OPTIMIZATION RESEARCH OF IMPROVED GENERALIZED NORMAL DISTRIBUTION OPTIMIZATION ALGORITHM IN WSN COVERAGE

Biling HU[1] *, Mengshu HOU[2], Fei XIE[3], Yanan LIU[4]

A coverage control strategy based on an improved generalized normal distribution optimization algorithm is proposed for coverage optimization of sensor networks. Firstly, IGNDO uses a combination of Logistic and Tent chaotic mappings to initialize the original population in the initialization phase; secondly, it uses nonlinear control parameters to adjust the probability of local search and global search being selected in the iterative phase, and perturbs the update of individuals after iterative update with partial Lévy flight to Improve their ability to escape from their local best; Finally, the algorithm's search ability is further improved by using random regression transgression. In the experimental phase, the improved algorithm is firstly tested with benchmark functions, and secondly IGNDO is employed to sensor network coverage optimization, and the optimal results are obtained in comparison with other optimization algorithms.

Keywords: wireless sensor network; coverage optimization; generalized normal distribution

1. Introduction

A large number of stationary or mobile sensor nodes form a wireless sensor network, which collaborate with each other to sense, gather, analyze and deliver information about the monitored area in a multi-hop wireless communication, and finally send the information to the control terminal and users. Due to its ease of deployment, self-organization and rapid movement it is now widely used in agriculture, industry, construction, aviation, the environment, etc. But due to the low energy level of sensor nodes, limited processing capacity and communication bandwidth, how to organize the location of sensor nodes to better perform tasks such as environment sensing and information acquisition has become a key issue in sensor network applications, namely the problem of coverage optimization [1-2].

[1] Lecturer, School of Computer science, Hefei Normal University, China, e-mail: bilinghu@hfnu.edu.cn
[2] Prof., School of Computer Science and Engineering, University of Electronic Science and Technology of China, China, e-mail: mshou@uestc.edu.cn
[3] Prof., School of Computer science, Hefei Normal University, China, e-mail: xiefei9815057@sina.com
[4] Associate Prof., School of Computer science, Hefei Normal University, China, e-mail: liuyn1210@163.com

Intelligent optimization algorithms have been gradually applied to wireless sensor coverage problems to optimize network coverage in recent years. In the literature [3], a particle swarm algorithm was applied to optimize the node distribution of wireless sensor networks, which improved the coverage to some extent. In literature [4], an artificial fish swarm algorithm was used, which achieved better coverage optimization than particle swarm and genetic algorithms. The literature [5] investigated the optimization performance of extrapolated artificial bee colony algorithm on sensor network coverage. The literature [6] improved the gray wolf algorithm by using chaotic algorithm to improve the initialized population, and also improved the convergence factor and local extreme, and applied the improved gray wolf algorithm to sensor network node deployment and obtained superior coverage control than the gray wolf algorithm. In the literature [7], an enhanced sparrow search algorithm was proposed and used to improve the node layout of the sensing network. Better coverage as well as convergence speed were obtained compared to the compared algorithms, but there are still coverage blind areas. The literature [8] improves the Ant-Lion algorithm by incorporating multiple strategies and using it for optimizing the network coverage. This improvement is better than the previous improved version for network coverage, but the effective coverage of nodes is still low and its stability and accuracy are not satisfactory. The literature [9] proposes an improved whale algorithm and applies it to WSN coverage optimization and obtains higher coverage than the comparison algorithm, but the coverage vulnerability is still large. The literature [10] investigated the optimization performance of the water wave optimization algorithm for sensor network coverage. In the literature [11], the Improved Artificial Bee Colony (ABC) algorithm was combined with the teaching strategy of TLBO optimization algorithm to balance the global search and local search and obtained better optimized coverage than the Improved Artificial Bee Colony and TLBO. The above research results show that for the coverage problem of WSN networks, the network coverage can be optimized by using intelligent optimization algorithms, but the overall effect of optimization still needs to be improved.

The theoretical basis of the generalized normal distribution optimization algorithm is derived from the generalized normal distribution model. Unlike most meta-heuristic algorithms, using this optimization algorithm to solve optimal problems requires only population size and ending conditions, no special control parameters are required. However, similar to the problems of other intelligent algorithms, the generalized normal distribution also suffers from slow convergence and is prone to be trapped in a local optimum, and its ability to find an optimum still needs to be improved. In this paper, we provided a hybrid policy to update the generalized normal distribution optimization (IGNDO) algorithm, and also applied the improved IGNDO algorithm to the coverage of sensor networks to maximize the network coverage performance.

