Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7732989, 23 pages
https://doi.org/10.1155/2022/7732989

Research Article

A Metaheuristic Algorithm for Coverage Enhancement of Wireless Sensor Networks

Zhigang Wang,[1] **Liqin Tian**,[1,2] **Wenxing Wu**,[2] **Lianhai Lin**,[1] **Zongjin Li**,[1] **and Yinghua Tong**[1]

[1]*School of Computer, Qinghai Normal University, Xining 810000, China*
[2]*School of Computer, North China Institute of Science and Technology, Beijing 101601, China*

Correspondence should be addressed to Liqin Tian; tianliqin@ncist.edu.cn

Received 2 March 2022; Revised 16 April 2022; Accepted 26 April 2022; Published 12 May 2022

Academic Editor: Xiaohui Yuan

When wireless sensors are randomly deployed in natural environments such as ecological monitoring, military monitoring, and disaster monitoring, the initial position of sensors is generally formed through deployment methods such as air-drop, and then, the second deployment is carried out through the existing optimization methods, but these methods will still lead to serious coverage holes. In order to solve this problem, this paper proposes an algorithm to improve the coverage rate for wireless sensor networks based on an improved metaheuristic algorithm. The sensor deployment coverage model was firstly established, and the sensor network coverage problem was transformed into a high-dimensional multimodal function optimization problem. Secondly, the global searching ability and searching range of the algorithm are enhanced by the reverse expansion of the initial populations. Finally, the firefly principle is introduced to reduce the local binding force of sparrows and avoid the local optimization problem of the population in the search process. The experimental results showed that compared with ALO, GWO, BES, RK, and SSA algorithms, the EFSSA algorithm is better than other algorithms in benchmark function tests, especially in the test of high-dimensional multimodal function. In the tests of different monitoring ranges and number of nodes, the coverage of EFSSA algorithm is higher than other algorithms. The result can tell that EFSSA algorithm can effectively enhance the coverage of sensor deployment.

1. Introduction

In the field of sensor monitoring and communication engineering, the development of wireless sensor network (WSN) technology has become the main driving force of the Internet of Things (IoT) technology [1]. With the development of communication and software technology, sensor networks have become a hot spot in essential application fields such as monitoring the ecological environment [2], agricultural production [3], military battlefield [4], and disasters and accidents [5, 6]. Generally, the hardware equipment used for monitoring is composed of multiple sensors with clear functions, low energy, small shape, and portability. The hardware completes the deployment of wireless sensor networks through sensing and communication functions.

In recent years, many new technologies and methods have been adopted to improve the quality of service (QoS) in wireless sensor networks [7]. The coverage effect of the sensor monitoring area is an essential indicator of network quality of service [8], which can measure the perception ability of the sensor to be monitored area under different deployment structures. Usually, the number of sensors is limited and randomly placed in the target monitoring range, and the deployment of sensor nodes is uneven, resulting in the problem of coverage holes, which ultimately affects the network's quality of service. The optimal deployment of sensor nodes is the premise to ensure the data acquisition, transmission, processing, and reliable application of sensor networks. It is an enduring conundrum in sensor networks [9]. For sensor deployment, the main goal is to use limited sensor resources to cover the target area as evenly and widely

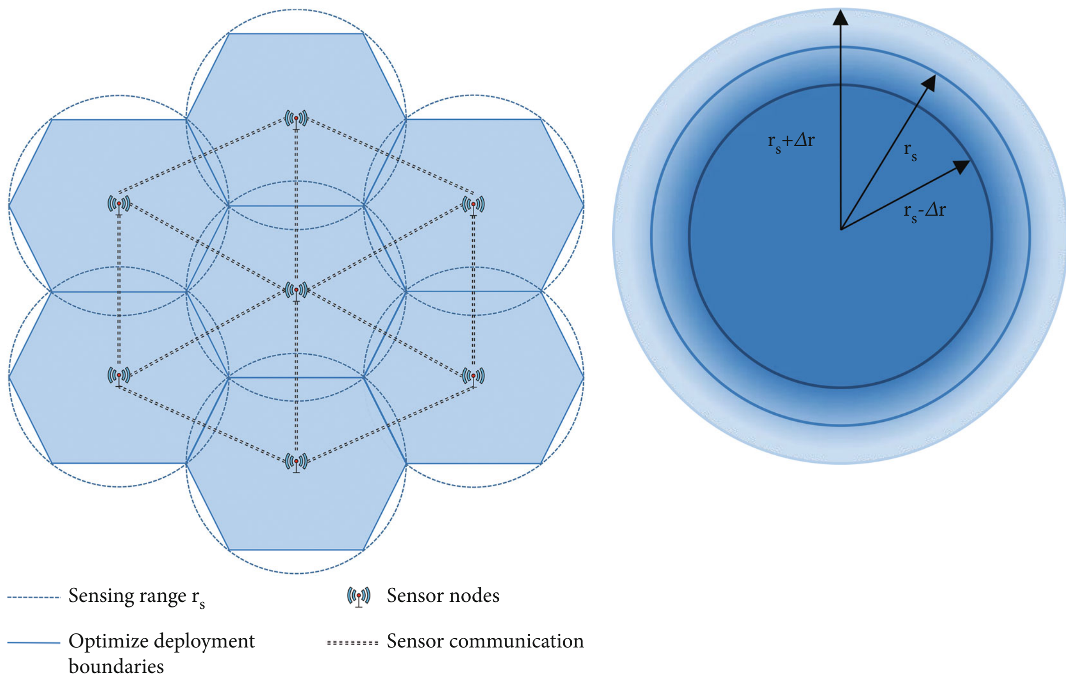


Figure 1: Deployment model. (a) Optimal deployment structure. (b) Probabilistic deployment structure.

/***Initial the Sparrow Search Population***/
  The Number of Population: n
  Maximum Iterations: $I_{\max}$
  Initial Finder Number: F_{num}
  Initial Detection and Early Warning Ratio: $DE_{\max}$
  Initial Warning Value : 0.8
  Firefly Parameter: $\alpha = 0.2, \beta_0 = 1, \gamma = 1$
/***Iterative search***/
  while $(t < I_{\max})$
    According to Equation (12) and (13), the elite reverse strategy is implemented for the initial population ($2n$ individuals);
    Rank the individuals in the population according to fitness values;
    Select the optimal n individuals as the first population;
    for $i = 1 : F_{num}$
    Based on the results of the last iteration of the finder, update the position according to Equation (8);
    end for
    for $i = F_{num} + 1 : n$
      The position of the joiner is disturbed according to the Equation (10);
    end for
    for $i = 1 : DE_{\max}$
      Based on the results of the last iteration of the detection and early warning, update the position according to equation (11) and (17);
    end for
    $t = t + 1$;
  end while
  return the best solution.

Algorithm 1: EFSSA.

as possible [10]. The coverage effect of the area to be monitored can be divided into point coverage, area coverage, and fence coverage [11]. WSN coverage in this paper is based on the method of point coverage. Most studies assume that the points to be monitored are evenly distributed in two-dimensional (2D) or three-dimensional (3D), and these areas are covered by a circular or spherical monitoring range centered on the sensor.

/∗Initialization∗/
Initialization EFSSA parameters are consistent with Algorithm 1
Set the number of sensor nodes d
Initialize the sensor node position s_i as population individuals
Set the sensor deployment scope(m^2):
900,2500,4900,8100
/∗Iterative search∗/
while $(t < I_{\max})$
The elite reverse strategy is implemented for the initial sensor position array;
Rank the individuals in the population according to fitness values;
Select the optimal n individuals as the first population;
for$i = 1 : F_{num}$
Update the position of finder;
end for
for$i = F_{num} + 1 : n$
Update the position of joiner;
end for
for$i = 1 : DE_{\max}$
Update the detection and early warning with firefly strategy;
end for
Obtain the current sensor node deployment results by calculating node coverage in Equation (6);
If the new overlay mode is better than the last overlay result, update it;
$t = t + 1$;
end while
return Optimal WSNs deployment results.

Algorithm 2: WSNs deployment based on EFSSA.

The research on sensor deployment is mainly divided into two types: deterministic deployment and random deployment [12, 13]. First of all, the deployment method is primarily used under the conditions of a good geographical environment, such as plains, wheat fields, grasslands, and lakes. In [14], sensors are deployed using a regular lattice model that can be full of space, but the actual monitoring environment cannot be an ideal 2D or 3D space. The environment often has flat terrain and is easy to plan and deploy. Random deployment is mainly used when the sensor nodes are difficult to fix quickly or reach the designated position. The first deployment is formed in a complex or harsh natural environment by aircraft throwing [15]. Obviously, this kind of method will cause a large number of sensor coverage redundancy and coverage holes. It will seriously waste sensor network resources and damage the network's quality of service. In fixed deployment mode, when the sensor does not have mobility, mobile sink can be used to improve the energy consumption and delay of the network [16]. On the other hand, when the sensor is mobile, researchers use the method of moving sensors in a small range to adjust the deployment structure and realize the second deployment of sensors [17]. There are also studies on monitoring target points or target regions by moving sensor nodes with strong mobility [18, 19]. There are also studies that address the deployment reliability of sensors; reference [20] divides sensor coverage methods into centralized and distributed. Wireless sensors are costly to deploy and must be used to maximize coverage for resource-constrained problems [21, 22]. After random deployment of sensor nodes in different scenarios, it is difficult to immediately meet the requirements of network coverage, and it is a hot issue to use the limited mobility of sensors for secondary deployment. However, most methods cannot achieve the expected results, resulting in a waste of resources and energy.

In this paper, we propose a coverage enhancement method for wireless sensor networks based on improved metaheuristic algorithm, which can effectively improve the coverage performance of wireless sensor networks. The main contributions are as follows:

(1) The location model of the area to be monitored and the sensor location model has been established, and the sensor coverage problem was transformed into the solution model of intelligent optimization algorithm

(2) An initial solution construction method is proposed. The quality of the initial solution of the population iteration is improved by an inverse elite solution to further make the location distribution of the random sensors as uniform as possible

(3) According to the fluorescence effect in the firefly intelligent optimization algorithm, the population individual optimization method in order to improve the local binding force during the population iteration and prevent from falling into local optimization is improved

(4) In experimental simulations, on the one hand, a comparison with other metaheuristic algorithms for sensor coverage is made to demonstrate the advantage of the algorithm on high-dimensional

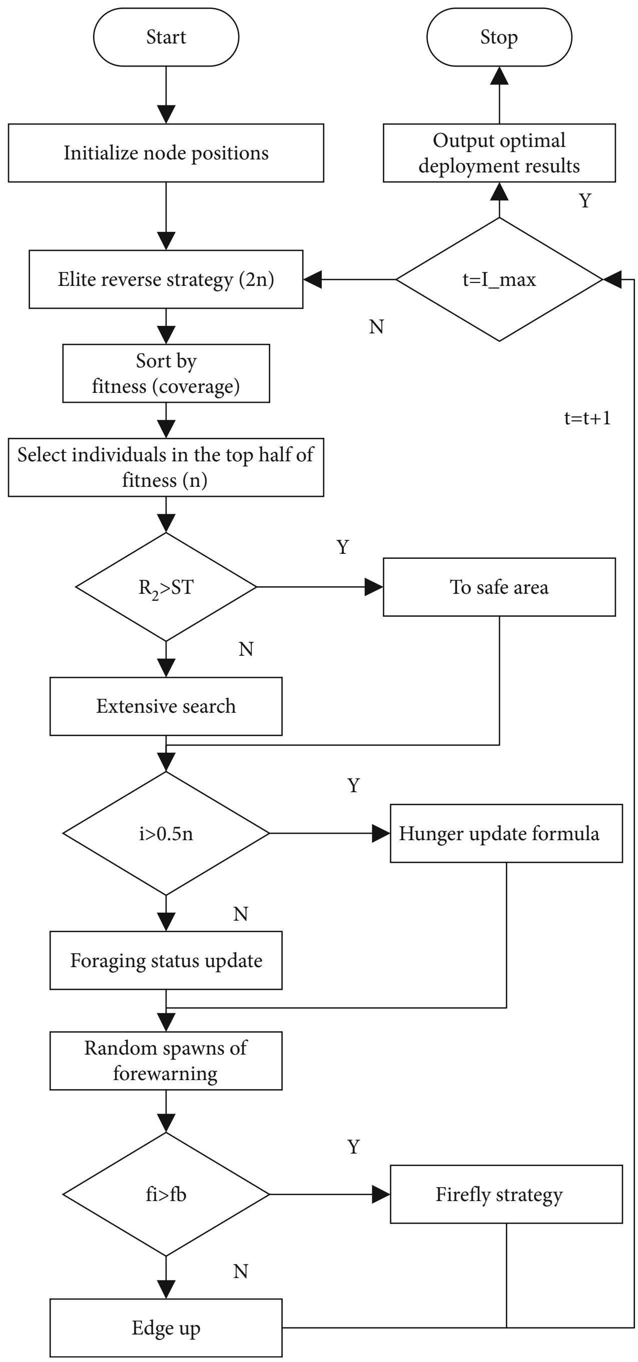


Figure 2: Flowchart for EFSSA wireless sensor network deployment.

multimodal reference functions. On the other hand, it verifies the advantages of deploying coverage in wireless sensor networks

The structure of this paper to enhance the WSN coverage is as follows: Section 2 introduces the current situation of coverage optimization at home and abroad. Section 3 introduces the coverage model in detail. Section 4 discusses the improved sparrow search algorithm. Section 5 presents the performance of the algorithm in the benchmark function and the effect in the application of sensor coverage and analyzes and discusses it. Section 6 gives conclusions and future prospects.

