & \text{if } r_6 < \Delta d < r_3 \\ (0, 0, 0) & \text{if } \Delta d \leq r_6 \end{cases} \quad (37)$$

or

$$\vec{F}_{iA_{m'}} = \begin{cases} (0, 0, 0) & \text{if } \Delta d \geq r_c \\ B_5 & \text{if condition 1} \\ (0, 0, 0) & \text{if } \Delta d \leq r_t''' \cap r_e \leq \Delta x'' < r_h' \\ B_4 & \text{otherwise} \end{cases} \quad (38)$$

$$B_4 = (r_4\gamma_1(\varphi_2\theta_2), r_4\gamma_2(\varphi_2\theta_2), r_4\gamma_3(\varphi_2)) \quad (39)$$

$$B_5 = (r_5\gamma_1(\varphi_2\theta_2), r_5\gamma_2(\varphi_2\theta_2), r_5\gamma_3(\varphi_2)) \quad (40)$$

$$r_4 = \omega_{A_{m'}} p_{A_{m'}} \quad (41)$$

$$r_5 = \omega_{A_{m'}} p_{A_{m'}} \cdot \Delta d \quad (42)$$

$$r_6 = r_s - r_e \quad (43)$$

$$r_t''' = \sqrt{\left(1 + \frac{r_{cs}^2}{h^2}\right)(x_{A_{m'}3} - x_{i3})^2 - r_e} \quad (44)$$

$$\Delta d = d(s_i, A_{m'}) - r_{A_{m'}} \quad (45)$$

$$\Delta x'' = |x_{A_{m'}3} - x_{i3}| \quad (46)$$

where condition 1 is $r_t''' < \Delta d < r_t''' + 2 \cdot r_e \cap 0 \leq \Delta x'' < r_h$; $\omega_{A_{m'}}$ is weight; $r_{A_{m'}}$ is the radius of PA $A_{m'}$; $p_{A_{m'}}$ is the importance level of PA $A_{m'}$; $d(s_i, A_{m'})$ is the Euclidean distance between SN s_i and PA $A_{m'}$; φ_2 is the intersection angle between coordinate plane ZOX and the plane that goes through the Z-axis and the line segment from s_i to $A_{m'}$; θ_2 is the intersection angle between the line segment from SN s_i to PA $A_{m'}$ and the Z-axis and $x_{A_{m'}3}$ is the coordinate of PA $A_{m'}$ on the Z-axis. (37) is suitable for general SNs and (38) is suitable for ground-based SNs.

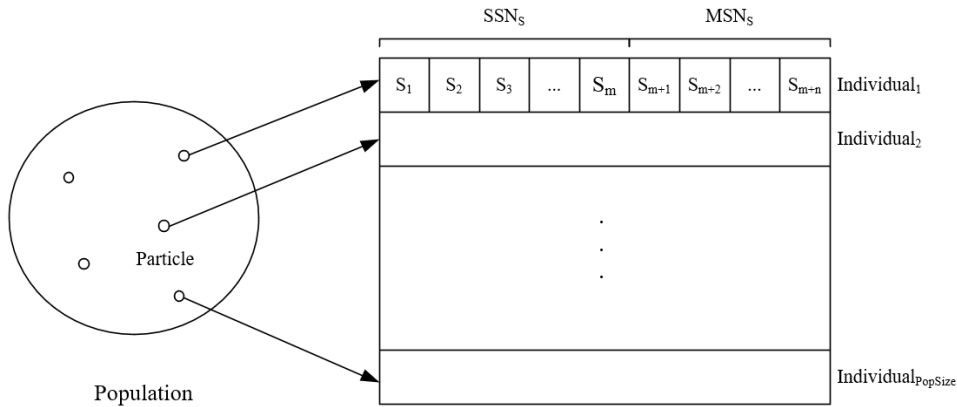


FIGURE 1. One-to-one match with particles in the DPSOVF algorithm.

Similar to [6], we assume obstacles exert RFs on the SNs while PAs exert AFs on the SNs. The update of the position of SN s_i is given by:

$$x_{i1}(t + 1) = x_{i1}(t) + \frac{F_{i1}(t)}{F(t)} \cdot \text{maxstep} \cdot e^{\frac{-1}{F(t)}} \quad (47)$$

$$x_{i2}(t + 1) = x_{i2}(t) + \frac{F_{i2}(t)}{F(t)} \cdot \text{maxstep} \cdot e^{\frac{-1}{F(t)}} \quad (48)$$

$$x_{i3}(t + 1) = x_{i3}(t) + \frac{F_{i3}(t)}{F(t)} \cdot \text{maxstep} \cdot e^{\frac{-1}{F(t)}} \quad (49)$$

where F_{i1} , F_{i2} and F_{i3} are the components of the total VF in the directions of the X-axis, Y-axis and Z-axis, respectively; maxstep indicates the maximal length of a step for an SN.

C. THE PROPOSED SENSOR DEPLOYMENT ALGORITHM

The proposed sensor deployment algorithm is based on a combination of the DPSO algorithm and the 3D VF algorithm proposed above. We name it the DPSOVF algorithm. In the DPSOVF algorithm, each candidate solution of the sensor deployment problem is a concatenation of the positions of all SNs. Each solution has a one-to-one match with a particle in the DPSOVF algorithm. Fig.1 illustrates the match relation.

SSNs are initially deployed randomly and remain stationary afterwards. The state of SSN s_i can be expressed as:

$$v_{ij}(t + 1) = 0 \quad (50)$$

$$x_{ij}(t + 1) = x_{ij}(t) \quad (51)$$

Relocation of the MSNs is used to improve the coverage and connectivity of the WSN. Considering the CL, in this article, the global best position (x^G) in (1) is replaced using a known best position (x^K). In general, the known best position is different from the global best position in practice. We make this change because, when communication is limited, not every particle can communicate with each other to receive timely information about the newly found global best position.

When guiding particles to search for the global optimal solution, to improve the search efficiency, jump out of the local optimum and avoid obstacles, a 3D VF oriented term is

added to the velocity update equation of each particle. Thus, the update equations of the velocity and position of MSN s_i in DPSOVF can be mathematically expressed as:

$$v_{ij}(t + 1) = \omega(t) \cdot v_{ij}(t) + c_1 \cdot r_1 \cdot (x_{ij}^L(t) - x_{ij}(t)) + c_2 \cdot r_2 \cdot (x^K(t) - x_{ij}(t)) + c_3 \cdot r_3 \cdot g_{ij}(t) \quad (52)$$

$$x_{ij}(t + 1) = x_{ij}(t) + v_{ij}(t + 1) \quad (53)$$

$$g_{ij}(t) = \begin{cases} \frac{F_{i1}^{(i, \frac{j+2}{3})}(t)}{F^{(i, \frac{j+2}{3})}(t)} \cdot \text{maxstep} \cdot e^{\frac{-1}{F^{(i, \frac{j+2}{3})}(t)}} & j = 3n - 2 \\ \frac{F_{i2}^{(i, \frac{j+1}{3})}(t)}{F^{(i, \frac{j+1}{3})}(t)} \cdot \text{maxstep} \cdot e^{\frac{-1}{F^{(i, \frac{j+1}{3})}(t)}} & j = 3n - 1 \\ \frac{F_{i3}^{(i, \frac{j}{3})}(t)}{F^{(i, \frac{j}{3})}(t)} \cdot \text{maxstep} \cdot e^{\frac{-1}{F^{(i, \frac{j}{3})}(t)}} & j = 3n \end{cases} \quad (54)$$

where $n = 1, \dots, N$ in (54); r_1 , r_2 and r_3 are random numbers within the range of $[0, 1]$; c_1 is the self-cognitive factor which reflects the effect of the history best position on velocity; c_2 is the social-cognitive factor which reflects the effect of the known best position on velocity; c_3 is the VF-cognitive factor which reflects the effect of VF on velocity, proper selections of c_1 , c_2 and c_3 are important to the performance of the DPSOVF algorithm; F is the VF vector exerted on particle par_i ; v_i and x_i are the velocity and position of particle par_i , respectively.