2. WSN model

2.1 Basic assumptions

Assume that the WSN sensing area is a two-dimensional plane with an area of $S = m \times n$, and the nodes are located randomly over the monitoring area, the set of nodes is defined as $C = \{C_1, C_2, \cdots, C_N\}$, and the coordinates of the node position are $(x_i, y_i), i = 1, 2, \cdots, N$. The sensing radius of each node is r, the communication radius is R, and $R = 2r$. At the same time, the sensing area of all sensor nodes is abstracted into a closed circle with itself as the center and a fixed radius r. Further, the two-dimensional sensing area is gridded into a set of pixel points to be covered, which is $T = \{T_1, T_2, \cdots, T_{m\times n}\}$. The coordinates of the geometric centroid of each pixel point are $(x_j, y_j), j = 1, 2, \cdots, m \times n$, i.e., the location of the coverage optimization target.

2.2 Coverage model

Based on the above basic assumptions, the Euclidean distance from the node to the target point is shown in equation (1):

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

Ideally, the probability that a target being sensed by a node is defined as

$$P\left(C_i, T_j\right) = \begin{cases} 1, & d(C_i, T_j) \le r \\ 0, & otherwise \end{cases} \tag{2}$$

In practical applications, due to environmental factors, the actual perceptual model presents a probability distribution, i.e.

$$P\left(C_i, T_j\right) = \begin{cases} 1, & d(C_i, T_j) \le r \\ \dfrac{-\alpha_1 \beta_1 \lambda_1}{e^{\alpha_2 + \beta_2 \lambda_2}}, & r - r_e < d(C_i, T_j) < r + r_e \\ 0, & otherwise \end{cases} \tag{3}$$

Where r_e is the measurable reliability of the node, and the range meets $0 < r_e < r$, $\alpha_1, \alpha_1, \beta_1, \beta_2$ is the relevant measurement parameter of the sensor node itself, λ_1, λ_2 is the input parameter and is expressed as

$$\lambda_1 = r_e - r + d(C_i, T_j) \tag{4}$$

$$\lambda_2 = r_e + r - d(C_i, T_j) \tag{5}$$

Then for a target monitoring point T_j , the joint probability of all sensor nodes sensing it is

$$P_s(C,T_j)=1-\prod_{i=1}^{N}\left(1-P(C_i,T_j)\right) \tag{6}$$

Knowing the deployment of sensor nodes in the sensing area, the ratio of the total number of pixels that are covered by the set C of sensor nodes to the total number of all pixels in the area is the coverage ratio, defined as:

$$P_{\text{cov}}=\frac{\sum_{j=1}^{j=m\times n}P_s(C,T_j)}{m}\times n \tag{7}$$

How to optimize the placement of WSN nodes to maximize the value of equation (7) is our concern.

3. Generalized normal distribution optimization algorithm

The locations of all individuals in the generalized normal distribution optimization algorithm [12] are regarded as stochastic variables that comply with a normal distribution.

The GNDO algorithm updates the position of individuals based on the constructed generalized normal distribution formula. Its search process consists of a local search and a global search, both of which have the same chance of being chosen.

The work of population initialization in GNDO is defined by equation (8).

$$x_{i,j}^{t}=l_j+(u_j-l_j)\times\lambda_i, i=1,2,3,...,N, j=1,2,3,...,D \tag{8}$$

The D is the dimension of the problem solution, l_j is the bottom bound of the j dimension, u_j is the top bound of the j dimension, and λ_5 is a stochastic number ranging from 0 to 1.

The mechanism for screening individuals in a population is shown in Equation (9):

$$x_i^{t+1}=\begin{cases}v_i^t, if\ f(v_i^t)<f(x_i^t)\\ x_i^t, otherwise\end{cases} \tag{9}$$

Where x_i^{t+1} is the position of the ith individual at generation $t+1$ and v_i^t is the updated position of the ith individual at generation$t$. v_i^t is then obtained by local and global search.

The local search is built on the model of the generalized normal distribution constructed from the current optimal and average positions. The formula for the updated position of an individual under the local search strategy is shown by the following equation (10).

$$v_i^t = \mu_i + \delta_i \times \eta, i = 1, 2, 3, \ldots, N \tag{10}$$

μ_i is the generalized mean position of the ith individual, δ_i is the generalized standard variance, of which η is the penalty parameter. where μ_i, δ_i, η are defined by equations (11), (12), and (13) below, respectively.

$$\mu_i = \frac{1}{3}\left(x_i^t + x_{Best}^t + M\right) \tag{11}$$

$$\delta_i = \sqrt{\frac{1}{3}\left[\left(x_i^t - \mu\right)^2 + \left(x_{Best}^t - \mu\right)^2 + \left(M - \mu\right)^2\right]} \tag{12}$$

$$\eta = \begin{cases} \sqrt{-\log(\lambda_1)} \times \cos(2\pi\lambda_2), if \ a \le b \\ \sqrt{-\log(\lambda_1)} \times \cos(2\pi\lambda_2 + \pi), otherwise \end{cases} \tag{13}$$