2. Related Work

The approximate solution of the metaheuristic optimization algorithm based on intelligent optimization has strong applicability in practical application compared with an accurate solution, especially in the research of sensor deployment optimization. Intelligent optimization algorithms have become the primary means of research in this direction.

In the deterministic deployment study, when there are few sensor nodes, a grid-based distributed sensor node deployment strategy was used in reference [23] to determine the location of deployed sensor nodes. Xu et al. [24] divided the monitoring area into multiple triangular grids to complete the coverage of the target by adjusting the distance relationship between the nodes and applied it to the underwater sensor network. In reference [25], on the basis of coverage, the connectivity between sensors is considered at the same time, which is able to keep the network connection even if some sensors fail. On the basis of guaranteeing sensor coverage and connectivity, some researchers in the literature [26, 27] further guarantee the connectivity and coverage of deterministic deployment networks by introducing relay nodes. Compared to random deployment, deterministic deployment was aimed at achieving target area coverage mainly with the minimum number of nodes and is closely related to the deployment space structure.

[28] proposed a metaheuristic algorithm based on antlion optimization (ALO) for sensor coverage and sensor sensing perception performance. This method transforms coverage into a function maximization problem, which can effectively and quickly achieve reliable deployment of wireless sensors, and proves that the deployment strategy of this algorithm outperforms sensor coverage by genetic algorithm (GA_WSN) and particle swarm optimization (PSO_WSN) algorithm for sensor deployment applications. Reference [29] proposed a sensor random deployment method based on grey wolf optimizer (GWO) to address the problem of low sensor coverage. Liao et al. [30] used the firefly swarm algorithm (FA) [31] to establish two detailed coverage models, central deployment and overlay deployment, for the sensor deployment coverage problem, and compared the coverage efficiency and mobility problems of these two models. These studies are based on the initial random deployment, so before considering the secondary deployment of sensors, the random nodes initially generated are close to the real scene, which is conducive to the implementation of the application. [32] not only generates scientific random deployment nodes of sensor nodes but also proposes the method of generating data packets, which has strong applicability in WSN performance imitation.

In terms of algorithm performance, although the intelligent optimization algorithm has strong optimization ability, few design parameters, and fast running speed, it still has some shortcomings. Compared with other traditional

Table 1: Reference function.

Name	Functions	d	Domain	$f_{\min}$
Sphere	$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
Schwefel-1	$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[-10,10]	0
Schwefel-2	$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \lvert x_i \rvert \right)^2$	30	[-100,100]	0
Tablet	$F_4(x) = \max \{ \lvert x_i \rvert, -1 \le i \le n \}$	30	[-100,100]	0
Rosenbrock	$F_5(x) = \sum_{i=1}^{n} \left[100 * \left(x_{i+1} - x_i^2 \right)^2 + (x_i - 1)^2 \right]$	30	[-30,30]	0
Step	$F_6(x) = \sum_{i=1}^{n} (\lvert x_i + 0.5 \rvert)^2$	30	[-100,100]	0
Quadric	$F_7(x) = \sum_{i=1}^{n} ix^4 + \text{random}[0, 1)$	30	[-1.28,1.28]	0
Schwefel	$F_8(x) = \sum_{i=1}^{n} - x_i \sin \left(\sqrt{\lvert x_i \rvert} \right)$	30	[-500,500]	0
Rastrigrin	$F_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10 \cos (2\pi x_i) + 10 \right]$	30	[-5.12,5.12]	0
Ackley	$F_{10}(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e$	30	[-32,32]	0
Griewank	$F_{11}(x) = \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	30	[-600,600]	0
Penalized	$F_{12}(x) = 4x_1^2 - 2.1x_1^2 + \frac{1}{3} x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	0
Foxholes	$F_{13}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \left(j + \sum_{i=1}^{2} \left(x_i - a_{ij} \right)^6 \right)^{-1} \right)^{-1}$	2	[-65,65]	0
Kowalik	$F_{14}(x) = \sum_{j=1}^{11} \left[a_i - \frac{x_1 \left(b^2 + b_i x_2 \right)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5,5]	0
Hartman	$F_{15}(x) = -\sum_{j=1}^{4} c_i \exp \left[-\sum_{j=1}^{3} ai_j \left(x_j - p_{ij} \right)^2 \right]$	3	[0,1]	0

algorithms, SSA proposed by Xue and Shen [33] has certain advantages in parameter design and solution accuracy. However, there are still problems of poor population diversity and individual populations easy to fall into local optimization. Sensor network coverage itself is a multidimensional problem. In this paper, the objective function of coverage is established in the high-dimensional mathematical model. Therefore, it is essential to solve poor population diversity and make it easy to fall into local optimization. This paper uses elite reverse strategy and firefly algorithm to solve this problem, and an improved sparrow search algorithm based on firefly (EFSSA) is proposed. [34] studied the optimization ability of population diversity of elite reverse strategy in particle swarm optimization algorithm. Sengathir et al. [35] combined the firefly algorithm with the artificial bee colony algorithm to extend the lifetime of the clustering problem of wireless sensor networks. Some researchers have also used BSA algorithm [36] combined with SSA algorithm to extend the lifetime of wireless sensor networks [37]. For the robot path planning problem, Ref. [38] proposed three different improved methods in SSA. Based on sparrow search, this paper combines elite reverse strategy and firefly so that the intelligent optimization algorithm can have a better performance effect on the problem of sensor coverage.

In recent years, some new evolutionary algorithms have been proposed and have good performance in common test functions. For example, the bald eagle search (BES) algorithm [39] simulates the hunting strategy and intelligent

Table 2: Benchmark function test results.

Test function	Statistical value	ALO	GWO	BES	RK	SSA	EFSSA
F1	Mean	8.050E-05	1.286E-33	1.187E-37	7.230E-201	8.262E-51	6.636E-90
	Std	2.736E-05	1.835E-33	3.498E-37	0.000E+00	2.493E-50	2.003E-89
F2	Mean	3.113E+01	5.521E-20	2.608E-27	1.364E-109	9.793E-47	2.308E-48
	Std	3.903E+01	4.386E-20	3.441E-27	4.081E-109	3.703E-46	7.015E-48
F3	Mean	2.062E+03	1.210E-08	3.702E-05	4.726E-165	7.054E-33	2.650E-80
	Std	7.633E+02	1.384E-08	1.126E-04	0.000E+00	1.496E-32	1.436E-79
F4	Mean	1.542E+01	1.044E-09	2.864E-01	1.707E-34	7.898E-17	1.073E-35
	Std	4.745E+00	2.270E-03	2.460E-01	5.185E-34	1.875E-16	5.879E-35
F5	Mean	9.807E+01	2.658E+01	1.780E+01	2.290E+01	5.913E-04	1.132E-06
	Std	1.228E+02	5.720E-01	4.735E+00	1.034E+00	1.100E-03	2.742E-06
F6	Mean	1.511E-04	4.501E-01	7.296E-29	1.191E-09	2.089E-06	1.483E-11
	Std	1.694E-08	2.843E-01	1.955E-28	4.113E-10	4.154E-06	7.506E-12
F7	Mean	1.514E-01	1.620E-03	1.900E-03	1.278E-04	3.634E-04	1.490E-03
	Std	4.370E-02	8.694E-04	1.350E-03	6.935E-05	3.110E-04	7.961E-04
F8	Mean	-5.519E+03	-6.609E+03	-4.994E+03	-8.222E+03	-1.048E+04	-8.915E+03
	Std	7.357E+02	5.895E+02	4.588E+02	7.791E+02	2.096E+03	1.286E+03
F9	Mean	8.453E+01	7.981E-01	1.344E+01	0.000E+00	0.000E+00	0.000E+00
	Std	1.469E+01	1.644E+00	2.464E+01	0.000E+00	0.000E+00	0.000E+00
F10	Mean	2.179E+00	4.233E-14	4.440E-15	8.880E-16	8.880E-16	8.882E-16
	Std	6.775E-01	1.678E-15	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F11	Mean	2.162E-02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
	Std	1.270E-02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
F12	Mean	1.151E+01	2.836E-02	8.881E-26	4.664E-10	3.319E-07	2.184E-13
	Std	3.072E+00	1.233E-02	2.696E-25	1.462E-10	6.935E-07	1.330E-13
F13	Mean	4.504E+00	3.196E-01	3.467E-02	4.310E-03	2.741E-06	4.144E-12
	Std	1.133E+01	1.558E-01	3.895E-02	7.170E-03	3.240E-06	1.397E-12
F14	Mean	1.989E+00	2.562E+00	9.980E-01	1.591E+00	4.220E+00	9.980E-01
	Std	7.847E-01	2.921E+00	1.312E-02	9.286E-01	4.552E+00	1.312E-02
F15	Mean	2.740E-03	2.310E-03	2.340E-03	6.112E-04	3.270E-04	3.073E-04
	Std	5.980E-03	6.120E-03	6.110E-03	4.378E-04	4.029E-05	4.498E-07

information interaction behavior of condor when looking for prey. Ahmadianfar et al. [40] proposed Runge Kutta (RK) optimizer algorithm, which proposed the global optimization search mechanism in the feature space and enhanced solution quality (ESQ) to avoid the local optimization mechanism.

3. Problem Definition

Suppose that the sensor nodes of the wireless sensor networks are distributed in the two-dimensional space of $L \times L(m^2)$, N isomorphic sensors are deployed, and the sensing radius of the sensor is defined as r_s. This study draws on the previous literature [41], an image that divides the space into $N' = m \times n$ pixels, and the coordinates of each point are $p_j = (x_j, y_j), j = 1, 2 \cdots, N$. In addition, assuming that the location set of sensor nodes is expressed as $S = \{s_1, s_2, \cdots, s_N\}$, the location information of each node can be expressed as $s_i = (x_i, y_i)$. To better calculate the coverage of the sensor, therefore, the distance d between node s_i and spatial point p_j in the sensor network is

$$d\left(s_i, p_j\right) = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}. \tag{1}$$

Assuming that the sensing model is a Boolean model, the sensor node s_i can sense point p_j; according to literature

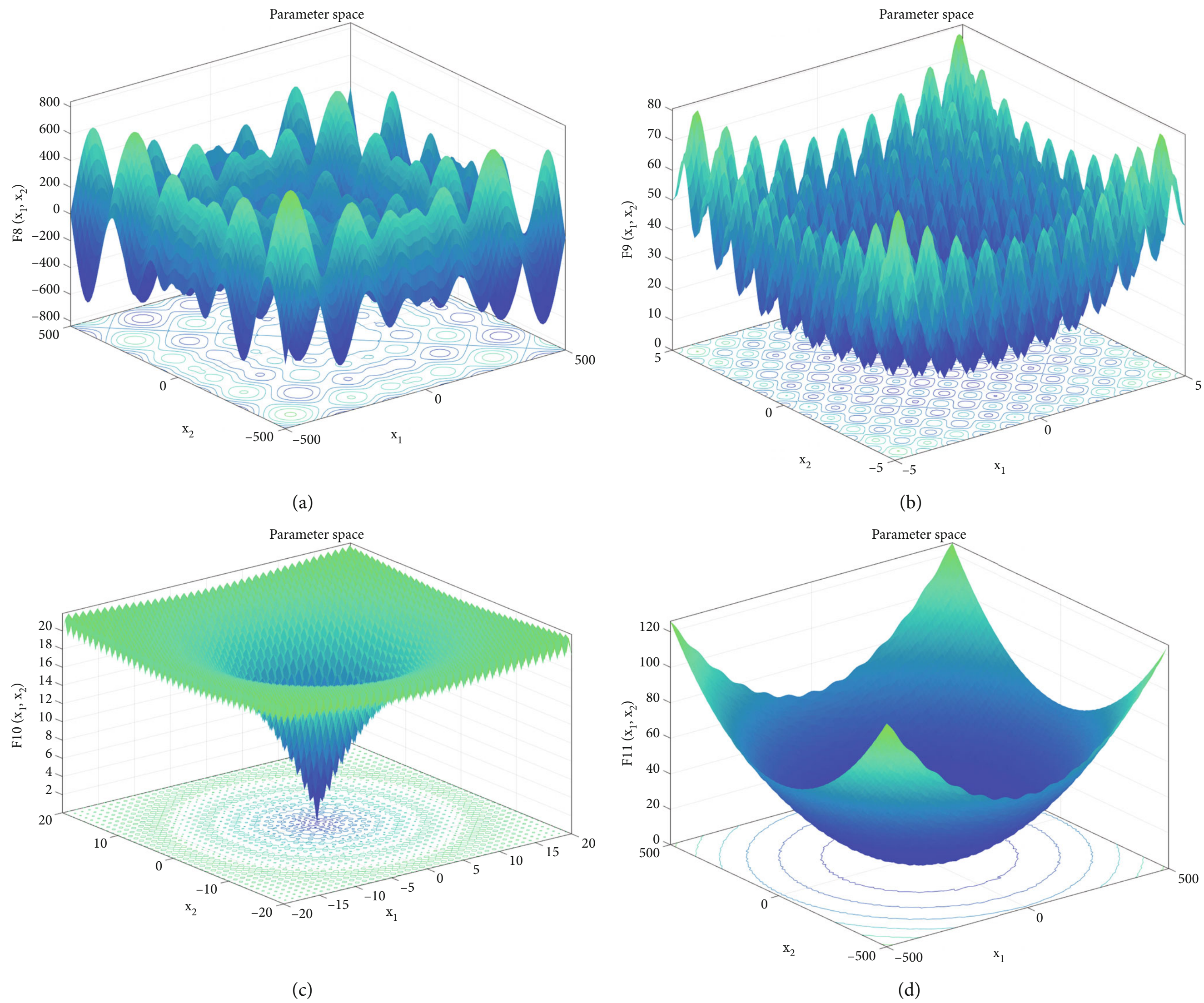


FIGURE 3: High-dimensional multimodal test functions. (a) Schwefel-F8. (b) Rastrigrin-F9. (c) Ackley-F10. (d) Griewank-F11.