Considering that the PSO algorithm contains both global and local searches of the solution space, to balance the exploration and exploitation abilities, the inertial weight in (52) is designed to decrease linearly.

$$\omega(t) = \omega_{max} - (\omega_{max} - \omega_{min}) \frac{t}{maxiter} \quad (55)$$

where ω_{max} and ω_{min} are the upper and lower bound of ω , respectively; $maxiter$ denotes the maximal iteration number. This designation of the weight supports the global exploration at the beginning and ensures local exploitation at the end.

The update of the history best position of each particle in (52) is based on (3). Next, we discuss how to update the known best position considering the CL of the SNs. To obtain a known best position that is more approximate to the global best position during the implementation of DPSOVF, a cluster-based sensor network architecture is introduced to strengthen connectivity. Considering the CL of the SNs, the positions of some SNs cannot be known by other SNs since some SNs are not located inside of the communication ranges of the other SNs. Thus, the traditional methods that are based on k-means or Voronoi diagrams cannot be directly applied to SN clustering.

We develop a heuristic algorithm that clusters SNs effectively in limited communication environments. In this algorithm, each cluster consists of a CH and some member SNs. The aim of clustering is to try to contain all SNs within clusters while maintaining the SNs ability to communicate with their CHs. In this way, if all CHs remain connected to the BS, all SNs inside of clusters can be connected to the BS. Thus, the best position can be easily transmitted and known by more SNs. Here, an SN is said to be connected to the BS if there is at least one communication path from the SN to the BS in either direct or indirect ways. The BS can broadcast the best information to the SNs connected to it.

In the proposed clustering algorithm, all CHs are selected from SNs. First, SNs that are inside of the communication range of the BS are selected as CHs. Second, the SNs that are inside of the communication ranges of the selected CHs are selected as CHs.

We denote the total number of CHs as N_c and the set of all CHs as S_c . After the selection of the CHs, each CH clusters all its communication SNs (CSNs) into one cluster. A CSN is defined as:

Definition 1 (CSN): SN s_j is said to be a CSN of SN s_i at t if $s_j \in C_i(t)$ where

$$C_i(t) = \{s_k \in S \mid d(s_i, s_k)(t) \leq 2 \cdot r_c \text{ for all } k \neq i\} \quad (56)$$

S denotes the set of all SNs, $d(s_i, s_k)(t)$ is the distance between SN s_i and SN s_k at t . Since all MSNs update their positions each iteration, the set of CSNs of an SN is a variable set.

Fig. 2 and Fig. 3 illustrate the CH selection process. In Fig. 2 and Fig. 3, the squares represent the BS, the small circles represent the SNs, and the triangles represent the selected CHs. To distinguish the CHs selected in the first step from those selected in the second step, the triangles in Fig. 3 are divided into different colors. Green triangles represent the CHs selected in the first step.

The data flow of this cluster-based sensor network is shown in Fig. 3. The directions of the arrows in Fig. 3 represent the directions of the data flows. After SN clustering, each SN transmits its position data to its CH. Each CH transmits the received data to the BS. Then, the BS performs data aggregation, determines the best position of the particles and broadcasts this best position data to all the SNs connected to it.

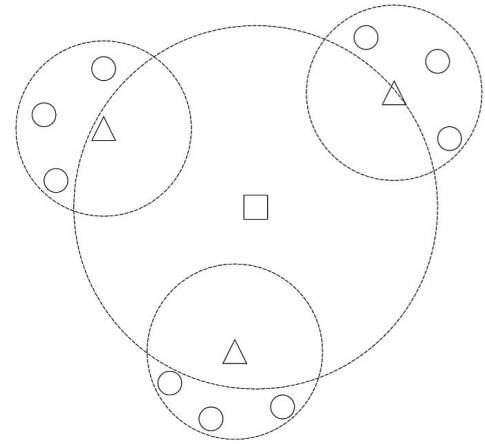


FIGURE 2. First selection of CHs.

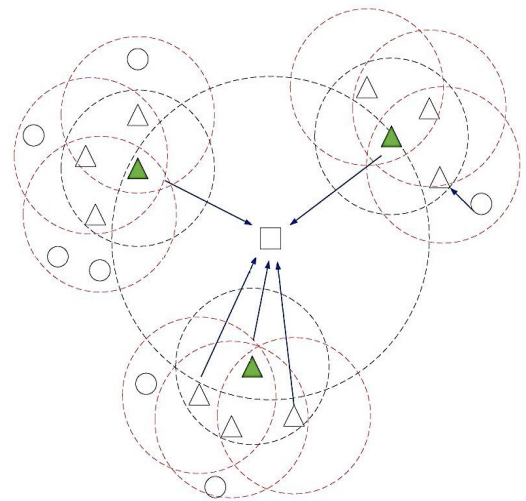


FIGURE 3. Second selection of CHs.

The clustering effect depends on two aspects. One is the total number of SNs inside the clusters, and the other is the total number of the overlapped clustering SNs (OSNs).

Definition 2 (OSN): An SN is called an OSN if it belongs to more than one cluster simultaneously.

The greater the number of SNs inside a cluster is, the less the information loss of the particles, and thus, the closer the known best position is to the global best position. However, OSNs lead to unnecessary data transmission, namely, an OSN needs to transmit its position data to more than one CH, which leads to more energy consumption of the WSN. Considering the energy consumption model of WSN in [43]:

$$E_{Tx} = \begin{cases} E_{elec} \times b + E_{fs} \times b \times d^2 & d \leq d_0 \\ E_{elec} \times b + E_{mp} \times b \times d^4 & d > d_0 \end{cases} \quad (57)$$

$$d_0 = \sqrt{\frac{E_{fs}}{E_{mp}}} \quad (58)$$

where E_{Tx} is the transmission energy; E_{elec} is the energy dissipated to run the transmitter or receiver circuitry;

E_{fs} is the energy used for reception; E_{fs} and E_{mp} depend on the transmitter amplifier model; d is the data transmission distance and b is the length of the data transmitted.

After the initial clustering, to increase the number of SNs inside of the clusters and decrease the number of OSNs, some of the CHs adjust their positions. Since in each cluster, the CH can know the total number of SNs that belong to its cluster, the CHs who have relatively few neighbor SNs perform position adjustments to cover more SNs. We denote the set of CHs who need to perform position adjustments as S_{CM} .

Definition 3 (Neighbor CH (NCH)): CH CH_i is said to be a NCH of CH CH_j at t if $CH_j \in \mathcal{U}_i(t)$ where

$$\mathcal{U}_i(t) = \{CH_k \in S_c \mid d(CH_i, CH_k)(t) < r_c\} \quad (59)$$

$d(CH_i, CH_k)(t)$ is the Euclidean distance between CH_i and CH_k at t . For $CH_j \in \mathcal{U}_i(t)$,

$$\vec{FC}_{ij}(t) = (r_4\gamma_1(\varphi_3\theta_3), r_4\gamma_2(\varphi_3\theta_3), r_4\gamma_3(\varphi_3)) \quad (60)$$

$$r_4 = \omega_1 \left(\frac{1}{d(CH_i, CH_j)(t)} - \frac{1}{2r_c} \right) \quad (61)$$

$$\vec{FC}_i(t) = \sum_{CH_j \in \mathcal{U}_i(t)} \vec{FC}_{ij}(t) \quad (62)$$

where ω_1 is the weight; φ_3 is the intersection angle between coordinate plane ZOX and the plane that goes through the Z-axis and the line segment from CH_i to CH_j ; θ_3 is the intersection angle between the line segment from CH_i to CH_j and the Z-axis.

To guarantee that all CHs can be connected to the BS, for each CH_i that belongs to S_{CM} , $\exists CH_j \in S_c \setminus S_{CM}$ satisfies $d(CH_i, CH_j) < 2r_c$ or $d(CH_i, BS) < r_c + r_{bc}$. $d(CH_i, BS)$ is the distance between CH_i and the BS. r_{bc} is the communication radius of the BS. The manner in which the positions of the CHs are updated can be referred to in (47), (48) and (49) under the condition that all CHs can be connected to the BS.