Where $a, b, \lambda_1, \lambda_2$ are stochastic numbers ranging from 0 to 1, x_{Best}^t represents the current best individual position, and M is defined as the average position of all individuals, which is calculated by equation (14).

$$M = \frac{\sum_1^N x_i^t}{N} \tag{14}$$

In the GNDO algorithm, the global search strategy is based on an arbitrary selection of three individuals, and the update formula for the individual positions can be expressed as equation (15).

$$v_i^t = x_i^t + \beta \times \left(|\lambda_3| \times v_1\right) + (1 - \beta) \times \left(|\lambda_4| \times v_2\right) \tag{15}$$

where λ_3 and λ_4 are two arbitrary numbers that follow a standard normal distribution, β is a number between 0 and 1, called the adjustment parameter, and v_1 and v_2 are two trajectory vectors, which are calculated as shown in (16) and (17).

$$v_1 = \begin{cases} x_i^t - x_{p1}^t, if \ f(x_i^t) < f(x_{p1}^t) \\ x_{p1}^t - x_i^t, otherwise \end{cases} \tag{16}$$

$$v_2 = \begin{cases} x_{p2}^t - x_{p3}^t, if \ f(x_{p2}^t) < f(x_{p3}^t) \\ x_{p3}^t - x_{p2}^t, otherwise \end{cases} \tag{17}$$

where $p1$、$p2$ and $p3$ are three arbitrary integers chosen within 1 to N, consistent with $p1 \neq p2 \neq p3 \neq i$. After initializing the initial population and the max value of iterations, the GNDO algorithm finds the location of the optimal individual and outputs it by using local and global search strategies with equal probability.

4. Improved optimization algorithm for generalized normal distribution

4.1 Logistic-Tent chaos mapping

The original population of the basic GNDO algorithm prior to the iteration is generated randomly. The result is that populations tend to be unevenly distributed and poorly diversified, which in turn affects the optimization results of the algorithm. In contrast, chaotic motion has the characteristics of regularity, periodicity and randomness [13], which can easily motivate the optimization algorithm out of local optima solutions and improve the global search ability.

In this paper, a combination of Logistic mapping and Tent mapping models is introduced into the initialization phase of the IGNDO algorithm. Using the characteristics of regularity, periodicity and randomness of the two mappings, the chaotic sequences generated by the mappings are transformed into the solution space of the IGNDO algorithm instead of the original population by establishing mapping relations, which lays a good foundation for global search.

The expression for a logistic-tent chaotic mapping is as follows:

$$x_{k+1} = \begin{cases} \left[r x_k (1 - x_k) + \frac{(4-r)}{2} x_k \right] \mathrm{mod} 1, if\ x_k < 0.5 \\ \left[r x_k (1 - x_k) + \frac{(4-r)}{2} (1 - x_k) \right] \mathrm{mod} 1, if\ x_k < 0.5 \end{cases} \tag{18}$$

Where mod is the residual function, [ ] represents rounding, $x = (x_1, x_2, \cdots, x_d)$ is the chaotic sequence generated by the Logistic-Tent mapping, and d represents dimensionality.

4.2 Non-linear control parameters

In the GNDO algorithm, the choice between local and global search is not related to the number of iterations performed but is chosen by judging if the generated random number is larger than 0.5. When the random number is larger than 0.5, the local search is performed, otherwise the global search is performed. The update strategy has a large randomness. In order to further balance the global exploration and local exploitation capability of the algorithm [14] and obtain better performance of the search. In this paper, a non-linear control argument is considered to replace the fixed value of 0.5 in the original algorithm, with the aim of enhancing the probability of the global search being selected in the early iterations to facilitate the search for the best solution and improving the probability of local exploitation in the later iterations, facilitating the convergence of the algorithm.

Non-linear control parameters are defined as

$$a(t) = a_{ini} - (a_{ini} - a_{fin}) \times (t / Maxit)^2 \tag{19}$$

Where the current round number is t, a_{ini} is the initial value of the control parameter, a_{fin} is the final value of the control parameter. $Maxit$ is the maximum number of iterations.

4.3 Levy Flight Strategy

As the iterations progress, it is found that the basic generalized normal distribution optimization algorithm is difficult to jump out of the local optimum. To address this problem, in this paper, we consider the introduction of Lévy flight [15] perturbation strategy after the global search or local search of population individuals to improve the algorithm prone to local optimum and the occurrence of premature convergence.