[42], the perception probability can be defined as

$$\mathrm{B}\left(s_i, p_j\right) = \begin{cases} 1, & \text{if } d\left(s_i, p_j\right) \leq r_s, \\ 0, & \text{others } d\left(s_i, p_j\right) > r_s. \end{cases} \tag{2}$$

In the ideal state, the optimal structure for isomorphic sensor node deployment is the coverage model [43] with the sensor node as the center, the sensing distance as the radius of the circumscribed circle of the positive hexagon, and the tiled area to be monitored (e.g., in Figure 1(a)).The sensor equipment will be disturbed by noise and the physical environment in the actual scene, following a specific regular probability distribution [40]. At this time, the relationship between probability and distance is (see Figure 1(b))

$$P\left(s_i, p_j\right) = \begin{cases} 1, & \text{if } d\left(s_i, p_j\right) \leq r_s - \Delta r, \\ \exp\left(-\alpha\lambda^{\beta}\right), & \text{if } \left|d\left(s_i, p_j\right) - r_s\right| \leq \Delta r, \\ 0, & \text{others}. \end{cases} \tag{3}$$

In the equation, α and β are related to the physical characteristics of the sensor itself, $\Delta r \epsilon (0, r_s)$ represents the reliable sensing range parameter with sensor coverage change, and λ is an input parameter. Usually, the calculation method of λ is

$$\lambda = d\left(s_i, p_j\right) + \Delta r - r_s. \tag{4}$$

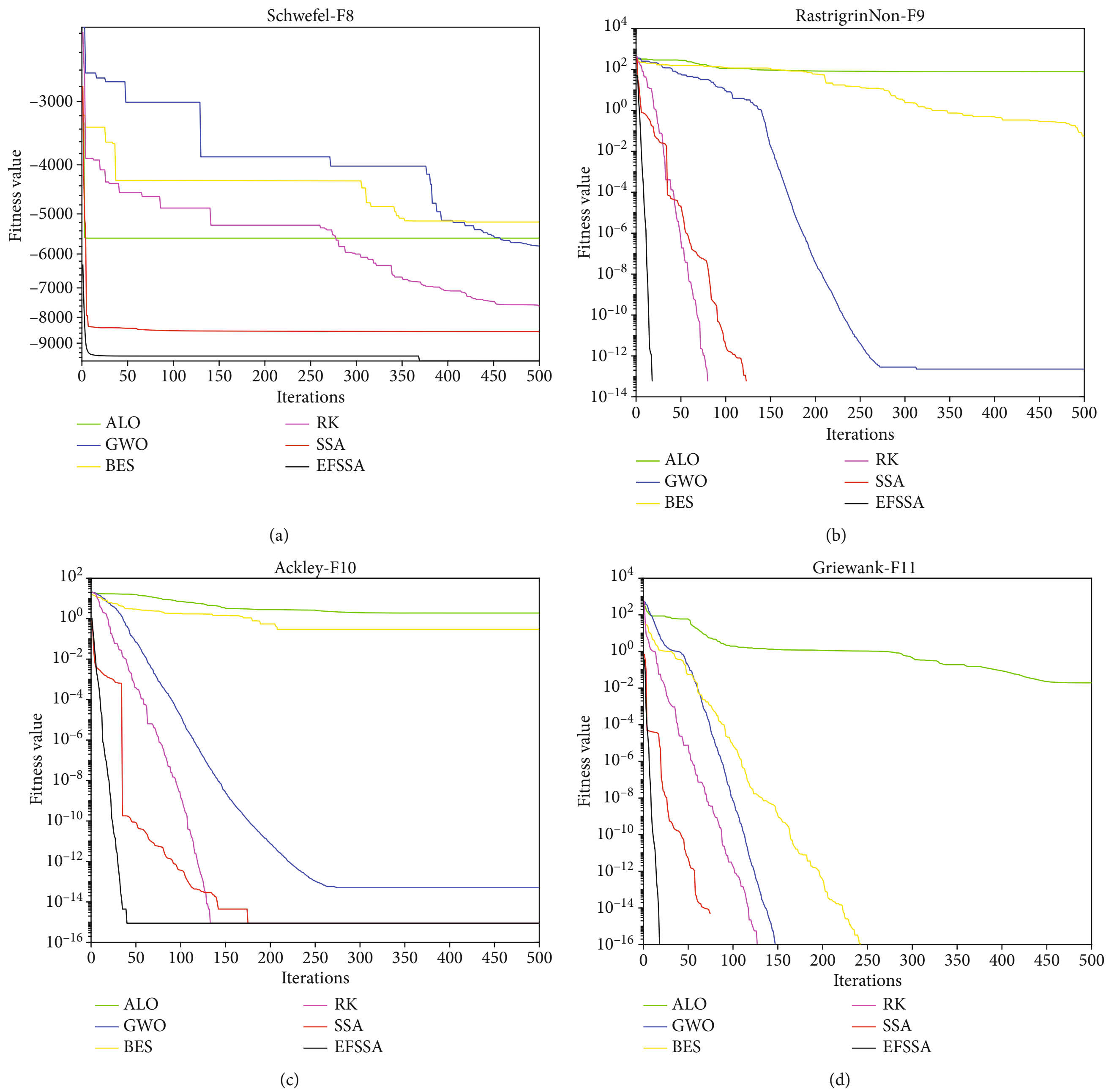


FIGURE 4: High-dimensional multi-modal test function curve. (a) Schwefel-F8. (b) RastrigrinNon-F9. (c) Ackley-F10. (d) Griewank-F11.

The same point p_j in the area may be sensed by multiple sensors simultaneously because as long as each spatial point is sensed by one sensor, it is considered to cover the area successfully. Then, the perceived probability of a node p_j in the perceived area N' can be further expressed as

$$B_{N'}\left(S, p_j\right) = 1 - \prod_{i=1}^{N}\left(1 - B\left(s_i, p_j\right)\right). \tag{5}$$

Then, the calculation equation for the total coverage R_{cov} of the monitoring area is

$$R_{cov} = \frac{\sum_{j=1}^{N'} B_{N'}\left(S, p_j\right)}{N'}. \tag{6}$$

At the same time, in order to better reflect the redundancy, utilization, and distribution uniformity of sensor network coverage, the ratio C_e between the monitoring range of real sensor coverage and the total area that all sensors can cover is taken as an evaluation index, and the equation is

Table 3: Simulation parameters.

Parameters	Value
Monitoring area size	30m × 30m 50m × 50m 70m × 70m 90m × 90m
Iterations	500
Perceived radius (r_s)	7m
Probability perception (Δr)	0.5m
Single-step moving distance	1m
The population size	50
Repetitions	30
Finders ratio	0.2
Reconnaissance ratio	0.15
Early warning value	0.8

expressed as

$$C_e = \frac{R_{\text{cov}} \times L \times L}{N \times (\pi \times r_s^2)}. \quad (7)$$

4. Proposed Method

4.1. Overview of Sparrow Search Algorithm. In 2020, a metaheuristic method SSA based on large-scale bird foraging and early warning was proposed [33]. In this algorithm, the whole sparrow group can be abstracted into two groups: entrants and finders, and a reconnaissance and early warning mechanism is added to the search. The finder usually has a high energy reserve. When areas with more food are found, they are able to provide flight directions for the joiners. Joiners always have access to good food sources from the information of the finder. At the same time, in SSA algorithm, the population has the strategy to realize the threat and to adopt antipredatory behavior.

4.1.1. Update Method of the Finder. Firstly, the definition of the population is given. Assuming that each population size is n in SSA, the population matrix $X = [X_1, X_2, \cdots, X_i, \cdots, X_n]^T$ is formed for the d-dimensional coverage problem, where $X_i = [X_{i,1}, X_{i,2}, \cdots, X_{i,j}, \cdots, X_{n,d}]$. All finder is responsible for finding food for the whole population and providing the migration direction of the foraging area for the joiners. Therefore, in SSA, the finder has a large search area. According to the rules, in the iterative process of the algorithm, the location update of the finder is described as follows [33]:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \bullet \exp\left(-\dfrac{i}{\alpha \bullet T_{\max}}\right), & \text{if } R_2 < \text{ST}, \\ X_{i,j}^t + Q \bullet L, & \text{if } R_2 \geq \text{ST}, \end{cases} \quad (8)$$

where $T_{\max}$ represents the final number of iterations and $X_{i,j}^t$ denotes the position of the i-th individual at iteration t-th. Both α and Q are random in the range $(0, 1]$. L is a unit matrix of size $1 \times d$. ST indicates a dangerous cut-off value. When $R_2 < ST$, it means that the population is safe; that is, there is no predator around, and the finder can search more widely to find a more suitable foraging place; when $R_2 \geq \text{ST}$, the group found the predator and sounded the alarm. The group immediately stopped foraging, made antipredation behavior, and quickly approached the safety zone.

It should be noted that when the condition $R_2 < \text{ST}$ is satisfied in Equation (1), the critical component of the finder updating the following location around the current location is independently expressed as y (see Figure 1):

$$y = \exp\left(-\frac{i}{\alpha \bullet T_{\max}}\right). \quad (9)$$

In Equation (9), as the search algorithm is computed, the iteration around y of the current node changes in the range of $(0, 1)$. However, the range of population variation becomes smaller in the later iterations. Because the individuals in the population remains unchanged, when iterations increases, the finders are more densely distributed in the smaller y interval than in the larger interval, so the local search ability is stronger. However, the reduction of the search range will also lead to the algorithm not being able to obtain the optimal global solution in complex problems.

4.1.2. Update Method of Joiners. In the sparrow population, after removing the finder, the rest is the joiner, and the joiner will follow the finder. Joiners will also monitor the finders. Once they find a better foraging location, they will abandon their current foraging area and fly to an area with more food based on the information. The equation for updating the position of joiners is expressed as [33]

$$X_{i,j}^{t+1} = \begin{cases} Q \bullet \exp\left(\dfrac{X_{\text{worst}}^t - X_{i,j}^t}{\alpha \bullet T_{\max}}\right), & \text{if } i > \dfrac{n}{2}, \\ X_{\text{best}}^{t+1} + \left|X_{i,j}^t - X_{\text{best}}^{t+1}\right| \bullet A^+ \bullet L, & \text{if } i \leq \dfrac{n}{2}. \end{cases} \quad (10)$$

In Equation (10), X_{worst}^t represents the global worst individual position in the iteration results of the previous generation. Similarly, X_{best}^{t+1} represents the global best position in the current iteration result, that is, the elite solution. A^+ has the same structural dimension as Q above. A is a $d \times d$ matrix composed of 1 or -1. A^+ has the characteristics of $A^+ = A^T (AA^T)^{-1}$. Conditions $i > n/2$ and $i \leq n/2$ in Equation (10) indicate that the sparrow is hungry for foraging and has been in the best foraging position.

4.1.3. Update Methods of Reconnaissance and Early Warning. In the simulation experiment, sparrows with conscious danger signals did not exceed 20% of the total number and their selection was randomized. The update formula for

Table 4: Coverage of different nodes and monitoring areas.

Area	Nodes	Initial	ALO	GWO	BES	RK	SSA	EFSSA
30m × 30m	7	0.52222	0.90667	0.91556	0.88667	0.91222	0.85333	0.92444
	9	0.67889	0.96778	0.99667	0.96556	0.99889	0.91000	1.00000
	11	0.75111	0.99778	1.00000	0.99333	1.00000	0.93444	1.00000
50m × 50m	20	0.58480	0.86080	0.94280	0.84160	0.93640	0.83280	0.96120
	25	0.68280	0.96778	0.98000	0.88600	0.96800	0.84240	0.99600
	30	0.72720	0.97680	0.99640	0.94640	0.99000	0.92800	1.00000
70m × 70m	40	0.64041	0.87490	0.90673	0.79163	0.87551	0.87490	0.95224
	50	0.71429	0.90330	0.80600	0.81050	0.90070	0.80890	0.96920
	60	0.76714	0.95959	0.97755	0.91347	0.98750	0.97000	0.99980
90m × 90m	64	0.65321	0.84543	0.75370	0.76407	0.86753	0.77210	0.92136
	80	0.71568	0.88100	0.79490	0.80500	0.78990	0.80040	0.94460
	96	0.78309	0.96346	0.86185	0.88642	0.94543	0.87099	0.99284

these individuals is as follows [33]:

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^{t} + \beta \bullet \left| X_{i,j}^{t} - X_{\text{best}}^{t} \right|, & \text{if } f_i > f_b, \\ X_{\text{best}}^{t+1} + K \bullet \dfrac{\left| X_{i,j}^{t} - X_{\text{best}}^{t+1} \right|}{(f_i - f_w) + \varepsilon}, & \text{if } f_i = f_b. \end{cases} \tag{11}$$

Similar to Equation (9), X_{best}^{t+1} originates from the whole population. The function of β is to control the step size, obeying the 0~1 distribution. $K \in (0, 1]$. f represents the fitness; f_i, f_b, and f_w represents the current value, the global best, and the worst value, respectively. The denominator part is by adding ε make sure it is not 0. When $f_i = f_b$, it means that the early warning sparrow is in the middle; it should move immediately to the edge of the population to avoid being attacked. When $f_i > f_b$, another strategy of moving in smaller steps should be adopted.

