Mathematical Biosciences and Engineering

MBE, 21(2): 2787–2812.
DOI: 10.3934/mbe.2024124
Received: 14 October 2023
Revised: 21 December 2023
Accepted: 02 January 2024
Published: 24 January 2024

http://www.aimspress.com/journal/MBE

Research article

An optimization method for wireless sensor networks coverage based on genetic algorithm and reinforced whale algorithm

Shuming Sun, Yijun Chen and Ligang Dong*

School of Information and Electronic Engineering (Sussex Artificial Intelligence Institute), Zhejiang Gongshang University, Hangzhou 310018, China

* **Correspondence:** Email: jiangxian@zjgsu.edu.cn; Tel: +8615988495509.

Abstract: In response to the problem of coverage redundancy and coverage holes caused by the random deployment of nodes in wireless sensor networks (WSN), a WSN coverage optimization method called GARWOA is proposed, which combines the genetic algorithm (GA) and reinforced whale optimization algorithm (RWOA) to balance global search and local development performance. First, the population is initialized using sine map and piecewise linear chaotic map (SPM) to distribute it more evenly in the search space. Secondly, a non-linear improvement is made to the linear control factor 'a' in the whale optimization algorithm (WOA) to enhance the efficiency of algorithm exploration and development. Finally, a Levy flight mechanism is introduced to improve the algorithm's tendency to fall into local optima and premature convergence phenomena. Simulation experiments indicate that among the 10 standard test functions, GARWOA outperforms other algorithms with better optimization ability. In three coverage experiments, the coverage ratio of GARWOA is 95.73, 98.15, and 99.34%, which is 3.27, 2.32 and 0.87% higher than mutant grey wolf optimizer (MuGWO), respectively.

Keywords: wireless sensor network; coverage optimization; energy saving; whale optimization algorithm; genetic algorithm

1. Introduction

1.1. Background

Wireless Sensor Network (WSN) [1] is a network composed of a series of spatially distributed sensor nodes used to collect, process, and transmit various types of data. Due to the large number of nodes in WSN and the relatively short communication distance between nodes, the design and implementation of WSN face many challenges and difficulties, such as energy constraints, low coverage, and redundant node distribution [2]. Among these challenges, coverage optimization has always been one of the most fundamental issues plaguing WSNs. The coverage optimization problem in WSN can be described as the node deployment problem within a specified monitoring area while ensuring the connectivity of the sensor network. It determines the service quality of the wireless sensor network. A reasonable and effective node deployment strategy can not only reduce network costs but also greatly improve network efficiency [3]. In most cases, sensor nodes are randomly placed in the target monitoring area, resulting in uneven distribution of sensor nodes and low coverage. Therefore, improving the coverage of WSN is of great significance for the future development of wireless sensor network applications.

The traditional deployment of nodes in WSN faces problems of low coverage and redundant node distribution. In recent years, many intelligent optimization algorithms have been applied to WSN coverage optimization. Common optimization algorithms include the grey wolf optimizer (GWO) [4], the whale optimization algorithm (WOA) [5], and the genetic algorithm (GA) [6]. GWO can balance local optimization and global search, but it has the drawbacks of being prone to premature convergence, low convergence accuracy, and insufficient convergence speed for complex problems. WOA has fast convergence speed and strong global search capability, but it is prone to premature convergence for certain problems. GA has strong global search capability, but it has a slow convergence speed. Optimization algorithms face a series of challenges in practical applications, including low solution accuracy, a slow optimization process, susceptibility to local optima, and a narrow search space range.

To overcome the aforementioned limitations, many scholars have conducted valuable research on optimizing WSN coverage using intelligent optimization algorithms. Jin et al. proposed the improved sparrow search algorithm (MSSA) that integrates multiple strategy enhancements [7]. In the initialization stage of the sparrow search algorithm, they introduced an excellent point-set strategy for population initialization to ensure diversity and thoroughness. MSSA is then applied to optimize WSN coverage, addressing the two key issues of network coverage and network lifespan balance. Zhang et al. introduced the hybrid particle swarm butterfly algorithm (HPSBA) and designed a control strategy for parameter ‘c’ based on logistic mapping [8]. HPSBA optimizes coverage with a higher coverage ratio, effectively reducing node redundancy and extending the lifespan of WSN. Trong-The Nguyen et al. used elite backward learning, multi-directional strategy modification, and updating equations to propose the improved honey-badger algorithm (IHBA) [9]. The IHBA algorithm exhibits effective optimization performance, convergence speed, and increased feasible coverage. Zeng et al. proposed an improved wild horse optimizer algorithm (IWHO) [10], which uses backward learning and a Cauchy mutation strategy to escape local optima and broaden the search space, achieving better and more effective sensor connectivity and coverage. Sajjad et al. improved the existing GWO by mutating some of the original agents and designing a mutant grey wolf optimizer (MuGWO) [11]. It enhances resource utilization by maximizing coverage while maintaining connectivity.

At present, scholars both domestically and internationally have conducted extensive research on the coverage optimization problem in planar area wireless sensor networks and have made certain progress. The main research methods can be divided into two categories: coverage optimization based on geometric methods and coverage optimization based on intelligent algorithms. Mahboubi et al. [12] proposed a node deployment algorithm based on the Voronoi diagram, using virtual forces applied by polygon vertices and boundaries to find new positions for nodes. Simulation results demonstrate that this algorithm can enhance coverage. Liu et al. [13] optimized node deployment using an improved virtual force algorithm. This algorithm, considering the interaction forces between nodes based on traditional virtual force algorithms, can drive sensor nodes to cover the entire monitoring area. However, it still faces challenges in achieving high coverage ratio.

In recent years, with the development and widespread application of intelligent algorithms, researchers have found that intelligent algorithms can effectively address the coverage optimization problem in wireless sensor networks. Intelligent algorithms have the advantages of simple computation and strong search capabilities. Mahnaz et al. [14] proposed an improved Whale Optimization Algorithm for WSN coverage optimization. It tackles complex coverage issues through methods like exploration development, spiral attack, and bubble-net attack. Experimental results demonstrate that this algorithm can prolong the network's lifespan, but it is prone to getting stuck in local optima. Kavita et al. [15] introduced a node deployment method based on GWO. This approach maximizes coverage while ensuring network connectivity by enhancing the fitness function and population position updating method. Yin et al. [16] presented a wireless sensor network coverage optimization method based on a Yin-Yang Crow-inspired optimization algorithm, which improves coverage and convergence. Bacanin et al. [17] introduced a quasi-reflective learning algorithm that overcame the shortcomings of the original WOA and was used for WSN localization. Wang et al. [18] designed an efficient routing algorithm based on elite hybrid metaheuristic optimization algorithm to maximize the survival time of wireless sensor network routing with aggregation nodes. Zivkovic et al. [19] proposed an improved version of GWO, that had been applied to improve the network lifetime optimization. Xue et al. [20] proposed a self-adaptive particle swarm optimization algorithm for feature selection, especially for large-scale feature selection. Xue et al. [21] proposed a multi-objective evolutionary algorithm with a probability stack for neural architecture search, which considered the two objects of precision and time consumption. Hu et al. [22] proposed a fuzzy multiobjective feature selection method with particle swarm optimization for a feature selection problem with fuzzy cost. Banoth et al. [23] proposed a new energy-aware algorithm for the coverage and connectivity of the sensor nodes to maximize the number of cover sets and energy-aware connectivity. Kumar et al. [24] presented various machine learning algorithms for WSNs with their advantages, drawbacks, and parameters effecting the network lifetime. Chaturvedi et al. [25] provided the classification of coverage approaches especially related to that of target coverage. However, the efficiency of these algorithms is low, and they cannot be applied in complex environments.

1.2. Contributions

Recent research on coverage optimization in WSN has achieved some results, but there are still some issues. In recent years, most researchers have focused on studying specific scenarios without considering the optimization performance of algorithms in multiple scenarios. This makes it difficult to apply them effectively in various complex environments. Although the WOA algorithm has successfully addressed some problems in the field of wireless sensor networks, it still struggles to effectively handle multi-objective optimization problems and is prone to getting trapped in local optima. To effectively enhance WSN coverage optimization, the genetic algorithm and reinforced whale optimization algorithm (GARWOA) is proposed, which can achieve high coverage ratio in various coverage environments.

The main contributions can be outlined as follows:

1) We designed an reinforced whale optimization algorithm (RWOA). First, sine map and piecewise linear chaotic map (SPM) initializes the population, ensuring a more even distribution in the search space. Next, we implemented a non-linear improvement on the linear parameter "a" in WOA to enhance the efficiency of algorithm exploration and development. Finally, we incorporated a Levy flight mechanism to mitigate the algorithm's tendency to fall into local optima and premature convergence phenomena.

2) We applied the GARWOA algorithm for WSN coverage optimization to enhance node coverage ratio.

3) We compared the performance of GARWOA and five other algorithms on standard functions to validate its optimization capabilities.

4) We compared the network coverage results of GARWOA and five other algorithms in different environments to assess their effectiveness in improving network coverage.

1.3. Paper organization

This paper is structured as follows: Section 2 presents the WSN coverage and energy consumption model. Section 3 outlines the related concepts used in the GARWOA algorithm, while Section 4 details the steps taken in the optimization of WSN coverage using the GARWOA algorithm. Furthermore, Section 5 includes experimental simulations that confirm the algorithm's effectiveness. Finally, Section 6 summarizes the entire paper and provides an outlook for potential future research.

2. System model

This section will describe the coverage model and energy consumption model adopted by WSN.

2.1. WSN coverage model

The node coverage model is an important issue in WSN, involving how to select the positions and quantity of nodes to achieve the maximum coverage area under a certain number of nodes and complete coverage of the target area. The WSN node coverage optimization problem refers to the scenario where each sensor can only sense within a fixed sensing radius at their expected deployment positions. When deploying sensors in the required monitoring area, each sensor can only perform sensing and exploration within its sensing radius [26].

Therefore, the deployment of each node must adhere to the restricted sensing radius to ensure communication between each other and with the entire network. The optimization objective of this problem is to find the coverage problem of its internal objects within the possible range.