With the introduction of Lévy flight, the individual is updated with the following equation for position.

$$x_i^l(t) = x_i(t) + \alpha \oplus Levy(\lambda) \quad i = 1, 2, \cdots, n \tag{20}$$

where i denotes the individual number, $x_i^l(t)$ denotes the position of the ith individual after the Lévy flight perturbation, $x_i(t)$ is the position of the ith individual after the local or global search, $\oplus$ is the dot product, α is the step control parameter, $Levy(\lambda)$ denotes the random search path, denoted as

$$Levy \sim u = t^{-\lambda} \quad 1 < \lambda \leq 3 \tag{21}$$

For the random step of the Lévy flight, the Mantegna algorithm is currently used to simulate it, and is calculated as

$$s = \frac{u}{|v|^{1/\beta}} \tag{22}$$

Where the parameter β equals to 1.5, parameters u and v obey a normal distribution, defined as follows.

$$u \sim N(0, \sigma_u^2) \tag{23}$$

$$v \sim N(0, \sigma_v^2) \tag{24}$$

Of which

$$\sigma_u = \left[\frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma\left[\frac{1+\beta}{2}\right] \beta \, 2^{\frac{(\beta-1)}{2}}} \right]^{1/\beta} \tag{25}$$

$$\sigma_v = 1 \tag{26}$$

In order to avoid too many perturbations that cause the algorithm to stall, a random number $rand$ is generated before each Lévy flight perturbation, and the perturbation probability is set to 0.5, and the individual position is perturbed only when the random number $rand > 0.5$. At the same time, a greedy selection strategy is also employed, with the perturbed individual retained if its fitness is better than the original updated position and discarded otherwise.

4.4 Random regression processing out of bounds

Intelligent optimization algorithms usually have individuals crossing the boundaries in the process of iterative individual search, and it is now common practice to directly assign upper or lower bound values to the positions of individuals located outside the search region. In this paper, we adopt a random regression method of out-of-bounds treatment, as shown in equation (27).

$$x_i^{'}(t) = \begin{cases} ub - \min(x_i(t) - ub, ub - lb).\gamma, x_i(t) > ub \\ lb + \min(lb - x_i(t), ub - lb).\gamma, x_i(t) < lb \end{cases} \tag{27}$$

where γ is the random number of the [0,1] interval distribution, ub 、 lb is the upper and lower bound of the defined boundary, $x_i(t)$ is the location of the individual after the ith generation update, and $x_i{'}(t)$ is the position of the individual after the boundary crossing process.

5. Coverage optimization design

If the IGNDO algorithm is used for sensor network coverage optimization, the fitness function $f(x)$ should be set to the network coverage P_{COV} of the sensor network nodes. The optimization goal is to find the optimal deployment of sensor nodes to maximize network coverage. Each individual in the algorithm represents a node distribution. Define the number of sensor nodes is N, then the dimension of an individual is $2N$. The specific coverage optimization is designed in the following steps:

Step 1: Set the range of the sensor node monitoring area s, the population size n, the number of sensor nodes N, the total number of iterations $Maxit$, the detection radius of the sensor nodes r , the radius of communication R, and the parameters related to the sensing model.

Step 2: Generate the initial population with Logistic-Tent mapping.

Step 3: For each individual in the population, the fitness value is computed, and the current best individual is located in terms of the fitness value.

Step 4: Generate a random number and compare it with the current value of Eq. (19); if the number is less than $a(t)$, the individual position is updated according to the global search defined by Eq. (15)(16)(17); otherwise, the individual position is updated according to the local search defined by Eq. (11)(12)(13).

Step 5: For the iterated individual position firstly, a random regression crossing process is performed, followed by an individual update according to equation (9).

Step 6: To apply the Lévy flight perturbation to the updated individual position according to whether the generated random number is greater than 0.5, and compare it with the position before the perturbation after the boundary crossing process, and keep it if the adaptation is better, otherwise discard the position after the perturbation.

Step 7: Judging whether the number of maximum iterations is met, if it is met, the global best solution will be output, i.e. the coordinates of the corresponding node position and the optimal coverage, otherwise turn to step 3 for the next iteration.

6. Experimental simulation

6.1 Benchmark function optimization performance testing

Based on the selected four benchmark functions, the search accuracy and convergence performance of the GNDO and IGNDO algorithms were compared and verified at 1000 iterations, with each benchmark function run independently for 30 times and averaged. The benchmark function information is listed in Table 1 below, and the comparison results are presented in Table 2.