Each solution of this clustering problem can be expressed as the concatenation of positions of all CHs: $(x_{CH_1}, x_{CH_2}, \dots, x_{CH_{N_c}})$. The evaluation criterion of the cluster solution is based on:

$$f_c = n_i - n_o \quad (63)$$

where n_i denotes the number of SNs inside of the clusters and n_o denotes the number of OSNs. The termination condition is that there is no improvement in the value of f_c after 15 successive iterations. The pseudocode of the proposed SN cluster algorithm is shown in algorithm 1.

In algorithm 1, cs denotes the cluster solution, cs_u denotes the updated cluster solution and cr is the optimal cluster result.

To eliminate holes in coverage and enhance the connectivity of the WSN, the regions whose coverage and connectivity are poor need more SNs. We use RSNs selected according to certain criteria to fulfill this need. Regions that need more SNs can be used to guide the position updates of the RSNs.

Algorithm 1 SNs Cluster

```

1: Initialize the positions of SNs and BS
2: Select CHs
3: Select CHs again based on the first selection
4: Cluster CSNs of each CH
5: Evaluate  $cs$  based on (63)
6: if the termination condition is not met then
7:   Determine  $S_{CM}$ 
8:   for  $CH_i \in S_{CM}$  do
9:     for  $CH_j \in \mathcal{U}_i$  do Compute  $\vec{FC}_{ij}$  based on (60)
10:    end for
11:   Compute  $\vec{FC}_i$  based on (62)
12:   Update the position of  $CH_i$  based on (47), (48)
   and (49)
13:   end for
14:   Cluster CSNs of each CH
15:   if  $f_c(cs_u) > f_c(cs)$  then
16:      $cr = cs_u$ 
17:   else
18:      $cr = cs$ 
19:   end if
20: end if
21: Return cluster result

```

We name such regions R_n s. First, we give two definitions based on which R_n s are selected.

Definition 4 (Coverage Level):

$$L_c(A)(t) = \sum_{s_i \in S} \frac{c_A(s_i)(t)}{\sum_{A \in ROI} c_A(s_i)(t)} \quad (64)$$

where $L_c(A)(t)$ denotes the coverage level (COL) of the region A at t ; $c_A(s_i)(t)$ is the sensing probability to region A from SN s_i at t .

Definition 5 (Connectivity Level (CNL)): First, we define an indicator function to denote whether SN s_i is connected to the BS or not at t .

$$I_i(t) = \begin{cases} 1 & \text{if } s_i \text{ is connected to the BS} \\ 0 & \text{otherwise} \end{cases} \quad (65)$$

The CNL of region A at t is:

$$L_n(A)(t) = \sum_{s_i \in A_s} I_i(t) \quad (66)$$

where A_s denotes the set of all SNs located inside of region A .

If the CNL of region A increases, the connectivity of the SNs inside of region A to the BS increases, and then the known best position is more likely to be well transmitted. The lower the COL and CNL of a region, the greater the demand for SNs in the region is. Thus, regions that have low COLs and CNLs are selected as R_n s to guide the position updates of the RSNs. The pseudocode of selection of R_n s is shown in algorithm 2.

The selection of the RSNs is based on three factors: the overlap level (OL) of an SN, the residual energy of an SN

Algorithm 2 Select R_n s

- 1: Initialize positions of SNs and BS
- 2: Divide the ROI based on l
- 3: SNs cluster according to algorithm 1
- 4: **for** each cubic region (A) in ROI **do**
 Compute $L_c(A)$ based on (64)
 Compute $L_n(A)$ based on (66)
- 5: **end for**
- 6: Return regions with low COLs and CNLs

and the distance between an SN and the nearest R_n in its communication range.

Definition 6 (OL of an SN):

$$L_{o_{s_i}} = \bigcup_{\forall j \neq i, s_j \in S} (R_{s_i} \cap R_{s_j}) \quad (67)$$

where $L_{o_{s_i}}$ denotes the OL of SN s_i ; R_{s_i} is the sensing region of SN s_i . The higher the value of the OL of an SN is, the more likely that the SN will be selected as an RSN.

There are two different deployment cases in Fig. 4 and Fig. 5, and the OL of SN s_1 in Fig. 4 is larger than that in Fig. 5. Thus, SN s_1 is more likely to be an RSN in the case of Fig. 4 than in Fig. 5. We denote the set of selected RSNs as S_r .

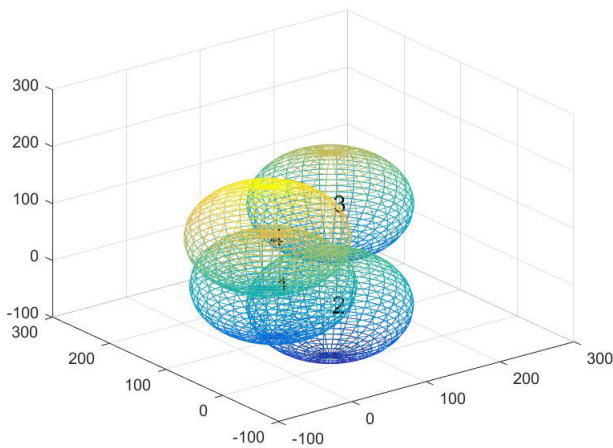


FIGURE 4. Deployment of four SNs in one case.

Considering the OL can help to decrease SN overlap and improve coverage of the WSN. The residual energy is considered in order to avoid using an SN with low energy too many times, which can help to extend the network life.

Inspired by [44], a routing scheme was designed to reduce the energy consumption of unmanned aerial vehicles in data collection. Considering that we can learn from equation (57), a long distance of data transmission can lead to large energy consumption. To reduce the energy consumption of RSNs, we take the distance between an RSN and its nearest R_n into consideration and use the nearest R_n to guide the velocity update of an RSN.

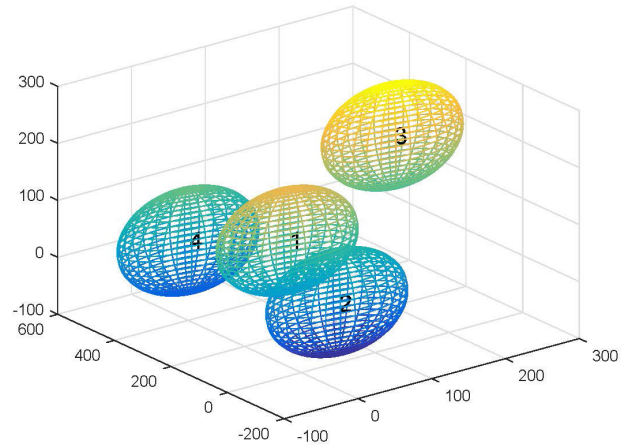


FIGURE 5. Deployment of four SNs in another case.

Consequently, the criterion used to select RSNs can be expressed as follows:

$$RS_{sele} = arg \max_{s_i \in U_{mn}} \left\{ \frac{L_{o_{s_i}} E_{s_i}}{d_n} \right\} \quad (68)$$

$$U_{mn} = \{s_k \in U_m \mid d(s_i, s_k) \leq r_c \text{ for all } k \neq i\} \quad (69)$$

where U_m denotes the set of all MSNs; E_{s_i} is the residual energy of SN s_i ; d_n is the distance between SN s_i and its nearest R_n .

All RSNs are selected from MSNs. The nearest R_n to the i th RSN (rs_i) is selected to guide the velocity and position updates of rs_i . We know that there may be more than one R_n inside of an RSN's communication range. The updates of velocity and position of rs_i can be mathematically expressed as follows:

$$v_{ij}(t+1) = \omega_2 \cdot v_{ij}(t) + c_1 \cdot r_1 \cdot (\bar{x}_{A_j} - x_{ij}(t)) \quad (70)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (71)$$

where ω_2 is the weight; c_1 is a learning coefficient; r_1 is a random number within the range of [0, 1]; \bar{x}_{A_j} is the position in the j th dimension of the geometry center of the nearest R_n . Through (70) and (71), RSNs can move effectively to regions whose demand for SNs is great.