The WSN monitoring model is illustrated in Figure 1. Assuming the WSN is set up in a two-dimensional monitoring area of $S = H\times W(m^2)$, N sensor nodes are randomly initialized. Let the set of nodes be denoted as M, represented as $M=\{M_i, i=1,2,\cdots,N\}$. The coordinates of M_i are $\{(M_i(x), M_i(y)), i = 1,2,\cdots,N\}$. The sensing range of sensor nodes is a circle with a sensing radius of R_p.

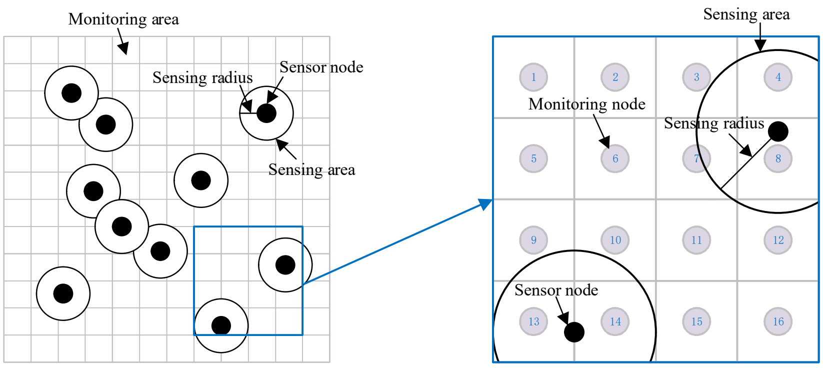


Figure 1. WSN monitoring model.

The two-dimensional WSN monitoring model is based on the following assumptions [27]:

1) The sensing radius and communication radius of each sensor node are R_p and R_c, respectively, satisfying $R_c \geqslant 2R_p$.

2) Each sensor node possesses the same parameters and communication capabilities.

3) The positions and deployment of nodes remain unchanged over a certain period, meaning that node mobility or failure are not considered.

To facilitate calculations, the rectangular area of the deployed network is divided into $H\times W$ grids of equal area. The nodes to be monitored, denoted as $G=\{G_j, j=1,2,\cdots,L= H\times W\}$, are located at the center of each grid. The coordinates of G_j are $\{(G_j(x), G_j(y)), j=1,2,\cdots,L\}$. The Euclidean distance between the sensor node M_i and the target monitoring node G_j is defined as:

$$d\left(M_i,G_j\right)=\sqrt{\left(M_i(x)-G_j(x)\right)^2+\left(M_i(y)-G_j(y)\right)^2} \tag{1}$$

When the distance between the target monitoring node G_j and any sensor node is less than or equal to the sensing radius R_p, it indicates that the monitoring node is sensed by the sensor. Therefore, the sensing probability p is as follows:

$$p(M_i,G_j)=\begin{cases}1, & d(M_i,G_j)\le R_p\\ 0, & d(M_i,G_j)> R_p\end{cases} \tag{2}$$

In general, multiple sensor nodes can simultaneously sense a target monitoring node. The joint probability P that the target monitoring node is sensed by all deployed sensor nodes is introduced, which can increase the probability of the target monitoring node being sensed. The joint probability P is defined as follows:

$$P(M_i,G_j)=1-\prod_{i=1}^{N}[1-p(M_i,G_j)] \tag{3}$$

The network coverage ratio can be defined as the ratio of the sum of monitored target monitoring nodes in the monitoring area to the sum of deployed target monitoring nodes in the monitoring area, defined as follows:

$$Cov = \frac{\sum_{j=1}^{L} P(M, G_j)}{H \times W} \tag{4}$$

2.2. Energy consumption model

The energy consumption model [28] for the wireless radio network used in this section is shown in Figure 2.

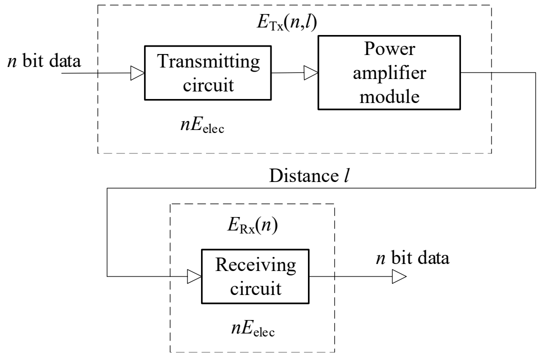


Figure 2. Radio network energy consumption model.

According to this energy consumption model, when the sender transmits n bits of data to the receiver located at a distance of l, the energy consumed by the sender is given by:

$$E_{\text{Tx}}(n,l) = \begin{cases} nE_{\text{elec}} + n\varepsilon_{\text{fs}} l^2, & l < l_0 \\ nE_{\text{elec}} + n\varepsilon_{\text{mp}} l^4, & l \geq l_0 \end{cases} \tag{5}$$

$$l_0 = \sqrt{\frac{\varepsilon_{\text{fs}}}{\varepsilon_{\text{mp}}}} \tag{6}$$

where E_{elec} is the unit energy consumption for receiving or transmitting data, ε_{fs} and ε_{mp} are the power amplification loss factors, and l_0 is the threshold distance. When the transmission distance is less than l_0, it follows the free space model; otherwise, it follows the multi-path fading model.

The energy consumed by the receiver to receive an n-bit data packet is:

$$E_{\text{Rx}}(n) = nE_{\text{elec}} \tag{7}$$

If each regular node sends an S-bit data round to the cluster head, the total energy consumed in the round in the network is:

$$E_{\text{Round}} = S\left(2NE_{\text{elec}} + NE_{\text{DA}} + k\varepsilon_{\text{mp}} l_{\text{CB}}^4 + k\varepsilon_{\text{fs}} l_{\text{CH}}^2\right) \tag{8}$$

where k is the number of clusters, l_{CB} is the average distance between all cluster heads and the base station, and l_{CH} is the average distance between cluster members and cluster heads, as shown in the following equation:

$$l_{\text{CB}} = 0.765\frac{M}{2}, \quad l_{\text{CH}} = \frac{M}{\sqrt{2\pi k}} \tag{9}$$

3. Design of GARWOA

This section will introduce the concepts of GA and WOA and then combine these two algorithms to design the GARWOA optimization algorithm for WSN coverage optimization.

3.1. GA

GA is an algorithm that simulates natural genetic operations in the biological world. It conducts a random search in the population by simulating natural selection and genetic mechanisms. It iteratively repeats key operations such as individual encoding, selection, crossover, mutation, and fitness function calculation, continuously updating and improving individuals, and gradually approaching the optimal solution as shown in Figure 3. By randomly searching the entire solution space, GA can find relatively optimal solutions to complex problems and has a certain global search ability.

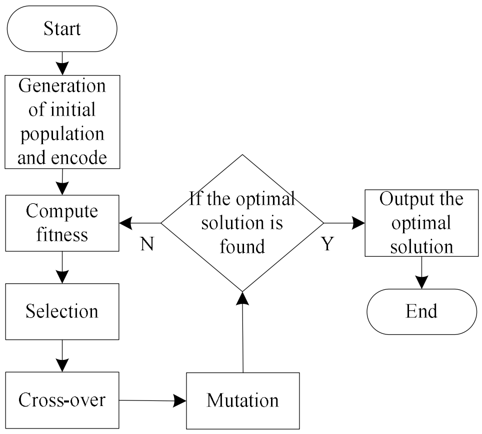


Figure 3. Genetic algorithm flow chart.

3.1.1. Selective operation

Selection is an important step in GA, determining whether a specific individual participates in the reproduction process. The probability of an individual being selected for reproduction depends on its fitness. Individuals with higher fitness are more likely to be selected, while those with lower fitness may be eliminated. In this paper, the roulette wheel selection method is adopted, which allocates all possible strings to a wheel in proportion. Based on their fitness values, a portion of the wheel is allocated to them. The wheel is then rotated to select specific solutions to participate in the formation of the next generation. According to the roulette wheel selection method, the probability of an individual i being selected is:

$$p_i = \frac{F_i}{\sum_{j=1}^{N} F_j} \tag{10}$$

where F_i is the fitness value of individual i, and N is the number of individuals in the population.

3.1.2. Crossover operation

A crossover operation generates new individuals based on a certain crossover probability and method. Specifically, two individuals are randomly selected from the population, and by exchanging parts of the chromosomes, the excellent features of the parent individual are passed on to the offspring individual, thereby producing an individual with new advantages. In this paper, real number encoding is used, so the crossover operation adopts the real number crossover method. The crossover operation for the k-th chromosome a_k and the l-th chromosome a_l at the j-th position is as follows:

$$\begin{cases} a_{kj} = a_{kj}(1-b) + a_{lj}b \\ a_{lj} = a_{lj}(1-b) + a_{kj}b \end{cases} \tag{11}$$

where b is a random number in the range [0, 1], a_{kj} and a_{lj} represent the genes of the k-th chromosome and the l-th chromosome at the j-th position.

3.1.3. Variational operation

A mutation operation randomly selects an individual from the population and mutates at a selected point on that individual to generate an individual with better characteristics. Such mutation operations help introduce new genetic variations, allowing the population to better explore the solution space, thereby increasing the potential searchability of the evolutionary algorithm. The mutation operation for the j-th gene a_{lj} of the i-th individual is as follows:

$$a_{ij} = \begin{cases} a_{ij} + (a_{ij} - a_{\max}) * f(g), r \geq 0.5 \\ a_{ij} + (a_{\min} - a_{ij}) * f(g), r < 0.5 \end{cases} \tag{12}$$

$$f(g) = r_2 (1 - g / G_{\max})^2 \tag{13}$$

where $a_{\max}$ and $a_{\min}$ represent the upper and lower bounds of the gene a_{lj}, r_2 is a random number, g is the current iteration number, $G_{\max}$ is the maximum number of evolutions, and r is a random number in the range [0,1].

3.2. RWOA

This section will introduce RWOA. Firstly, the SPM chaotic mapping [29] is utilized to initialize the population, ensuring a more uniform distribution in the search space. Next, a non-linear improvement is applied to the linear parameter ‘a’ in WOA to enhance the algorithm’s efficiency in exploration and development. Finally, a Levy flight mechanism is incorporated to address the issues of the algorithm getting stuck in local optima and premature convergence.