Table 1

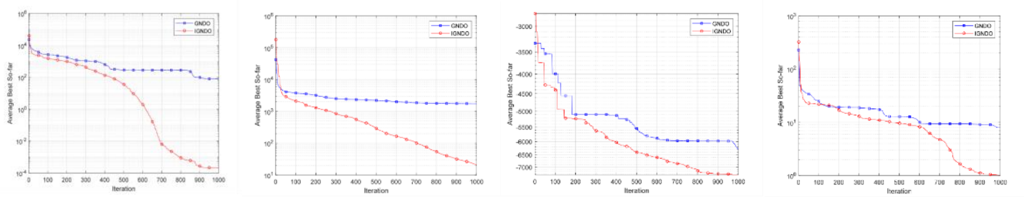
Benchmarking functions

Function name	Expression	Search range	Optimum value
SphereModel	$f_1(x)=\sum_{i=1}^{n} x_i^2$	[-100,100]	0
Schwefel's problem 1.2	$f_2(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j^2\right)$	[-100,100]	0
Generalized Schwefei's problem 2.26	$f_3(x)=\sum_{i=1}^{n} -x_i \sin\sqrt{\lvert x_i \rvert}$	[-500,500]	-418.9829n
Generalized Griewank Function	$f_8(x)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos\frac{x_i}{\sqrt{i}}+1$	[-600,600]	0

Table 2

Basic function optimization results

F	GNDO	IGNDO
	Ave	Ave
F1	78.418	0.00021841
F2	1720.7449	21.0978
F3	-6284.5402	-7329.1481
F4	7.9212	1.0125



(a)F1 (b)F2 (c)F3 (d)F4

Fig. 1. Comparison of optimal convergence curves of reference functions

The results in Table 2 and Fig. 1 demonstrated that the IGNDO algorithm exhibits better search performance and convergence results than the basic GNDO algorithm. For single-peaked test functions F1 and F2 with only one global optimal solution, the IGNDO algorithm shows stronger convergence performance as the number of iterations increases. The multi-peaked test functions F3 and F4 have many local optima, and the power of the algorithm to explore is critical to its ability to achieve better optimization. From the convergence curves in Fig. 1(c)(d) it can be seen that compared to GNDO, IGNDO exhibits a better ability to escape from the local optimum the local optimum and thus obtain a better global optimum solution.

6.2 Comparison of coverage performance of different algorithms

The improved optimization algorithms IGNDO and improved whale optimization algorithm [9], extrapolated artificial bee colony algorithm[5], particle swarm algorithm[3], artificial fish swarm algorithm(AF)[4] and the basic generalized normal distribution optimization algorithm in this paper were simulated for different scenarios of coverage to compare the coverage performance of different algorithms. The simulation experimental parameters of several algorithms are consistent, and the simulation experiments of each comparison algorithm are run 20 times independently to take the average coverage. Set the number of iterations to 500 and the population size to 30. The simulation parameters for each scenario are presented in Table 3.

Table 3

Parameter Setting

Parameters	Value
Area Size $m*n$(m^2)	[20*20;50*50;100*100]
Number of nodes N	[24;35;40]
Node Perception Radius r(m)	[2.5;5;10]
Node communication radius R(m)	[5;10;20]

(1) Coverage comparison with monitoring area of 20*20

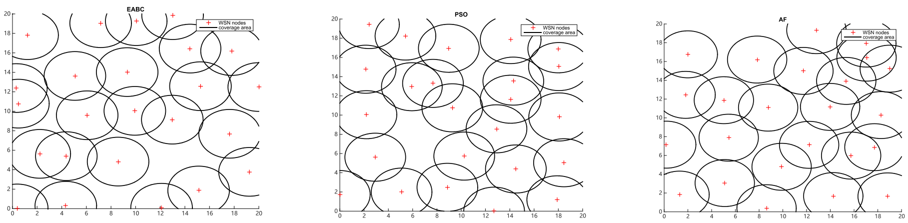
The comparison of the average coverage optimization results for 20 runs of the EABC algorithm, PSO algorithm, AF algorithm, IWOA algorithm, GNDO and IGNDO algorithm with the monitoring area of 20*20 is listed in Table 4. Fig.

2 depicts the node distribution of EABC algorithm, PSO algorithm, AF algorithm, IWOA algorithm, GNDO algorithm and IGNDO algorithm after running the optimization can be viewed, and Fig. 5(a) shows the coverage convergence curve of each comparison algorithm.