Once an SN is selected as an RSN, it will be updated based on (70) and (71) instead of (52) and (53). It is worth noting that a set of RSNs is dynamic because the relative positions of SNs change with increasing iteration. An SN that is selected as an RSN at the t th iteration may still be an RSN or it may become an ordinary MSN at the $t+1$ th iteration.

Obviously, there are three different update manners for SNs in the proposed DPSOVF algorithm, which are illustrated in Fig. 6. Circles with characters S , M and R represent the SSNs, MSNs and RSNs, respectively.

The dispersion degree and the district-difference degree are introduced to overcome the overgathering of SNs. The dispersion degree of the SNs is calculated by a Gaussian function:

$$H_1 = e^{(-\frac{1}{2}(\delta' - \mu)^T V^{-1}(\delta' - \mu))} \quad (72)$$

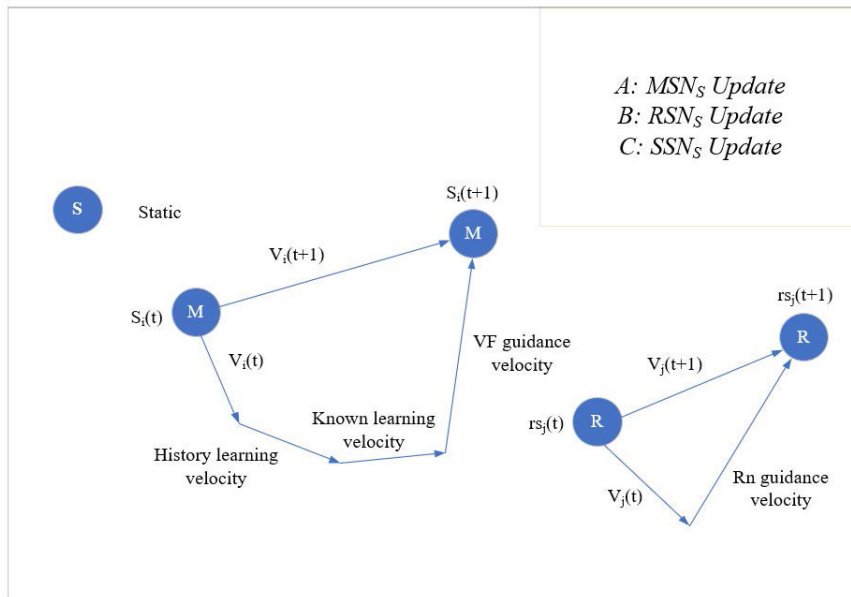


FIGURE 6. Three different update manners in the proposed DPSOVF algorithm.

$$\delta' = \frac{\bar{d}}{\sqrt{L^2 + W^2 + H^2}} \quad (73)$$

$$\mu = \frac{1}{t} \sum_{k'=1}^t \delta'_{k'} \quad (74)$$

$$V = \frac{1}{2} [\max(\delta'_{k'}) - \min(\delta'_{k'})] \quad (75)$$

$$\bar{d} = \frac{2}{N_l(N_l - 1)} \sum_{i=1}^{N_l} \sum_{j=i+1}^{N_l} d(s_i, s_j) \quad (76)$$

where H_1 is the dispersion degree; \bar{d} is the real time average distance between the SNs; k' is a counter in the V calculation and $k' = 1, \dots, t$; N_l is the number of SNs that are connected to the BS; and L , W and H are the length, width and height of the ROI, respectively.

The district-difference degree is defined as:

$$H_2 = \frac{A_u}{ROI} + \sum_{i=1}^{N_l} I(R_i) \quad (77)$$

$$I(R_i) = \begin{cases} 1, & \text{if SN } s_i \text{ in the } i\text{'th part of the search region} \\ 0 & \text{otherwise} \end{cases} \quad (78)$$

where A_u denotes the unexplored search region known by the BS. $I(R_i)$ is used to judge whether SN s_i is in the i 'th part of the search region; N_g is the number of divided parts of the search region and $i' = 1, \dots, N_g$.

To control the population diversity and appropriately strengthen the exploration ability, u defined below, is used to evaluate the particles in the DPSOVF algorithm.

$$u = \frac{H_1 H_2}{\omega_{H_1} H_2 + \omega_{H_2} H_1} \quad (79)$$

where ω_{H_1} and ω_{H_2} are weights. The values of ω_{H_1} and ω_{H_2} depend on applications.

In addition, the coverage ratio and connectivity ratio are used as criteria to evaluate particles in the DPSOVF algorithm. The coverage ratio is defined as:

$$CR = \frac{n_c}{n_x \cdot n_y \cdot n_z} \quad (80)$$

where n_c denotes the number of covered cubic grids; n_x , n_y and n_z represent the number of grids on the X-axis, Y-axis and Z-axis, respectively. The connectivity ratio is defined as:

$$CO = \frac{L_n(ROI)(t)}{m+n} = \frac{\sum_{s_i \in S_{RS}} I_i(t)}{m+n} \quad (81)$$

where S_{RS} denotes the set of all SNs located inside of ROI.

Thus, all candidate solutions are evaluated through a fitness function defined as follows:

$$f = \mu_1 \cdot CR + \mu_2 \cdot CO + \mu_3(t) \cdot u \quad (82)$$

where μ_1 , μ_2 and μ_3 are weighting factors. μ_3 is designed to be dynamic to guarantee the population diversity at an early stage and convergence speed of the proposed algorithm at a later stage.

$$\mu_3(t) = \mu_{3_{max}} - (\mu_{3_{max}} - \mu_{3_{min}}) \frac{t}{maxiter} \quad (83)$$

Fig. 7 shows the flowchart of the whole procedure.

The pseudocode of the proposed DPSOVF algorithm is shown in algorithm 3.

In algorithm 3, U_s denotes the set of all SSNs; U_r denotes the set of all RSNs and U_m denotes the set of all MSNs.

