Hindawi
Journal of Sensors
Volume 2022, Article ID 7483148, 20 pages
https://doi.org/10.1155/2022/7483148

Research Article

Multiobjective Optimization Strategy of WSN Coverage Based on IPSO-IRCD

Juan Wu[1] **Hongbing Li**[2] **Lei Luo**[3] **Jun Ou**[1] **and Ying Zhang**[2]

[1]*Chongqing Key Laboratory of Geological Environment Monitoring and Disaster Early-Warning in Three Gorges Reservoir Area, Chongqing Three Gorges University, Wanzhou, 404120, China*
[2]*Key Laboratory of Intelligent Information Processing and Control of Chongqing Municipal Institutions of Higher Education, Chongqing Three Gorges University, Wanzhou, 404120, China*
[3]*School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China*

Correspondence should be addressed to Hongbing Li; sxxylhb@163.com

Received 4 April 2022; Revised 8 May 2022; Accepted 12 May 2022; Published 24 May 2022

Academic Editor: Cong-Bin Fan

The nonuniform distribution characteristic of randomly deployed mobile nodes will lead to the coverage hole and redundancy in wireless sensor networks (WSNs). To solve this problem, we propose a multiobjective optimization algorithm for WSN based on Improved Particle Swarm Optimization-Increment of the Ratio of Coverage Rate to Move Distance (IPSO-IRCD), and a network node coverage optimization model is formulated to maximize the coverage rate of the target area while reducing the moving distance of nodes. In each iteration of IPSO, the population fitness value is calculated and compared with the historical optimal value, when the arbitrary dimensional location information of each node is updated, which can avoid the standard PSO algorithm loses the optimal solution, and IPSO will determine the candidate deployment location of nodes. Based on which, IRCD node coverage scheduling optimization algorithm is proposed, so that the final deployment location can be determined iteratively by calculating IRCD of nodes. Simulation results indicate that, for the nodes initial coverage state follows random distribution and Gaussian distribution, IPSO-IRCD can, respectively, improve 4.6% and 7.4% coverage ratio compared with the suboptimal algorithm in other five similar algorithms and reduce 809.59 m and 626.63 m nodes moving distance.

1. Introduction

Wireless sensor networks (WSNs), which consist of a series of deployed sensor nodes in the monitoring area, are a multihops and self-organization wireless network system. It can real-time surveil, perceive, and collect the monitored object's information [1]. For large-scale WSN, numerous sensor nodes are usually randomly distributed in the target area, resulting in uneven node distribution and increasing network cost [2]. Therefore, to design a reasonable coverage strategy of sensor nodes is vital for improving the system performance of WSN.

One of the fundamental problems in WSN is node coverage, as it has a direct impact on the sensors moving distance and energy consumption [3]. The key challenge to solve the coverage problem is how to optimize the deployment algorithm and adjust the deployment location of sensor nodes, so that the quality of service in the target area can be guaranteed. Based on which, an efficient method to realize WSN coverage control is to deploy mobile sensor nodes in target area, where the mobile sensor nodes can be scheduled to the areas with low coverage rate, thus healing coverage holes and enhancing coverage of the target area [4].

On the other hand, the sensor node will consume a certain energy when moving in the target area, and its energy consumption is positively correlated with the moving distance [5]. For the WSN composed of mobile sensor nodes, the node moving distance should be minimized to reduce the energy consumption, while ensuring the coverage rate of the target area. Meanwhile, the two metrics are affected by the sensor node's sensing range. It is obvious that the larger the sensing range, the higher the node's energy consumption and network coverage rate. A smaller sensing range leads to lower energy consumption and network coverage rate [6].

Network coverage is an important indicator to evaluate WSN performance. In particular, coverage rate and node moving distance have been used as the performance monitors of WSN [7]. In this paper, we investigate the sensor node coverage problem with the objective of coverage rate and node moving distance and propose a multiobjective optimization algorithm for WSN based on Improved Particle Swarm Optimization-Increment of the Ratio of Coverage Rate to Move Distance (IPSO-IRCD).

1.1. Related Works. Many researchers have carried out researches in the field of WSN coverage, and some typical intelligent algorithms are widely used to solve the node coverage problem in WSN, such as simulated annealing algorithm, genetic algorithm, standard particle swarm optimization algorithm (SPSO), and sparrow search algorithm (SSA) [8, 9]. For WSN coverage optimization problem in different scenarios, the work in [10–24] adopted the improved intelligent algorithms to optimize the node coverage with the goal of network coverage rate. Aiming to improve the algorithm convergence and maximize the coverage rate, an enhanced sparrow search algorithm (ESSA) for WSN coverage research was proposed in [10], where the convergence factor, Cauchy operator, and cross-border processing method were introduced to modify the basic SSA, so that the performance of the algorithm could be further improved. In [11], an improved whale optimization algorithm for improving network coverage was proposed. In order to maximize the network coverage rate and maintain connectivity for WSN, a combination of the distributed PSO and virtual force algorithm was carried out in [12]. Coverage and energy are indispensable for WSN to perform specific monitoring tasks. The work in [13] focused on a balance between coverage rate and energy cost based on PSO. In [14], PSO and genetic algorithm were, respectively, introduced to optimize the coverage rate and energy consumption. To reduce the energy consumption of nodes close to the sink, an enhanced power-efficient gathering in sensor information systems algorithm was proposed to adjust the communication distance and range of nodes in [15], and a routing protocol based on smart energy management and throughput maximization for clustered WSNs was proposed in [16] to increase the network lifetime. In [17], a clustering by social spider optimization and fuzzy logic algorithm and mobile sink was proposed for maximizing energy efficiency. The global levy flight of cuckoo search with PSO was proposed in [18] to get improved network performance incorporating balanced energy dissipation and results in the formation of optimum number of clusters and minimal energy consumption. An energy-efficient algorithm for point and area sweep-coverage in WSN was proposed in [19], where the goal was to visit the point of interest and cover the area of interest with few number of mobile sensor nodes. The work in [20] focused on the issue of designing data routing techniques in WSNs to minimize the energy consumed during the data communication. To save energy of the mobile WSN, an improved low-energy adaptive clustering hierarchy protocol for mobile WSN was proposed in [21] to prolong the network lifetime and reduce the packet loss. Similarly, a clustering algorithm for energy-efficient adaptive scheme was proposed in [22] to extend the life time of WSNs by minimizing the distance of data communication. Furthermore, a fuzzy-logic-based clustering approach with an extension to the energy predication was proposed in [23] to prolong the lifetime of WSNs by evenly distributing the workload. In addition, machine learning algorithms were also utilized to address the issue for WSNs in [24].

Note that PSO can be simply implemented and has high-quality solution; many scholars have applied it to WSN and achieved the great results in WSN coverage problem. However, since SPSO is prone to fall into the dilemma of local optimal and poor convergence in WSN coverage optimization, some researches have conducted IPSO to improve network performance [25–30]. A PSO based on dynamic acceleration coefficients (PSO-DAC) algorithm was proposed in [25] to maximize the coverage rate. The algorithm adopted decreasing inertia weight coefficients and introduced dynamic acceleration coefficients, which overcame the slow solving speed and the dilemma to go down in the local extreme value. In [26], the method to handle the cross-border particles was generalized to improve the applicability of PSO. Then, the explorative capability enhancement PSO algorithm was proposed to improve the performance of PSO. Combined with the idea of virtual force, an adaptive virtual force PSO algorithm was proposed in [27], and the mobile nodes were scheduled to perform the node redeployment location distribution. In order to improve the coverage performance of WSN, a quasiphysical PSO algorithm based on inertia weight was presented in [28]. The algorithm fostered the convergence compared with the SPSO, while a part of the nodes were overstepped the boundary. Since the insufficient coverage of specified target points in coverage distribution of WSN based on PSO, an IPSO algorithm based on virtual potential field was proposed in [29] to optimize coverage distribution of WSN. However, the researchers not investigated the convergence and coverage rate compared with other algorithms, and the coverage performance can be further improved. In [30], a hybrid PSO-butterfly algorithm (HPSBA) was proposed for solving the node optimization deployment problem in WSN. This algorithm designed logistic mapping and adaptive adjustment strategies to control parameter values, which could effectively improve the coverage rate.

The above improved algorithms have produced ideal effect on node deployment and coverage optimization, while the algorithm convergence and network coverage remain to be further improved. Meanwhile, it also indicates that applying the improved intelligent algorithms to WSN coverage problem is a promising strategy.

Besides the network coverage rate, the moving distance of sensor nodes is also an important indicator to evaluate the performance of WSN [31–35]. In [31], aiming to reduce the moving distance and find the optimal location for sensor nodes scheduling, the sensor nodes were efficiently deployed by using simulated annealing and PSO. Then, the sensor nodes were scheduled by applying the simulated annealing and Dempster Shafer Theory. The sensing field was partitioned into multiple grids, and the coverage rate of each grid was calculated in [32]. Then, PSO was utilized to guide the nodes to move to the grid with lower coverage rate, so as to reduce the moving distance of the nodes effectively. The work in [33] concerned with the nodes coverage hole, and a node redeployment strategy based on the firefly algorithm was proposed, where the reduction of moving distances of nodes was achieved through moving the redundant nodes effectively. In [34], sensor node deployment problem was decomposed into target coverage and network connection subproblems, and the target coverage subproblem was solved by the proposed clique partition and greedy algorithm based on Voronoi partition of the deployment region, in order to reduce the total movement distance of sensor nodes. Moreover, to shorten the average moving distance of nodes, an area coverage optimization algorithm of WSN based on virtual force perturbation and cuckoo search was investigated in [35].

1.2. Motivation and Contribution. In this paper, we take coverage rate and node moving distance as our optimization target of sensor nodes deployment in WSN. We consider the multiple mobile sensor node deployment scenario, in which, the coverage rate and moving distance of sensor nodes are directly related to the final deployment location of mobile sensor nodes. If the coverage ranges are overlapped for multiple sensor nodes or the coverage holes are existed in target area, sensor nodes need to move to the areas with low coverage rate. Obviously, the coverage rate and moving distance of sensor nodes are correlative. As we have summarized the related work above, there are several works focusing on the coverage rate and node moving distance optimization. However, there are few papers considering the coverage rate and moving distance of sensor nodes jointly. Moreover, the increment of coverage rate induced by moving sensor nodes from initial deployment location to candidate deployment location is possibly too low or even unchanged; thus, the IRCD is also an important element that affects the performance of WSN. Most works in WSN coverage optimization have ignored the IRCD of sensor nodes.