4.2. Improvement Strategy. In the previous studies, the SSA algorithm has been proved to have good convergence and robustness [33], but there are still some deficiencies. For the convergence speed and local optimum problem of SSA algorithm, this paper will improve the SSA algorithm through elite reverse learning and firefly algorithm and propose a sparrow search algorithm (EFSSA) with elite reverse and firefly crossover. These two strategies are introduced in detail below.

4.2.1. Elite Reverse Strategy. The initial solution of the traditional sparrow search algorithm is initialized in a random way, and the population diversity is poor. Therefore, in this paper, the elite inverse method is added to the position initialization process. The elite individuals are constructed to reverse the sparrow individuals so that the algorithm has a better initial solution. In 2005, Professor Tizhoosh first proposed the concept of reverse learning (OBL) [44]. This paper points out that the initial values of most intelligent algorithms are mainly based on guessing and then finding or close to the optimal solution after many iterations. Randomly generating the initial solution will greatly impact the solution results. Suppose the random value at the beginning of each iteration is far from the optimal solution or even the opposite. In that case, it will greatly impact the algorithm and consume a lot of update time [45].

Therefore, this paper introduces the reverse solution, which can expand the search area of the algorithm. However, the original solution is higher than the reverse solution for those sparrows with high fitness values. If the reverse solution space is searched, it will be a waste of time, and the original domain search should be strengthened. The value of reverse region search is higher for sparrows whose reverse solution is higher than the original solution. The definition of the inverse solution is given below:

(1) Definition of elite solution: suppose $X_i(t) = [X_{i,1}, X_{i,2}, \cdots, X_{i,j}, \cdots, X_{n,d}]$ as solutions for the t-th iteration, and the solution of its reverse learning is $X_i(t)^*$. We calculate the fitness functions $f(X_i(t))$ and $f(X_i(t)^*)$ of the current solution and elite reverse learning solution, respectively. When $f(X_i(t)) \geq f(X_i(t)^*)$, $X_i(t)$ is an ordinary individual in the current iteration. On the contrary, $X_i(t)$ is the elite individual in the current iteration, which is recorded as $E_i(t)$. The elite solution composed of p elite individuals [34]:

$$\{E_1(t), E_2(t), \cdots, E_p(t)\} \epsilon \{X_1(t), X_2(t), \cdots, X_{\text{n}}(t)\} \tag{12}$$

(2) Definition of inverse solution: assuming that $X_{i,j}$ is an ordinary individual in $X_i(t)$ and its individual inverse solution is represented by $X_{i,j}(t)^*$, then

$$X_{i,j}(t)^* = k(a_j(t) + b_j(t)) - X_i(t) \tag{13}$$

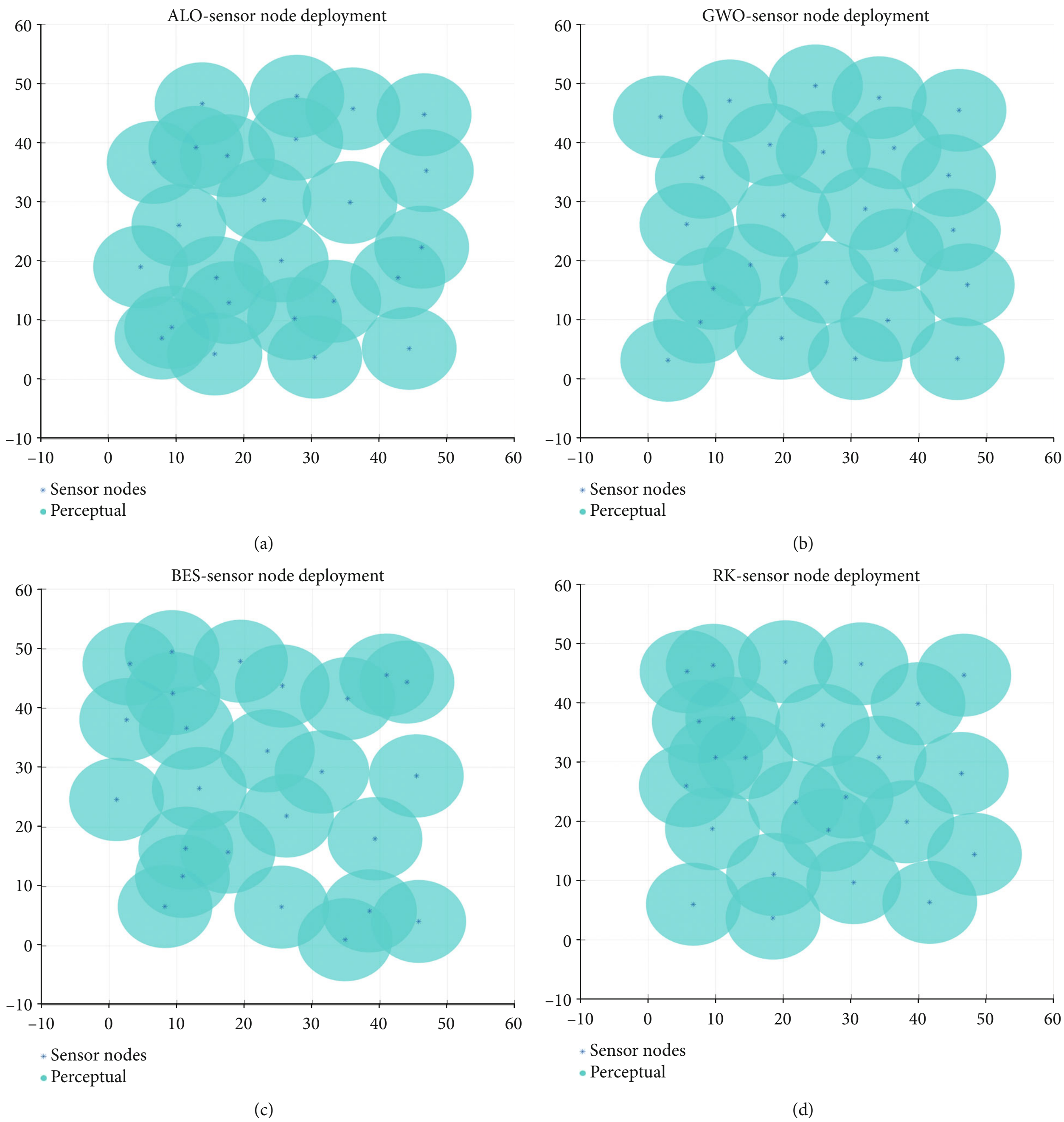
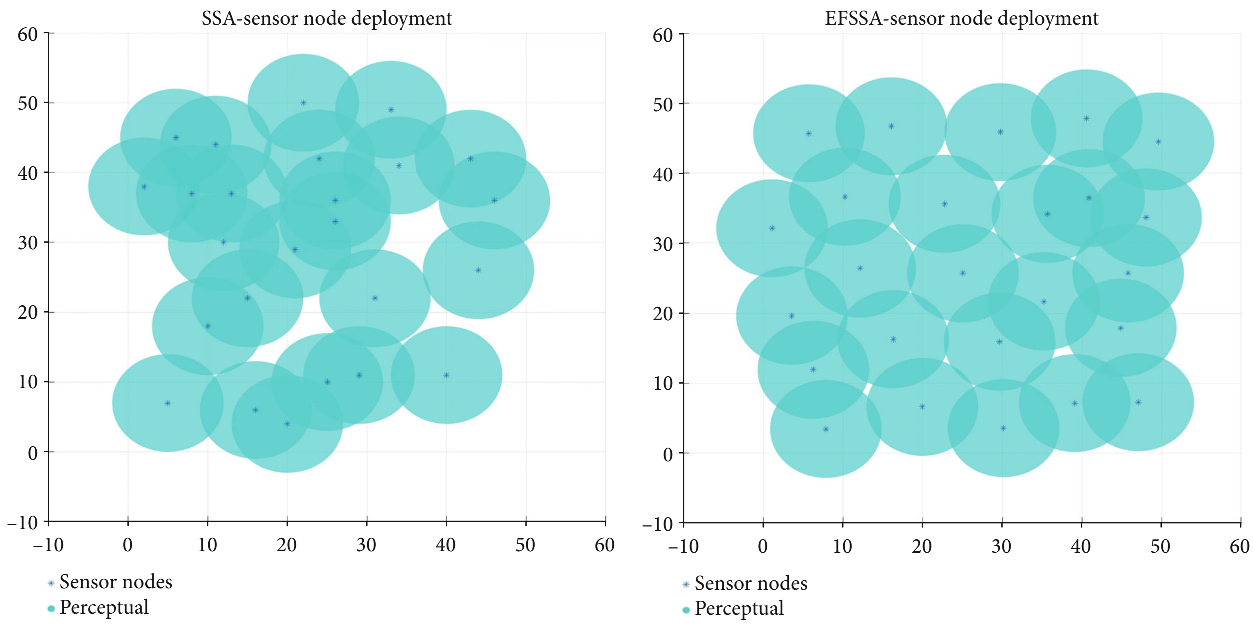


FIGURE 5: Continued.



(e) (f)

Figure 5: Covering effect based on Boolean model. (a) ALO deployment. (b) GWO deployment. (c) BES deployment. (d) RK deployment. (e) SSA deployment. (f) EFSSA deployment.

where $k \in (0,1]$ control step size; the range of the elite group is in interval $[a_j(t), b_j(t)]$, where $a_j(t) = \min \{E_1(t), E_2(t), \cdots, E_p(t)\}$ and $b_j(t) = \max \{E_1(t), E_2(t), \cdots, E_p(t)\}$

4.2.2. Firefly Disturbance Strategy. SSA algorithm is easy to fall into local optimization in the later stage of iteration, which is usually caused by sparrow individuals falling into local optimization in a specific dimension in the calculation process. The firefly algorithm is introduced into the algorithm. All sparrows and the optimal sparrow are disturbed by the algorithm to update the position to improve its search performance. The sparrow after disturbance is compared with the sparrow before disturbance. If it is better, the sparrow position is updated. The main parameters disturbed by the firefly intelligent optimization algorithm include the fluorescence brightness, attraction, and update position of the firefly. The equations are as follows:

Firstly, the expression equation of fluorescence brightness principle based on firefly strategy [35] is

$$I = I_0 \bullet e^{-\gamma r_{i,j}}, \tag{14}$$

where I_0 represents the maximum brightness generated by the firefly population, depending on the function to be optimized. γ is the light intensity absorption coefficient of fireflies. The farther away from the distance, the smaller the coefficient value, and vice versa. $r_{i,j}$ represents the distance between adjacent fireflies. The distance is Cartesian, i.e.,

$$r_{i,j} = \left\| x_i - x_j \right\| = \sqrt{\sum_{k=1}^{d} \left(x_{ik} - x_{jk} \right)^2}. \tag{15}$$

Therefore, in the firefly strategy, the attraction equation of firefly [35] is

$$\beta = \beta_0 \bullet e^{-\gamma r_{i,j}^2}. \tag{16}$$

It can be concluded that the attraction refers to the principle of the fluorescence brightness equation; β_0 indicates the maximum attraction of fireflies.

Suppose the position of firefly i is x_i. When firefly i is attracted by firefly j, it updates its position immediately. The new position calculation equation is

$$X_i^{t+1} = X_i^t + \beta_0 \bullet e^{-\gamma r_{i,j}^2} \bullet \left(x_i - x_j \right) + \alpha \bullet (R - 0.5), \tag{17}$$

where α represents the step size coefficient in the range [0,1] and $R \in [0, 1]$ which is uniformly distributed.

4.2.3. Sensor Deployment Based on Metaheuristic Algorithm. This section gives the flow of EFSSA algorithm (see Algorithm 1 for the specific implementation process).