3.2.1. Standard WOA

The WOA algorithm is an efficient algorithm designed based on the behavioral characteristics of whales when hunting prey. This algorithm leverages the diversity and global optimization capability of whale populations and exhibits good local search capabilities. During the iterative process of the algorithm, whales continuously adjust their positions and states based on their own positions and the information of the surrounding environment to search for better solutions. Therefore, each position of a whale corresponds to a feasible solution, and the goal of the algorithm is to find the optimal one among these solutions. When hunting, each whale adopts different strategies. Some whales may encircle the prey, while others may approach other whales. Another method involves using bubble nets, a cooperative hunting technique. Whales will exhale a circle of bubbles in the water to surround the prey in the center and then swim towards the center together to gather the prey, making it easier for the whales to hunt. In each hunting action, whales randomly choose a method to prey on. When a whale besieges its prey, it will select the target based on its own condition and the surrounding environment, but this selection is not entirely random.

WOA updates individual positions based on the position update strategy of whale bubble attack behavior and finally obtains the current global optimal solution position based on prey search behavior as follows:

$$c_i^{t+1} = \begin{cases} s_i^{t+1}, p >= 0.5 \\ \left|C_{\text{best}}^t - c_i^t\right| * e^{bl} * \cos(2\pi l) + C_{\text{best}}^t, p < 0.5 \end{cases} \tag{14}$$

The hunting behavior of whales is determined based on the probability factor p. When $p < 0.5$, whales adopt a strategy of updating their positions in a spiral manner; when $p \geqslant 0.5$, whales use a surrounding prey strategy to update their positions. In Eq (14), c_i^{t+1} represents the spatial position of whale i in the current iteration round t+1, b is a constant, p is a random number in the range of [0, 1], l is a random number uniformly distributed in the range of [−1, 1], C_{best}^t represents the position of the best whale individual in the current iteration round n, and s_i^{t+1} represents the prey searching behavior as follows:

$$s_i^{t+1} = \begin{cases} C_{\text{rand}}^t - A\left|C * C_{\text{rand}}^t - c_i^t\right|, |A| >= 1 \\ C_{\text{best}}^t - A\left|C * C_{\text{best}}^t - c_i^t\right|, |A| < 1 \end{cases} \tag{15}$$

In prey searching behavior, when the coefficient $|A|<1$, the whale moves towards any other whale in the population; when the coefficient $|A| \geqslant 1$, the whale chooses to move towards the position of the best whale seen so far. In Eq (15), C_{rand}^t represents the position of any whale individual in the current iteration round t, C is a random number uniformly distributed in the range of [0, 2], and A is a random number uniformly distributed in the range of [−a, a]. The definitions of A, C, and a are as follows:

$$A = 2ar_1 - a \tag{16}$$

$$C = 2r_2 \tag{17}$$

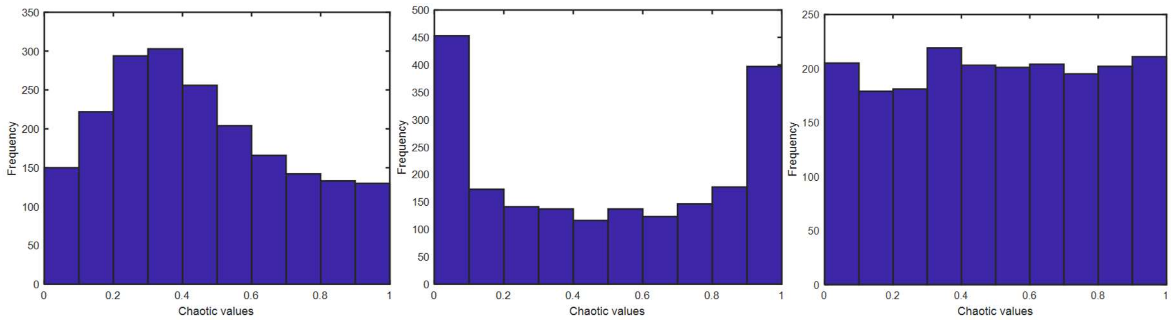
$$a = 2 - \frac{2t}{\text{maxIterate}} \tag{18}$$

Here, r_1 and r_2 are random numbers within the range of [0, 1], maxIterate represents the maximum number of iterations, and a represents a control parameter that linearly decreases from 2 to 0 with the increase of iteration round t.

3.2.2. SPM chaotic mapping

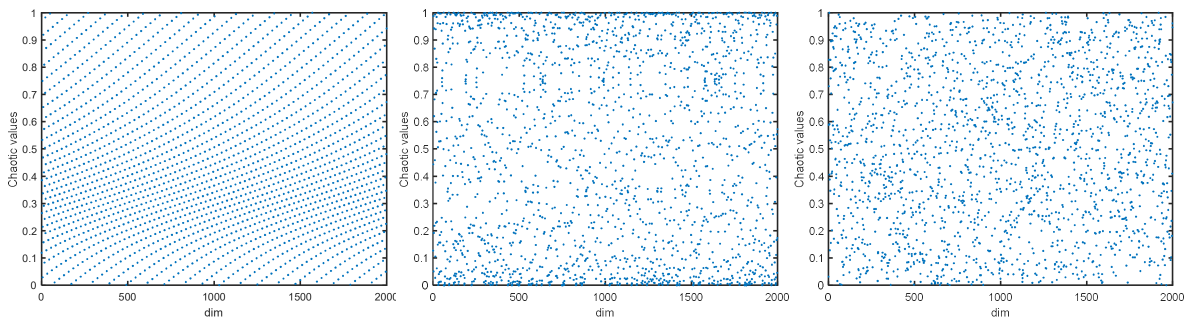
The initialization of the population has a significant impact on the efficiency of most current intelligent optimization algorithms. A uniformly distributed population can moderately expand the search range to improve convergence speed and solution accuracy. The unpredictability and aperiodicity of chaos can be used to enhance algorithmic performance. The original WOA algorithm initializes the population with the rand function, resulting in uneven population distribution and overlapping individuals, leading to a rapid decrease in population diversity in later iterations. Chaos is a unique and widely present form of non-cyclic motion in nonlinear systems. Due to its traversing and random characteristics, it is widely used in population-based intelligent algorithms to optimize population diversity. In this paper, the SPM chaotic mapping model is introduced, which has excellent chaotic and traversing properties. The expression is as follows:

$$x(t+1)=\begin{cases}\mathrm{mod}\left(\dfrac{x(t)}{\eta}+\mu\sin(\pi x(t))+r,1\right) & 0\le x(t)<\eta\\ \mathrm{mod}\left(\dfrac{x(t)/n}{0.5-\eta}+\mu\sin(\pi x(t))+r,1\right) & 0\le x(t)<0.5\\ \mathrm{mod}\left(\dfrac{1-x(t)/n}{0.5-\eta}+\mu\sin(\pi(1-x(t))+r,1\right) & 0.5\le x(t)<1-\eta\\ \mathrm{mod}\left(\dfrac{1-x(t)}{\eta}+\mu\sin(\pi(1-x(t)))+r,1\right) & 1-\eta\le x(t)<1\end{cases} \tag{19}$$



(a) Circle mapping (b) Logistic mapping (c) SPM mapping

Figure 4. Chaotic mapping histogram.



(a) Circle mapping (b) Logistic mapping (c) SPM mapping

Figure 5. Chaotic mapping scatterplot.

This section employs commonly used logistic and circle mappings. Under the condition of the same initial value and 2000 iterations, these mappings are compared with the SPM mapping. Figure 4 displays histograms of three chaotic mappings, where the horizontal axis represents the chaotic values and the vertical axis represents the frequency of each chaotic value. The results demonstrate that the SPM mapping exhibits superior chaotic performance and traversing capability. Therefore, the SPM mapping is selected to enhance population diversity and achieve a more uniform distribution. Figure 5 illustrates the population distributions when different chaotic mappings are applied to the appearance of the population.

3.2.3. Non-linear improvement

Since the control parameter *a* in the WOA algorithm does not undergo nonlinear changes, it is difficult to reflect the actual optimization process, and it is prone to getting stuck in local optima. Therefore, this paper proposes a nonlinear factor updating formula as follows:

$$a = 1 + \sin\left(\frac{\pi}{2} + \pi \times \frac{t}{\text{maxIterate}}\right) \tag{20}$$

where *t* represents the current iteration. This nonlinear factor is compared with the original *a* function in standard WOA under the condition of setting the maximum iteration times to 100. As shown in Figure 6, in the first 50 rounds, the value of *a* used for exploration in this function is larger, indicating a broader exploration range compared to the original function. In the latter 50 rounds, the value of *a* used for exploitation in this function is smaller, allowing the algorithm to converge more quickly [30].

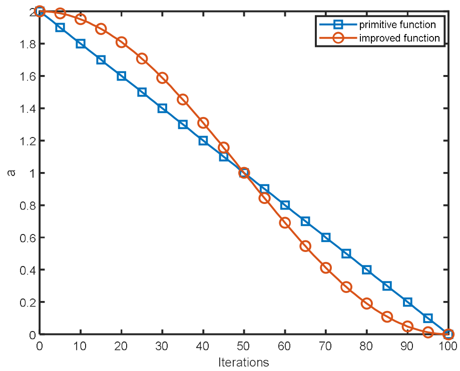


Figure 6. Comparison plot of improved parameter *a* with original WOA parameter *a*.