Motivated by this, we jointly consider the two metrics of coverage rate and moving distance of sensor nodes in WSN coverage optimization and propose a multiobjective optimization algorithm for WSN based on IPSO-IRCD. The main contributions of this paper are summarized as follows:

(i) We study a multiobjective optimization problem for WSN. To characterize the fundamental trade-off between the coverage rate and the moving distance of sensor nodes, we define the objective function as the weighted sum of the two metrics, to maximize the coverage rate while reducing the moving distance of the sensor nodes

(ii) We propose an IPSO-IRCD algorithm to solve the formulated multiobjective optimization problem. Firstly, to maximize the coverage rate, IPSO is utilized to determine the candidate deployment location of mobile nodes by calculating the population fitness value and comparing with the historical optimal value. Then, to reduce the moving distance of the sensor nodes, the proposed node scheduling algorithm based on IRCD is utilized to determine the final deployment location of mobile nodes. Finally, the corresponding mobile nodes are moving from initial deployment location to the final deployment location

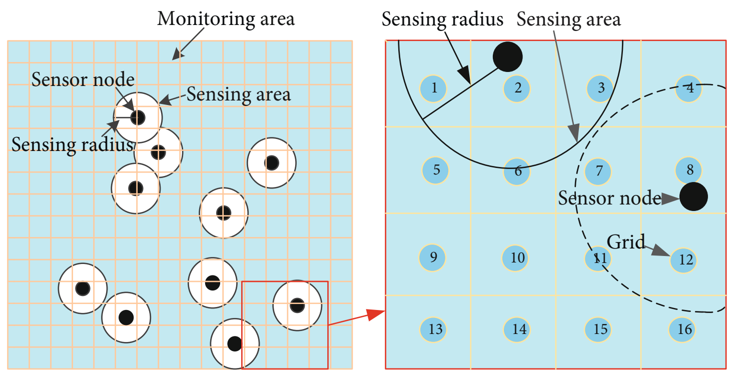
(iii) Since SPSO may drop the optimization solutions in the process of updating the dimensional information of the particle, IPSO is utilized to produce more optimization solutions of sensor nodes and higher coverage rate, by calculating the fitness value of each node after each location status is updated. However, the search space of the optimization solution is increased in IPSO, thus increasing the computational complexity for IPSO

(iv) Simulation results indicate that the proposed algorithm obtains the suboptimal solution with a dozen iterations. Moreover, for the nodes initial coverage state follows random distribution and Gaussian distribution, IPSO-IRCD can, respectively, improve 4.6% and 7.4% coverage ratio compared with the suboptimal ESSA algorithm proposed in [10] and reduce 809.59 m and 626.63 m nodes moving distance

The rest of this paper is organized as follows. The system model is presented in Section 2. In Section 3, we first briefly illustrate the SPSO, and then, the IPSO is proposed to determine the candidate deployment location of sensor nodes. Section 4 presents the node scheduling algorithm based on IRCD. Algorithm computational complexity analysis is summarized in Section 5. Simulation results and analysis are provided in Section 6. Finally, conclusions are drawn in Section 7.

2. System Model

2.1. Network Model. As shown in Figure 1, we consider a WSN composed of multiple sensor nodes. Suppose that the set of sensor nodes in the monitoring area is $S = \{S_j\}, j = 1, \cdots, n$, where n is the number of sensor nodes. All sensor nodes are randomly deployed in the monitoring area, and they firstly switch to the sleep mode and remain static after deployment. Then, we awake the corresponding sensor nodes from sleep mode to fix coverage holes while the final deployment locations of the nodes are determined. Furthermore, we assume that all sensor nodes are with the same sensing radius R_s and communication radius $R_c \geq 2R_s$, and each sensor node in the monitoring area can acquire its own location information from self-localization of WSN and then broadcasts to the entire network. If a point in the monitoring area is located in the sensing range of the sensor node, it is considered that the point is covered by the sensor node. Therefore, the monitoring area can be divided into



(a) Grid discretization for the monitoring area (b) Magnification for the grid points in (a)

Figure 1: Sensor nodes deployment model of WSN.

covered area and uncovered area. Specifically, the covered area is that covered by at least one sensor node, and the uncovered area is that not covered by any sensor node.

2.2. Coverage Sensing Model. Suppose that the monitoring area A is a finite two-dimensional plane, which is discretized into $A \times B$ grid points, and each grid point is with the same size. Without loss of generality, sensor nodes are deployed at different locations of grid points to supervise the monitoring area. The center coordinate of the grid point G_i is denoted as (x_i^G, y_i^G), and the coordinate of the sensor node S_j is denoted as (x_j^S, y_j^S). Hence, the Euclidean distance between the sensor node S_j and the grid point G_i is $d(S_j, G_i) = \sqrt{(x_j^S - x_i^G)^2 + (y_j^S - y_i^G)^2}$. Based on Boolean sensing model, the probability of grid points G_i covered by sensor node S_j is [13]

$$P(S_j, G_i) = \begin{cases} 1, d(S_j, G_i) \le R_s 1, \\ 0, \text{otherwise}. \end{cases} \quad (1)$$

Meanwhile, G_i might be covered by multiple sensor nodes simultaneously, and the probability that G_i is covered by at least one sensor node is [13]

$$P(S, G_i) = 1 - \prod_{j=1}^{n} \left(1 - P(S_j, G_i)\right). \quad (2)$$

Therefore, the total number of grid points covered in the monitoring area is $N = \sum_{i=1}^{A \times B} P(S, G_i)$.

2.3. Objective Function. Network coverage rate is an important indicator of WSN coverage performance, which is defined as the ratio of the effective coverage grid points to the total grid points in the monitoring area [13], i.e.,

$$\eta = \frac{N}{A \times B}, \quad (3)$$

where η is the coverage rate of the monitoring area and $A \times B$ is the total number of grid points in the monitoring area.

In addition, by moving the sensor nodes can improve the coverage rate of the monitoring area. However, due to the energy consumption induced by the movement of sensor nodes, the coverage performance of WSN is limited. Therefore, we jointly consider the coverage rate and the moving distance of sensor nodes and define the objective function as the weighted sum of the two metrics. It is assumed that the initial location of all nodes in S is denoted as (x_{ini}^S, y_{ini}^S), and the final deployment location is expressed as (x_{fin}^S, y_{fin}^S). Furthermore, the objective function $g(x)$ can be formulated as

$$g(x) = \alpha\eta + \beta\lambda \frac{1}{d(S)}, \quad (4)$$

where α and β are the weighting factors with $\alpha + \beta = 1$ and λ is the distance trade-off coefficient to ensure η and $1/d(S)$ are comparable. $d(S) = \sqrt{(x_{ini}^S - x_{fin}^S)^2 + (y_{ini}^S - y_{fin}^S)^2}$ is the moving distance of all sensor nodes.

3. IPSO-Based Node Candidate Deployment Location Optimization Algorithm

3.1. SPSO. SPSO is a swarm intelligence algorithm that simulates the behavior of bird flock preying. Suppose that bird flock is randomly distributed in the monitoring area where only one piece of food in this area, the task of bird flock is to find that food without knowing its location and distance. Each individual bird updates its current location by means of history information of individual and bird flock. As expected, the bird flock confirms the exact location of food by updating location iteratively. Motivated by this, SPSO algorithm is proposed to solve optimization problem. Since SPSO algorithm uses internal information to update speed and location, the parameters of SPSO are fewer and the implementation is easier. Moreover, SPSO algorithm is mainly applied to continuous optimization problem and may converge to the optimal solution more rapidly.

Therefore, SPSO algorithm is suitable for WSN coverage optimization problem [36].

The particle swarm of SPSO is randomly located in the decision space, and each particle is a potential solution to optimization of problem. These particles are evaluated by fitness value which is decided by optimization objective function, and each particle decides its own speed and location according to history best fitness value of its own and population. Then, each particle moves in entire decision space to search the optimization solution. In other words, each particle exchanges information with other particles to get heuristic information and guide the movement of the whole population, thus getting optimum solution to optimization problem.

It is assumed that the decision space is D, population size of particle swarm is M, wherein the location of particle m at iteration k is denoted as $x_m^k = (x_{m1}^k, x_{m2}^k, \cdots, x_{md}^k)$, and speed of particle m is $v_m^k = (v_{m1}^k, v_{m2}^k, \cdots, v_{md}^k)$. The optimal location of particle m experienced in the iteration is $b_m^k = (b_{m1}^k, b_{m2}^k, \cdots, b_{md}^k)$. Therefore, based on the coverage rate η, the global optimal location for all particles in the population can be calculated as [36]

$$g^k = \left\{ b_l^k \middle| \max \left\{ \eta\left(b_l^k\right) \right\}, l = 1, 2, \cdots, M \right\}. \tag{5}$$

The speed and location of particle m in the iteration are updated as follows [36]:

$$v_m^{k+1} = \omega v_m^k + c_1 r_1 \left(b_m^k - x_m^k\right) + c_2 r_2 \left(g^k - x_m^k\right), \tag{6}$$

$$v_m^{k+1} = v_m^{k+1} = \begin{cases} v_{\max}, v_m^{k+1} > v_{\max}, \\ v_{\min}, v_m^{k+1} < v_{\min}, \end{cases} \tag{7}$$

$$x_m^{k+1} = x_m^k + v_m^{k+1}, \tag{8}$$

where ω is the inertia weight, and it can be employed to balance the global search and local search ability of SPSO. c_1 and c_2 are learning factors with the value in range $[1, 2]$. r_1 and r_2 are two random numbers in range $[0, 1]$. $v_{\max}$ and $v_{\min}$ are, respectively, the maximum and minimum flight speed of particles, which can be used to limit the flight speed of particles to improve the search ability. In addition, the optimal location of particle m experienced in the iteration is updated as

$$b_m^{k+1} = b_m^{k+1} = \begin{cases} b_m^k, \eta\left(x_m^{k+1}\right) \le \eta\left(b_m^k\right) b_m^k, \\ x_m^{k+1}, \eta\left(x_m^{k+1}\right) > \eta\left(b_m^k\right). \end{cases} \tag{9}$$

3.2. IPSO. It is worth noting that each particle P in IPSO contains all sensor nodes, that is, $P = \{S_1, S_2, \cdots, S_n\}$. In particular, each scheme of particle represents a feasible solution, i.e., a node deployment strategy for WSN. All nodes in each particle are moving in two-dimensional space, and their speed and location information are, respectively, given by

$$v_m = \left(v_{m1,x}, v_{m1,y}, v_{m2,x}, v_{m2,y}, \cdots, v_{mn,x}, v_{mn,y}\right), \tag{10}$$

$$x_m = (x_{m1}, y_{m1}, x_{m2}, y_{m2}, \cdots, x_{mn}, y_{mn}), \tag{11}$$

where $v_{mn,x}$ and $v_{mn,y}$ are, respectively, the speed of the nth node in the mth particle along with the x-axis and y-axis directions, x_{mn} and y_{mn} denote the x-coordinate and y-coordinate of the nth node in the mth particle, respectively. Since the movement of the sensor nodes can improve the coverage rate of WSN, the fitness function $f(x)$ of IPSO is defined as coverage rate, i.e.,

$$f(x) = \eta = \frac{\sum_{i=1}^{A \times B} P(S, G_i)}{A \times B}. \tag{12}$$

It should be mentioned that the fitness values are calculated and compared when all the dimensional information of each particle are updated according to Equations (6)–(8) in each iteration of the SPSO, that is, the speed and location information of all nodes in the particle are updated to determine the fitness value. However, the method may drop the optimization solutions in the process of updating the dimensional information of the particle, which reduces the search space of the optimization solution. Therefore, the convergence of the algorithm and the coverage rate of WSN are both decreased. Moreover, the redundant moving distances of some sensor nodes are increased simultaneously.

The basic idea of IPSO is that the location status of each node in the particle is firstly updated in each iteration, which include three location statuses, i.e., updating the location information of each node in x-axis and y-axis directions independently and jointly. Then, we calculate the fitness value of each node after each of three location statuses is updated, while the locations of other nodes remain unchanged, and the fitness values for three location statuses are, respectively, denoted by $f(x_{mj}^{k+1})$, $f(y_{mj}^{k+1})$, and $f(x_{mj}^{k+1}, y_{mj}^{k+1})$, $j = 1, \cdots, n$. Finally, the optimal fitness value $f^*(x_m^{k+1})$ is calculated by the updated fitness value of all nodes in each location status, i.e.,

$$f^*\left(x_m^{k+1}\right) = \max_{j=1,\cdots,n} \left(\max \left\{ f\left(x_{mj}^{k+1}\right), f\left(y_{mj}^{k+1}\right), f\left(x_{mj}^{k+1}, y_{mj}^{k+1}\right) \right\} \right). \tag{13}$$

In addition, $f^*(x_m^{k+1})$ is compared with the historical fitness value. In case that $f^*(x_m^{k+1})$ is better than its historical optimal fitness value $f^b(x_m)$, the location of the node that corresponds to the optimal fitness value and the historical optimal value of the particle are updated, i.e.,

$$x_{mj}^{k+1} = \begin{cases} x_{mj}^k + v_{mj,x}^{k+1}, f^*\left(x_{mj}^{k+1}\right) > f^b(x_m), \\ x_{mj}^k, f^*\left(y_{mj}^{k+1}\right) > f^b(x_m) x_{mj}^k, \\ x_{mj}^k + v_{mj,x}^{k+1}, f^*\left(x_{mj}^{k+1}, y_{mj}^{k+1}\right) > f^b(x_m), \end{cases} \tag{14}$$

$$y_{mj}^{k+1} = \begin{cases} y_{mj}^{k}, f^*\left(x_{mj}^{k+1}\right) > f^b(x_m), \\ y_{mj}^{k} + v_{mj,y}^{k+1}, f^*\left(y_{mj}^{k+1}\right) > f^b(x_m), \\ y_{mj}^{k} + v_{mj,y}^{k+1}, f^*\left(x_{mj}^{k+1}, y_{mj}^{k+1}\right) > f^b(x_m), \end{cases} \quad (15)$$

$$f^b(x_m) = f^*\left(x_m^{k+1}\right). \quad (16)$$

The above process is repeated for all sensor nodes in the particle until $f^*(x_m^{k+1})$ is no longer better than the historical optimal fitness value $f^b(x_m)$. Meanwhile, the optimal location b_m^{k+1} that the particle experienced is updated according to Equation (9). Similarly, the above process is also repeated for all particles in the population, and the location of sensor node and the historical optimal value of the particle are updated according to Equations (13)–(16). Finally, the global optimal locations g^{k+1} of all particles are updated according to Equation (5).