In Algorithm 1, n individuals are randomly formed in the initial quigroup, and n new individuals are generated in this paper by using elite reverse strategy. Then, the $2n$ individuals are sorted according to the objective function or fitness function, and the first n are selected. The fitness function index R_{cov} of EFSSA applied to WSN in this paper has been given in Equation (6). Next, the location of the finder is updated according to the threshold of the warning value; according to the order of the i-th follower, the joiners are divided into hungry searching individuals and extensive searching individuals. Finally, for the reconnaissance and early warning, whether it is at the population boundary is judged. When the altered individual is

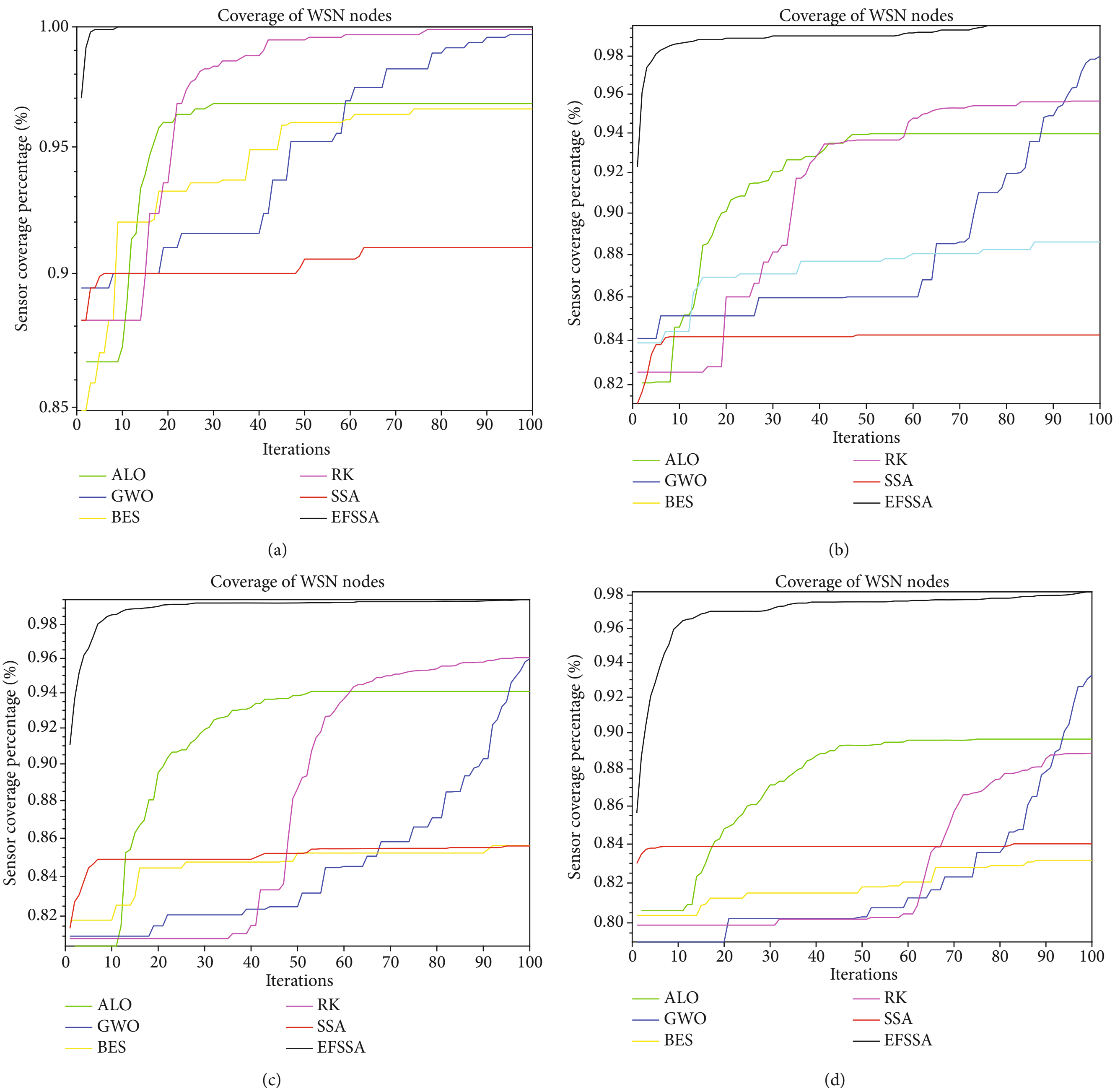


Figure 6: Coverage curve with the number of iterations in balanced deployment mode. (a) 9 nodes are in 30m × 30m. (b) 25 nodes are in 50m × 50m. (c) 50 nodes are in 70m × 70m. (d) 80 nodes are in 90m × 90m.

Table 5: Average rate of increase in coverage of Boolean coverage model.

Nodes	ALO (%)	GWO (%)	BES (%)	RK (%)	SSA (%)	EFSSA (%)
Sparse (-20%)	46.714	48.377	38.571	51.081	40.157	57.782
Balance (9/25/50/80)	33.463	28.561	24.484	31.343	20.625	40.211
Dense (+20%)	28.821	26.910	23.665	29.683	22.422	31.941

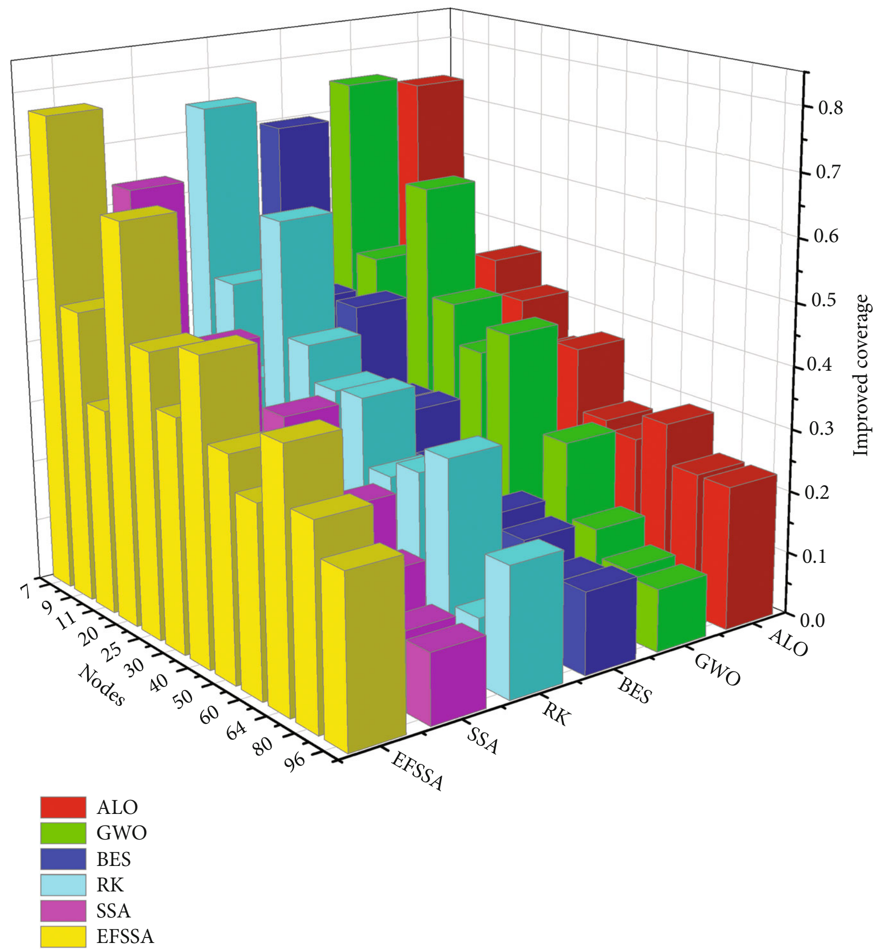


Figure 7: Increased coverage in the Boolean model.

Table 6: Real coverage rate C_e of different nodes and monitoring areas.

Area	Nodes	Initial	ALO	GWO	BES	RK	SSA	EFSSA
30m × 30m	7	0.43617	0.75726	0.76469	0.74056	0.76190	0.71271	0.77211
	9	0.44102	0.62868	0.64745	0.62724	0.64889	0.59115	0.64961
	11	0.39922	0.53032	0.53150	0.52796	0.53150	0.49666	0.53150
50m × 50m	20	0.47487	0.69898	0.76557	0.68339	0.76037	0.67625	0.78051
	25	0.44356	0.62868	0.63662	0.57556	0.62882	0.54723	0.64701
	30	0.39366	0.52878	0.53939	0.51233	0.53593	0.50237	0.54134
70m × 70m	40	0.50962	0.69622	0.72155	0.62996	0.69671	0.69622	0.75777
	50	0.45473	0.57506	0.51312	0.51598	0.57340	0.51496	0.61701
	60	0.40698	0.50908	0.51861	0.48461	0.52389	0.51460	0.53041
90m × 90m	64	0.53705	0.69508	0.61967	0.62819	0.71325	0.63479	0.75751
	80	0.47073	0.57946	0.52283	0.52947	0.51954	0.52645	0.62129
	96	0.42922	0.52808	0.47239	0.48586	0.51820	0.47740	0.54419

at the population boundary, firefly strategy is adopted to move position. Finally, judge whether the maximum number of iterations has been reached $I_{\max}$. If not, continue the iterative calculation; otherwise, output the optimal deployment result.

The optimization performance of SSA algorithm has been proved in literature [33]. This paper will further explore the computational time complexity. Compared with the original SSA algorithm, the time complexity of EFSSA is acceptable. The time complexity of the algorithm is denoted

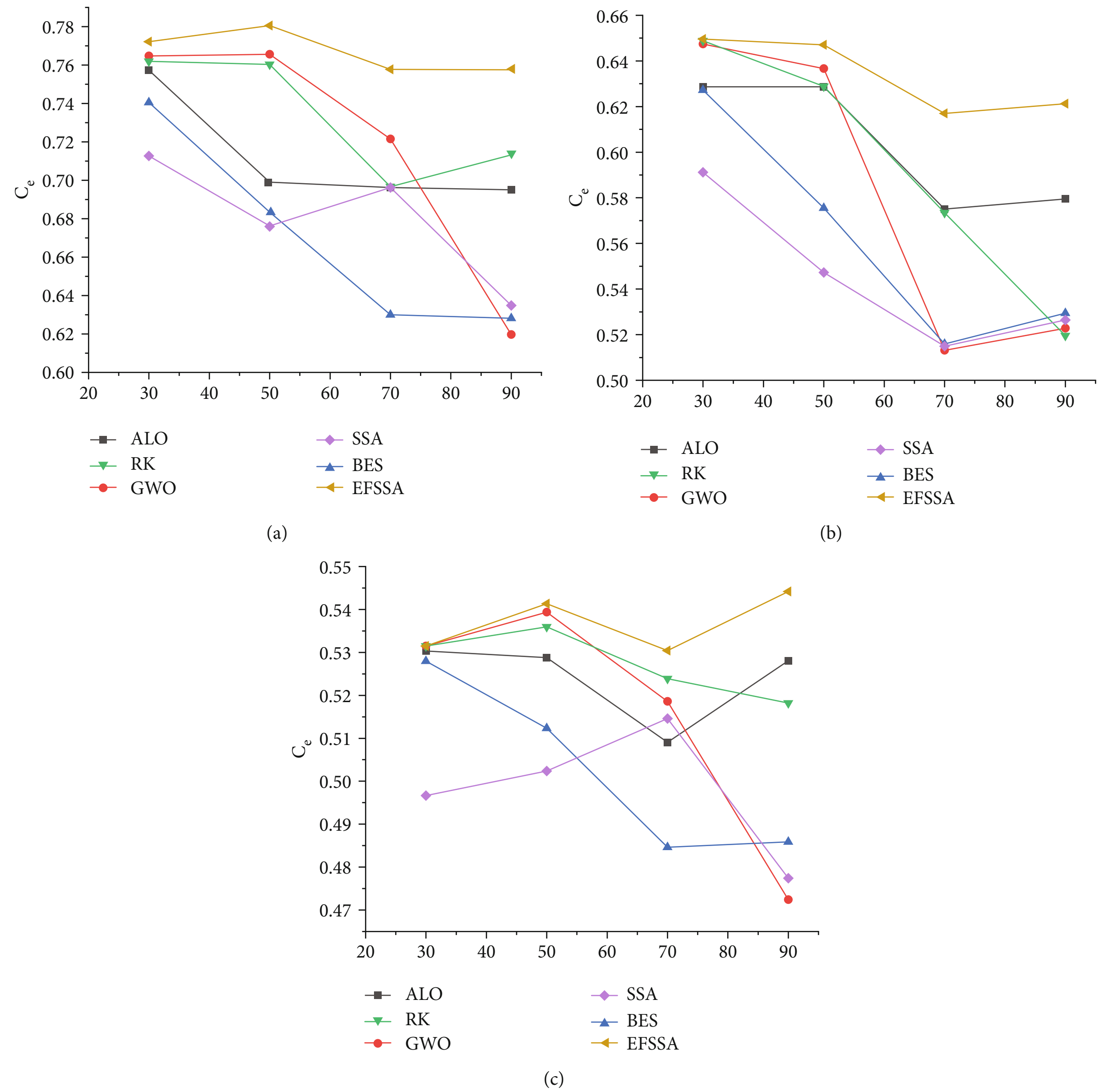


FIGURE 8: Coverage index C_e. (a) Sparse deployment. (b) Balance deployment. (c) Dense deployment.

as $O(\bullet)$, the dimension of individuals in the population is assumed to be m, and the number of populations is n. In the process of population iteration, individuals need to conduct fitness sorting, and the time complexity in the sorting process is $O(n^2)$. Then, the time complexity of finders update is $O(m * F_{num})$, the time complexity of joiners is $O(m * (n - F_{num}))$, and the time complexity of detection and early warning is $O(m * n * DE_{max})$. It can be concluded that the sum of time complexity of finder and follower is $O(mn)$. The proportion of reconnaissance and early warning is 10%~20% [33]. The total time complexity is $O(n * n + mn + mn * DE_{max})$. The calculation results ignore the constant part, and the time complexity after simplification is $O(n^2 + mn)$.