3.2.4. Levy flight perturbation strategy

Based on the study of the foraging trajectories of birds and insects by biologists, it has been found that the probability of straight-line segments appearing in the flight paths of some organisms is similar to the basic characteristics of a Levy distribution, indicating a Levy flight behavior. Levy flight belongs to a class of random walk strategies where the step lengths follow a heavy-tailed stable distribution, resulting in alternating movements between relatively short and relatively long distances. Therefore, incorporating a Levy flight search mechanism in bio-inspired swarm intelligence optimization algorithms can effectively increase the exploration range of the search space, enhance the diversity of population search, and make it possible for the search algorithm to escape local optima [31]. In the WOA algorithm, the introduction of the Levy flight mechanism disrupts the position update process during the optimization process, improving the algorithm' s tendency to fall into local optima and premature convergence. Through the Levy flight, the new position update formula is as follows:

$$x(t+1) = x(t) + \alpha \oplus Levy \tag{21}$$

where α is the step size control factor set to 0.01. The calculation formula for the random step length, *Levy*, is as follows:

$$Levy = \frac{\alpha_1}{|\alpha_2|^{\frac{1}{\beta}}} \tag{22}$$

where α_1 and α_2 are both $1 \times d$ random matrices following a normal distribution, with $\alpha_1 \sim (0,\sigma^2)$ and $\alpha_1 \sim (0,1)$. β is a constant, set to 1.5, and σ is defined as follows:

$$\sigma = \left[\frac{\Gamma(1+\beta) \cdot \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \cdot \beta \cdot 2^{(\frac{\beta-1}{2})}} \right]^{\frac{1}{\beta}} \tag{23}$$

3.3. GARWOA

GA tends to converge to local optima in the final iterations, making it unable to perform proper local searches. The local search capability of the RWOA compensates for the shortcomings of GA. Therefore, we design the GARWOA, that combines the global search ability of GA with the local search ability of RWOA, to enhance the performance of the optimization process in finding approximate global solutions.

GARWOA consists of two stages. In the first stage, GA performs initial optimization by searching the design space for a limited number of iterations, and the optimal solution found in this stage is denoted as C_{best}. In the second stage, RWOA is used to conduct a more refined search on C_{best} through a limited number of iterations.

GA constructs an initial population randomly distributed across the entire space, while in RWOA, C_{best} is directly transformed into the optimal initial population.

Algorithm 1 represents the pseudocode of the GARWOA algorithm, described as follows:

Figure **7** represents the GARWOA flow chart.

Algorithm 1: The proposed GARWOA algorithm

Input: Population size N, initialization coefficients a, r, $G_{\max}$ and η, etc., maximum number of iterations Max, maxIterate

Output: The optimal solution C_{best}

1. The objective function $f(x)$
2. Initialize the population and produce an initial population of N chromosomes $Y_i (i = 1, 2, ..., N)$
3. Calculate the degree of adaptation $f(x)$
4. **while** (t < Max) **do**
5. Selection of a pair of chromosomes according to the fitness according to Eq (10)
6. Crossover operation on selected pairs according to Eq (11) crossover probability
7. According to Eq (12) the mutation is applied to the offspring with mutation probability
8. Updating populations and recalculating fitness
9. **end while**
10. The objective function $f(x) = C_{best}$
11. Initialize the population C_0 according to Eq (19)
12. **while** (t < maxIterate) **do**
13. **for** $i \in [1, N]$
14. Update the parameters a, A, C, l, r_1, r_2 and p
15. **if** ($p \geq 0.5$) **then**
16. **if** ($|A|<1$) **then**
17. Update whale position by moving toward the current best position according to Eq. (14)
18. **else** ($|A| \geq 1$)
19. Updating whale positions by random movement according to Eq (14)
20. **end if**
21. **else if** (p < 0.5)
22. Spiral update of whale positions according to Eq (13)
23. **end if**
24. **end for**
25. Levy flight perturbation to update whale position according to Eq (21)
26. Finding the optimal solution G_{best}
27. **end while**
28. **return** The optimal solution G_{best}

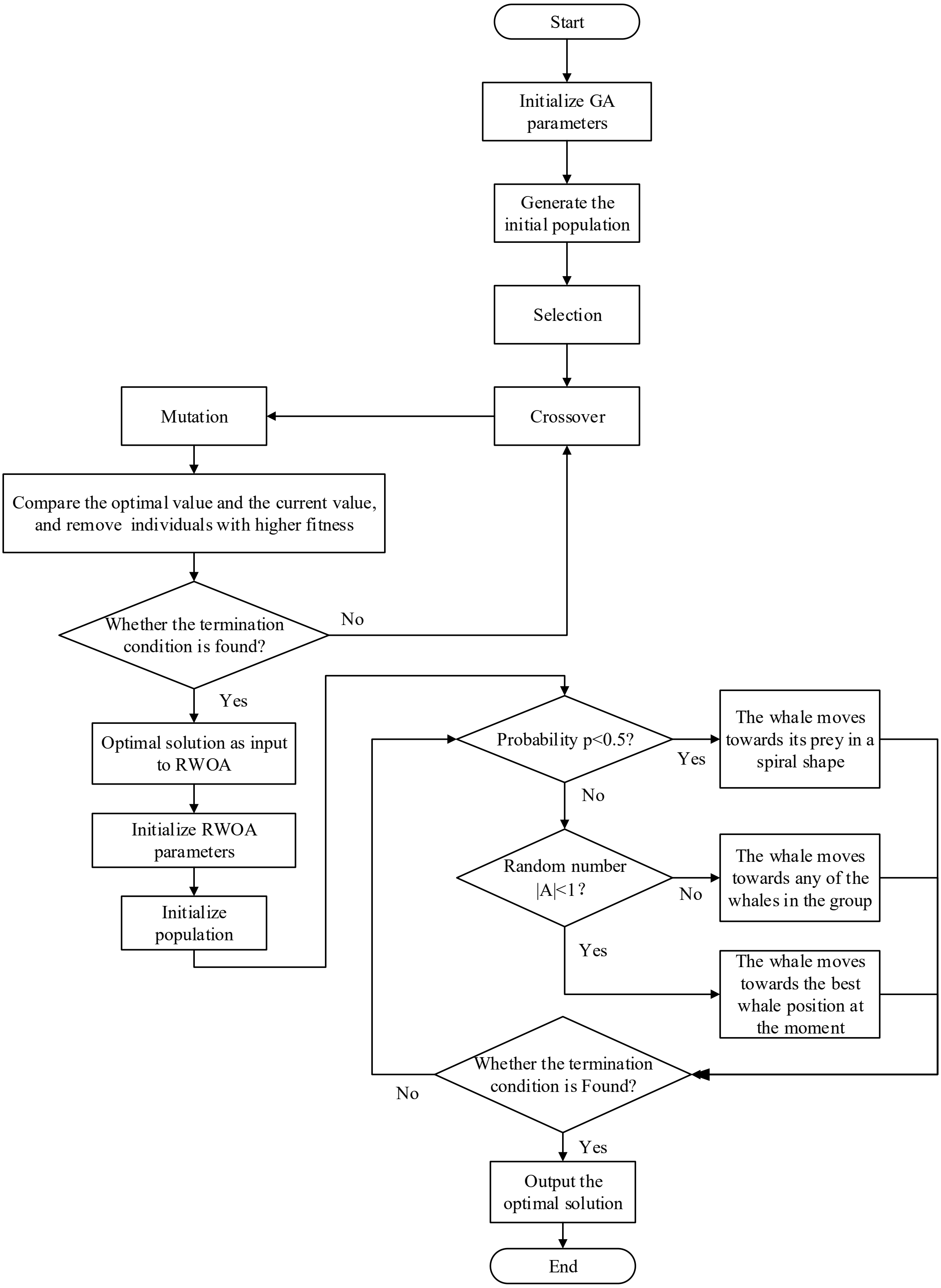


Figure 7. GARWOA flow chart.

3.4. GARWOA time complexity analysis

In this section, analyze the time complexity of GARWOA algorithm. Assuming the maximum number of iterations of the algorithm is T, the population size is N, and the problem dimension is D. The time complexity of GA algorithm is O($N*logN*D*T$), while the time complexity of WOA algorithm is O($N*D*T$). Therefore, the time complexity analysis of the algorithm in this article is as follows:

1) The initialization time complexity of the GA population is O($N*D$), and the time complexity of calculating the fitness of various populations is O(N).

2) The time complexity of the GA selection operation is O($N*logN*D*T/2$), while the time complexity of the crossover and mutation operations is O($N*D*T/2$).

3) The initialization time complexity of the RWOA population is O($N*D$).

4) The time complexity of RWOA position update strategy is O($N*D*T/2$), and the time complexity of Levy flight strategy is O ($D*T/2$).

Therefore, the time complexity of GARWOA is O($N*logN*D*T$), which is the same as GA and adds a certain amount of computational complexity compared to WOA algorithm. But the coverage ratio of this algorithm is better than other algorithms.

4. WSN optimized coverage based on GARWOA

Based on GARWOA, the objective of optimizing coverage in Wireless Sensor Networks (WSNs) is to cover a larger area using the same number of sensors while maintaining effective communication. The population consists of the coordinates of sensor nodes. Initially, GA is used for preliminary optimization to obtain the optimal solution, C_{best}. C_{best} is then utilized as the optimal initial population for RWOA, which conducts a more refined search to obtain a more precise optimal solution, G_{best}. G_{best} represents the deployment positions of sensor nodes that maximize coverage.

The following are the steps for WSN coverage optimization based on GARWOA:

Step1 Input the size of the WSN detection area, the number of sensors, the sensing radius, and the communication radius. Initialize various parameters in the GA algorithm.

Step2 Generate the initial population, where each individual represents a coverage scheme. In this step, sensors are dispersed around the detection area.

Step3 Evaluate each individual in the population. Calculate fitness based on Eq (4) and select the best individual.

Step4 Apply crossover operations to the current individual based on Eq (11) to obtain offspring individuals.

Step5 Apply mutation operations to the current individual based on Eq (12) to obtain offspring individuals.

Step6 Compare the optimal value of the parent population with the current individual. Iterate with the higher-fitness individual.

Step7 Determine if the termination condition is met. If not, loop back to **Step4**. If yes, use the optimal solution, C_{best}, as input for RWOA.

Step8 Initialize various parameters in the RWOA algorithm and initialize the population based on Eq (19).

Step9 Decide the hunting behavior of whales based on whether the probability, p, is less than 0.5.

Step10 If the probability, p, is greater than or equal to 0.5, whales perform prey encircling. If the absolute value of random number A is less than 1, the whale selects the best position among the whales encountered so far to move towards. If the absolute value of random number A is greater than 1, the whale moves towards any other whale in the population.