For the IPSO iteration, the search ability of IPSO is affected by the inertia weight ω. In particular, the global search ability of IPSO can be effectively improved with larger ω, while the smaller ω is conducive to local search. Therefore, Sigmoid function is utilized to modify the inertia weight, and a nonlinear decreasing inertia weight strategy is proposed, that is,

$$\omega(k) = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \frac{1}{1 + \exp(a - bk)}, \quad (17)$$

where $\omega_{\max}$ and $\omega_{\min}$ are, respectively, the maximum and minimum values of inertia weight, a and b are constants, and k is the number of current iterations.

Besides, the search ability of IPSO is also affected by the learning factors c_1 and c_2. In order to provide different evolutionary strategies for different particles in the same generation, we adopt adaptive dynamic variation to modify the learning factors, i.e.,

$$c_{1,m}^{k} = 4 - c_{2,m}^{k}, \quad (18)$$

$$c_{2,m}^{k} = c_{\min} + (2 - c_{\min}) * \exp\left(\frac{-f\left(b_m^k\right)}{f\left(g^k\right)}\right), \quad (19)$$

where $c_{\min}$ is the minimum value of learning factor, $f(b_m^k)$ is the current fitness value of particle m, and $f(g^k)$ is the global optimal fitness value of the population.

3.3. Candidate Deployment Location Optimization Strategy for Mobile Sensor Nodes. To solve the problem of coverage redundancy and holes in the monitoring area and improve the network coverage, IPSO is proposed to calculate the candidate deployment location of mobile sensor nodes iteratively. In each iteration, the current location of the particle is regarded as the target location of the mobile sensor nodes, and the fitness function value $f(x)$ of the particle at the current location is calculated. Moreover, to update the historical optimal fitness function value and population optimal value of particles, $f(x)$ is compared with the two values. In case that the algorithm reaches the maximum number of iterations $k = K$, the location of the optimal particle in population is set as the candidate deployment location of mobile sensor nodes. Based on the idea of IPSO proposed in Section 3.2, the steps to determine the candidate deployment location of mobile sensor nodes are as follows:

(i) Initialize the number n and location (x_{ini}^S, y_{ini}^S) of mobile sensor nodes, population size M, the maximum number of iterations K, maximum\minimum inertia weight $\omega_{\max} \setminus \omega_{\min}$, minimum learning factor $c_{\min}$, particle location x_m^0 and speed v_m^0, and so on

(ii) Calculate fitness value $f(x)$ of each particle in population, i.e., the network coverage rate η. Take current location x_m^k and fitness value $f(x_m^k)$ of each particle as its historical optimal location and fitness value. The particle location x_m^k with the optimal fitness value max $\{f(x_m^k)\}$ is regarded as the historical optimal location g^k of the population, and the corresponding fitness value max $\{f(x_m^k)\}$ is set as the optimal fitness value of the population

(iii) Update the inertia weight ω and learning factor c_1 and c_2 according to Equations (17), (18), and (19)

(iv) Update the speed v_m^{k+1} and location x_m^{k+1} of particles according to Equations (6) and (8)

(v) Calculate fitness values $f(x_{mj}^{k+1})$, $f(y_{mj}^{k+1})$, and $f(x_{mj}^{k+1}, y_{mj}^{k+1})$ of each node in particle m according to Equation (12)

(vi) Calculate the optimal fitness value $f^*(x_m^{k+1})$ of all nodes in particle m according to Equation (13)

(vii) If $f^*(x_m^{k+1})$ is better than its historical optimal fitness value $f^b(x_m)$, the node location corresponding to the optimal fitness value and the historical optimal value $f^b(x_m)$ of the particle are updated according to Equations (14)–(16). Otherwise, the optimal location b_m^k experienced by the particle m is updated according to Equation (9)

(viii) If all particles have updated their historical optimal value $f^b(x_m)$ and optimal location b_m^k, then the global optimal location g^k of the population is updated according to Equation (5). Otherwise, return to step (iii) to update the historical optimal value and optimal location of the next particle

(ix) If the algorithm reaches the maximum number of iterations $k = K$, then the historical optimal location g^k and optimal fitness value max $\{f^b(x_m), m = 1, 2, \cdots, M\}$ of the population are obtained, and

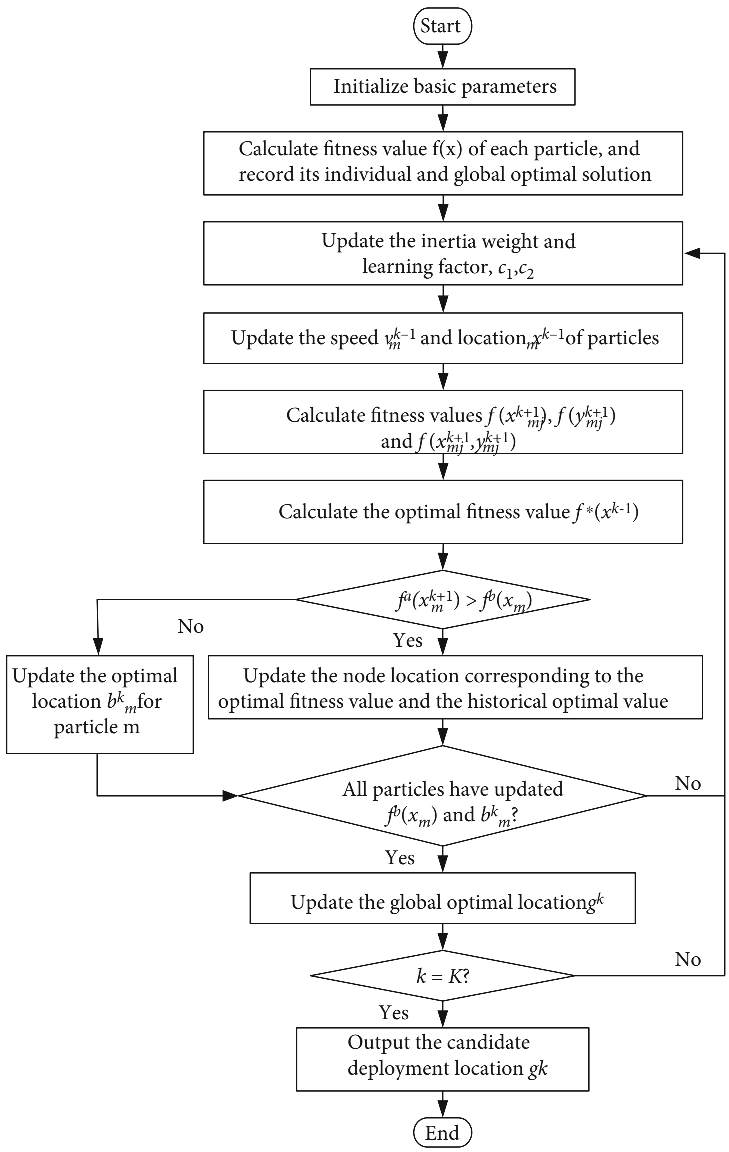
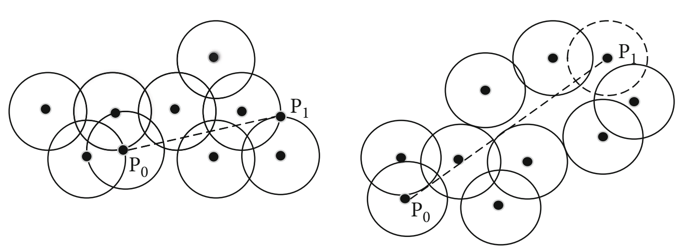


Figure 2: Algorithm procedure of node candidate deployment location optimization based on IPSO.



(a) Small increasement of coverage rate (b) Long moving distance of nodes

Figure 3: Optimization of the number of mobile sensor nodes.

the historical optimal location g^k of the population is set as the candidate deployment location of mobile sensor nodes; otherwise, go to step (iii)

The procedure of IPSO-based node candidate deployment location optimization algorithm is shown in Figure 2.

4. IRCD-Based Node Coverage Scheduling Optimization Algorithm

The object of node coverage scheduling optimization is to reduce the number and distance of mobile sensor nodes. It is assumed that the node is virtually moving to the candidate

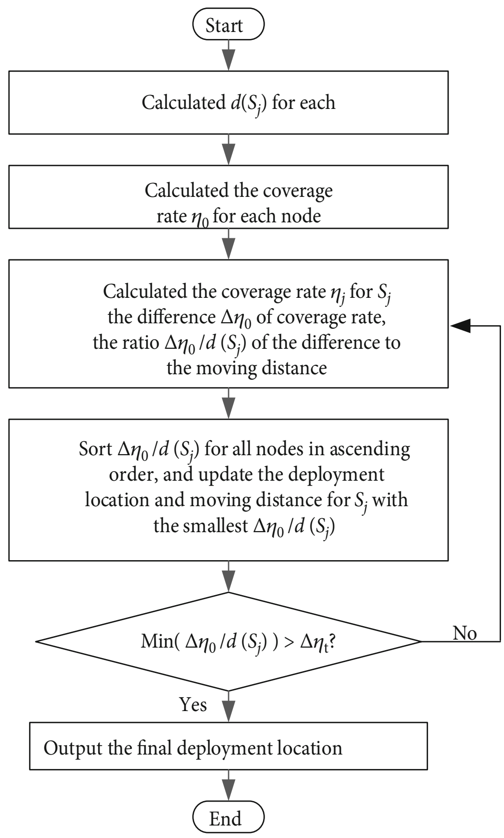


Figure 4: Node scheduling algorithm procedure based on IRCD.

deployment location when the candidate deployment location is determined. Moreover, the node will awake from the sleep mode and move to the final deployment location. As shown in Figure 3, there are two situations while the mobile sensor node S_j moving from P_0 to P_1: (a) the increment of network coverage rate is small; (b) the moving distance of node is too long, that is, the kind of movement is redundant. In this case, the mobile sensor node S_j is set as sleep mode, i.e., S_j keeps static status, so as to reduce the number and distance of mobile sensor nodes.

To further reduce the moving distance of mobile sensor nodes, a node scheduling optimization algorithm based on IRCD is proposed in this paper. The main idea of IRCD is firstly to calculate the network coverage rate η_0, where the mobile sensor nodes are located at the candidate deployment location. Then, we calculate the network coverage rate η_j when node S_j is in the initial location, and other nodes are in the candidate deployment location. Finally, the increment of coverage rate $\Delta\eta_{0j}$ is calculated based on η_0 and η_j, and the moving distance for S_j from the initial location to the candidate deployment location is also be calculated. Similarly, for each independent node in S, the coverage rate η_i, $i \neq j$ is repeatedly calculated when independent node is in the initial location and other nodes are in the candidate deployment location, until the coverage rates for all independent nodes in the initial location are calculated. Then, the increment of coverage rate $\Delta\eta_{0i}$ and the moving distance $d(S_i)$ for each independent node are calculated.