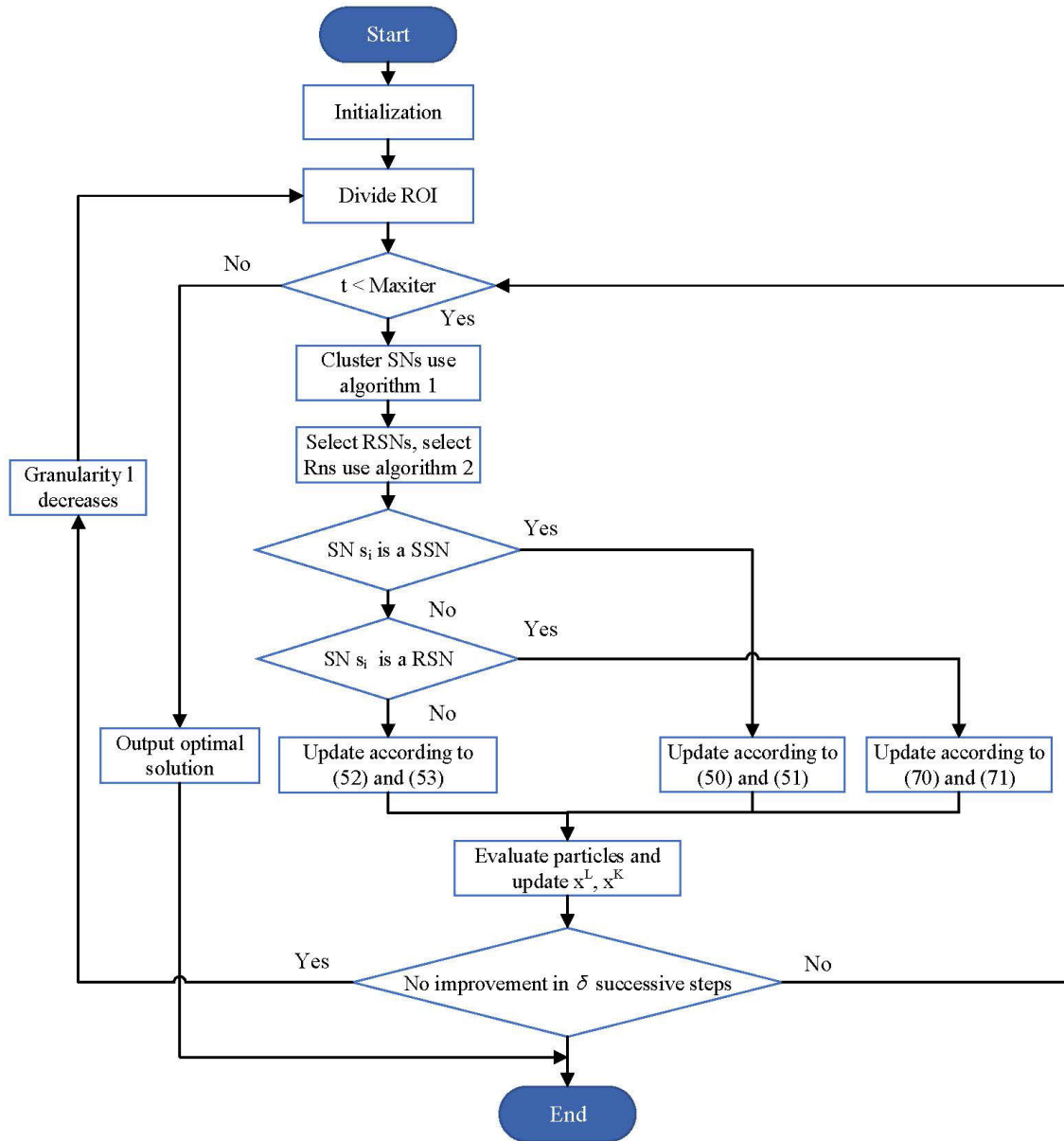


FIGURE 7. Flowchart of the whole procedure.

V. EXPERIMENTS AND ANALYSIS

To verify the effectiveness of the proposed algorithm, we use MATLAB (R2016a) to perform simulation experiments. The simulations are performed on an Intel(R) Core(TM) i5-8250U CPU (1.80 GHz) PC. We simulate a hybrid WSN consisting of 4 SSNs, 10 MSNs and a BS. The proposed DPSOVF algorithm is compared with some existing algorithms for deploying sensors in 3D terrain. For simplicity, we assume no obstacles exist. To guarantee the accuracy of comparisons, average values are taken from 30 independent simulation runs. The parameter setting is shown in table 1.

Three different cases are considered to evaluate the performance of the DPSOVF algorithm. These cases differ in the size of the ROI, number of particles (popsize) and maximal

iteration (maxiter), as shown in Table 2. The unit of the data related to the ROI is meter.

In practical applications, the termination condition (reach maxiter in this article) can be defined as when there is no improvement in the quality of the solution in 15 successive iterations.

Figs. 8, 9 and 10 show the comparison of coverage ratio when using the original PSO algorithm, DPSO algorithm [33], CPSO algorithm [35], PSOVF algorithm, IDPSOVF algorithm [6] and proposed DPSOVF algorithm in different environmental cases.

Fig. 8 depicts the comparison of the coverage ratio in case 1. Because the values of the coverage ratios are very small, logarithms of the coverage ratios are used to reflect

Algorithm 3 DPSONV Algorithm

```

1: Initialize SNs, BS, maxiter,  $l$ ,  $x^L$  and  $x^K$ 
2: Divide the ROI based on  $l$ 
3: if  $t < \text{maxiter}$  then
4:   SNs cluster according to algorithm 1
5:   Select RSNs based on (68)
6:   Select Rns according to algorithm 2
7:   for  $\text{SN} \in U_s$  do
8:     Update velocity and position based on
      (50) and (51)
9:   end for
10:  for  $\text{SN} \in U_r$  do
11:    Update velocity and position based on
      (70) and (71)
12:  end for
13:  for  $\text{SN} \in (U_m \setminus U_r)$  do
14:    Update velocity and position based on
      (52) and (53)
15:  end for
16:  Evaluate each candidate solution based on (82)
17:  Update  $x_{ij}^L(t)$  based on (3)
18:  Update  $x_{ij}^K(t)$ 
19:   $t = t + 1$ 
20:  if no improvement in value of  $f$  in (82) in  $\delta$  successive
      steps then
21:     $l$  decreases
22:    Divide the ROI based on new  $l$  and return to 3
23:  end if
24: end if
25: Return the optimal solution
    
```

TABLE 1. Parameter setting.

Parameter	Value	Parameter	Value
r_s	37m	r_{cs}	18 m
h	28m	α	0.9
r_c	$2.3 * r_s$	β	1
r_e	$0.4 * r_s$	μ_1	1
c_1, c_2, c_3	1	$\omega_{A_{m'}}$	1
d_{th}	$2 * r_s$	μ_2	1
l	5	ω_{Ob}	3
ω_{max}	0.8	ω_R	3
ω_{min}	0.1	ω_1, ω_2	0.5
maxstep	$0.5 * r_s$	$\omega_{H_1}, \omega_{H_2}$	1
E_{elec}	82nJ/bit	E_{fs}	13pJ/bit/m ²
E_{mp}	0.0013pJ/bit/m ⁴	Initial energy	17J

TABLE 2. Three different simulation cases.

Parameter	Case 1	Case 2	Case 3
ROI	700 * 700 * 700	150 * 150 * 150	200 * 200 * 200
Popsize	4	5	5
Maxiter	5	10	50

the changes in the coverage ratios as the number of iterations increases. The coverage ratios are very small in this case due to an ROI that is too large, an insufficient number of SNs and a small maxiter. From Fig. 8, we can see that the coverage ratio

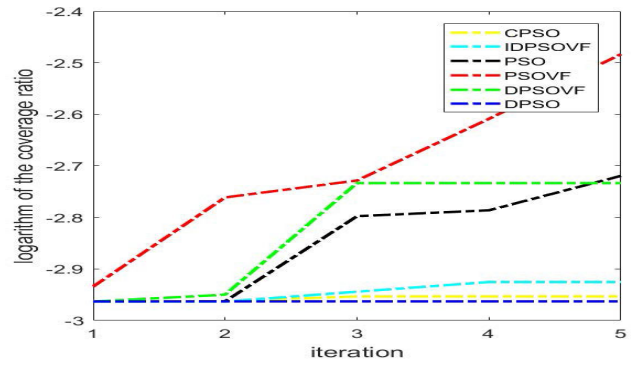


FIGURE 8. Coverage performance comparison of six algorithms in case 1.

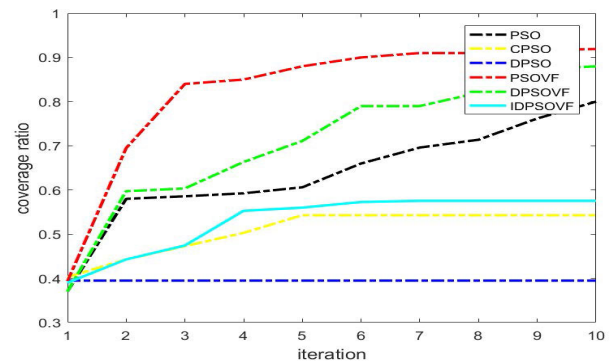


FIGURE 9. Coverage performance comparison of six algorithms in case 2.

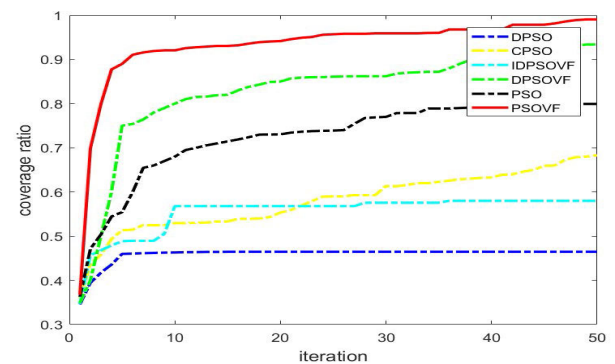


FIGURE 10. Coverage performance comparison of six algorithms in case 3.

represented by the green line has a higher convergence speed than that of the black line, because of the contribution of VF. However, the final coverage ratio when using the DPSONV algorithm is lower than that when using the PSO algorithm. This is because the CL cannot be well overcome due to an ROI that is too large and insufficient SNs and maxiter.