Step11 If the probability, p, is less than 0.5, whales move in a spiral shape towards the prey.

Step12 If the termination condition is met, the algorithm stops and outputs the current optimal solution, G_{best}. Otherwise, loop back to **Step9**.

Step13 Output the coverage scheme with the highest coverage ratio for sensor nodes.

5. WSN optimized coverage based on GARWOA

This section employs MATLAB 2021b as the simulation software. Firstly, by comparing the performance of five algorithms (MuGWO, WOA, IWHO, PSO, and GARWOA) on ten standard test functions, the convergence of GARWOA is verified. On this basis, the effectiveness of GARWOA in WSN node deployment is validated through a comparison of network coverage under three experimental conditions against MuGWO, WOA, IWHO, and PSO algorithms.

5.1. Experiments of standard test functions

The standard functions used for testing are listed in Table 1. Here, $f_1(x) \sim f_6(x)$ represent unimodal functions, which can be used to examine the convergence speed and accuracy of the optimization algorithm, whereas $f_7(x) \sim f_{10}(x)$ represent multimodal functions, which can be used to assess the global search capability of the optimization algorithm. The dimensionality of all test functions is set to 30, and their optimal values are all 0.

5.1.1. Settings of experimental parameter

The population size for all algorithms is set to 100, with a total of 500 iterations, and the shared parameters remain consistent. The parameter settings for each algorithm are shown in Table 2.

5.1.2. Result of standard test functions

To avoid result bias caused by experimental contingencies, each of the five algorithms was independently run 50 times on each test function, and the experimental results were averaged. This evaluation method effectively reduces the influence of random factors, enhancing the reliability and repeatability of the experimental results. The test results are shown in Table 3. From the results, it can be observed that for functions $f_1(x)$, $f_2(x)$, $f_7(x)$, $f_9(x)$, and $f_{10}(x)$, GARWOA demonstrates the best optimization performance. For functions $f_3(x)$, $f_4(x)$, and $f_6(x)$, IWHO performs the best. For function $f_5(x)$, the average of GARWOA outperforms other algorithms.

Table 1. Standard functions.

Function name	Math expression	Range	Dim	Optimum value
Sphere Model	$f_1(x)=\sum_{i=1}^{n} x_i^2$	$[-100,100]$	30	0
Schwefel's problem 2.22	$f_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	$[-10,10]$	30	0
Schwefel's problem 1.2	$f_3(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$	$[-100,100]$	30	0
Schwefel's problem 2.21	$f_4(x)=\max_i\left\{\lvert x_i\rvert,1\leqslant i\leqslant n\right\}$	$[-100,100]$	30	0
Generalized Rosenbrock's Function	$f_5(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	$[-30,30]$	30	0
Step function	$f_6(x)=\sum_{i=1}^{n}\left(\left[x_i+0.5\right]\right)^2$	$[-100,100]$	30	0
Quartic Function	$f_7(x)=\sum_{i=1}^{n} i x_i^4+\text{random}\left[0,1\right)$	$[-1.28,1.28]$	30	0
Generalized Rastrigin's Function	$f_8(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos\left(2\pi x_i\right)+10\right]$	$[-5.12,5.12]$	30	0
Ackley's Function	$f_9(x)=-20\exp\left[-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right]-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)\right)+20+e$	$[-32,32]$	30	0
Generalized Penalized Function	$f_{10}(x)=\frac{\pi}{n}\left\{10\sin\left(\pi y_1\right)+\sum_{i=1}^{n-1}\left(y_i-1\right)^2\left[1+10\sin^2\left(\pi y_{i+1}\right)\right]+\left(y_n-1\right)^2\right\}+\sum_{i=1}^{n}\left(x_i,10,100,4\right)$ $y_i=1+\frac{x_i+1}{4}$ $u\left(x_i,a,k,m\right)=\begin{cases} k\left(x_i-a\right)^m, x_i>a \\ 0,-a<x_i<a \\ k\left(-x_i-a\right)^m, x_i<a \end{cases}$	$[-50,50]$	30	0

Table 2. Experimental parameter settings for each algorithm.

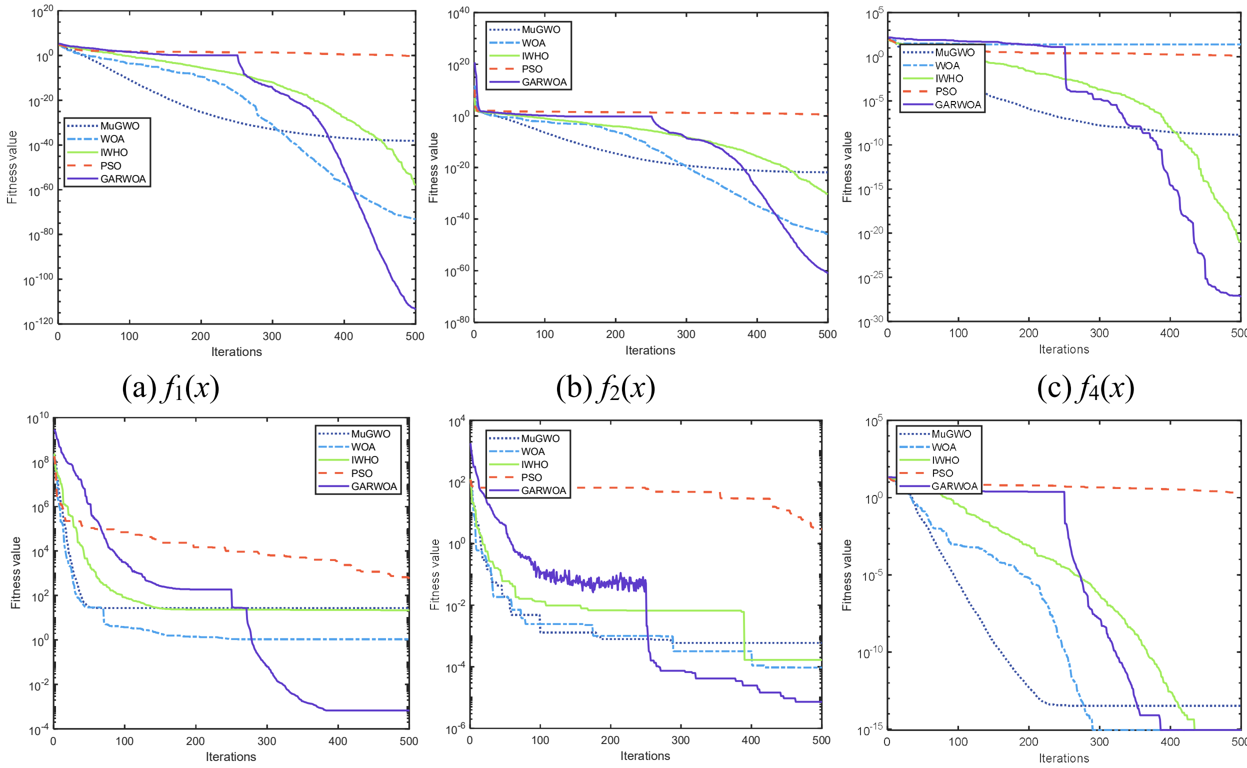
Algorithms	Parameters
MuGWO	mutationDimensions = 2 , mutationCount = 30 , freedomRate = 0.3
WOA	a decreases linearly from 2 to 0
IWHO	PS = 0.1 , PC = 0.13 , $a = \pi$, $b = \pi$, $\tau = (\sqrt{5} - 1) / 2$
PSO	$c1 = c2 = 2$, $\omega_{\max} = 0.9$, $\omega_{\min} = 0.6$, $V_{\max} = 6$
GARWOA	pc = 0.999 , pm = 0.001 , $b = 1$, $\lambda = 0.2$, $\tau = 0.5$

Table 3. Test function experimental results.