Table 1: Simulation parameters.

Parameters	Value
Number of sensor nodes (n)	40
Population size (M)	20
Sensing radius (R_s)	5 m
Maximum number of iterations (K)	80
Maximum\minimum inertia weight ($\omega_{\max}$, $\omega_{\min}$)	0.9, 0.4
Minimum learning factor ($c_{\min}$)	1.2
Maximum\minimum flight speed ($v_{\max}$, $v_{\min}$)	3, -3
The value of threshold ($\Delta\eta_t$)	0.05%
Weighting factors (α, β)	0.8, 0.2

Furthermore, the ratio of coverage rate increment and moving distance $\Delta\eta_{0j}/d(S_j)$ for all sensor nodes are sorted in ascending order. Meanwhile, we record the S_j with the smallest $\Delta\eta_{0j}/d(S_j)$, i.e., $S_j = \underset{j\in\{1,2,\cdots,n\},S_j\in S}{\operatorname{argmin}} \Delta\eta_{0j}/d(S_j)$, and update the candidate deployment location and moving distance $d(S_j)$ for S_j. The above process is repeated until the minimum $\Delta\eta_{0j}/d(S_j)$ is greater than the threshold $\Delta\eta_t$ for the increment of the ratio between coverage rate and moving distance, and the final node deployment location and total moving distance are obtained. In this case, the corresponding mobile sensor nodes are scheduled to the final deployment location. The steps of IRCD-based node scheduling algorithm are as follows:

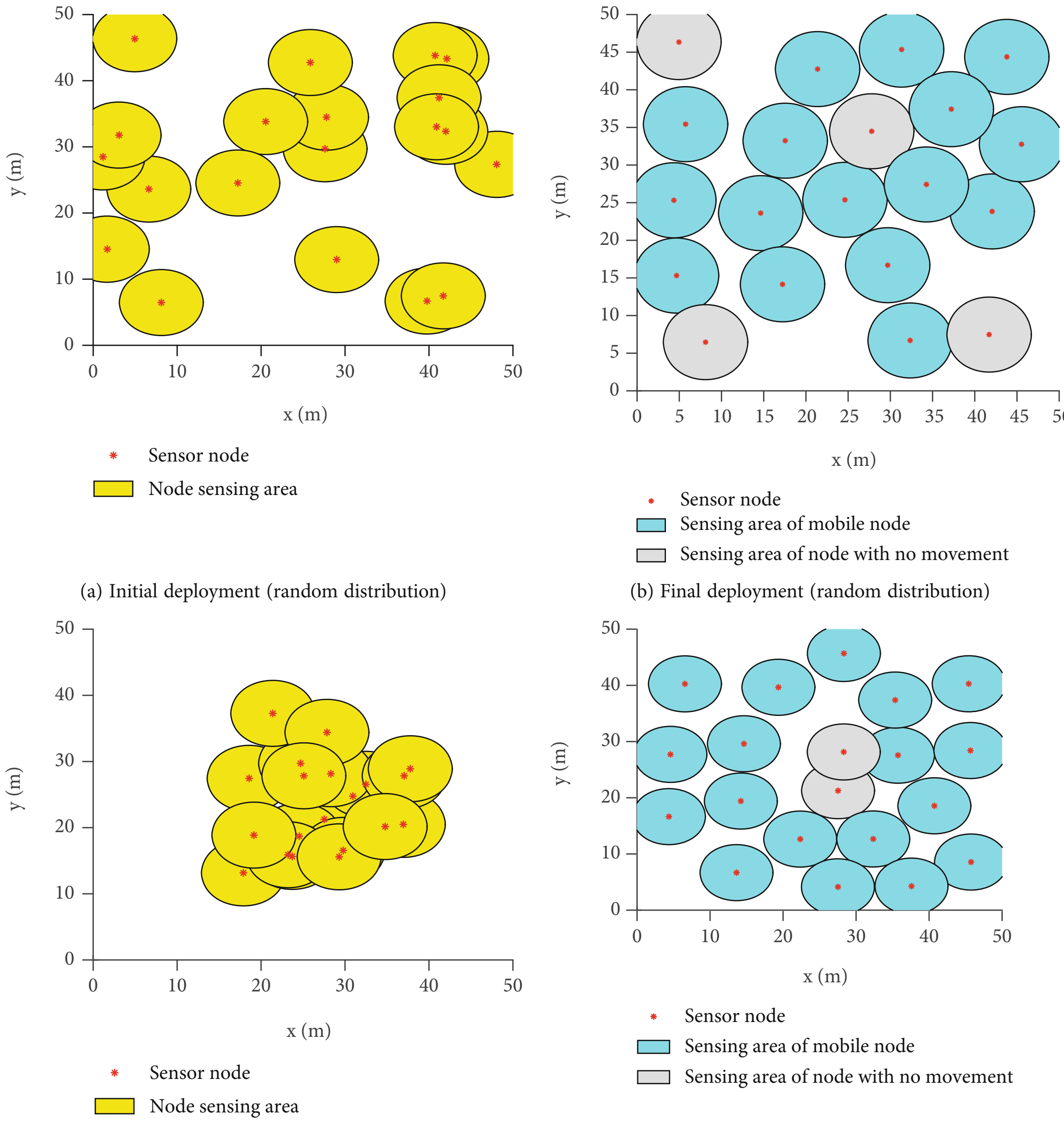
(i) Calculate the distance $d(S_j), S_j \in S$ from the initial location to the candidate deployment location of each node

(ii) Calculate the coverage rate η_0 when each node is located at the candidate deployment location according to Equation (3)

(iii) Calculate the coverage rate η_j when one node $S_j \in S$ is in the initial location and all other nodes are in the candidate deployment location according to Equation (3), and the difference $\Delta\eta_{0j} = \eta_0 - \eta_j$ between η_0 and η_j, as well as the ratio $\Delta\eta_{0j}/d(S_j)$ of the difference to the moving distance

(iv) Repeat step (iii) until the η_i, $i \neq j$ of each node in the initial location, and the difference $\Delta\eta_{0i}$ between η_0 and η_i, as well as the ratio $\Delta\eta_{0i}/d(S_i)$ of the difference to the moving distance are calculated

(v) Sort $\Delta\eta_{0j}/d(S_j)$ for all nodes in ascending order, and the candidate deployment location for S_j with the smallest $\Delta\eta_{0j}/d(S_j)$ is updated to the initial location, i.e., $(x^S_{j,fin}, y^S_{j,fin}) = (x^S_{j,ini}, y^S_{j,ini})$. Then, the corresponding moving distance $d(S_j)$ is updated to 0



(a) Initial deployment (random distribution)

(b) Final deployment (random distribution)

(c) Initial deployment (Gaussian distribution)

(d) Final deployment (Gaussian distribution)

FIGURE 5: Nodes deployment ($n = 20$).

(vi) For the node with $d(S_j) \neq 0$, if the minimum $\Delta\eta_{0j}/d(S_j)$ is greater than the threshold $\Delta\eta_t$, the algorithm ends and outputs the final node deployment location (x^S_{fin}, y^S_{fin}) and total moving distance $d(S)$; otherwise, go to step (iii)

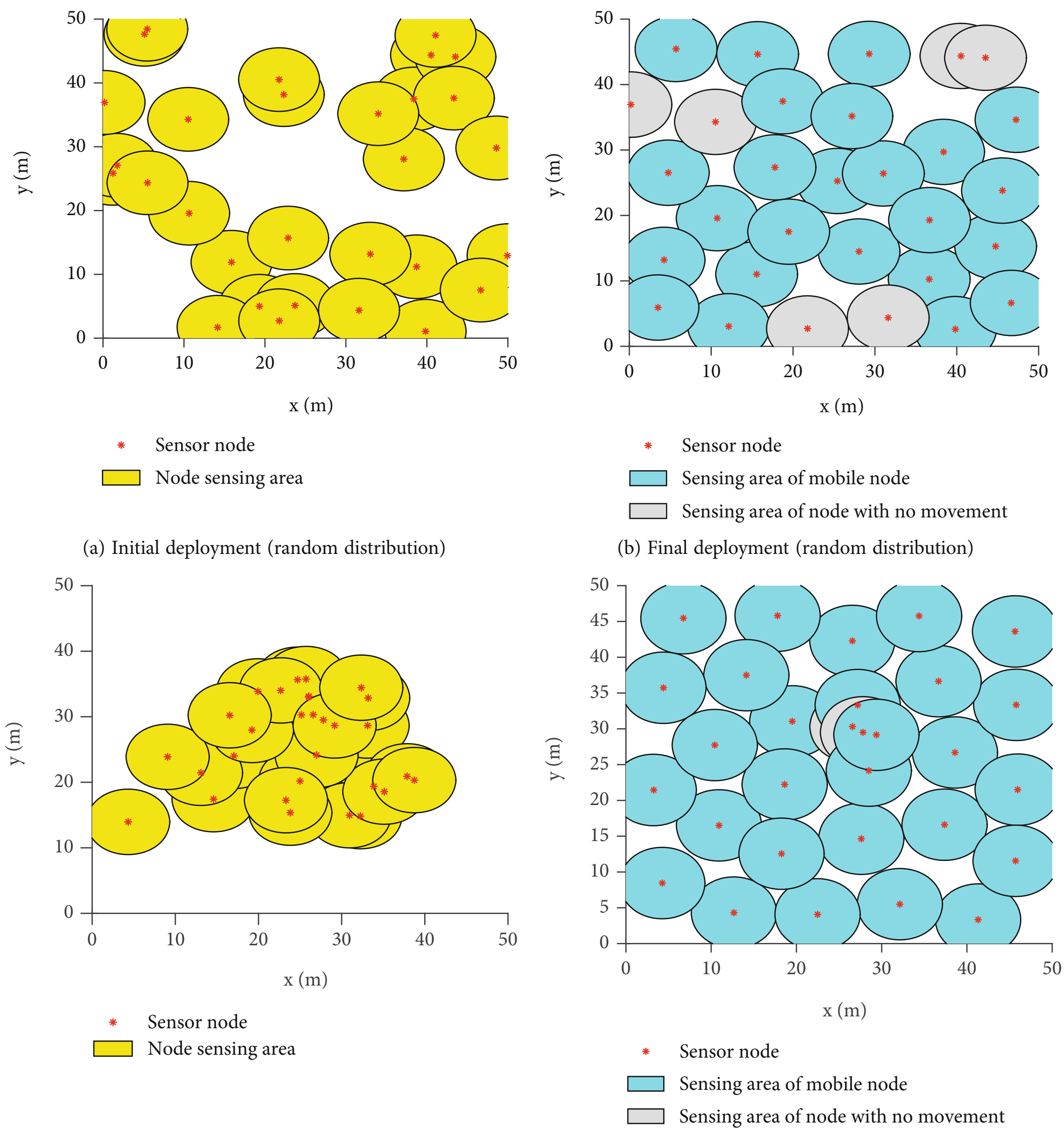
The procedure of IRCD-based node scheduling algorithm is shown in Figure 4.

5. Complexity Analysis

In this section, we analyze the computational complexity of our proposed IPSO-IRCD algorithm. In particular, we first conduct the mobile sensor node candidate deployment location optimization based on IPSO to maximize network coverage rate. Then, IRCD-based node coverage scheduling optimization algorithm is implemented to reduce the number and moving distance of mobile sensor nodes. Therefore, we analyze the computational complexity of IPSO and IRCD, respectively.

For IPSO, the computational complexity for population initialization is $O(nM)$, the calculation for fitness value is with the computational complexity $O(nM)$, the computational complexity for calculating the optimal fitness value in all sensor nodes is given as $O(n)$, and the computational complexity for updating nodes global location is $O(M)$. Hence, the computational complexity of IPSO is $O(nMK)$, where K denotes the iteration number.

For IRCD, the computational complexity for calculating the moving distance of each node is $O(n)$, the computational complexity of the ratio of coverage rate increment to moving



(a) Initial deployment (random distribution)

(b) Final deployment (random distribution)

(c) Initial deployment (Gaussian distribution)

(d) Final deployment (Gaussian distribution)

FIGURE 6: Nodes deployment ($n = 30$).