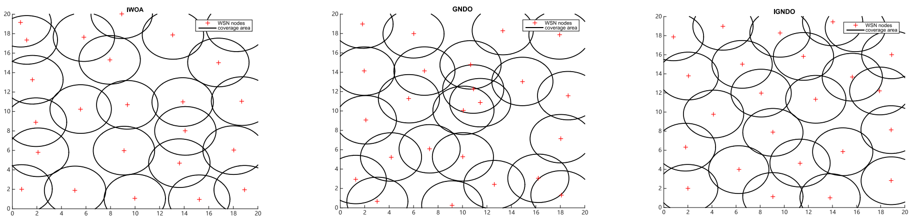
Table 4

Comparison of coverage optimization results

Algorithm	Average coverage（20rounds）
EABC	78.05%
PSO	83.46%
AF	86.91%
IWOA	90.70%
GNDO	85.71%
IGNDO	92.06%



(a)EABC (b) PSO (c)AF



(d) IWOA (e) GNDO (f) IGNDO

Fig. 2 Optimized node distribution diagram

The data in Table 4 shows that the IGNDO algorithm has improved coverage by 14.01%, 8.6%, 5.15%, 1.36%, and 6.35% compared to EABC, PSO, AF, IWOA, and GNDO for its average 20 runs, respectively. It can also be observed from Fig. 3 that IGNDO has a better performance in terms of optimization, more uniform node distribution, and presents less coverage vulnerabilities and redundant areas.

From Fig. 5(a), we can see that compared with the other five algorithms, IGNDO has a better performance in the optimization search ability and can moves beyond the local optimum quickly. Under the parameter setting of monitoring area size of 20*20^{m2} , the average coverage rate of IGNDO algorithm reaches 92.06%

in 20 runs. In contrast, the extrapolated artificial bee swarm algorithm, particle swarm algorithm, and artificial fish swarm optimization algorithm had almost no improvement in coverage from iteration number 100 in 500 iterations. The basic GNDO algorithm presents poorer search ability and convergence performance than the artificial fish swarm algorithm, the improved whale optimization algorithm, and IGNDO. The search ability of IGNDO is weaker than IWOA when the number of iterations is less than 300, but as the number of iterations increases, it quickly moves beyond the local optimum, showing better exploitation and optimization results.

(2) Coverage comparison with monitoring area of 50*50

The average coverage optimization results of each optimization algorithm for 20 runs with the monitoring area of 50*50 are compared as shown in Table 5.

Table 5

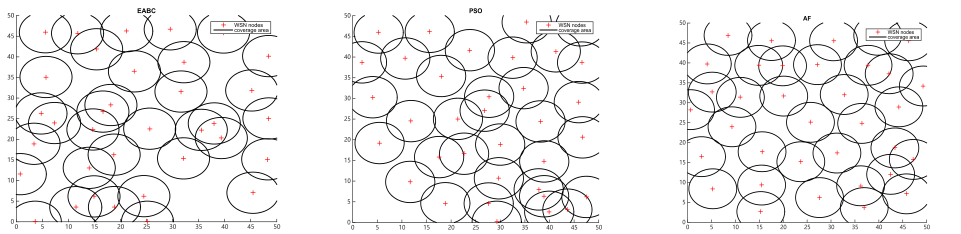
Comparison of coverage optimization results

Algorithm	Average coverage（20rounds）
EABC	75.85%
PSO	81.04%
AF	84.19%
IWOA	85.06%
GNDO	79.89%
IGNDO	88.85%

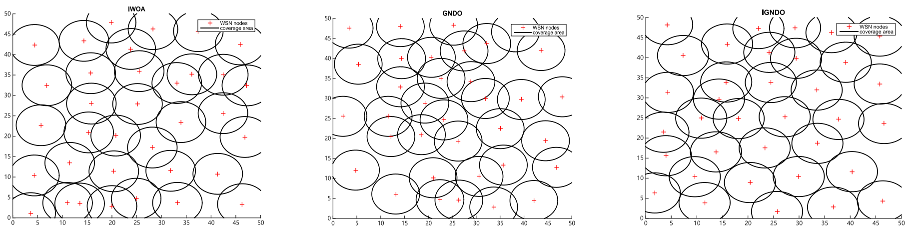
Fig. 3 shows the distribution of nodes optimized by each optimization algorithm, and Fig. 5(b) shows the coverage convergence curve of each comparison algorithm.

The data in Table 5 also reveals that the IGNDO algorithm performs 20 optimizations with an average coverage increase of 13%, 7.81%, and 4.66% compared with EABC algorithm, PSO algorithm, and AF algorithm, respectively. Compared with the basic GNDO improvement, the average coverage of IGNDO improves by 8.96%, and even for IWOA, which has better optimization performance, IGNDO shows a higher coverage finding performance.

It can also be seen from Fig. 3 that the IGNDO algorithm provides better coverage with less redundancy and smaller coverage blind areas. In addition, from the convergence curve Fig. 5(b), it can be seen that the IGNDO algorithm has a more stable search power. Although the algorithm has lower coverage than IWOA until 150 iterations, IGNDO can leave the local optimum quickly and iterate steadily. In contrast, the extrapolated artificial bee colony algorithm and particle swarm algorithm almost stop improving their coverage after 200 iterations.