Function	Criteria	IWHO	WOA	MuGWO	PSO	GARWOA
$f_1(x)$	Mean	6.37×10^{-49}	1.30×10^{-71}	1.53×10^{-37}	1.09	$\mathbf{1.67 \times 10^{-111}}$
	Std	3.90×10^{-48}	8.22×10^{-71}	2.09×10^{-37}	0.549	$\mathbf{1.02 \times 10^{-110}}$
	Mid	4.99×10^{-53}	2.58×10^{-76}	1.07×10^{-37}	0.940	$\mathbf{1.35 \times 10^{-114}}$
	Max	1.46×10^{-47}	2.85×10^{-71}	1.44×10^{-36}	2.54	$\mathbf{3.26 \times 10^{-110}}$
	Min	8.33×10^{-58}	3.42×10^{-81}	2.27×10^{-39}	0.210	$\mathbf{9.34 \times 10^{-119}}$
$f_2(x)$	Mean	1.95×10^{-28}	7.57×10^{-42}	5.71×10^{-22}	2.67	$\mathbf{2.20 \times 10^{-60}}$
	Std	5.91×10^{-28}	2.40×10^{-42}	4.37×10^{-22}	0.688	$\mathbf{5.45 \times 10^{-60}}$
	Mid	1.77×10^{-29}	3.92×10^{-43}	5.20×10^{-22}	2.53	$\mathbf{2.84 \times 10^{-61}}$
	Max	2.55×10^{-27}	1.48×10^{-40}	2.35×10^{-21}	4.16	$\mathbf{1.62 \times 10^{-58}}$
	Min	1.30×10^{-32}	3.83×10^{-46}	1.40×10^{-22}	1.10	$\mathbf{4.60 \times 10^{-64}}$
$f_3(x)$	Mean	$\mathbf{3.27 \times 10^{-25}}$	1.76×10^{4}	8.06×10^{-11}	82.0	3.51×10^{-4}
	Std	$\mathbf{1.41 \times 10^{-24}}$	8.60×10^{3}	2.02×10^{-10}	27.0	7.55×10^{-4}
	Mid	$\mathbf{3.00 \times 10^{-29}}$	1.77×10^{4}	1.91×10^{-11}	81.4	1.88×10^{-5}
	Max	$\mathbf{2.26 \times 10^{-24}}$	4.03×10^{4}	1.22×10^{-9}	1.43×10^{2}	3.00×10^{-3}
	Min	$\mathbf{5.61 \times 10^{-34}}$	4.68×10^{3}	1.66×10^{-14}	27.4	6.33×10^{-8}
$f_4(x)$	Mean	$\mathbf{1.47 \times 10^{-18}}$	7.72	3.07×10^{-9}	1.54	2.63×10^{-9}
	Std	$\mathbf{3.68 \times 10^{-18}}$	17.3	4.48×10^{-9}	0.229	1.86×10^{-8}
	Mid	$\mathbf{1.57 \times 10^{-19}}$	1.00×10^{-2}	1.36×10^{-9}	1.50	3.96×10^{-9}
	Max	$\mathbf{1.74 \times 10^{-16}}$	81.5	1.10×10^{-8}	1.95	2.38×10^{-8}
	Min	$\mathbf{5.04 \times 10^{-22}}$	1.01×10^{-11}	1.41×10^{-10}	1.11	4.83×10^{-10}
$f_5(x)$	Mean	24.6	22.7	26.5	5.67×10^{2}	**5.82**
	Std	1.20	10.2	**0.844**	5.39×10^{2}	9.43
	Mid	24.6	27.6	26.2	3.98×10^{2}	$\mathbf{5.96} \times 10^{-4}$
	Max	96.3	27.8	28.5	9.91×10^{2}	**22.1**
	Min	23.1	7.00×10^{-2}	25.2	1.42×10^{2}	$\mathbf{9.07} \times 10^{-5}$
$f_6(x)$	Mean	$\mathbf{1.31} \times 10^{-11}$	1.74×10^{-2}	0.326	1.07	2.61×10^{-7}
	Std	$\mathbf{3.18} \times 10^{-11}$	6.98×10^{-3}	0.255	0.440	9.52×10^{-8}
	Mid	$\mathbf{4.12} \times 10^{-12}$	2.00×10^{-2}	0.250	0.890	2.26×10^{-7}
	Max	$\mathbf{4.27} \times 10^{-10}$	3.00×10^{-2}	1.01	2.72	7.93×10^{-7}
	Min	$\mathbf{1.66} \times 10^{-13}$	1.00×10^{-2}	2.64×10^{-5}	0.320	8.86×10^{-8}

Continued on next page

Function	Criteria	IWHO	WOA	MuGWO	PSO	GARWOA
	Mean	4.57×10^{-4}	4.79×10^{-4}	7.49×10^{-4}	8.12	$\mathbf{6.46 \times 10^{-6}}$
	Std	2.84×10^{-4}	5.83×10^{-4}	4.19×10^{-4}	5.94	$\mathbf{8.89 \times 10^{-6}}$
$f_7(x)$	Mid	5.04×10^{-4}	3.50×10^{-4}	7.44×10^{-4}	5.58	$\mathbf{3.32 \times 10^{-6}}$
	Max	1.96×10^{-3}	3.47×10^{-3}	1.68×10^{-3}	38.8	$\mathbf{6.24 \times 10^{-5}}$
	Min	8.94×10^{-5}	4.38×10^{-5}	2.56×10^{-4}	1.09	$\mathbf{2.92 \times 10^{-7}}$
	Mean	-2.90×10^{2}	-2.90×10^{2}	-2.50×10^{2}	$\mathbf{-1.49 \times 10^{2}}$	-2.90×10^{2}
	Std	1.24×10^{-8}	**0.00**	27.8	27.8	2.93×10^{-14}
$f_8(x)$	Mid	-2.90×10^{2}	-2.90×10^{2}	-2.52×10^{2}	$\mathbf{-1.48 \times 10^{2}}$	-2.90×10^{2}
	Max	-2.90×10^{2}	-2.90×10^{2}	-1.89×10^{2}	**−99.1**	-2.90×10^{2}
	Min	$\mathbf{-2.90 \times 10^{2}}$	$\mathbf{-2.90 \times 10^{2}}$	$\mathbf{-2.90 \times 10^{2}}$	$\mathbf{-2.10 \times 10^{2}}$	$\mathbf{-2.90 \times 10^{2}}$
	Mean	1.17×10^{-15}	4.37×10^{-15}	3.50×10^{-14}	1.96	$\mathbf{1.87 \times 10^{-16}}$
	Std	9.74×10^{-16}	2.54×10^{-15}	4.54×10^{-15}	0.504	$\mathbf{2.42 \times 10^{-17}}$
$f_9(x)$	Mid	$\mathbf{8.88 \times 10^{-16}}$	4.44×10^{-15}	3.29×10^{-14}	1.83	4.44×10^{-15}
	Max	$\mathbf{4.44 \times 10^{-15}}$	7.99×10^{-15}	4.00×10^{-14}	2.64	7.99×10^{-15}
	Min	$\mathbf{8.88 \times 10^{-16}}$	$\mathbf{8.88 \times 10^{-16}}$	2.93×10^{-14}	0.810	$\mathbf{8.88 \times 10^{-16}}$
	Mean	8.29×10^{-3}	1.32×10^{-3}	2.99×10^{-2}	1.57×10^{-2}	$\mathbf{3.92} \times 10^{-8}$
	Std	4.61×10^{-2}	5.33×10^{-4}	1.62×10^{-2}	2.29×10^{-2}	$\mathbf{1.23} \times 10^{-8}$
$f_{10}(x)$	Mid	$\mathbf{3.93} \times 10^{-14}$	0.00	2.00×10^{-2}	1.00×10^{-2}	4.09×10^{-8}
	Max	0.100	0.00	3.52	0.130	$\mathbf{7.46} \times 10^{-8}$
	Min	$\mathbf{1.43} \times 10^{-15}$	5.58×10^{-4}	1.00×10^{-2}	0.00	1.03×10^{-8}



(a) $f_1(x)$ (b) $f_2(x)$ (c) $f_4(x)$

(d) $f_5(x)$ (e) $f_7(x)$ (f) $f_9(x)$

Figure 8. Comparison of partial convergence curves.

To further verify the convergence of GARWOA, it is necessary to compare the iterative convergence curves of the test functions for the five algorithms. Some of the convergence curves are shown in Figure 8. It can be observed that GARWOA exhibits superior convergence performance compared to other algorithms. The results indicate that GARWOA possesses good practicality, fast convergence, and excellent optimization capabilities.

5.2. Experiments of network coverage simulation

To validate the WSN coverage optimization performance of the GARWOA algorithm, it is compared with the WOA, PSO, IWHO, and MuGWO algorithms in three experimental scenarios. This includes analyzing coverage ratio, convergence curves, and network coverage graphs.

5.2.1. Settings of simulation parameter

In Experiment 1, the area is relatively small with a lower number of nodes, which can simulate densely populated urban areas or small-scale monitoring tasks. In Experiment 2, the area is larger with a higher number of nodes, representing scenarios such as urban broadcast communication, environmental monitoring, or monitoring of large farms in the agricultural sector. In Experiment 3, the area is even larger, but with a relatively lower number of nodes, which may involve applications like remote environmental monitoring or field ecology research. The parameter settings [32] for these three experimental scenarios can be found in Table 4.

Table 4. Network coverage experiment parameter settings.

Parameters	Experiment 1 Value	Experiment 2 Value	Experiment 3 Value
Area size S	50 m × 50 m	100 m × 100 m	200 m × 200 m
Number of nodes N	40	80	50
The perceptual radius R_p	5 m	7.5 m	20 m
The communication radius R_c	10 m	15 m	40 m

5.2.2. Coverage experiment results analysis

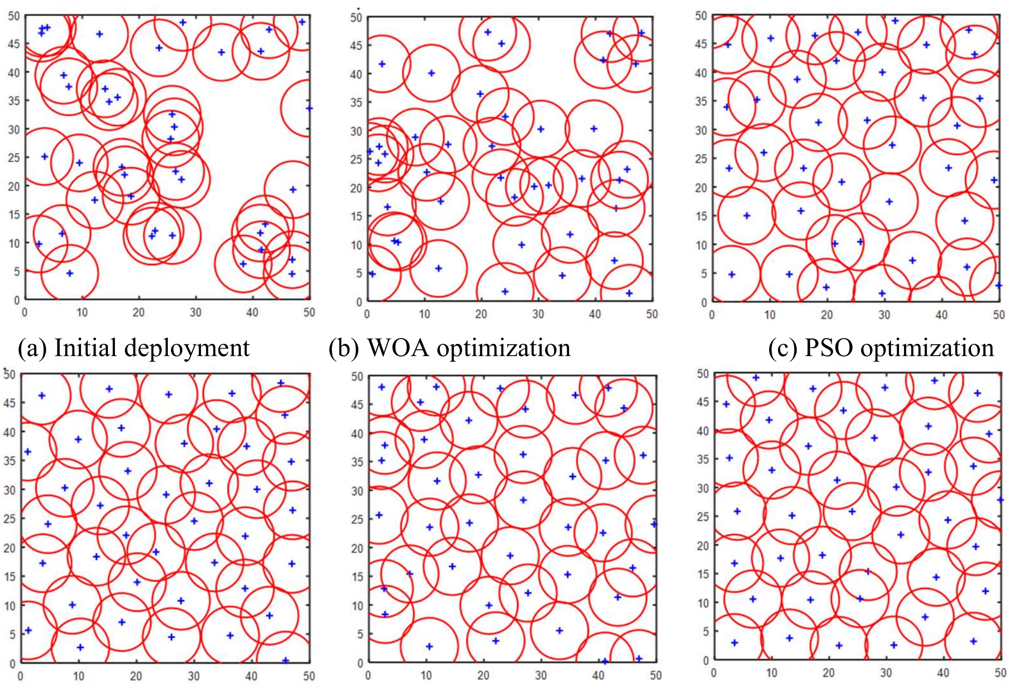
To validate the performance of GARWOA in optimizing coverage within WSN, three experiments were conducted by simulating different scenarios and adjusting various parameters. To demonstrate the effectiveness of the GARWOA algorithm, it was compared with four other algorithms. To avoid result bias caused by experimental contingencies, each of the five algorithms was independently run 30 times on each experiment, and the experimental results were averaged.