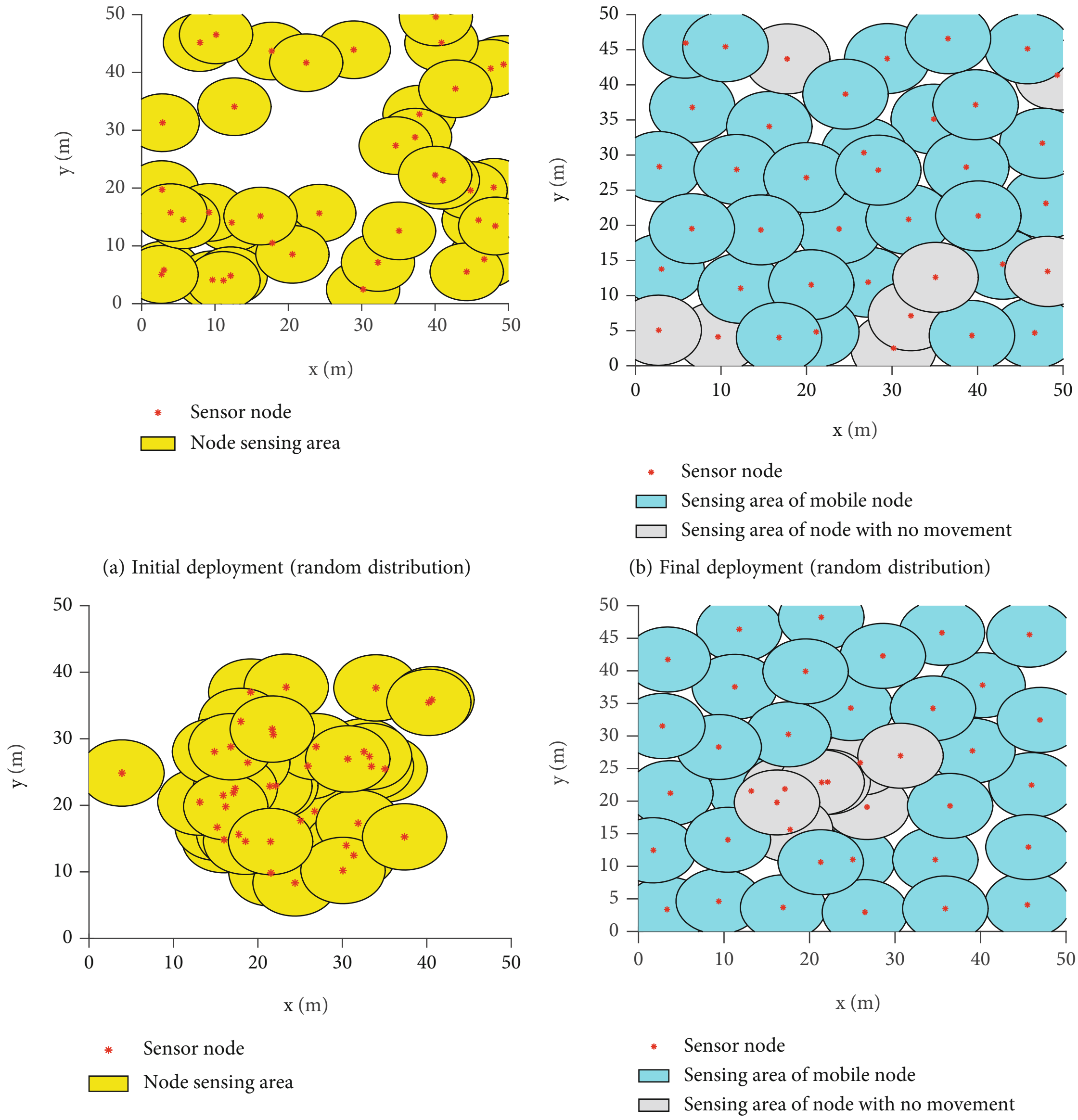
distance is given as $O(n)$, and to sort the ratio of coverage rate increment to moving distance for all nodes is with the computational complexity $O(n\log n)$. Thus, the computational complexity of IRCD algorithm is $O(n\log n)$.

Therefore, the total computational complexity of IPSO-IRCD is $O(nMK) + O(n\log n)$.

To illustrate the superior complexity of the proposed IPSO-IRCD algorithm, we compare the total computational complexity of IPSO-IRCD with exhaustive search algorithm. Specifically, based on the discretization of the monitoring area and to determine the deployment location of mobile sensor nodes, exhaustive search algorithm is conducted to search the $A \times B$ grid points in the monitoring area. Therefore, the total computational complexity to execute exhaustive search algorithm is $O(nAB)$. Moreover, to obtain the optimal WSN coverage strategy by applying the exhaustive search algorithm, the distance between the adjacent grid points in the monitoring area should be as small as possible, which increases the number of grid points. Accordingly, the total computational complexity $O(nAB)$ of exhaustive search algorithm is much greater than $O(nMK) + O(n\log n)$ of IPSO-IRCD.

6. Numerical Results

In this section, numerical results are provided to evaluate the performance of our proposed WSN coverage scheme. In the simulation, we consider that the sensor node initial coverage status follows random distribution and Gaussian distribution. The size of the simulation region is 50 m × 50 m. Without loss of generality, the performance of IPSO-IRCD is compared with SPSO, SSA, and ESSA proposed in [10], PSO-DAC



(a) Initial deployment (random distribution)

(b) Final deployment (random distribution)

(c) Initial deployment (Gaussian distribution)

(d) Final deployment (Gaussian distribution)

FIGURE 7: Nodes deployment ($n = 40$).

TABLE 2: Comparison of WSN performance under different number of nodes and different initial distribution.

	Number of sensor nodes	Initial network coverage rate	Final network coverage rate	Moving distance of nodes (m)
Random distribution	20	43.87%	60.9%	89.22
	30	58.13%	81.32%	156.58
	40	66.47%	92.85%	185.96
Gaussian distribution	20	26.84%	60.82%	190.27
	30	34.64%	81.89%	251.17
	40	42.56%	93.19%	294.11

proposed in [25], and HPSBA proposed in [30]. Unless otherwise stated, the numerical setup of the simulations is given in Table 1. To execute the proposed IPSO-IRCD algorithm, M particles are utilized to represent M distinct coverage strategy for WSN, where each particle P contains all n sensor node information. In particular, IPSO is iteratively applied to update the location information of each particle P with the goal of maximizing the coverage rate of the monitoring area. When the iterative process reaches the maximum number of iterations K, the sensor nodes coverage strategy of particle P with the maximum coverage rate is regarded as a candidate deployment strategy. However, for the candidate deployment strategy, the increment of network coverage rate may be small or the moving distance of sensor nodes is too long while the mobile sensor node S_j moving from the initial location to the candidate deployment location. To reduce the moving distance of mobile sensor nodes, IRCD is proposed to determine

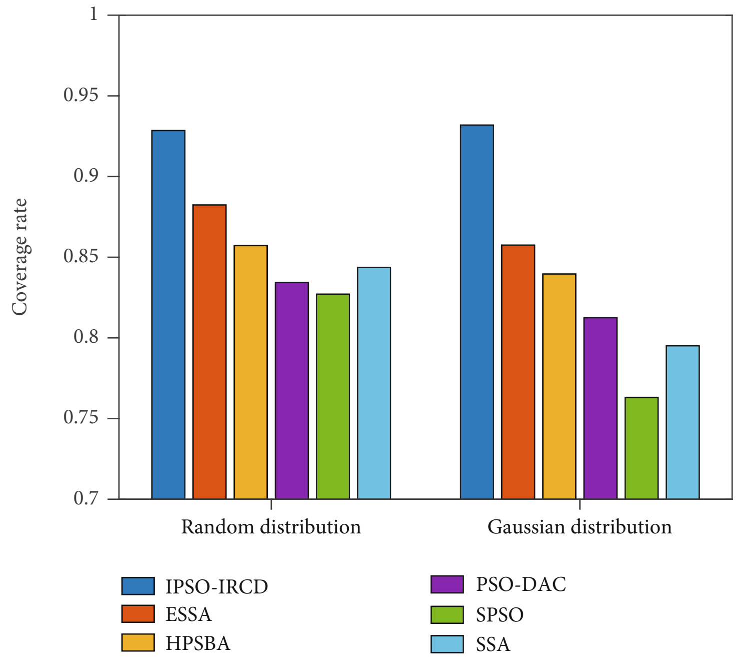


Figure 8: Influence of initial distribution category of nodes on WSN coverage rate.

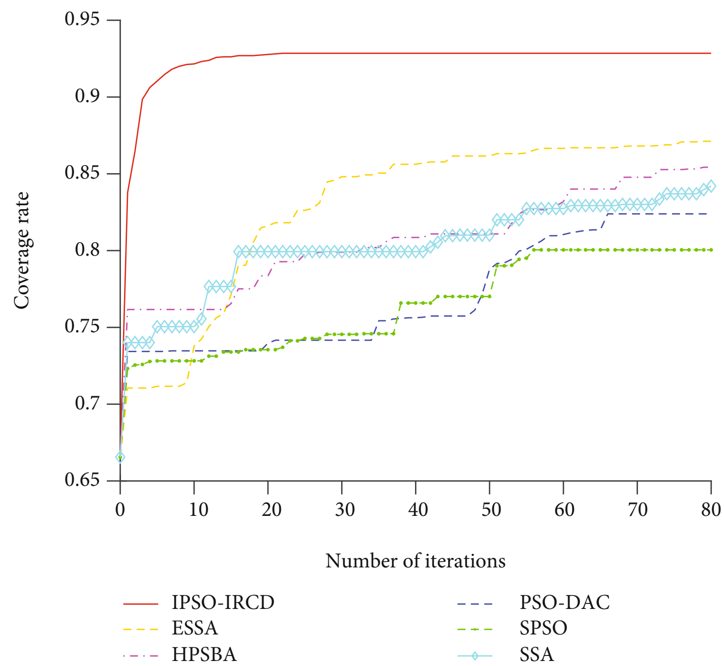
the final deployment locations of mobile sensor nodes in terms of the ratio of coverage rate increment and moving distance $\Delta\eta_{0j}/d(S_j)$. Specifically, in the case that $\Delta\eta_{0j}/d(S_j)$ of S_j is smaller than the threshold $\Delta\eta_t$, the initial location of S_j is set as its final deployment location, i.e., S_j is without movement. The process is repeated to obtain the final deployment locations of all n mobile sensor nodes until the minimum $\Delta\eta_{0j}/d(S_j)$ is greater than $\Delta\eta_t$. Finally, the corresponding mobile sensor nodes are scheduled to the obtained final deployment locations.

Since the different initial distribution of nodes may have an impact on the final coverage optimization of WSN. To demonstrate the robustness of IPSO-IRCD proposed in this paper, we firstly compare the final coverage effect of WSN with different number of sensor nodes under random initial distribution and Gaussian initial distribution.

