Ad Hoc Networks 150 (2023) 103284

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

A novel self-adaptive multi-strategy artificial bee colony algorithm for coverage optimization in wireless sensor networks ☆

Jin Wang [a], Ying Liu [a], Shuying Rao [a], Xinyu Zhou [b], Jinbin Hu [a,*]

[a] School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410004, China
[b] School of Computer and Information Engineering, Jiangxi Normal University, Nanchang 330022, China

ARTICLE INFO

Keywords:
Wireless sensor network
Coverage optimization
Multi-strategy
Self-adaptive
Artificial bee colony algorithm

ABSTRACT

Wireless sensor node coverage optimization is a critical issue in wireless sensor networks (WSN), which is a commonly typical NP-hard problem. To enhance the coverage of wireless sensor networks, coverage optimization refers to the prudent placement of resource-constrained wireless sensor nodes. Current coverage optimization techniques frequently result in local optimums and have poor optimization performance. Based on the excellent optimization performance of artificial bee colony (ABC) algorithm, this paper presents a novel self-adaptive multi-strategy artificial bee colony (SaMABC) algorithm, which designs an appropriate strategy pool and a fine-grained adaptive selection mechanism according to the coverage optimization problem. Furthermore, the algorithm is improved through using simulated annealing approach and the dynamic search step to enhance its ability to jump out of the local optimum. Compared with the state-of-the-art optimization algorithms, the evaluation results carried out in several scenarios show that SaMABC obtains the best performance in terms of coverage optimization. Specifically, the coverage of wireless sensor networks in SaMABC achieves around 99.1% and outperforms the initial coverage by up to 14.1%.

1. Introduction

Wireless sensor network (WSN) is a new computing and network model, which can be defined as a network composed of tiny, expensive and highly intelligent devices known sensor nodes [1,2]. WSN is a network structure made of several sensor nodes via wireless communication technology. The detection and monitoring of its main target areas have been widely used in industry, such as urban monitoring [3], environmental detection [4], military monitoring [5], mobile target tracking [6] and smart home [7]. However, sensor nodes have some limitations, such as high network cost and weak sensing range. To enhance WSN coverage, sensor node redundancy should be avoided while deploying sensor nodes.

A crucial issue in WSNs is the optimization of sensor node coverage, and the coverage has a substantial influence on the performance of the network [8]. Coverage Optimization aims to improve the network area that can be monitored with just a minimal number of sensors while reducing the amount of blind spots. Typically, sensor nodes are dispersed at random throughout the region that has to be detected. However, the high node density and redundancy caused by this random deployment technique result in low overall coverage, which further degrades the effectiveness of WSN monitoring. Therefore, it is essential to develop a practical sensor node deployment strategy that can not only increase the WSN's energy efficiency and service quality, but also achieve the load balancing of transmission inside the WSN [9–12].

The optimization of sensor node coverage is, in fact, a typical NP-hard problem because of the influence of network resources and coverage characteristics. Therefore, it is difficult to solve the problem using classical mathematical optimization techniques such as gradient descent method. In recent years, coverage problem of WSN has been studied by a large number of researchers, with the genetic algorithm (GA) [13], particle swarm optimization algorithm (PSO) [14], artificial bee colony algorithm (ABC) [15], and simulated annealing algorithm (SA) [16] being the most popular approaches. This type of optimization procedure is highly adaptable and has few restrictions for the problem's mathematical characteristics [17].

Even though some of the aforementioned heuristic algorithms have been successful at optimizing WSN coverage, in reality, all of them aim for an approximative optimal solution rather than the best possible one.

☆ This work is supported by the National Natural Science Foundation of China (62072056, 62102046, 61966019), and the Natural Science Foundation of Hunan Province, China (2022JJ30618, 2020JJ2029), the Hunan Provincial Key Research and Development Program, China (2022GK2019) , and the Scientific Research Fund of Hunan Provincial Education Department, China (22B0300).

* Corresponding author.
E-mail address: jinbinhu@csust.edu.cn (J. Hu).

https://doi.org/10.1016/j.adhoc.2023.103284
Received 27 November 2022; Received in revised form 15 July 2023; Accepted 20 August 2023
Available online 23 August 2023
1570-8705/

Table 1
Comparison among different state-of-the-art techniques.