Fig. 9 depicts the coverage performance comparison in case 2. The ascent rate of the blue-green line is faster than that of the yellow line. This is because the addition of the VF term in the velocity update equations of the particles improves the efficiency.

The coverage performance comparison in case 3 is shown in Fig. 10. The yellow line has a higher coverage ratio than that of the blue-green line at last. This is because the CPSO

algorithm tries to strengthen communications among particles by adding a communication behavior into the velocity update equation of each particle, while the IDPSOVF algorithm does not take CL into consideration. The participation of VF in the IDPSOVF algorithm only speeds up convergence but has no method to overcome the CL. The PSO algorithm performs better than the IDPSOVF algorithm because it is assumed that the CL can be overcome completely in the implementation of the PSO algorithm.

As seen from Figs. 8, 9 and 10, it is obvious that the performance of the PSOVF algorithm, which is represented by red lines, is the most effective in guiding the SN deployment of a WSN in 3D terrain. Similar to the PSO algorithm, during the implementation of the PSOVF algorithm, it is assumed that CL can be overcome completely or that no CL exists. All particles can communicate with each other, and the global best information can be known precisely by all particles. The PSO algorithm in the PSOVF algorithm is used to guide the global search for the optimal solution of SN deployment. The VF algorithm in the PSOVF algorithm helps to avoid local optima and improve the convergence speed. This combination can lead to more effective deployment performance.

However, the PSOVF algorithm is not realistic, because CL is not a negligible factor and can never be overcome completely in a distributed manner. The coverage performance of the proposed DPSOVF algorithm is second only to that of the PSOVF algorithm. The DPSOVF algorithm can alleviate the impact of CL of the SNs and has a great improvement in coverage performance when compared with other algorithms. In a word, when considering CL, the proposed DPSOVF algorithm has the best deployment effect in terms of coverage among the six algorithms.

When the CL is not severe, the coverage performance of the IDPSOVF algorithm is better than that of the CPSO algorithm since the IDPSOVF algorithm absorbs the advantages of the PSO algorithm and VF algorithm. The coverage performance of the DPSO algorithm is the worst of all the compared algorithms because it neither considers CL nor has the contribution of VF. CL leads to unprecise global best information, which has a poor influence on the updates of the particles in the DPSO algorithm.

Figs. 11-13 depict part of the evolutionary process of the DPSOVF algorithm. This experiment is done in case 2. The sphere centers represent positions of the SNs. The area inside of a sphere represents the probable sensing area of an SN. The SNs update their positions according to the DPSOVF algorithm. As shown in Figs. 11-13, along with the increase in iteration, the coverage ratio obviously increases.

Table 3 illustrates the connectivity performance comparison when using the PSO algorithm, DPSO algorithm, CPSO algorithm, PSOVF algorithm, IDPSOVF algorithm and proposed DPSOVF algorithm. This comparison is done in experiment case 3. In table 3, the connectivity ratios that correspond to 10% of the maxiter, 50% of the maxiter, and the maxiter when using the six algorithms are shown. When using the PSO algorithm and PSOVF algorithm, the values

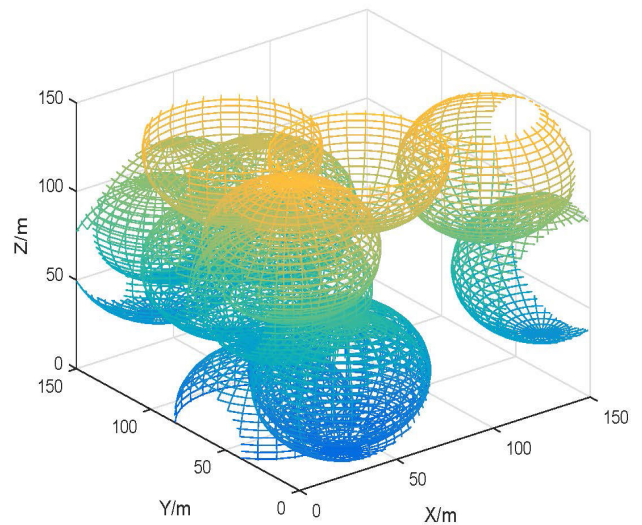


FIGURE 11. Random deployment of SNs at initiation.

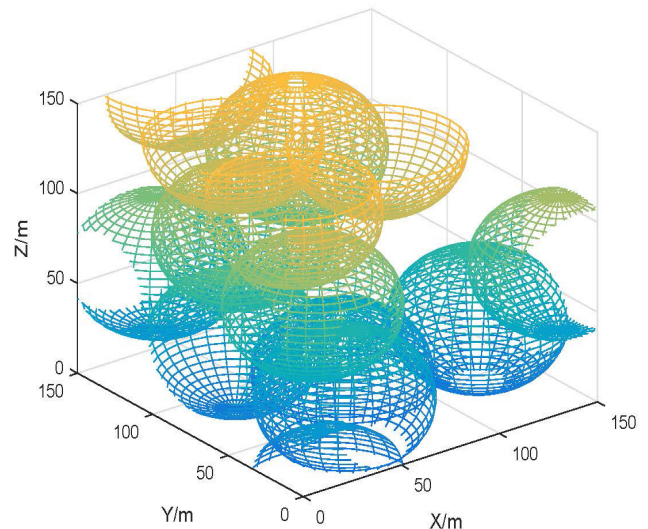


FIGURE 12. Deployment of SNs in the 5th generation.

TABLE 3. Connectivity performance comparison of six algorithms.

Algorithms	Connectivity ratio		
	10% maxiter	50% maxiter	maxiter
PSO	1	1	1
DPSO	0.0714	0.1428	0.1428
CPSO	0.0714	0.2857	0.4285
PSOVF	1	1	1
IDPSOVF	0.1428	0.2143	0.2857
Proposed	0.2857	0.6428	0.9286

of the connectivity ratios equal one at any time, because it is assumed that CL can be overcome completely during the implementations of the PSO algorithm and PSOVF algorithm. Thus, all SNs are connected to the BS. However, that is

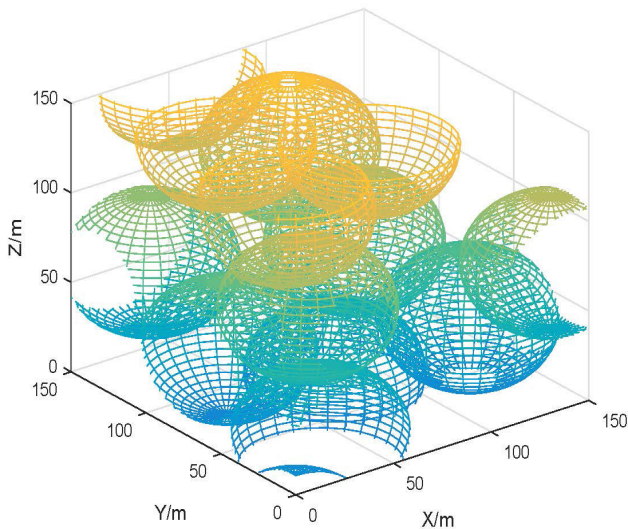


FIGURE 13. Deployment of SNs in the 10th generation.

only an ideal assumption and is not realistic. The connectivity performance of the DPSO algorithm is the worst because it does not consider CL and has no VF effect. Obviously, when considering CL, the proposed DPSOVF algorithm can effectively strengthen the connectivity of the WSN.

To verify the effectiveness of the proposed update method for the selected RSNs, three different update approaches for RSNs are used for comparison. They are the update approach shown in (52) and (53), which is proposed for ordinary MSNs; the random update approach, namely, RSNs update their positions randomly inside of the ROI; and the proposed update approach expressed in (70) and (71). This comparison is performed in experiment case 3.