The objective function is ratio of coverage as shown in Eq (4). Table 5 shows the experimental results. It can be seen from Table 5 that GARWOA outperforms other algorithms in all three experiments in terms of indicators.

The comparison of coverage maps for the three experimental scenarios can be seen in Figure 9–11. By comparing the coverage maps of various algorithms, it can be observed that GARWOA performs better in all three experimental environments. It provides superior coverage in the monitoring area, with a more even distribution of nodes and fewer redundant nodes. It proves to be adaptable to different environments, demonstrating excellent applicability.

Table 5. Network coverage experiment results.

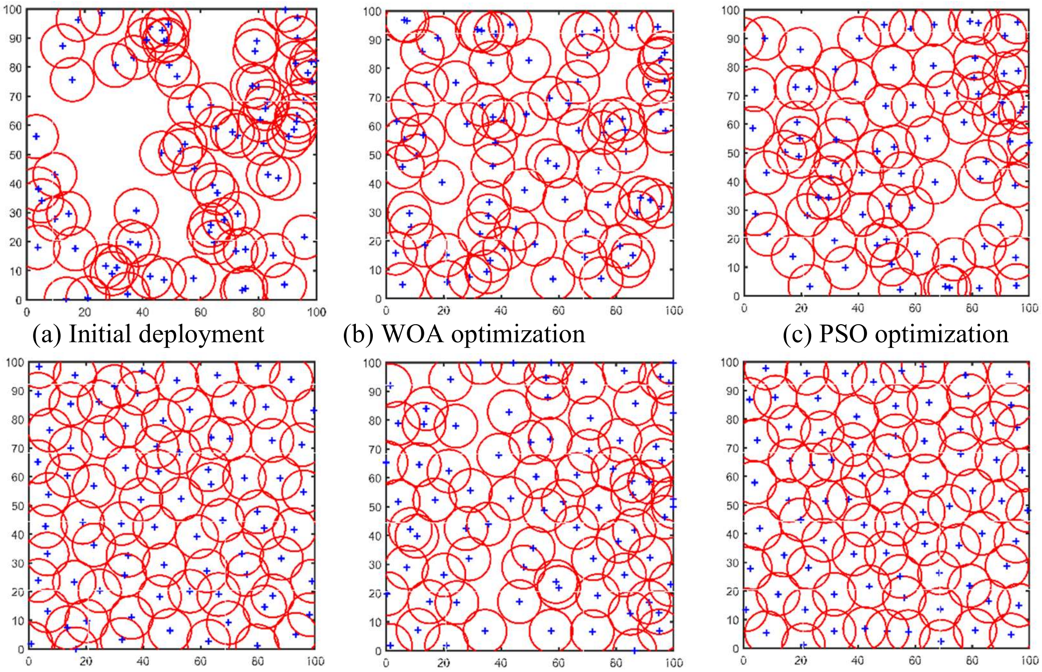
Experiment	Criteria	IWHO	WOA	MuGWO	PSO	GARWOA
Experiment 1	Mean	0.8743	0.7628	0.9246	0.8931	**0.9573**
	Std	0.0173	0.0134	0.0071	0.0137	**0.0051**
	Mid	0.8875	0.7664	0.9235	0.8929	**0.9564**
	Max	0.9208	0.7966	0.9369	0.9196	**0.9658**
	Min	0.8504	0.7520	0.9093	0.8574	**0.9450**
Experiment 2	Mean	0.8970	0.8159	0.9583	0.8589	**0.9815**
	Std	0.0106	0.0108	0.0054	0.0398	**0.0048**
	Mid	0.9062	0.8391	0.9572	0.8838	**0.9749**
	Max	0.9231	0.8726	0.9663	0.9231	**0.9837**
	Min	0.8831	0.8069	0.9481	0.8017	**0.9666**
Experiment 3	Mean	0.9539	0.8838	0.9847	0.8860	**0.9934**
	Std	0.0095	0.0192	0.0021	0.0592	**0.0012**
	Mid	0.9641	0.8721	0.9840	0.9198	**0.9964**
	Max	0.9699	0.9177	0.9888	0.9743	**0.9968**
	Min	0.9466	0.8581	0.9819	0.8280	**0.9914**



(a) Initial deployment (b) WOA optimization (c) PSO optimization

(d) MuGWO optimization (e) IWHO optimization (f) GARWOA optimization

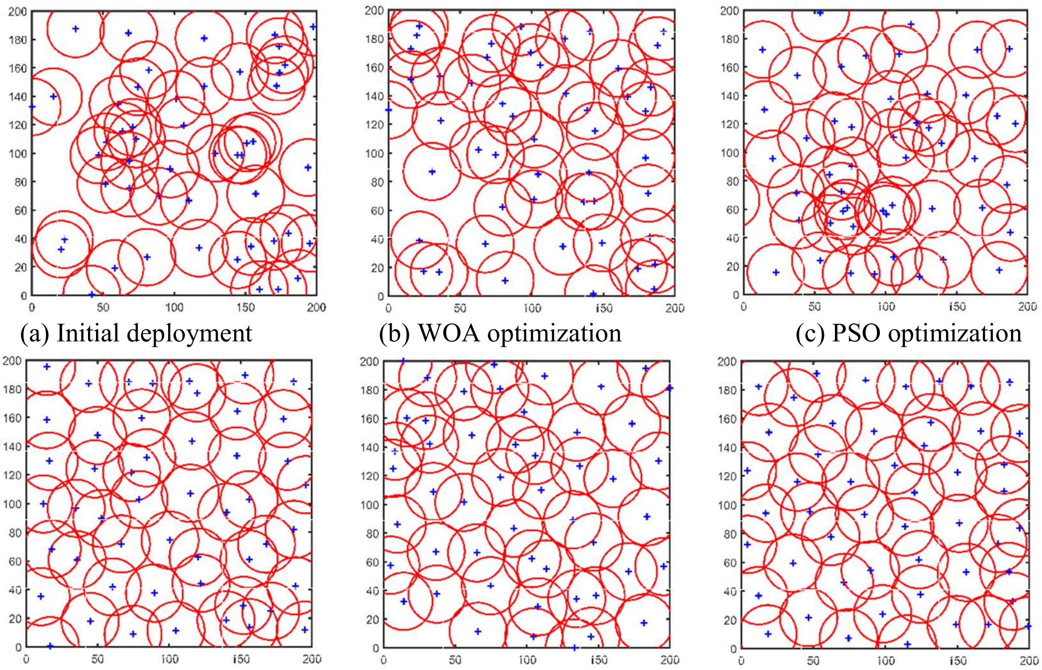
Figure 9. Coverage optimization comparison chart (experiment 1).



(a) Initial deployment (b) WOA optimization (c) PSO optimization

(d) MuGWO optimization (e) IWHO optimization (f) GARWOA optimization

Figure 10. Coverage optimization comparison chart (experiment 2).



(a) Initial deployment (b) WOA optimization (c) PSO optimization

(d) MuGWO optimization (e) IWHO optimization (f) GARWOA optimization

Figure 11. Coverage optimization comparison chart (experiment 3).

In each of the three experimental environments, all algorithms underwent 300 iterations, and their convergence curves are depicted in Figure 12. Notably, GARWOA exhibited superior optimization capabilities compared to the other four algorithms. While the other four algorithms tended to converge after 100 iterations and struggled to escape local optima, GARWOA continued to refine its search for a better optimum.

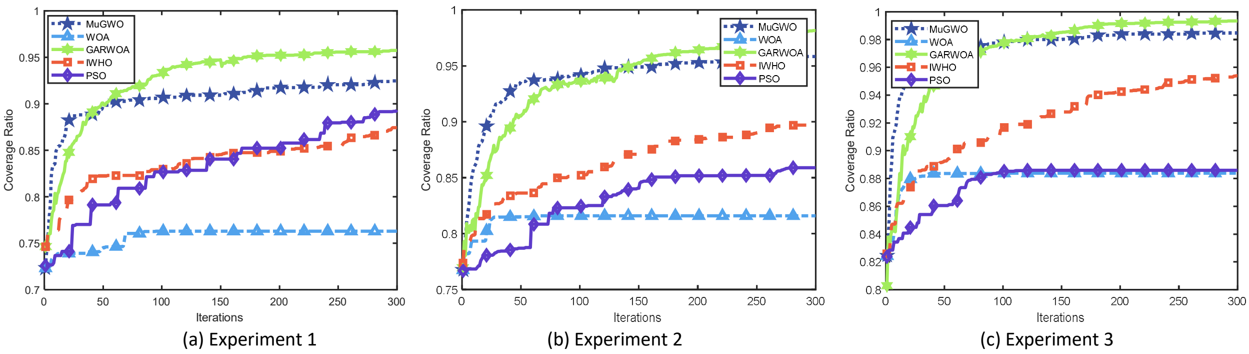


Figure 12. Algorithm coverage optimization convergence curves.

Furthermore, the coverage results for all algorithms in the three experimental environments are presented in Figure 13. The results indicate that GARWOA ultimately outperforms the other four algorithms in terms of coverage. In Experiment 1, the coverage ratio reached 95.73%, representing a 3.27% improvement over the best-performing MuGWO algorithm and a 19.45% improvement over the least effective WOA algorithm. In Experiment 2, the coverage ratio was 98.15%, indicating a 2.32% improvement over MuGWO and a 16.56% improvement over WOA. In Experiment 3, the coverage ratio reached 99.34%, signifying a 0.87% improvement over MuGWO and a 10.96% improvement over WOA. Therefore, GARWOA exhibits excellent performance in optimizing WSN coverage.