EFSSA adopts elite reverse strategy, so it increases the size of sorting. When sorting, the time complexity of EFSSA is $O((2n)^2)$. The time complexity of finders and joiners is the same as that of SSA. In the part of reconnaissance and early warning, the algorithm uses firefly strategy for some individuals in order to jump out of local optimal. This strategy also uses the distance between two individuals in Equation (11). Hardly increasing the calculation time, the ratio of this part will not exceed DE_{max}. Therefore, the time complexity is $O(4n * n + mn + mn * DE_{max})$, and the simplified complexity is still $O(n^2 + mn)$.

Then, the coverage optimization framework is established according to the sensor deployment model (Algorithm 2). Input parameters related to the sparrow search algorithm and coverage are initialized, and the number of sensor nodes and deployment area is set. Then, the coordinates of the sensor nodes are used as the individuals of the population. Under the calculation of the number of

Table 7: Coverage of different nodes and monitoring areas.

Area	Nodes	Initial	ALO	GWO	BES	RK	SSA	EFSSA
30m × 30m	7	0.54471	0.94119	0.95447	0.93452	0.9427	0.86135	0.95785
	9	0.71403	0.98971	0.99860	0.99970	0.99707	0.92022	1.00000
	11	0.77412	0.99986	1.00000	0.99995	0.99995	0.97898	1.00000
50m × 50m	20	0.61927	0.89713	0.95495	0.85740	0.94564	0.86233	0.98819
	25	0.69384	0.98028	0.98480	0.91030	0.99061	0.87222	0.99845
	30	0.78252	0.99571	0.93871	0.95022	0.99581	0.93016	1.00000
70m × 70m	40	0.67723	0.90482	0.92735	0.83029	0.91366	0.80171	0.97860
	50	0.72356	0.94383	0.96897	0.88245	0.95941	0.88160	0.99971
	60	0.79525	0.97713	0.98469	0.94323	0.98750	0.94301	0.99992
90m × 90m	64	0.68236	0.89391	0.79155	0.78534	0.88172	0.81118	0.95985
	80	0.74505	0.94384	0.94525	0.86679	0.94197	0.87512	0.98739
	96	0.80732	0.98012	0.89001	0.92053	0.96962	0.89005	0.99814

iterations $I_{\max}$, the optimal solution with coverage as the objective function is finally solved through the elite reverse strategy and firefly optimization strategy. The algorithm flow is shown in Figure 2.

5. Experimental Analysis

5.1. Benchmark Function Test. In this section, in order to test the optimization performance of EFSSA algorithm, first compare the algorithm with four related evolutionary optimization algorithms on the benchmark function (shown in Table 1), including ALO [28] algorithm, GWO algorithm [29], BES algorithm [39], RK algorithm [40], and SSA algorithm [33]. The experiment was carried out in the test environment of Intel (R) core i7-8750h CPU, 2.20 GHz, 16 GB memory, and windows10 64 bit and was written with MATLAB 2020b software. Table 1 gives the names, equations, dimensions, independent variable ranges, and optimal global values of 15 classical benchmark functions on cec2008, cec2017, and cec2020.

F1~F7 are unimodal high-dimensional test functions, F8~F11 (see Figure 2) are multimodal test functions, and F12~F15 are fixed low-dimensional test functions. In order to avoid errors in the computation of all the given algorithms due to different parameters, the size of the population was set to 50 and all the experiments were repeated 30 times having 500 iterations independently. Through the benchmark function experiment, the optimal value and standard deviation of each algorithm on the same benchmark function are obtained, so as to evaluate their optimization and stability.

As shown in Table 2, the average value of the optimal optimization of the current function is marked in bold in each row of data. The optimization ability of EFSSA algorithm is better than that of other functions in most test functions. Among the 15 test functions, EFSSA algorithm outperforms others in 9 of them. The high-dimensional unimodal test EFSSA has the best optimization effect in F4 and F5 functions. The mean values of F1~F3 and F7 are second only to those of RK algorithm. This is because RK algorithm calls Runge Kutta method for the function to calculate the logic of slope change, so it is better in single-peak function.

Because the F6 function is a concave canyon function with a strong locality, the variation range in the valley is small, and the optimal value is in the only local region, so the effect of the BES algorithm is better. Among the high-dimensional multimodal test functions, EFSSA algorithm has obvious optimization and stability superior to other algorithms. This is because the multimodal function is easy to make the traditional algorithm fall into local solutions and needs a dispersed and uniform population distribution. Here, the applicability of EFSSA algorithm can be obviously reflected. Finally, for low-dimensional functions F12~F15, EFSSA algorithm can basically maintain the superiority of the algorithm, although the performance of F12 is worse than that of BES, it is still superior to other algorithms.

Because functions F8~F11 and sensor network coverage are similar high-dimensional multimodal function problems, we further analyze the testing effect of EFSSA algorithm in this kind of problem. Figures 3 and 4 show the objective function curves and the function graph of benchmark functions Schwefel, Rastrigrin, Ackley, and Griewank with the number of iterations.

As shown from the Figure 4, the EFSSA algorithm has better convergence with the increase of iteration times compared with ALO, GWO, BES, RK, and SSA algorithms. Its optimization performance is also the best. In general, EFSSA performs better than other algorithms in testing functions, especially in high-dimensional multimodal function problems. In the next section, this paper will further verify the effect of EFSSA algorithm in the sensor coverage deployment experiment.

5.2. Simulation Experiment of Sensor Network Coverage. In this section, in order to test the optimization performance of EFSSA algorithm, firstly, the critical parameters of EFSSA algorithm and five related evolutionary optimization algorithms are given. In the experiment, the wireless network perception model uses Boolean model and probability model. Then, the simultaneous interpreting of the two deployment effects of sensors with different monitoring areas and different sensor numbers is compared.

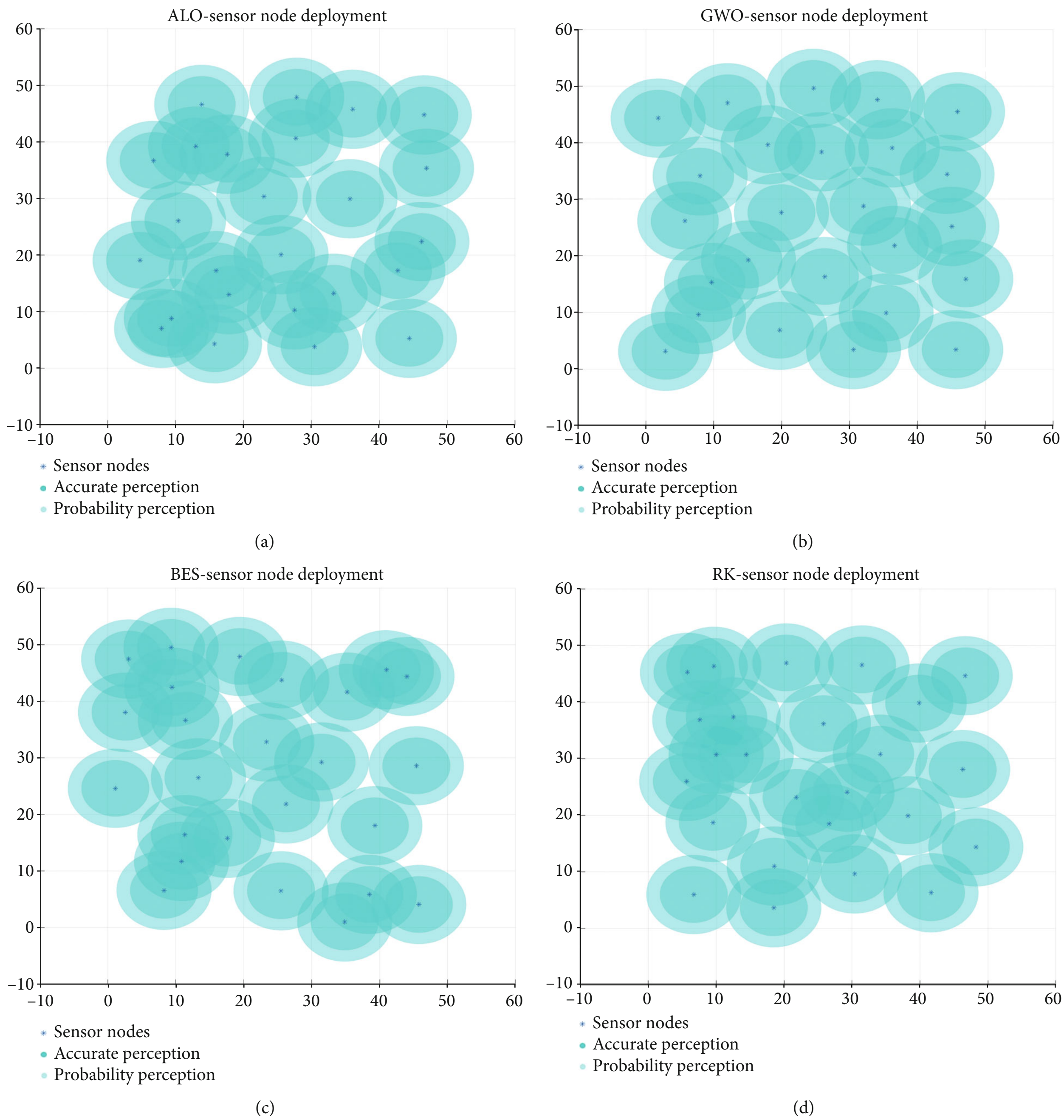
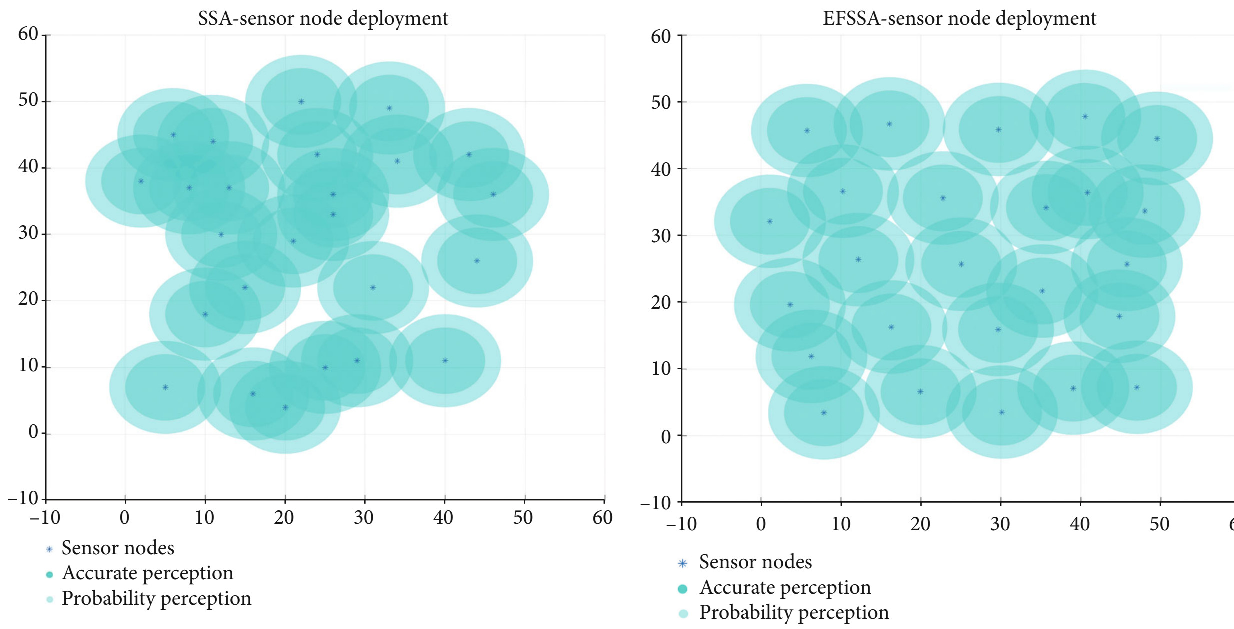


Figure 9: Continued.



(e) (f)

FIGURE 9: Covering effect based on probability model. (a) ALO deployment. (b) GWO deployment. (c) BES deployment. (d) RK deployment. (e) SSA deployment. (f) EFSSA deployment.

5.2.1. Parameter Setting. In this section, the ability of EFSSA algorithm to solve the coverage problem of wireless sensor networks will be evaluated through a series of simulation experiments. The software and physical platforms tested by the above benchmark algorithm are still used, and the MATLAB experimental environment is used to simulate sensor the coverage effect of sensor deployment. The critical parameters used in the experiment of ALO [28] algorithm, GWO algorithm [29], BES algorithm [39], RK algorithm [40], and SSA algorithm [33] are set by default. For example, the parameters in SSA and FA algorithm come from literature [33, 35]. The parameters of monitoring area parameters and the EFSSA algorithm are shown in Table 3.