Fig. 14 shows the comparison result of coverage performance when using the three update approaches for the

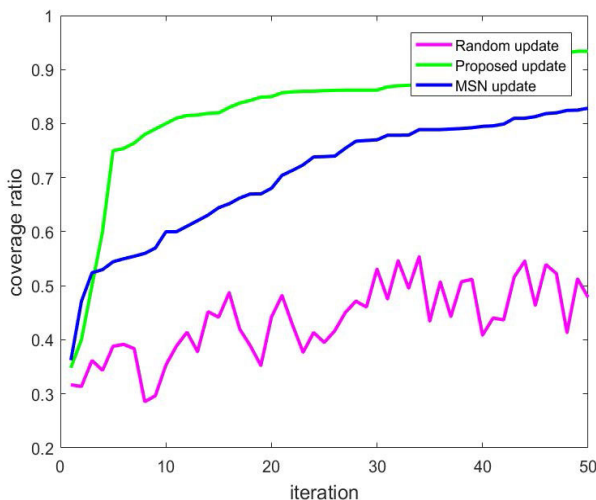


FIGURE 14. Coverage performance comparison of three update approaches for RSNs.

selected RSNs. The fuchsia line represents the coverage ratio when all selected RSNs update according to the random update approach during the implementation of the proposed DPSOVF algorithm. Obviously, the coverage performance of the random update approach is the worst. The values of the coverage ratios fluctuate irregularly, which means that random updates of the RSNs do not effectively improve the coverage of the WSN. However, the general change trend of the fuchsia line is rising, which is mainly the result of the updates of the MSNs. The blue line represents the coverage performance when all the RSNs are updated in the same way as the ordinary MSNs. As seen from Fig. 14, this approach is better than the random update approach in terms of improving coverage. Compared with the other two approaches, the coverage performance of the proposed approach represented by the green line is the most effective. The coverage ratio is effectively and rapidly improved by using the proposed update approach for RSNs.

The network lifetime denotes the time elapsed until the first node in the network dies [45]. In this study, we test the network lifetime by testing the time elapsed until the first node in the network dies and the time elapsed until the last node in the network dies. Three approaches are compared in terms of their effect on the network lifetime. Different from the proposed approach, the first approach does not consider the residual energy when selecting the RSNs. Also different from the proposed approach, the second approach does not consider the distance between an SN and its nearest Rn when selecting the RSNs. The third approach is the proposed sensor deployment approach. As seen from Fig. 15, compared with the other two deployment approaches, the network lifetime is the longest when using the proposed sensor deployment approach.

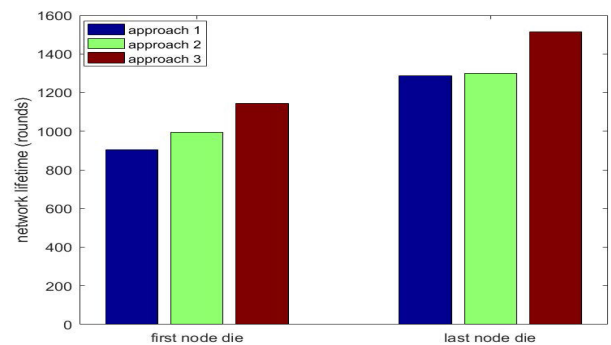


FIGURE 15. Network lifetime comparison when using different sensor deployment approaches.

In the proposed DPSOVF algorithm, the coverage ratio of the WSN is improved by the position updates of the MSNs. The influence of the number of MSNs on the performance of the DPSOVF algorithm is analyzed. Except for the change in the number of MSNs, other parameters are set the same as in case 3. The mean coverage ratio is the average value computed from 30 independent simulation runs. As can be observed from Fig. 16, the mean coverage ratio of the WSN

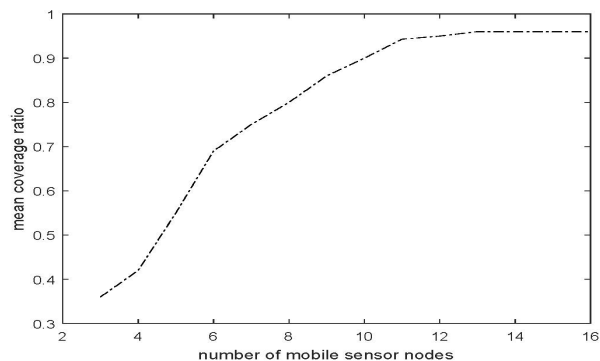


FIGURE 16. Relation of mean coverage ratio with the number of MSNs.

increases with the increase in the number of MSNs. It is easy to see that when the coverage ratio increases to a certain level, the increase in MSNs will have no obvious influence on the coverage ratio of the WSN.

VI. CONCLUSION

In this article, we focus on the problem of sensor deployment of a hybrid WSN in 3D terrain. A novel sensor deployment algorithm considering the CL of SNs is proposed. The algorithm is based on a combination of the DPSO algorithm and a newly proposed 3D VF algorithm. In the proposed DPSOVF algorithm, which considers CL, a heuristic algorithm is proposed to cluster the SNs in each generation to strengthen the connectivity. In addition, the RSNs are selected from the MSNs. Rns are selected to guide the relocation of the RSNs, which improves the deployment efficiency. All particles update their velocities and positions according to the proposed DPSOVF algorithm to improve the coverage and connectivity of the WSN. It has been shown that the DPSOVF algorithm can perform more effectively than any other compared algorithm in the sensor deployment of a WSN in 3D terrain when CL is considered.