(a) EABC (b) PSO (c)AF



(d) IWOA (e)GNDO (f) IGNDO

Fig. 3 Optimized node distribution diagram

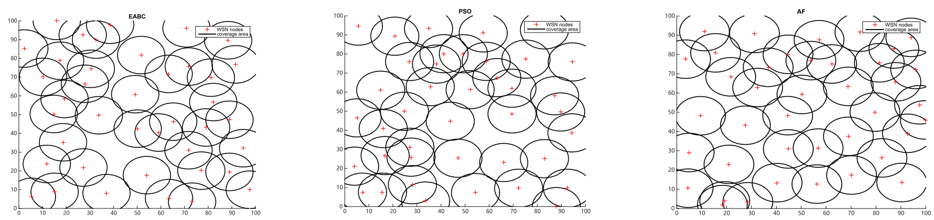
*(3) Coverage comparison with monitoring area of 100*100*

Table 6 provides the average coverage of each optimization algorithm for a monitoring area of 100*100. Fig. 4 illustrates the distribution of nodes after optimization for each optimal algorithm, and Fig. 5(c) presents the convergence curve of coverage for each algorithm compared.

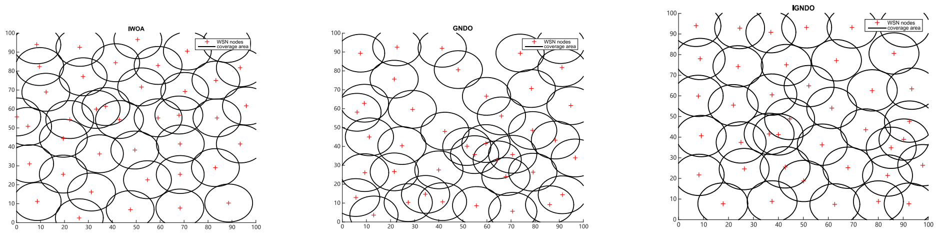
Table 6

Comparison of coverage optimization results

Algorithm	Average coverage（20rounds）
EABC	83.98%
PSO	85.13%
AF	84.91%
IWOA	92.90%
GNDO	86.82%
IGNDO	95.58%



(a)EABC (b) PSO (c)AF



(d) IWOA (e)GNDO (f) IGNDO

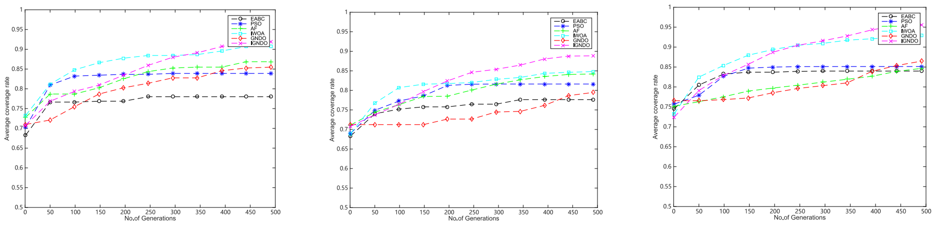
Fig. 4 Optimized node distribution diagram

As shown in Table 6, the average coverage rate of IGNDO algorithm for 20 optimizations is 11.6%, 10.45%, 10.67%, 2.68%, and 8.76% higher than that of EABC algorithm, PSO algorithm, AF algorithm, IWOA, and the basic GNDO algorithm, respectively, and Fig. 4 presents the optimized nodes of each algorithm in the monitoring area of 100*100. Again, it can be clearly seen that the nodes are more evenly distributed, with less coverage vulnerability and less redundancy. In addition, from the convergence curve Fig. 5(c), we can see that the convergence performance of IGNDO is more stable and still has good exploration ability after 300 iterations. In contrast, the extrapolated artificial bee swarm algorithm and particle swarm algorithm tend to converge after 150 iterations, and it’s hard to escape from the local optimum. The basic GNDO has the lowest coverage rate in the first 400 iterations, and the optimization-seeking ability only gradually increases after 400 iterations. And the growth trend of IWOA starts to slow down after 250 iterations. Comparing GNDO, IGNDO and the remaining optimization algorithms separately, it can be seen that the improved strategies effectively improve the algorithm's merit-seeking ability and convergence speed.