The monitoring area for sensor deployment is divided into 900m^2, 2500m^2, 4900m^2, and 8100m^2 two-dimensional monitoring space with three gradient sizes. In all experiments, except for the same monitoring area, the number of iterations of all algorithms is 100, the size of the population is 50, and the number of independent runs of algorithms is 30. Due to the high computational cost of solving practical problems and finding the optimal solution when the number of iterations is small, the number of iterations in literature [33] is set to 50. In order to make the experiment more convincing, this paper increases the number of iterations. In this paper, the finder proportion parameters, detection and early warning parameters, and early warning values of the SSA algorithm and EFSSA algorithm are set to 0.2, 0.15, and 0.8, respectively. In order to study the sensor deployment under different node densities, the experiments of sparse and dense deployment are designed, respectively, on the basis of regular hexagon-balanced node deployment (an approximate integer). The number of sparse and dense nodes is set to 80% and 120% of balanced deployment, respectively.

TABLE 8: Average rate of increase in coverage of probability coverage model.

Nodes	ALO (%)	GWO (%)	BES (%)	RK (%)	SSA (%)	EFSSA (%)
Sparse	45.566	45.591	36.927	47.474	33.660	55.146
Balance	34.254	35.644	27.376	35.360	23.471	38.661
Dense	25.577	20.754	20.602	25.334	18.784	27.373

5.2.2. Boolean Coverage Experiment. For Boolean coverage model in Equation (2), through the test results of the ALO, GWO, BES, RK, SSA, and EFSSA algorithms, this paper verifies the coverage efficiency of these six different algorithms in a sensor network monitoring environment.

Firstly, the monitoring area is completely covered without considering the boundary effect, and the optimal deployment strategy of a regular hexagon is used to calculate the sensor nodes that need to be evenly deployed within different monitoring area. When all sensor nodes are randomly deployed in a two-dimensional plane, the initial input positions are the same for all algorithms. It can be seen from Table 4 that when 9 nodes are deployed within $30\text{m} \times 30\text{m}$ monitoring area, the coverage rate (R_{cov}) of EFSSA is 92.444%. Deploying 25 nodes in the $50\text{m} \times 50\text{m}$ area, the coverage results of algorithms ALO, GWO, BES, RK, and SSA are 96.788%, 98%, 96.8%, and 84.24%, respectively. In

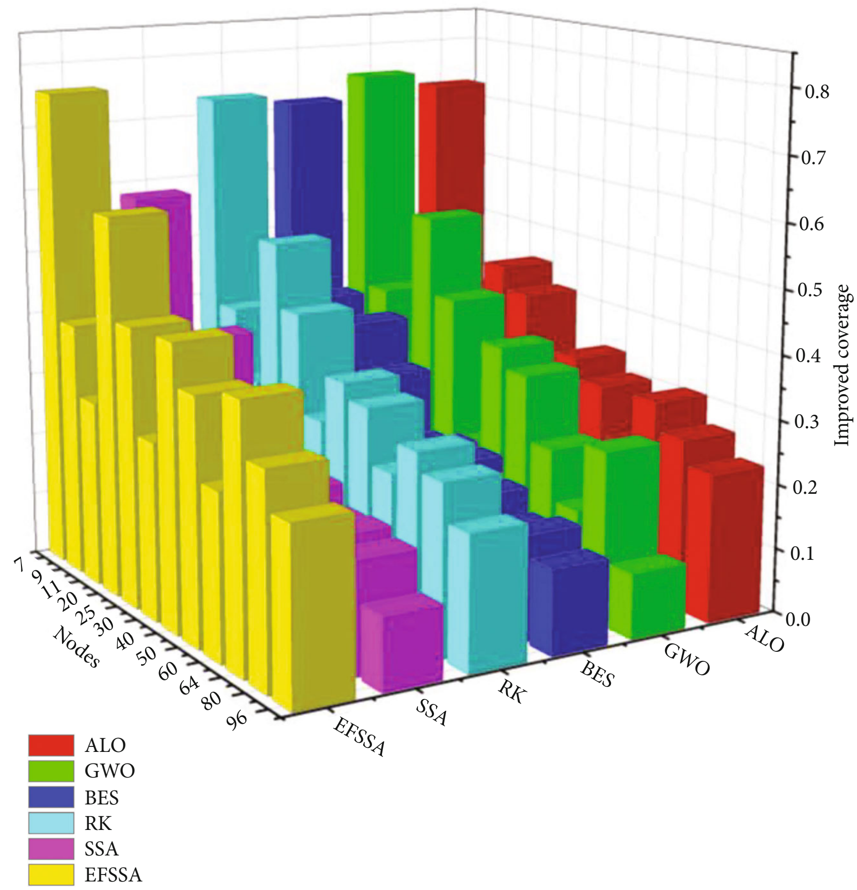


Figure 10: Increased coverage in the probability model.

contrast, the coverage of EFSSA algorithm is 99.6%. In 70 m × 70m area, the number of nodes under the optimal deployment mode is 50. With the second deployment of these sensors, the coverage of EFSSA algorithm reaches 96.92%, which is increased by 6.8%, 16.37%, 16.86%, 16.83%, and 16.54%, respectively, compared with the original coverage, and the coverage of ALO, GWO, BES, RK, and SSA algorithms is 90.33%, 81.05%, 80.58%, 80.6%, and 80.89%, respectively. In 90m × 90m deployment 80 sensor nodes, the coverage of ALO, GWO, BES, RK, SSA, and EFSSA is 88.1%, 79.49%, 80.5%, 78.99%, 80.04%, and 94.46%. In the simulation experiments of wireless sensor deployment optimization under the Boolean model, EFSSA significantly outperforms other algorithms in terms of coverage after secondary deployment of nodes.

In Figure 5, the effect of regional balanced deployment is shown. The uniformity of EFSSA with high coverage is obviously better than that of SSA and BES with low coverage. In Figure 5(f), the EFSSA sensor deployments are nearly universal, but some overlying voids can still be seen. After the optimization of these algorithms for second deployment, which is due to the fact that the experiments in this paper divide the space into two-dimensional pixel points and the noninteger coordinates are not included in the detection area.

In Figure 6, the variation of coverage of different algorithms with the number of iterations in the balanced deployment mode is statistically analyzed. As the monitoring area becomes larger, the convergence speed of each algorithm is affected, because the increase of dimension leads to the difficulty of solving. However, EFSSA algorithm can maintain the superiority of the algorithm and can always be quickly to complete convergence in the case of different monitoring area sizes. Other algorithms, such as GWO algorithm, increase the monitoring range from 900m^2 to 8100m^2, the number of iterations approaching the optimal solution increases from 50 to 90 times, and the final coverage is still less than EFSSA algorithm. All SSA algorithms without improved strategy fall into local optimal solution after less than 10 iterations.

This paper analyzes the coverage effect with the number of iterations under these three different monitoring scales (sparse, balanced, and dense). Assuming that the number of sensor nodes is changed, set the sensors to ±20% of the balanced deployment number. When seven nodes are deployed in 30m × 30m region, the coverage of EFSSA can still reach 92.444%, while the deployment effect of SSA algorithm is the worst. When the number of nodes increases to 11, GWO, RK, and EFSSA algorithms almost achieve complete coverage, and only EFSSA can achieve complete

Table 9: Real coverage rate C_e of different nodes and monitoring areas.

Area	Nodes	Initial	ALO	GWO	BES	RK	SSA	EFSSA
30m × 30m	7	0.45495	0.78610	0.79719	0.78053	0.78736	0.71941	0.80001
	9	0.46384	0.64293	0.64870	0.64942	0.64771	0.59779	0.64961
	11	0.41145	0.53143	0.53150	0.53147	0.53147	0.52033	0.53150
50m × 50m	20	0.50286	0.72848	0.77543	0.69622	0.76787	0.70022	0.80243
	25	0.45073	0.63680	0.63974	0.59134	0.64351	0.56660	0.64861
	30	0.42361	0.53902	0.50816	0.51440	0.53908	0.50354	0.54134
70m × 70m	40	0.53892	0.72003	0.73796	0.66072	0.72707	0.63798	0.77875
	50	0.46063	0.60086	0.61687	0.56179	0.61078	0.56124	0.63644
	60	0.42189	0.51838	0.52239	0.50040	0.52389	0.50028	0.53047
90m × 90m	64	0.56101	0.73494	0.65078	0.64568	0.72492	0.66692	0.78916
	80	0.49004	0.62079	0.62172	0.57012	0.61956	0.57559	0.64944
	96	0.44250	0.53721	0.48782	0.50455	0.53146	0.48785	0.54709

coverage among 60 nodes deployed in 50m × 50m. The optimization performance of RK and GWO algorithms proposed by researchers is second only to EFSSA, but their optimization speed is not as fast as EFSSA in sensor deployment problems.

Table 5 shows the percentage of average coverage improvements for these algorithms at different deployment densities. With the increase of node deployment density, the coverage improvement effect of the six algorithms decreases. The reason is the increase in the number of sensor nodes makes the coverage of the initial immediate deployment of sensors larger, so the improvement effect of reoptimization decreases. From another perspective, with the number of sensor nodes increases, the complexity of optimization increases, but the optimization effect decreases. EFSSA algorithm can still maintain a coverage improvement effect of more than 30% in dense mode, which fully demonstrates the applicability of this algorithm. In Figure 7, the data of specific coverage improvement is described, from which we can see the improvement effect of different nodes under different algorithms; EFSSA is superior to other methods.

In terms of node utilization, Table 6 shows the results of performance metrics C_e. In the case of random deployment within the region, 30m × 30m of balanced deployment is close to sparse deployment except for 30 regions. As the number of nodes increases, other regions gradually decrease, indicating that the effective coverage ratio decreases gradually. These optimization algorithms optimize the coverage of the target monitoring area while reducing the real coverage index of the sensor to varying degrees. To further reflect the relationship between sensor coverage cost and coverage, Figure 8 shows different density levels, and the abscissa in the figure represents the length of square deployment area (unit: meter). In the sparse deployment experiment (Figure 8(a)), the real deployment coverage C_e of EFSSA algorithm is higher than other algorithms, resulting in relatively small redundant coverage. In the process of increasing the deployment area, the coverage rate of GWO algorithm is inferior to EFSSA. In the case of 90m × 90m area, C_e of GWO is smaller than other algorithms. In the balanced deployment experiment (Figures 8(b) and 8(c)), the effective coverage C_e of EFSSA algorithm is still superior to other algorithms. RK algorithm and GWO algorithm are generally close to EFSSA, but they have poor performance in large-scale deployment experiments.

5.2.3. Probability Coverage Experiment. According to Equation (3), this section further analyzes the sensor coverage effects of different optimization methods under the probabilistic coverage model (see Table 7).

In the probabilistic perception model structure, the coverage of the sensor edge is strongly influenced by α and β. Here, we take the value of both 0.5. Compared with Table 7, the probability of sensor coverage generally increased because of the coverage edge change from r_s in the original Boolean model to $r_s + \Delta r$ in the probabilistic model, and in this paper, Δr is 0.5 m. In Table 7, the final probabilistic coverage solved by the EFSSA algorithm outperforms all other several algorithms.

In Figure 9, the final distribution result graphs of the six algorithms, ALO, GWO, BES, RK, SSA and EFSSA, are shown when 25 nodes are deployed in a 50m × 50m area, respectively. It is clear from the figure that the coverage effect of EFSSA algorithm is the most uniform. In the probabilistic coverage model, the probabilistic coverage edges are indicated by light colors.

Similar to the Boolean model experiment, the coverage improvement effect of different algorithms under different density deployment under probability model is also given in Table 8. Although the probabilistic perception model is not completely aware of some perception regions, the perception range is increased by 0.5 m. Therefore, the initial random coverage was increased, and the improvement rate of EFSSA algorithm in dense areas was decreased compared with that of Boolean model. In Figure 10, the percentage