REFERENCES

- [1] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, May 2000.
- [2] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Pers. Commun.*, vol. 7, no. 5, pp. 16–27, Oct. 2000.
- [3] M. Huang, K. Zhang, Z. Zeng, T. Wang, and Y. Liu, "An AUV-assisted data gathering scheme based on clustering and matrix completion for smart ocean," *IEEE Internet Things J.*, early access, Apr. 15, 2020, doi: 10.1109/JIOT.2020.2988035.
- [4] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 6, no. 4, pp. 621–655, Jun. 2008.
- [5] R. V. Kulkarni and G. Kumar Venayagamoorthy, "Particle swarm optimization in wireless-sensor networks: A brief survey," *IEEE Trans. Syst., Man, Cybern., C (Appl. Rev.)*, vol. 41, no. 2, pp. 262–267, Mar. 2011.
- [6] X. Wang, S. Wang, and J. Jie, "An improved co-evolutionary particle swarm optimization for wireless sensor networks with dynamic deployment," *Sensors*, vol. 7, no. 3, pp. 354–370, Mar. 2007.
- [7] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar, "Sensor placement for grid coverage under imprecise detections," in *Proc. 5th Int. Conf. Inf. Fusion (FUSION)*, Annapolis, MD, USA, Jul. 2002, pp. 1581–1587.
- [8] N. Boudriga, M. Hamdi, and S. Iyengar, "Coverage assessment and target tracking in 3D domains," *Sensors*, vol. 11, no. 10, pp. 9904–9927, Oct. 2011.
- [9] P. K. Sahoo and W. C. Liao HORA, "Distributed hole recovery algorithms for wireless sensor networks," *IEEE Trans.*, vol. 14, no. 7, pp. 1397–1410, Jul. 2015.
- [10] K. Mougou, S. Mahfoudh, P. Minet, and A. Laouiti, "Redeployment of randomly deployed wireless mobile sensor nodes," in *Proc. IEEE Veh. Technol. Conf. (VTC Fall)*, Quebec City, QC, Canada, Sep. 2012, pp. 1–5.
- [11] M. Yang, Y. Cao, L. Tan, and J. Yu, "A new self-deployment algorithm in hybrid sensor network," in *Proc. 2nd Int. Symp. Intell. Inf. Technol. Appl.*, vol. 2, Shanghai, China, Dec. 2008, pp. 268–272.
- [12] X. Wang, J.-J. Ma, S. Wang, and D.-W. Bi, "Prediction-based dynamic energy management in wireless sensor networks," *Sensors*, vol. 7, no. 3, pp. 251–266, Mar. 2007.
- [13] S. Yang, M. Li, and J. Wu, "Scan-based movement-assisted sensor deployment methods in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 8, pp. 1108–1121, Aug. 2007.
- [14] B. S. Manoj, A. Sekhar, and C. Siva Ram Murthy, "On the use of limited autonomous mobility for dynamic coverage maintenance in sensor networks," *Comput. Netw.*, vol. 51, no. 8, pp. 2126–2143, Jun. 2007.
- [15] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Trans. Mobile Comput.*, vol. 5, no. 6, pp. 640–652, Jun. 2006.
- [16] C.-H. Wu, K.-C. Lee, and Y.-C. Chung, "A delaunay triangulation based method for wireless sensor network deployment," *Comput. Commun.*, vol. 30, nos. 14–15, pp. 2744–2752, Oct. 2007.
- [17] Z. Chen, F. H. T. Xia, F. Bu, and H. Wang, "A localization method for the Internet of Things," *Comput. Commun.*, vol. 63, pp. 657–674, Dec. 2011.
- [18] H. Chizari, M. Hosseini, T. Poston, S. A. Razak, and A. H. Abdullah, "Delaunay triangulation as a new coverage measurement method in wireless sensor network," *Sensors*, vol. 11, no. 3, pp. 3163–3176, Mar. 2011.
- [19] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, vol. 4, Perth, WA, Australia, Dec. 1995, pp. 1942–1948.
- [20] F. Vandenbergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, Jun. 2004.
- [21] H. Huang, J. Zhang, R. Wang, and Y. Qian, "Sensor node deployment in wireless sensor networks based on ionic bond-directed particle swarm optimization," *Appl. Math. Inf. Sci.*, vol. 8, no. 2, pp. 597–605, Mar. 2014.
- [22] J. Wang, C. Ju, H.-J. Kim, R. S. Sherratt, and S. Lee, "A mobile assisted coverage hole patching scheme based on particle swarm optimization for WSNs," *Cluster Comput.*, vol. 22, no. S1, pp. 1787–1795, Jan. 2019.
- [23] E. Piña-Covarrubias, A. P. Hill, P. Prince, J. L. Snaddon, A. Rogers, and C. P. Doncaster, "Optimization of sensor deployment for acoustic detection and localization in terrestrial environments," *Remote Sens. Ecol. Conservation*, vol. 5, no. 2, pp. 180–192, Oct. 2018.
- [24] X. Wang, S. Wang, and J. Ma, "Dynamic deployment optimization in wireless sensor networks," *Intell. Control Automat.*, vol. 344, pp. 182–187, Oct. 2006.
- [25] A. S. Majid and E. Joelianto, "Optimal sensor deployment in non-convex region using discrete particle swarm optimization algorithm," in *Proc. IEEE Conf. Control, Syst. Ind. Informat.*, Sep. 2012, pp. 109–113.
- [26] X. Wang and S. Wang, "Hierarchical deployment optimization for wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 7, pp. 1028–1041, Jul. 2011.
- [27] L. Kong, M. Zhao, X.-Y. Liu, J. Lu, Y. Liu, M.-Y. Wu, and W. Shu, "Surface coverage in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 234–243, Jan. 2014.
- [28] N. Unaldi, S. Temel, and V. K. Asari, "Method for optimal sensor deployment on 3D terrains utilizing a steady state genetic algorithm with a guided walk mutation operator based on the wavelet transform," *Sensors*, vol. 12, no. 4, pp. 5116–5133, Apr. 2012.
- [29] S. Temel, N. Unaldi, and O. Kaynak, "On deployment of wireless sensors on 3-D terrains to maximize sensing coverage by utilizing cat swarm optimization with wavelet transform," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 1, pp. 111–120, Jan. 2014.
- [30] H. R. Topcuoglu, M. Ermis, and M. Sifyan, "Positioning and utilizing sensors on a 3-D terrain part II—Solving with a hybrid evolutionary algorithm," *IEEE Trans. Syst., Man, Cybern., C (Appl. Rev.)*, vol. 41, no. 4, pp. 470–480, Jul. 2010.
- [31] C. K. Ng, C. H. Wu, W. H. Ip, and K. L. Yung, "A smart bat algorithm for wireless sensor network deployment in 3-D environment," *IEEE Commun. Lett.*, vol. 22, no. 10, pp. 2120–2123, Oct. 2018.

- [32] V. Kumar, P. Khanna, and S. Bisht, "Adaptive PSO based algorithm for optimal WSN deployment in 3 dimensional terrains," *Found. Comput. Sci.*, vol. 2, no. 2, pp. 1–6, May 2012.
- [33] J. M. Hereford, "A distributed particle swarm optimization algorithm for swarm robotic applications," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jun. 2006, pp. 1678–1685.
- [34] G. W. K. B. Conant, D. R. Parhi, and B. Raju, "Advance particle swarm optimization-based navigational controller for mobile robot," *Arabian J. Sci. Eng.*, vol. 39, pp. 6477–6478, Jun. 2014.
- [35] L. Perreault, M. P. Wittie, and J. Sheppard, "Communication-aware distributed PSO for dynamic robotic search," in *Proc. IEEE Symp. Swarm Intell.*, Dec. 2014, pp. 1–8.
- [36] C. Ozturk, D. Karaboga, and B. Gorkemli, "Probabilistic dynamic deployment of wireless sensor networks by artificial bee colony algorithm," *Sensors*, vol. 11, no. 6, pp. 6056–6065, Jun. 2011.
- [37] Y. B. Türkoğullari, N. Aras, İ. K. Altinel, and C. Ersoy, "Optimal placement, scheduling, and routing to maximize lifetime in sensor networks," *J. Oper. Res. Soc.*, vol. 61, no. 6, pp. 1000–1012, Jun. 2010.
- [38] R. V. Kulkarni and G. K. Venayagamoorthy, "Bio-inspired algorithms for autonomous deployment and localization of sensor nodes," *IEEE Trans. Syst., Man, Cybern., C (Appl. Rev.)*, vol. 40, no. 6, pp. 663–675, Nov. 2010.
- [39] X. Cao, H. Sun, and L. Guo, "Potential field hierarchical reinforcement learning approach for target search by multi-AUV in 3-D underwater environments," *Int. J. Control*, vol. 93, no. 7, pp. 1677–1683, Sep. 2018.
- [40] M. Locatelli and U. Raber, "Packing equal circles in a square: A deterministic global optimization approach," *Discrete Appl. Math.*, vol. 122, nos. 1–3, pp. 139–166, Oct. 2002.
- [41] A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Proc. Distrib. Auton. Robotic Syst.*, Fukuoka, Japan, Jun. 2002, pp. 299–308.
- [42] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization based on virtual forces," in *Proc. 22nd Annu. Joint Conf. IEEE Comput. Commun. Societies (INFOCOM)*, San Francisco, CA, USA, Mar. 2003, pp. 1293–1303.
- [43] Y. Zhou, N. Wang, and W. Xiang, "Clustering hierarchy protocol in wireless sensor networks using an improved PSO algorithm," *IEEE Access*, vol. 5, pp. 2241–2253, 2017.
- [44] T. Li, W. Liu, T. Wang, Z. Ming, X. Li, and M. Ma, "Trust data collections via vehicles joint with unmanned aerial vehicles in the smart Internet of Things," *Trans. Emerg. Telecommun. Technol.*, Apr. 2020, doi: [10.1002/ett.3956](https://doi.org/10.1002/ett.3956).
- [45] B. Jiang, G. Huang, T. Wang, J. Gui, and X. Zhu, "Trust based energy efficient data collection with unmanned aerial vehicle in edge network," *Trans. Emerg. Telecommun. Technol.*, Mar. 2020, doi: [10.1002/ett.3942](https://doi.org/10.1002/ett.3942).



YANZHI DU is currently pursuing the Ph.D. degree with the School of Optical Electrical and Computer Engineering, University of Shanghai for Science and Technology, China. Her current research interests include intelligent algorithms, sensor deployment, and target search.

• • •