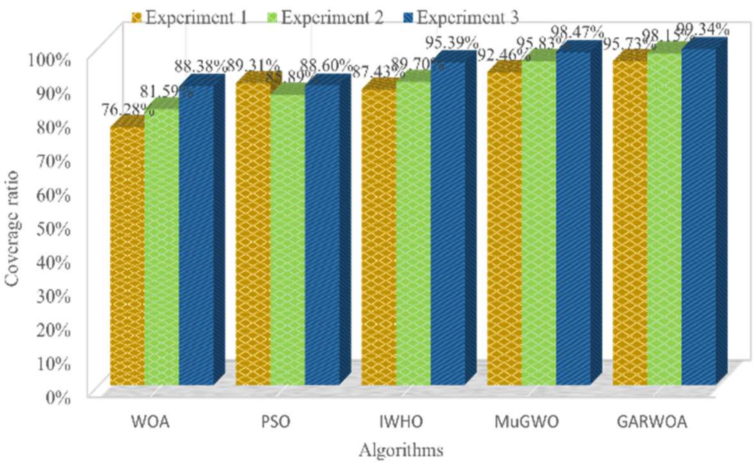


Figure 13. Comparison of coverage in different experiments with various algorithms.

6. Discussion and conclusions

This paper investigates the problem of WSN coverage optimization. The basic WOA is improved by initializing the population using SPM chaotic mapping, which ensures a more uniform distribution in the search space. The addition of the Levy flight mechanism helps to prevent the algorithm from getting stuck in local optima and avoids premature convergence. Furthermore, a non-linear factor is introduced to balance the algorithm's search and exploitation capabilities. By combining the improved WOA with the GA, a WSN coverage optimization algorithm named

GARWOA is designed. Through experimental simulations and analysis, the convergence performance and stability of the GARWOA algorithm are compared with other relevant algorithms, demonstrating its excellent optimization performance. Comparisons with other relevant algorithms also prove that it can effectively improve network coverage, exhibiting favorable optimization capabilities.

In the next stage of work, the integration of artificial intelligence with WSN will be considered, incorporating intelligent sensor and node technologies to enable autonomous learning of nodes and adaptation to environmental changes. This will elevate the level of intelligence of the nodes, leading to more intelligent data processing and decision-making.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by the Key Research and Development Program of Zhejiang under Grant No. 2021C01036, Zhejiang Provincial Key Laboratory of New Network Standards and Technologies (No.2013E10012), the National Natural Science Foundation of China (61871468) and School-level Teaching Project of Zhejiang Gongshang University No. 1120XJ2918335.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. J. J. Sumesh, C. P. Maheswaran, Energy efficient secure-trust-based ring cluster routing in wireless sensor network, *J. Interconnect. Networks*, **23** (2023). https://doi.org/10.1142/S0219265922500049
2. P. Chaturvedi, A. K. Daniel, A Comprehensive review on scheduling based approaches for target coverage in WSN, *Wireless Pers. Commun.*, **123** (2022), 3147–3199. https://doi.org/10.1007/s11277-021-09281-7
3. H. Chen, X. Wang, B. Ge, T. Zhang, Z. Zhu, A multi-strategy improved sparrow search algorithm for coverage optimization in a WSN, *Sensors Basel*, **23** (2023), 4124. https://doi.org/10.3390/s23084124
4. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software*, **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
5. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Software*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
6. M. Elyasi, M. E. Simitcioğlu, A. Saydemir, A. Ekici, O. Ö. Özener, H. Sözer, Genetic algorithms and heuristics hybridized for software architecture recovery, *Autom. Software Eng.*, **30** (2023). https://doi.org/10.1007/s10515-023-00384-y

7. J. Duan, A. N. Yao, Z. Wang, L. T. Yu, An improved sparrow search algorithm optimizes coverage in wireless sensor, *J. Jilin Univ.*, **52** (2022), 1–11.
8. M. Zhang, D. Wang, M. Yang, W. Tan, J. Yang, HPSBA: A modified hybrid framework with convergence analysis for solving wireless sensor network coverage optimization problem, *Axioms*, **11** (2022), 675. https://doi.org/10.3390/axioms11120675
9. T. N. Trong, T. D. Trong, T. N. Thi, V. N. Trinh, An improved honey badger algorithm for coverage optimization in wireless sensor network, *J Int. Technol*, **24** (2023), 363–377. https://doi.org/10.53106/160792642023032402015
10. C. Zeng, T. Qin, W. Tan, C. Lin, Z. Zhu, J. Yang, et al., Coverage optimization of heterogeneous wireless sensor network based on improved wild horse optimizer, *Biomimetics*, **8** (2023), 70. https://doi.org/10.3390/biomimetics8010070
11. S. Nematzadeh, M. Torkamanian-Afshar, A. Seyyedabbasi, F. Kiani, Maximizing coverage and maintaining connectivity in WSN and decentralized IoT: an efficient metaheuristic-based method for environment-aware node deployment, *Neural Comput. Appl.*, **35** (2023), 611–641. https://doi.org/10.1007/s00521-022-07786-1
12. M. Hamid, A. G. Aghdam, Distributed deployment algorithms for coverage improvement in a network of wireless mobile sensors: Relocation by virtual force, *IEEE Trans. Control Network Syst.*, **4** (2017). https://doi.org/10.1109/tcns.2016.2547579
13. S. Liu, R. Zhang, Y. Shi, Design of coverage algorithm for mobile sensor networks based on virtual molecular force, *Comput. Commun.*, **150** (2020). https://doi.org/10.1016/j.comcom.2019.11.001
14. M. Toloueiashtian, M. Golsorkhtabaramiri, S. Y. B. Rad, An improved whale optimization algorithm solving the point coverage problem in wireless sensor networks, *Telecommun. Syst.*, **79** (2022). https://doi.org/10.1007/s11235-021-00866-y
15. J. Kavita, A. Veena, A QoS aware optimal node deployment in wireless sensor network using Grey wolf optimization approach for IoT applications, *Telecommun. Syst.*, **78** (2021).
16. Y. Jianghao, D. Na, Z. Jindan, Wireless Sensor Network coverage optimization based on Yin–Yang pigeon-inspired optimization algorithm for Internet of Things, *Int. Things*, **19** (2022). https://doi.org/10.1016/j.iot.2022.100546
17. N. Bacanin, M. Antonijevic, T. Bezdan, M. Zivkovic, T. A. Rashid, Wireless sensor networks localization by improved whale optimization algorithm, in *Proceedings of 2nd International Conference on Artificial Intelligence: Advances and Applications*, 2022. https://doi.org/10.1007/978-981-16-6332-1_62
18. H. Wang, K. Li, W. Pedrycz, An elite hybrid metaheuristic optimization algorithm for maximizing wireless sensor networks lifetime with a sink node, *IEEE Sensors J.*, **20** (2020), 5634–5649. https://doi.org/10.1109/JSEN.2020.2971035
19. M. Zivkovic, N. Bacanin, T. Zivkovic, I. Strumberger, E. Tuba, M. Tuba, Enhanced grey wolf algorithm for energy efficient wireless sensor networks, in *2020 zooming innovation in consumer technologies conference (ZINC), IEEE*, 2020. https://doi.org/10.1109/ZINC50678.2020.9161788
20. Y. Xue, B. Xue, M. Zhang, Self-adaptive particle swarm optimization for large-scale feature selection in classification, *Trans. Knowl. Dis. From Data*, **13** (2019). https://doi.org/10.1145/3340848

21. Y. Xue, C. Chen, A. Slowik, Neural architecture search based on a multi-objective evolutionary algorithm with probability stack, *IEEE Trans. Evol. Comput.*, **27** (2023), 778–786. https://doi.org/10.1109/TEVC.2023.3252612
22. Y. Hu, Y. Zhang. D. Gong, Multiobjective particle swarm optimization for feature selection with fuzzy cost, *IEEE Trans. Cybern.*, **51** (2021), 874–888. https://doi.org/10.1109/TCYB.2020.3015756
23. S. Banoth, P. Donta, T. Amgoth, Target-aware distributed coverage and connectivity algorithm for wireless sensor networks, *Wireless Networks*, **29** (2023), 1815–1830. https://doi.org/10.1007/s11276-022-03224-1
24. D. Kumar, T. Amgoth, C. Annavarapu, Machine learning algorithms for wireless sensor networks: A survey, *Inf. Fusion*, **49** (2019), 1–25. https://doi.org/10.1016/j.inffus.2018.09.013
25. P. Chaturvedi, A.K. Daniel, A comprehensive review on scheduling based approaches for target coverage in wsn, *Wireless Pers. Commun.*, **123** (2022), 3147–3199. https://doi.org/10.1007/s11277-021-09281-7
26. W. Jin, L. Ying, R. Shuying, Z. Xinyu, H. Jinbin, A novel self-adaptive multi-strategy artificial bee colony algorithm for coverage optimization in wireless sensor networks, *Ad Hoc Network*, **150** (2023), 103284. https://doi.org/10.1016/j.adhoc.2023.103284
27. Z. Rakhshan, J. Tariq, A. Z. Anwar, U. Vali, Novel metaheuristic routing algorithm with optimized energy and enhanced coverage for WSNs, *Ad Hoc Network*, **144** (2023). https://doi.org/10.1016/j.adhoc.2023.103133
28. Q. Q. Ma, L. G. Dong, X. Jiang, Distributed high-efficiency entropy energy-saving clustering routing algorithm for SDWSN, *Telecommun. Sci.*, **39** (2023), 100–114. https://doi.org/10.11959/j.issn.1000-0801.2023024
29. D. Ban, X. Lv, X. Wang, Efficient image encryption algorithm based on 1D chaotic map, *Comput. Sci.*, **47** (2020), 278–284.
30. Y. Duan, X. Yu, A collaboration-based hybrid gwo-sca optimizer for engineering optimization problems, *Expert Syst. Appl.*, **213** (2023), 119017. https://doi.org/10.1016/j.eswa.2022.119017
31. H. Gao, Q. Zhang, J. Bu, J. Li, H. Zhang, Teaching-learning-based optimization algorithm based on cooperative mutation and Lévy flight strategy and its application, *J. Comput. Appl.*, **43** (2023), 1355–1364.
32. H. Zhang, D. Long, T. Qin, X. Wang, J. Yang, Coverage and connectivity optimization of WSN based on improved artificial bee colony algorithm, *Comput. Sci. Des.*, **43** (2022), 2701–2710. https://doi.org/10.16208/j.issn1000-7024.2022.10.001