7. Conclusion

A reasonable solution to the node coverage problem in WSN can be found based on a meta-heuristic algorithm. In this paper, we proposed a variant of the generalized normal distribution optimization algorithm that integrates initialization, dynamic adjustment of global and local search probability of being selected and incorporates the Levy flight variation and optimizes the boundary crossing treatment. Under the same number of iterations, compared to GNDO in terms of benchmark function test, IGNDO shows better search results and faster convergence as the number of iterations advances, both for single-peaked functions F1 and F2 and for multi-peaked functions F3 and F4. The IGNDO algorithm has a more uniform distribution of individuals in the initial population, which makes it easier to jump out of the local optimum and has a better global search capability; meanwhile, the probability of being selected for local and global search is dynamically adjusted based on the number of iterations, so that the algorithm has a higher probability of conducting global search at the beginning of the iteration, which enhances the individual search capability, and has a higher probability of conducting local search at the end of the iteration and thus converge more quickly than GNDO. The incorporation of the Levy flight variation further reduces the algorithm to fall into local optimum, while the random crossing process further enhances the individual variability and improves the global search performance of the algorithm. Comparing IGNDO with the selected other optimization algorithms in different size sensor networks under three different monitoring areas also shows that IGNDO obtains better average coverage as the iterations advance. Unlike the node distribution map of each optimization algorithm, the sensor nodes are more uniformly distributed in the monitoring area after using IGNDO for sensor network optimization coverage, with smaller coverage holes between nodes and lower node redundancy per unit coverage area. By comparing the convergence curves of coverage optimization, the global optimization capability of IGNDO algorithm of seeking and jumping out of local optimum is more verified.

In the future, further balanced optimization of the global and local search of the algorithm is planned, and the application of IGNDO to the optimization of the sensing network localization problem is explored.



(a)20*20 (b)50*50 (c)100*100

Fig. 5 Coverage optimization convergence curves

Acknowledgements

This work was supported by the Natural Science Research Projects in Anhui Universities in 2020 (Grant No. KJ2020A0090).

REFERENCES

[1]. Ren Yab, Zhang Sidong, Zhang Hongke. Coverage Control Theory and Algorithm in Wireless Sensor Networks. Journal of Software, 2006, 17(3):12.

[2] Wang Huaiyuan, Ding Xu, Wang Lili, et al. Target connected coverage algorithm in wireless sensor networks. Journal of nanjing university of science and technology, 2017, 9 (3): 285-293. The DOI: 10.14177 / j.carol carroll nki. 32-1397 n. 2017.41.03.003.

[3] Lin Zhuliang, Feng Yuanjing. Wireless Sensor Network Coverage Optimization Strategy Based on Particle Swarm Optimization algorithm. Computer Simulation, 2009(4):5.
[4] Huang Yuyue, Li Keqing. Coverage Optimization of Wireless sensor Networks Based on Artificial Fish Swarm Algorithm. Application Research of Computers, 2013, 30(2):3.
[5] Yu Wenjie, Li Xunbo, Yang Xing, et al. Application research of Artificial Bee Colony Algorithm in WSN Deployment Optimization. Instrument Technique and Sensor,2017(06):158-160+164.
[6] Hu Xiaoping,Cao Jing. Improved Grey Wolf Optimization Algorithm for WSN Node Deployment. Chinese Journal of Sensor and Actuators, 2018, 31(5):6
[7] Wang Zhendong, Wang Jiabao, Li Dahai. Study on WSN Optimization Coverage of an Enhanced Sparrow Search Algorithm. Chinese Journal of Sensor and Actuators, 2021, 34(6):11.
[8] Xu Qinshuai, He Qing, Wei Kangyuan. Modified Ant Lion Optimizer Based Coverage Optimization of Wireless Sensor Network. Chinese Journal of Sensor and Actuators, 2019(2):10.
[9] Song Tingting, Zhang Damin, Wang Yirou, et al. WSN Coverage Optimization based on improved Whale Optimization Algorithm. Chinese Journal of Sensor and Actuators, 2020, 33(3):8
[10] Wang Yi, Shen Xianhao, Tang Chaochen, et al. Research on Coverage of Wireless sensor Networks Based on Water Wave Optimization Algorithm. Journal of Nanjing University of Science and Technology, 2021, 45(6):7.
[11] Lu C, Li X, Yu W, et al. Sensor network sensing coverage optimization with improved artificial bee colony algorithm using teaching strategy. Computing, 2021(6).
[12] Zhang Y, Jin Z, Mirjalili S.Generalized normal distribution optimization and its applications in parameter extraction of photovoltaic models. Energy Conversion and Management, 2020, 224(9):113301.
[13] Wang Jianhao, Zhang Liang, Shi Chao, et al. Whale optimization algorithm based on chaotic search strategy. Control and Decision, 2019, 34(09):1893-1900.
[14] Long Wen, Wu Tiebin. Improved gray Wolf optimization algorithm for coordination of exploration and development capability. Control and Decision, 2017, 32(10):1749-1757.
[15] Mao Qinghua, Zhang Qiang, Mao Chengcheng, et al. Mixing Sine and Cosine Algorithm with Levy Flying Chaotic Sparrow Algorithm. Journal of Shanxi University (Natural Science Edition), 2021, 44(06):1086-1091