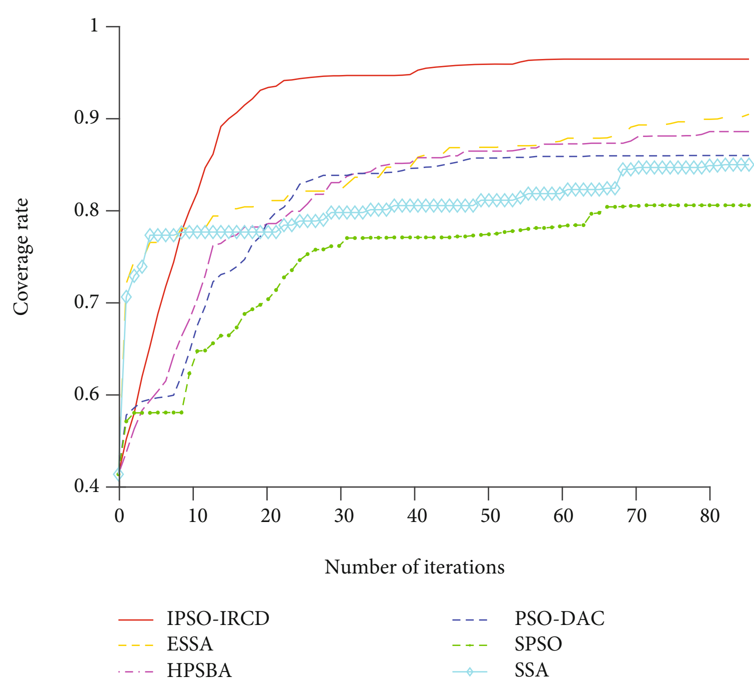
Figure 5 shows the node deployment status when $n = 20$. Note that "$*$" denotes the sensor node, and the circle area is the node sensing area. As can be seen in Figures 5(a) and 5(c), the coverage holes and overlaps are both in random initial distribution and Gaussian initial distribution, and the initial coverage rates are 43.87% and 26.84%, respectively. Figures 5(b) and 5(d) present the final deployment of sensor nodes with random distribution and Gaussian distribution, and the final coverage rates are 60.9% and 60.82%, respectively. It is observed that some mobile sensor nodes are set as sleep mode, i.e., keeping static status. Moreover, the number of mobile sensor nodes is reduced to 16 and 18, and the final nodes moving distances are 89.22 m and 190.27 m, respectively, for random distribution and Gaussian distribution. By comparing the initial deployment and final deployment of nodes under the two distributions, it can be seen that IPSO-IRCD algorithm can improve the coverage rates of 17.03% and 33.98%, respectively, while reducing the number and moving distance of mobile sensor nodes.

In Figure 6, we plot the node deployment status when $n = 30$. Specifically, Figures 6(a) and 6(c) show the initial deployment of random distribution and Gaussian distribution when the number of nodes is 30, and the initial coverage rates are 58.13% and 34.64%, respectively. Figures 6(b) and 6(d) provide the final deployment of sensor nodes under the two distributions, and the final coverage rates are 81.32% and 81.89%, respectively. Moreover, there are some mobile sensor nodes with no movement for both two distributions, and the number of nodes with no movement under Gaussian distribution is lower than random distribution. This stems from the fact that the initial node deployment of Gaussian distribution is more concentrated in a limited area, and more sensor nodes are scheduled to improve network coverage rate. The number of mobile sensor nodes under the two distributions is reduced to 24 and 28, and the final moving distances are 156.58 m and 251.17 m, respectively. By comparing the initial deployment and final deployment of nodes under the two distributions, IPSO-IRCD can improve the network coverage rates by 23.19% and 47.25%, respectively. Meanwhile, the number and moving distance of mobile sensor nodes are reduced.

Figure 7 depicts the node deployment under the two distributions when the number of nodes is 40. Among them, Figures 7(a) and 7(c) are, respectively, the initial deployment of sensor nodes under random distribution and Gaussian distribution. The initial coverage rates are 66.47% and 42.56% for the two distributions. Obviously, the problem of nodes redundancy and coverage holes are both in two distributions. Figures 7(b) and 7(d) give the final deployment of



(a) Random distribution

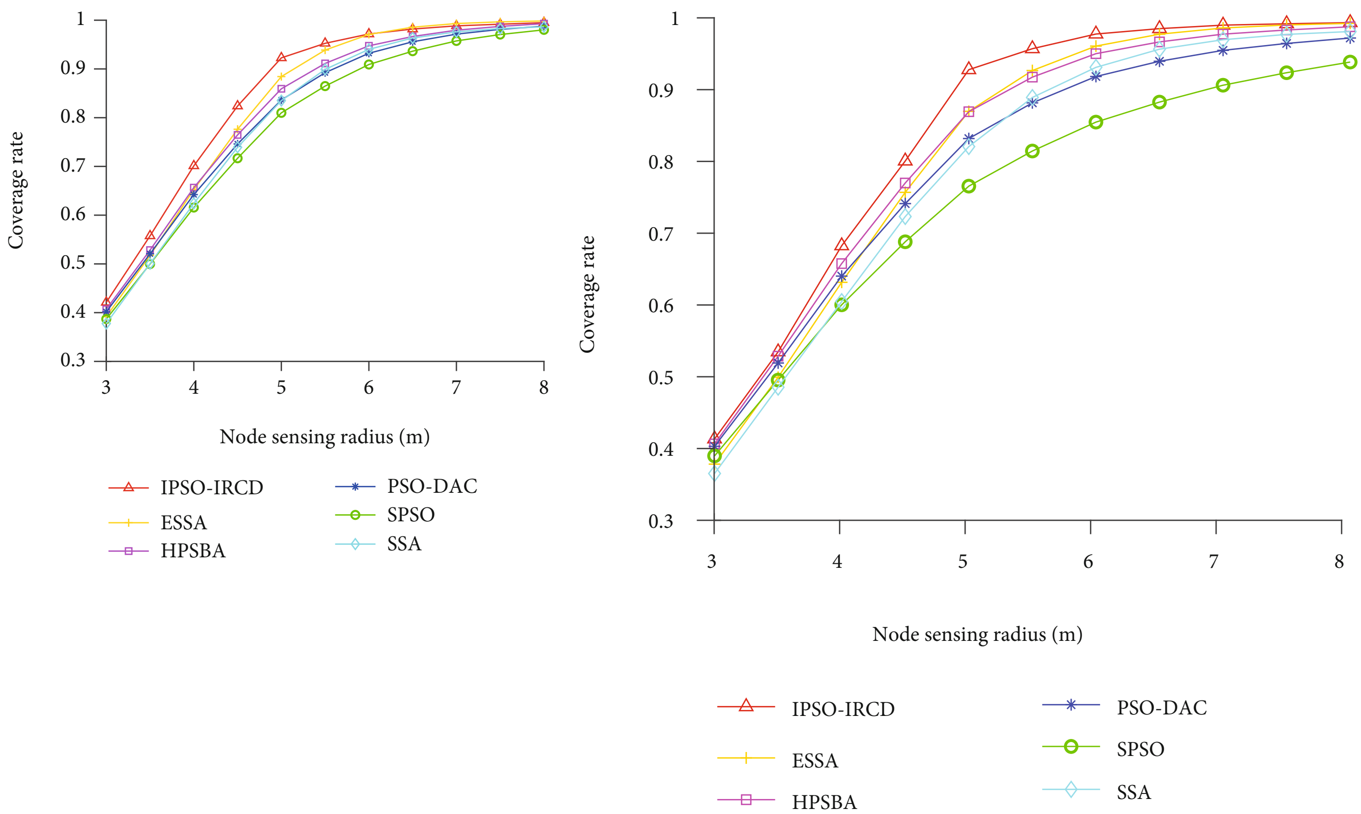


(b) Gaussian distribution

Figure 9: Network coverage rate versus iteration times.

sensor nodes under the two distributions, and the final coverage rates are 92.85% and 93.19%, respectively. It is worth noting that the sensor nodes with no movement under random distribution are relatively scattered in the monitoring area, and more concentrated in the center of the monitoring area under Gaussian distribution. The number of mobile sensor nodes is reduced to 32 for both two distributions, while the final moving distances are 185.96 m and 294.11 m, respectively. By comparing the initial deployment and final deployment of nodes under the two distributions, IPSO-IRCD can improve the network coverage rates by 26.38% and 50.63%, respectively, and reduce the number and moving distance of mobile sensor nodes.

The comparison of WSN performance under different number of nodes and different initial distribution is presented in Table 2. As can be seen from Table 2, the final



(a) Random distribution

(b) Gaussian distribution

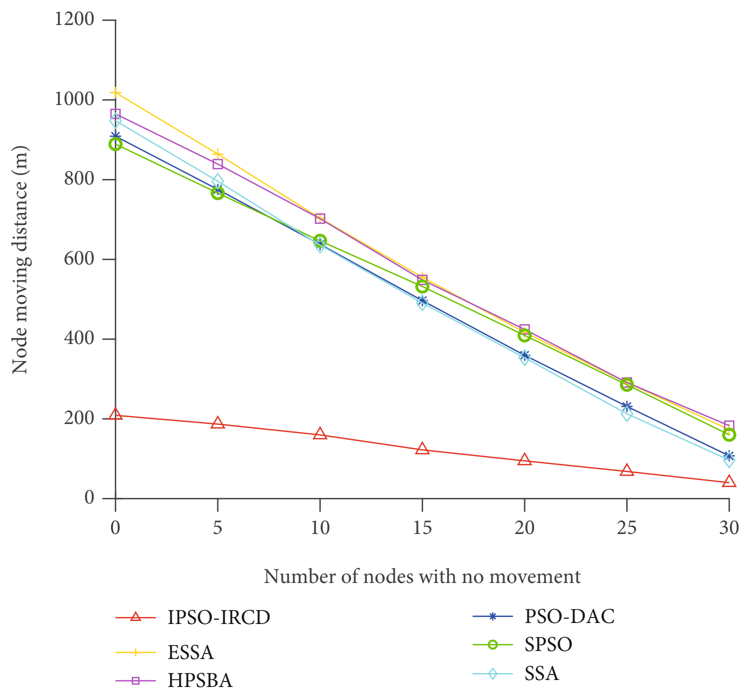
Figure 10: Network coverage rate versus node sensing radius.

network coverage rates can achieve a balance for two different initial distributions of sensor nodes, i.e., the coverage performance of WSN is not affected by the initial distribution of sensor nodes. However, since the nodes follow the Gaussian distribution are mainly concentrated in the limited area, the nodes are required to move farther to maximize the network coverage rate. Therefore, the moving distances of nodes follow the Gaussian distribution are greater than that of nodes with random distribution, while achieving the coverage rate approximatively.

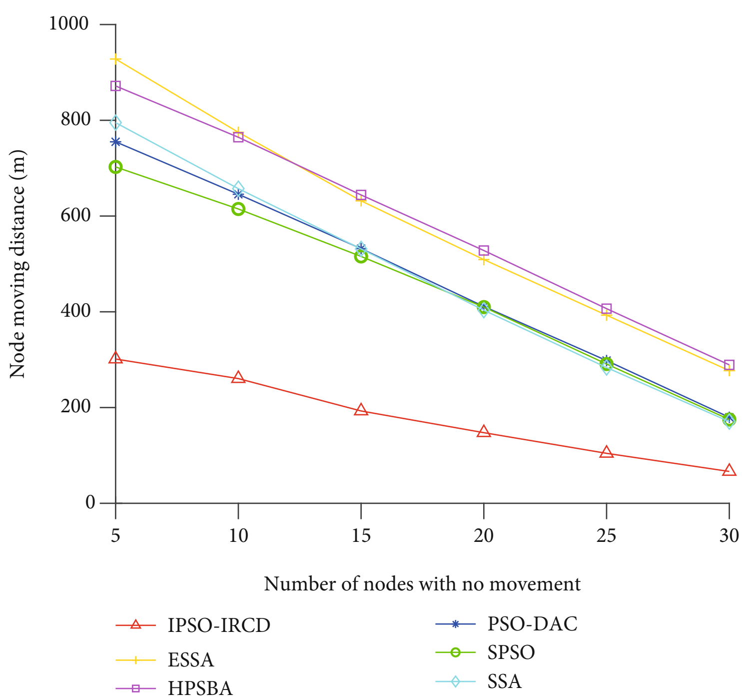
Figure 8 describes the impact of different initial node distribution on WSN coverage rate for different algorithms. From Figure 8, we can see that the coverage rate of other algorithms will be affected by the initial distribution category of nodes, while IPSO-IRCD can achieve the coverage rate approximatively for the two distributions. This is due to the fact that the location information of the node is updated for each dimension in our proposed scheme, i.e., the node with the highest coverage rate is selected for location update repeatedly, until the movement of mobile sensor nodes cannot improve the coverage rate. Therefore, IPSO-IRCD can avoid the node losing the optimization solution, and reduce the redundancy moving distance of the nodes simultaneously. Furthermore, IPSO-IRCD can achieve a comparable coverage rate for different initial node distributions. In addition, compared with the suboptimal algorithm ESSA, IPSO-IRCD can yield 4.6% and 7.4% gains, respectively, in terms of the coverage rate for random initial distribution and Gaussian initial distribution.