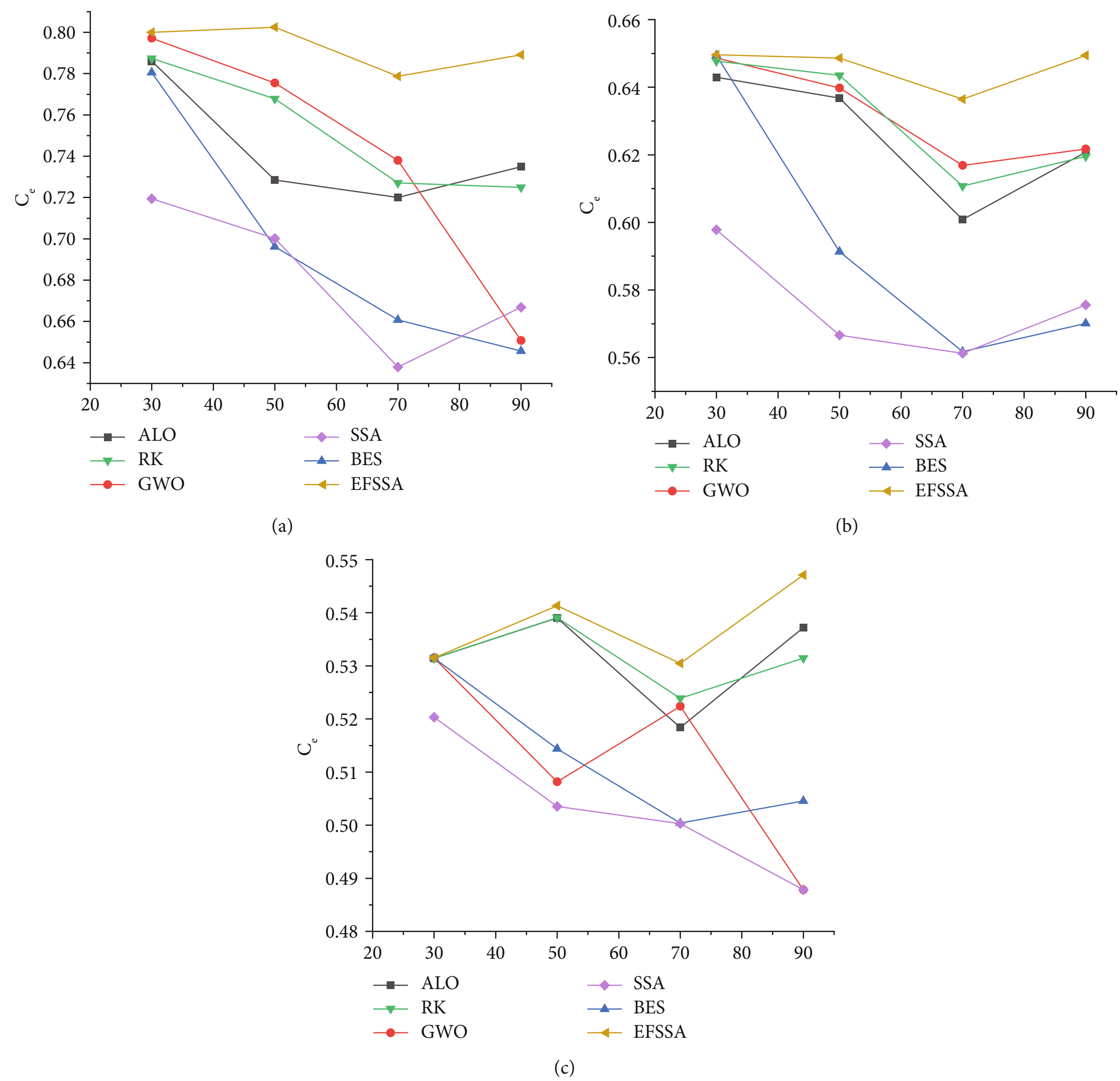


FIGURE 11: Coverage index C_e. (a) Sparse deployment. (b) Balance deployment. (c) Dense deployment.

improvement of these six algorithms compared to the initial coverage is shown. It can be obtained in the figure that the percentage improvement of the optimization algorithm is relatively high in the region with sparse number of nodes. And EFSSA algorithm has the highest coverage improvement compared to the other five algorithms.

In the probability model experiment, the evaluation index results C_e of real coverage are shown in Table 9. Combined with the results of EFSSA algorithm in Table 7 in deploying 9 nodes and 11 nodes, it can be found that when the sensor achieves full coverage deployment, the increase in the number of nodes will make the value of C_e drop rapidly. With three sensor deployments of different densities (see Figure 11), the C_e of the probabilistic model is generally superior to the Boolean model. The variation trend of the real coverage is almost the same, among which the variation of C_e of SSA algorithm changes greatly. Overall, the C_e of EFSSA is high and stable.

6. Conclusions

An improved metaheuristic algorithm EFSSA has been proposed and successfully applied to solve the node coverage problem of two-dimensional wireless sensor networks. Based on the original sparrow search algorithm, the elite reverse strategy and firefly strategy were combined to improve the generalization ability of the initial population and the global search ability of the population. The original sparrow search algorithm is prone to the stagnation of local optimization in the problem of high-dimensional and multilocal extremum of sensor deployment. The fluorescence effect of the firefly strategy can change the defect that the sparrow algorithm falls into local optimization. Simulation and experimental results show that the EFSSA algorithm can effectively accelerate the convergence speed of solving the optimal coverage and avoid local optima in high-dimensional problems. In Boolean model and probabilistic model experiments, the

EFSSA algorithm is more effective and feasible in applying second sensor deployment than other algorithms.

However, the method proposed in this paper has some limitations. In the experiment of sensor deployment index, the value of C_e will also drop sharply when the increase of the number of nodes almost achieves full coverage ($R_{cov} = 1$). Coverage rate R_{cov} and cost of sensor nodes are contradictory issues, and this paper does not balance this issue. It can be regarded as a multiobjective problem in the future research. In this problem, these two indicators and other possible indicators (such as network connectivity, throughput, and network delay) can be considered to establish a multiobjective model of WSN coverage, solve Pareto frontier, and give an effective scheme. There are still many challenges in the future work of sensor network deployment. For example, more constraints can be introduced, such as terrain constraints, heterogeneous sensor constraints, and deployment dimension constraints. Moreover, the optimization issues of applying EFSSA to IoT-based UAV data collection and edge computing-based sensor data collection also deserve further study.

Data Availability

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

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This research was supported in part by the China National Key Research and Development (No. 2018YFC0808306), Hebei Key Research and Development Program (No. 19270318D), Hebei Internet of Things Engineering and Technology Research Center (No. 3142018055), and Qinghai Internet of Things Key Laboratory (No. 2017-ZJ-Y21).

References

[1] A. Shahraki, A. Taherkordi, Ø. Haugen, and F. Eliassen, "A survey and future directions on clustering: from WSNs to IoT and modern networking paradigms," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 2242–2274, 2021.

[2] Z. A. A. Aiziz and S. Y. A. Ameen, "Air pollution monitoring using wireless sensor networks," *Journal of Information Technology and Informatics*, vol. 1, no. 1, pp. 20–25, 2021.

[3] Q. Yu, F. Xiong, and Y. R. Wang, "Integration of wireless sensor network and IoT for smart environment monitoring system," *Journal of Interconnection Networks*, vol. 2021, article 2143010, 2021.

[4] I. Ahmad, K. Shah, and S. Ullah, "Military applications using wireless sensor networks: a survey," *International Journal of Engineering Science and Computing*, vol. 6, no. 6, article 2143010, 2016.

[5] M. Erdelj, M. Król, and E. Natalizio, "Wireless sensor networks and multi-UAV systems for natural disaster management," *Computer Networks*, vol. 124, pp. 72–86, 2017.

[6] G. J. Han, X. Yang, L. Liu, W. B. Zhang, and M. Guizani, "A disaster management-oriented path planning for mobile anchor node-based localization in wireless sensor networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 1, pp. 115–125, 2020.

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

[8] K. Xu, Z. Zhao, Y. Luo, G. Hui, and L. Hu, "An energy-efficient clustering routing protocol based on a high-QoS node deployment with an inter-cluster routing mechanism in WSNs," *Sensors*, vol. 19, no. 12, p. 2752, 2019.

[9] A. N. Njoya, A. A. A. Ari, M. N. Awa et al., "Hybrid wireless sensors deployment scheme with connectivity and coverage maintaining in wireless sensor networks," *Wireless Personal Communications*, vol. 112, no. 3, pp. 1893–1917, 2020.

[10] 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.

[11] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "Span: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *Wireless Networks*, vol. 8, no. 5, pp. 481–494, 2020.

[12] F. Aznoli and N. J. Navimipour, "Deployment strategies in the wireless sensor networks: systematic literature review, classification, and current trends," *Wireless Personal Communications*, vol. 95, no. 2, pp. 819–846, 2017.

[13] M. Rout and R. Rajarshi, "Dynamic deployment of randomly deployed mobile sensor nodes in the presence of obstacles," *Ad Hoc Networks*, vol. 46, pp. 12–22, 2016.

[14] Z. Wang, L. Tian, L. Lin, and Y. Tong, "Lattice-based 3-dimensional wireless sensor deployment," *Journal of Sensors*, vol. 2021, Article ID 2441122, 14 pages, 2021.

[15] M. Kumar and V. Gupta, "A review paper on sensor deployment techniques for target coverage in wireless sensor networks," in *2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, pp. 452–456, Kumaracoil, India, 2016.

[16] P. K. Donta, T. Amgoth, and C. S. R. Annavarapu, "ACO-based mobile sink path determination for wireless sensor networks under non-uniform data constraints," *Applied Soft Computing*, vol. 69, pp. 528–540, 2018.

[17] M. R. Senouci, A. Mellouk, and A. Aissani, "Random deployment of wireless sensor networks: a survey and approach," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 15, no. 1/2/3, pp. 133–146, 2014.

[18] 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.

[19] C. Liu and H. Du, "t,K-Sweep coverage with mobile sensor nodes in wireless sensor networks," *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 13888–13899, 2021.

[20] D. Kandris, A. Alexandridis, T. Dagiuklas, E. Panaousis, and D. Vergados, "A survey and future directions on clustering: From WSNs to IoT and modern networking paradigms," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 4652801, 5 pages, 2020.

[21] K. Tarnaris, I. Preka, D. Kandris, and A. Alexandridis, "Coverage and k-coverage optimization in wireless sensor networks using computational intelligence methods: a comparative study," *Electronics*, vol. 9, no. 4, p. 675, 2020.

[22] A. Sangwan and R. P. Singh, "Survey on coverage problems in wireless sensor networks," *Wireless Personal Communications*, vol. 80, no. 4, pp. 1475–1500, 2015.

[23] S. S. Dhillon and K. Chakrabarty, "Sensor placement for effective coverage and surveillance in distributed sensor networks," *Wireless Communications and Networking, WCNC*, vol. 3, pp. 1609–1614, 2003.

[24] K. Xu, G. Takahara, and H. Hassanein, "On the robustness of grid-based deployment in wireless sensor networks,," in *Proceedings of the 2006 international conference on Wireless communications and mobile computings*, pp. 1183–1188, New York, United States, 2006.

[25] S. Miini, S. K. Udgata, and S. L. Sabat, "Sensor deployment and scheduling for target coverage problem in wireless sensor networks," *IEEE Sensors Journal*, vol. 14, no. 3, pp. 636–644, 2014.

[26] N. T. Hanh, P. L. Nguyen, P. T. Tuyen, H. Binh, and Y. Ji, "Node placement for target coverage and network connectivity in WSNs with multiple sinks," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, Flamingo Las Vegas, United States, 2018.

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

[28] W. Liu, S. Yang, S. Sun, and S. Wei, "A node deployment optimization method of WSN based on ant-lion optimization algorithm," in *Proceeding of IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACSSWS)*, pp. 88–92, Lviv, Ukraine, 2018.

[29] A. Shahraki, A. Taherkordi, Ø. Haugen, and F. Eliassen, "Node coverage optimization algorithm for wireless sensor networks based on improved grey wolf optimizer," *Journal of Algorithms & Computational Technology*, vol. 13, 2019.

[30] W. H. Liao, Y. Kao, and Y. S. Li, "A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12180–12188, 2011.

[31] X. S. Yang, "Firefly algorithms for multimodal optimization," in *International symposium on stochastic algorithms*, Springer, Berlin, Heidelberg, 2009.

[32] D. K. Sah, K. Cengiz, P. K. Donta, V. N. Inukollu, and T. Amgoth, "EDGF: empirical dataset generation framework for wireless sensor networks," *Computer Communications*, vol. 180, pp. 48–56, 2021.

[33] J. K. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.

[34] H. Zhang and Z. Li, "Energy-aware data gathering mechanism for mobile sink in wireless sensor networks using particle swarm optimization," *IEEE Access*, vol. 8, pp. 177219–177227, 2020.

[35] J. Sengathir, A. Rajesh, G. Dhiman, S. Vimal, C. A. Yogaraja, and W. Viriyasitavat, "A novel cluster head selection using hybrid artificial bee colony and firefly algorithm for network lifetime and stability in WSNs," *Connection Science*, vol. 34, no. 1, pp. 387–408, 2022.

[36] X. Wang, Y. Deng, and H. Duan, "Edge-based target detection for unmanned aerial vehicles using competitive bird swarm algorithm," *Aerospace Science and Technology*, vol. 78, pp. 708–720, 2018.

[37] C. Li, Y. Yue, and Y. Zhang, "A data collection strategy for heterogeneous wireless sensor networks based on energy efficiency and collaborative optimization," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 9808449, 13 pages, 2021.

[38] C. Ouyang, D. Zhu, and F. Wang, "A learning sparrow search algorithm," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 3946958, 23 pages, 2021.

[39] K. Kapileswa and P. Phani Kumar, "Energy efficient routing in IOT based UWSN using bald eagle search algorithm," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 1, 2022.

[40] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen, "RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method," *Expert Systems with Applications*, vol. 181, article 115079, 2021.

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

[42] H. M. Ammari, "Connected k-coverage in two-dimensional wireless sensor networks using hexagonal slicing and area stretching," *Journal of Parallel and Distributed Computing*, vol. 153, pp. 89–109, 2021.

[43] S. Oktug, A. Khalilov, and H. Tezcan, "3D coverage analysis under heterogeneous deployment strategies in wireless sensor networks," in *In 2008 Fourth Advanced International Conference on Telecommunications*, pp. 119–204, Athens, Greece, 2008.

[44] H. R. Tizhoosh, "A survey and future directions on clustering: from WSNs to IoT and modern networking paradigms," in *International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06)*, pp. 675–701, Sydney, Australia, 2006.

[45] X. Wu, X. Shen, J. Zhang, and Y. Zhang, "A wind energy prediction scheme combining Cauchy variation and reverse learning strategy," *Advances in Electrical and Computer Engineering*, vol. 21, no. 4, pp. 3–10, 2021.