The convergence speed is an important indicator to evaluate the performance of the algorithm. Figure 9 describes the relationship between the network coverage rate and the number of iterations for the six algorithms. As shown in Figure 9, we find that the network coverage rate can converge to a constant within 80 iterations based on IPSO-IRCD for two initial node distributions, which demonstrates the robustness of the proposed IPSO-IRCD node deployment algorithm. This is mainly attributed to the fact that the coverage rate increment of each node in three location updated methods is evaluated in IPSO-IRCD, which increases the search space of optimization solutions, and sensor nodes are faster closing to the optimal solution, thus improving the convergence speed of the algorithm. For the Gaussian initial distribution, the coverage rate of ESSA and SSA increased rapidly in the first 10 iterations, whereas both of them fall into the local optimal value quickly. Moreover, the coverage rate achieved by IPSO-IRCD is significantly better than the other five algorithms for two different initial distributions.

In Figure 10, network coverage rate versus node sensing radius is plotted for two different initial distributions. According to Figure 10, we find that the coverage rates for two initial distributions both increase as the sensing radius of nodes increases. Then, the coverage rate gradually tends



(a) Random distribution

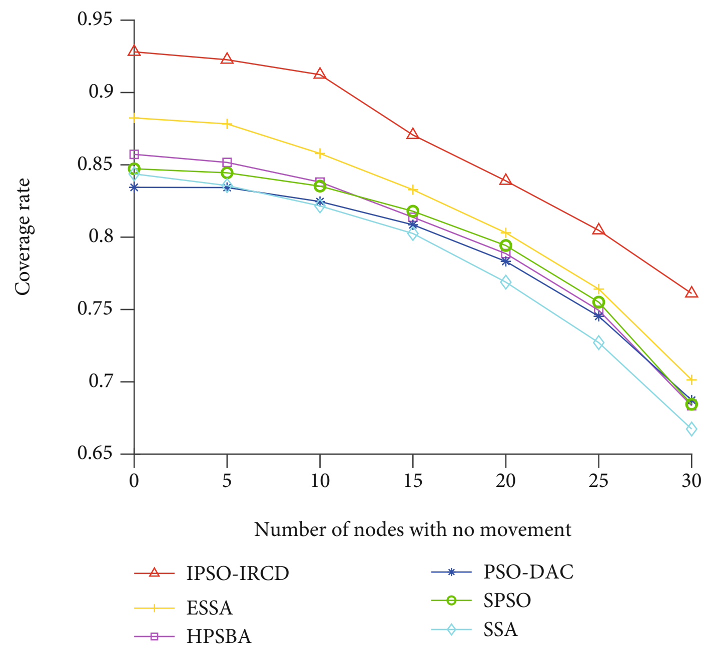


(b) Gaussian distribution

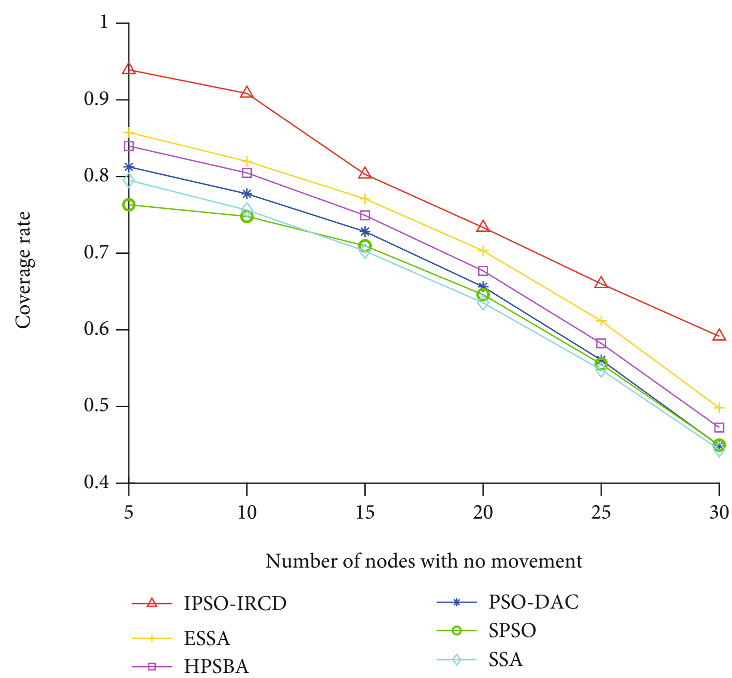
Figure 11: Node moving distance versus number of unmoved nodes.

to be stable when the sensing radius of nodes reaches a certain value. The reason is that the coverage area of each node is small with the smaller sensing radius, and there are large coverage holes in the monitoring area; thus, the network coverage rate can be effectively improved by increasing the sensing radius of nodes. As the sensing radius of nodes becomes larger, e.g., $R_s = 7.5$, the network coverage hole is small, and increasing the sensing radius of nodes may generate coverage overlaps in the monitoring area. Therefore, the growth of coverage rate tends to be stable gradually. In addition, compared with the other five algorithms, the IPSO-IRCD algorithm proposed in this paper can achieve higher network coverage rate with the same node sensing radius.

The moving distance of nodes and network coverage rate are interrelated. Figures 11 and 12 illustrate the impact of the number of unmoved nodes on nodes moving distance and



(a) Random distribution

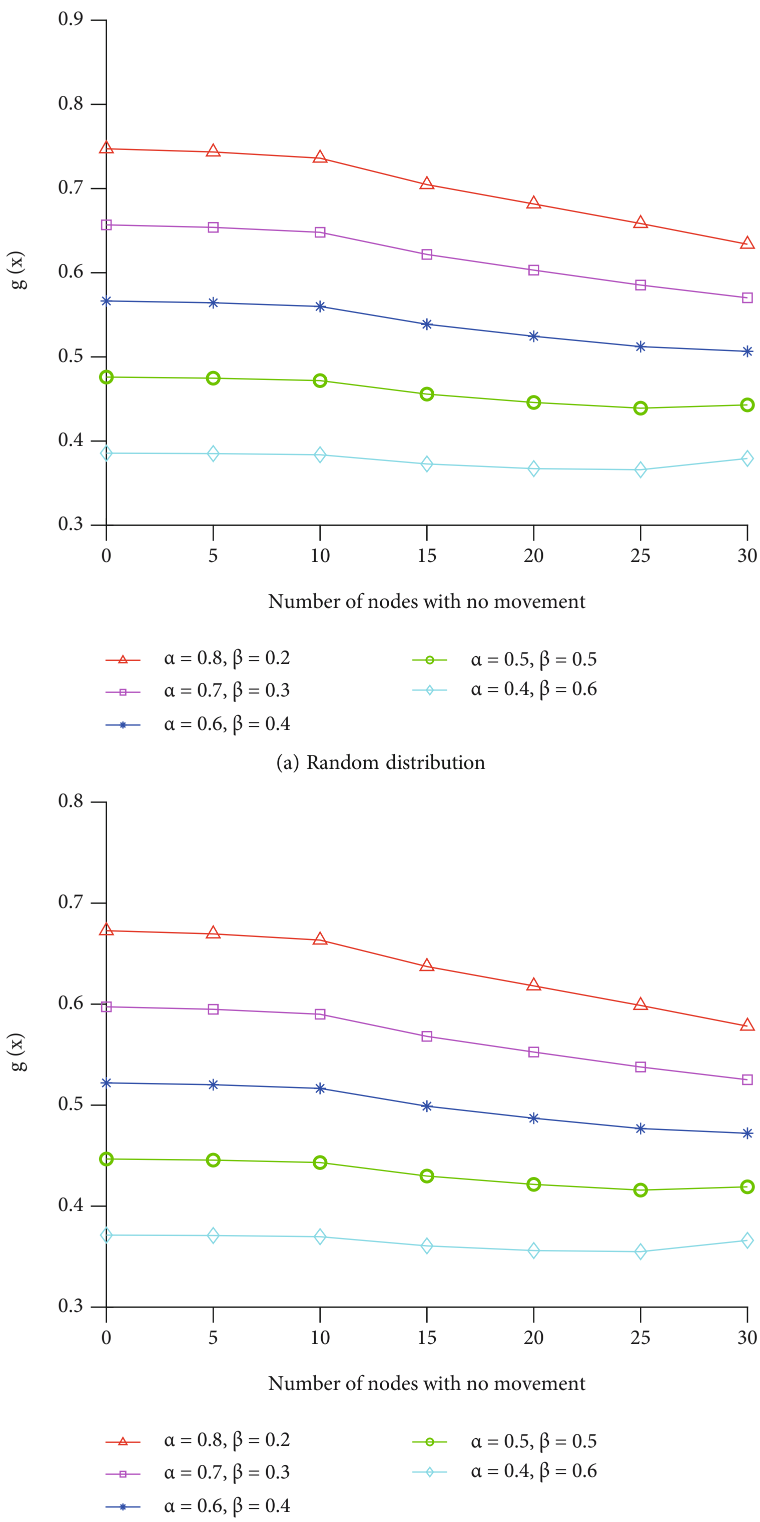


(b) Gaussian distribution

Figure 12: Network coverage rate versus number of unmoved nodes.

coverage rate for two different initial distributions. Among them, Figures 11(a) and 11(b) show the relationship between the nodes moving distance and the number of unmoved nodes for two different initial distributions. It can be observed that the moving distances of nodes for all algorithms are decreased while increasing the number of unmoved nodes. Meanwhile, the nodes moving distances of IPSO-IRCD in two initial node distributions are lower than that of the other five algorithms. This is due to the fact that IPSO-IRCD calculates the fitness function value for any updated location of each node. In case that the fitness value can be improved by updating the node location, the node is scheduled to the updated location, which reduces the redundancy moving distance of the node. However, the fitness value is calculated only after all the node locations in a population are updated in other five algorithms, which loses the optimization solution and increases the redundancy moving distance of nodes.

Figures 12(a) and 12(b) show the relationship between coverage rate and the number of unmoved nodes for two different initial distributions. It can be seen from the figure that



(b) Gaussian distribution

FIGURE 13: Objective function value with different number of unmoved nodes.

with the increase of the number of unmoved nodes, the coverage rate decreases, and the reduction of speed is increased gradually. The reason is that the coverage overlaps are generated among sensor nodes with the sufficient number of mobile sensor nodes. And the impact on coverage rate is small when the number of mobile sensor nodes is reduced, i.e., the reduction rate of coverage rate is slow. In addition, an increasing number of nodes are moving back from the better deployment location to their initial deployment location with the increase of the number of unmoved nodes, and the coverage holes become larger, thus resulting in the reduction of coverage rate. Since the mobile sensor nodes move back to their initial deployment location are according to the increment of coverage rate in the ascending order, the reduction rate of coverage rate increases gradually. Moreover, the coverage rate of IPSO-IRCD is better than the other five algorithms regardless of the initial node distribution and the number of unmoved nodes.

Figure 13 depicts the relationship between the objective function value of different weighting factors and the number of unmoved nodes for two initial node distributions. As shown in Figure 13, the value of the objective function $g(x)$ gradually decreases as the number of unmoved nodes increases. However, with the weighting factors corresponding to the coverage rate α decreases and the node moving distance β increases gradually, the reduction rate of $g(x)$ becomes slow. For the random initial distribution with $\alpha = 0.4$ and $\beta = 0.6$, the value of $g(x)$ first decreases as the number of unmoved nodes grows, and then increases when the number of unmoved nodes goes large. This is due to the fact that the impact of the nodes moving distances on $g(x)$ is greater than that of the coverage rate, while the value of β is large enough. Therefore, $g(x)$ begins to increase as the moving distances of mobile sensor nodes decrease. For Gaussian initial distribution, since the moving distance of nodes is higher than that of random initial distribution, the reduction rate of $g(x)$ is obviously with the increase of the number of unmoved nodes.

Obviously, the numerical results are affected by some important parameters (such as node sensing radius R_s, number of nodes with no movement, and weighting factors α, β, etc.). To provide a more effective illustration, we elaborate the impacts on numerical results under different values of these parameters.

Specifically, in Figure 10, the numerical setup of the simulations is given in Table 1 while the node sensing radius R_s is ranging from 3 to 8. In the case that $R_s = 3$, the coverage area of each node is small, and the network coverage rate is limited no matter how the mobile sensor nodes move. Besides, due to the coverage area is large enough of each node when $R_s = 8$, the network coverage rate can be improved by moving the mobile sensor nodes with lower moving distance. However, it should be mentioned that coverage redundancy is induced by the large R_s; thus, the appropriate R_s should be set to balance the network coverage rate and coverage redundancy. It can be observed that the moving distances of nodes increase as the coverage rate increases, by comparing Figures 11(a), 11(b), 12(a), and 12(b). Similarly, the coverage rate is also decreased when the moving distances of nodes decrease. Therefore, a compromise should be conducted between coverage rate and the moving distances of nodes by controlling the number of nodes with no movement. Furthermore, the numerical setup of the simulations in Table 1 is also utilized for Figure 13, while the weighting factors α and β are ranging from 0.4 to 0.8 and 0.2 to 0.6, respectively. It is obvious that the value of the objective function $g(x)$ increases as α increases and decreases as β increases. Fortunately, there is an optimal value of $g(x)$ under each set of α and β, which guides us to choose the applicable number of nodes with no movement to maximize the value of $g(x)$.

7. Conclusions

Since the sensor nodes are generally adopted random deployment to cover the monitoring area, coverage redundancy and hole are two urgent problems to be solved in WSN. In this paper, we propose a node coverage optimization algorithm based on IPSO-IRCD for WSN composed of multiple mobile sensor nodes. The IPSO-IRCD algorithm is divided into two stages, and the objective of the first stage is to improve the coverage rate of the monitoring area. By improving the update method of particle information in PSO, the search space of the solution is increased, and the convergence of the algorithm is improved. Meanwhile, the inertia weight and learning factor are improved to enhance the global and local search ability of the algorithm, so that the candidate deployment location of the mobile sensor nodes to maximize the coverage rate of the monitoring area can be determined. The objective of the second stage is to reduce the number and moving distance of mobile sensor nodes. Based on the IRCD of sensor nodes from the initial location to the candidate deployment location, the number and final deployment location of mobile sensor nodes are determined. Therefore, IPSO-IRCD can reduce the moving distances of nodes while ensuring the coverage rate of WSN. Simulation results indicate that the IPSO-IRCD algorithm proposed in this paper can effectively solve the problems of coverage redundancy and holes in WSN.

It should be mentioned that although the proposed IPSO-IRCD algorithm can achieve the improvement of WSN coverage performance, there are some limitations of the proposed work and can be extended to more general setups, e.g., the sensing radius R_s of sensor nodes may be different, or via the probability sensing model, which considers the uncertainty of node sensing, to characterize the feature of sensor nodes sensing. Furthermore, the energy of sensor nodes is limited, and the fault of sensor nodes may affect the reliability of WSN. For such general setups, sensor node coverage strategy design becomes much more involved and will be pursued in our future work by extending the results of this work.

Data Availability

The underlying data supporting the results can be found by contacting us.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was supported by the National Key R&D Program of China (2021YFB3901405), Scientific and Technological Research Program of Chongqing Municipal Education Commission (KJQN202101233 and KJQN202001229), Open Fund of Chongqing Artificial Intelligence and Intelligent Agriculture Subject Group (ZNNYKFB201901), Open Fund of Chongqing Key Laboratory of Geo-environment Monitoring and Disaster Early Warning of Three Gorges Reservoir Area (MP2020B0202).

References

[1] C. Luo, Y. Hong, D. Li, Y. Wang, W. Chen, and Q. Hu, "Maximizing network lifetime using coverage sets scheduling in wireless sensor networks," *Ad Hoc Networks*, vol. 98, p. 102037, 2020.

[2] T. O. Olasupo and C. E. Otero, "A framework for optimizing the deployment of wireless sensor networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1105–1118, 2018.

[3] R. Elhabyan, W. Shi, and M. St-Hilaire, "Coverage protocols for wireless sensor networks: review and future directions," *Journal of the Communications Network*, vol. 21, no. 1, pp. 45–60, 2019.

[4] M. Farsi, M. A. Elhosseini, M. Badawy, H. Arafat Ali, and H. Zain Eldin, "Deployment techniques in wireless sensor networks, coverage and connectivity: a survey," *IEEE Access*, vol. 7, pp. 28940–28954, 2019.

[5] W. Liu, K. Lu, J. Wang, G. Xing, and L. Huang, "Performance analysis of wireless sensor networks with mobile sinks," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 6, pp. 2777–2788, 2012.

[6] J. Amutha, S. Sharma, and J. Nagar, "WSN strategies based on sensors, deployment, sensing models, coverage and energy efficiency: review, approaches and open issues," *Wireless Personal Communications*, vol. 111, no. 2, pp. 1089–1115, 2020.

[7] R. Priyadarshi, B. Gupta, and A. Anurag, "Deployment techniques in wireless sensor networks: a survey, classification, challenges, and future research issues," *The Journal of Supercomputing*, vol. 76, no. 9, pp. 7333–7373, 2020.

[8] S. Pundir, M. Wazid, D. P. Singh, A. K. Das, J. J. P. C. Rodrigues, and Y. Park, "Intrusion detection protocols in wireless sensor networks integrated to Internet of Things deployment: survey and future challenges," *IEEE Access*, vol. 8, pp. 3343–3363, 2020.

[9] J. Wang, H. Han, H. Li, S. He, P. K. Sharma, and L. Chen, "Multiple strategies differential privacy on sparse tensor factorization for network traffic analysis in 5G," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1939–1948, 2022.

[10] Z. Wang, J. Wang, and D. Li, "Research on coverage optimization of wireless sensor network based on an enhanced sparrow search algorithm," *Chinese Journal of Sensors and Actuators*, vol. 34, no. 6, pp. 818–828, 2021.

[11] T. Song, D. Zhang, Y. Wang, H. Xu, Y. Fan, and L. Wang, "WSN coverage optimization based on improved whale optimization algorithm," *Chinese Journal of Sensors and Actuators*, vol. 33, no. 3, pp. 415–422, 2020.

[12] Y. Du, "Method for the optimal sensor deployment of WSNs in 3D terrain based on the DPSOVF algorithm," *IEEE Access*, vol. 8, pp. 140806–140821, 2020.

[13] J. Wang, C. Ju, Y. Gao, A. K. Sangaiah, and G. Kim, "A PSO based energy efficient coverage control algorithm for wireless sensor networks," *Computers, Materials and Continua*, vol. 56, no. 3, pp. 433–446, 2018.

[14] J. Wang, Y. Gao, C. Zhou, R. S. Sherratt, and L. Wang, "Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs," *Computers, Materials and Continua*, vol. 62, no. 2, pp. 695–711, 2020.

[15] 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, 9 pages, 2018.

[16] E. A. Hassan and A. Najid, "(SET) smart energy management and throughput maximization," in *Security Management in Mobile Cloud Computing*, pp. 1–28, IGI Global, 2017.

[17] S. Tabatabaei, "A novel fault tolerance energy-aware clustering method via social spider optimization (SSO) and fuzzy logic and mobile sink in wireless sensor networks (WSNs)," *Computer Systems Science and Engineering*, vol. 35, no. 6, pp. 477–494, 2020.

[18] K. Vijayalakshmi and P. Anandan, "Global levy flight of cuckoo search with particle swarm optimization for effective cluster head selection in wireless sensor network," *Intelligent Automation and Soft Computing*, vol. 26, no. 2, pp. 303–311, 2019.

[19] M. Srinivas, P. K. Donta, and T. Amgoth, "Efficient algorithms for point and area sweep-coverage in wireless sensor networks," in *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 315–320, Chennai, India, 2021.

[20] E. A. Hassan and A. Najid, "Optimization of energy efficiency in wireless sensor networks and internet of things: a review of related works," in *Nature-Inspired Computing Applications in Advanced Communication Networks*, pp. 89–127, IGI Global, 2020.

[21] J. Lee and C. Teng, "An enhanced hierarchical clustering approach for mobile sensor networks using fuzzy inference systems," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 1095–1103, 2017.

[22] E. A. Hassan and A. Najid, "EEA: clustering algorithm for energy-efficient adaptive in wireless sensor networks," *International Journal of Wireless Networks and Broadband Technologies (IJWNBT)*, vol. 7, no. 2, pp. 19–37, 2018.

[23] J. Lee and W. Cheng, "Fuzzy-logic-based clustering approach for wireless sensor networks using energy predication," *IEEE Sensors Journal*, vol. 12, no. 9, pp. 2891–2897, 2012.

[24] P. K. Donta, T. Amgoth, and C. Annavarapu, "Machine learning algorithms for wireless sensor networks: a survey," *Information Fusion*, vol. 49, pp. 1–25, 2019.

[25] Z. Teng, J. Lv, L. Guo, and Y. Xu, "Coverage strategy of wireless sensor network based on improved particle swarm optimization algorithm," *Journal of Guangxi Normal University(Natural Science Edition)*, vol. 36, no. 3, pp. 9–16, 2018.

[26] Y. Yang, X. Fan, Y. Gan, Z. Zhuo, S. Wang, and P. Zhao, "Coverage optimization of sensor network based on improved particle swarm optimization," *Journal of Systems Engineering and Electronics*, vol. 39, no. 2, pp. 310–315, 2017.

[27] W. Qi, H. Yu, G. Fan, and L. Chen, "WSN coverage optimization based on adaptive particle swarm optimization," *Computational science*, vol. 47, no. 7, pp. 243–249, 2020.

[28] C. Guo, Y. Yang, R. Hu, C. Xiao, J. Wang, and F. Xu, "WSN coverage optimization based on particle swarm optimization," *Application Research of Computers, Computer Engineering and Science*, vol. 37, no. 4, pp. 1170–1173, 2020.

[29] T. Wang and J. Sui, "Optimization of sensor network coverage distribution improved particle swarm optimization," *Journal of Liaoning Technical University (Natural Science)*, vol. 39, no. 3, pp. 280–286, 2020.

[30] M. Zhang, M. Wang, X. Wang, T. Qin, and J. Yang, "Research on WSN node deployment based on hybrid particle swarm-butterfly algorithm," *Application Research of Computers, Computer Engineering and Science*, pp. 1–9, 2021.

[31] D. P. Singh and B. Pant, "An approach to solve the target coverage problem by efficient deployment and scheduling of sensor nodes in WSN," *International journal of system assurance engineering and management*, vol. 8, no. 2, pp. 493–514, 2017.

[32] J. Wang, C. Ju, H. Kim, R. S. Sherratt, and S. Lee, "A mobile assisted coverage hole patching scheme based on particle swarm optimization for WSNs," *Cluster Computing*, vol. 22, pp. 1787–1795, 2019.

[33] H. Sun and H. Chen, "Redeployment strategy of wireless sensor network nodes based on firefly algorithm," *Journal of Computer Applications*, vol. 41, no. 2, pp. 492–497, 2021.

[34] Z. Liao, J. Wang, S. Zhang, J. Cao, and G. Min, "Minimizing movement for target coverage and network connectivity in mobile sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 7, pp. 1971–1983, 2015.

[35] G. Li and S. Hu, "Mobile sensor network coverage optimization algorithm based on VF-CS," *The Journal of Communication*, vol. 39, no. 3, pp. 95–107, 2018.

[36] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN'95-Int. Conf. Neural Netw.*, vol. 4, pp. 1942–1948, Perth, WA, Australia, 1995.