Category	Method	Description
Model optimization	Binary sensing model [18]	The most typical, simple and ideal sensing model
	Probability sensing model [19]	A sophisticated model based on environmental awareness
Improved optimization algorithms	NP-complete [20]	Using polynomial calculation and NP-complete mathematical method to solve
	GND-DE [21]	Adopt DE algorithm to solve the dynamic deployment of sensor nodes
	IWHO [22]	Whale optimization-based dynamic deployment algorithm
	SA&GD [23]	Hybrid optimization mechanism combining SA and GD
	YYPIO [24]	Hybrid optimization mechanism combining YYPO and PIO
Improved ABC algorithms	GABC [25]	Guided by global optimal solution
	GBABC [26]	An improved ABC algorithm for gaussian bare bones
	ECABC [27]	Optimization based on elite group guidance
	ABCVSS [28]	Multi-strategy guidance of variable search mechanism
	NABC [29]	Guided based on best neighbor and global neighborhood search
	ABC-MNT [17]	Guided optimization based on multi-neighborhood topology
	FLABC [30]	Online fitness landscape based on optimization problems

These algorithms' search techniques are also overly greedy. It is challenging to arrive at a solution of greater quality while the iteration is in the middle and late phases since it is so simple to settle on the problem's local optimal solution. Because of this, this work suggests an adaptive multi-strategy ABC algorithm, called SaMABC, from the standpoint of how to leave the local optimum, improve population diversity, and quicken algorithm convergence. For the coverage optimization problem, we design an appropriate strategic pool and fine-grained adaptive selection mechanism. Simultaneously, since the optimization process is prone to falling into a local optimum, SaMABC guide the algorithm in jumping out of the local optimum through simulated annealing and dynamic search steps, resulting in a higher quality solution.

The main contributions of this paper are as follows:

- We model and analyze the coverage optimization problem in WSNs, and then transform it into a solvable objective function. We solve the transformed objective function with a meta-heuristic algorithm and iterate continuously to get the best solution.
- We propose a novel self-adaptive multi-strategy artificial bee colony (SaMABC) algorithm, which select multiple suitable search strategies based on the characteristics of the problem to establish a multi-strategy pool.
- By combining the concept of simulated annealing, altering the dynamic search step length and the variable threshold *limit*, SaMABC enhances the algorithm's global search ability, making it possible to jump out of the local optimum and reach a higher quality solution.
- We simulate SaMABC in five wireless sensor network scenarios and analyze the results. According to the simulation results, SaMABC can successfully decrease sensor node redundancy, boost coverage by around 14.1%, and improve the coverage of WSN nodes.

The structure of this paper is as follows. In Section 2, we introduce the related work. In Section 3, we describe WSN node coverage optimization problem and artificial bee colony algorithm, as well as the main motivation of this paper. In Section 4, we present SaMABC algorithm, and describe the multi-strategy mechanism and algorithm framework in detail, including the algorithm framework and multi-strategy mechanism. In Section 5, we show the performance evaluation and analysis, including the analysis of relevant comparison results. The work of this study is reviewed in Section 6.

2. Related work

Coverage optimization is always the most challenging application problem in WSNs. Many related research work have been proposed in recent years to enhance WSN coverage. It can be roughly divided into three categories: (1) Optimization of WSN coverage model, (2) Solve with improved optimization algorithms, and (3) Solutions with improved ABC algorithms. Table 1 is the summary of technological achievements of related work.

(1) Optimization of WSN coverage model

In WSNs, each sensor node has the ability to sense and monitor the area to be monitored. The monitoring capability of a node is closely related to its physical attributes and coverage model. A more precise coverage model can increase sensor node sensing quality and monitoring capabilities. Sensor nodes in the binary sensing model [18] only detect the points to be monitored within the sensing range. However, the binary sensing model does not consider some uncertain factors and is considered to be the simplest coverage model. With increasing distance, the sensing ability of sensor nodes declines. A more realistic probabilistic sensing model is proposed. Elfes sensing model [19] effectively reduces the uncertainty of sensor node sense by introducing the fluctuation value of sensing radius. In addition, considering various factors affecting sense, more precise probabilistic sensing models have been proposed [31,32].

(2) Solutions with improved optimization algorithms

The random deployment mechanism can lead to the problem of sensor node redundancy and low coverage. To increase coverage of the monitoring region and minimize monitoring blind spots, optimization techniques may be utilized to choose the best place for node deployment.

Cardei et al. [20] proposed to solve this problem by polynomial calculation and NP-complete mathematical method. However, it is difficult to find a better deployment mechanism by using traditional mathematical methods. Many scholars have proposed meta-heuristic algorithms to obtain an approximate ideal solution of this problem. Wang et al. [21] proposed an improved Differential evolution (GND-DE) based on population topology to solve the dynamic deployment problem of sensor nodes. Zeng et al. [22] proposed an improved Wild Horse Optimization Algorithm (IWHO) to solve coverage optimization problems by combining multiple optimization techniques.

Many hybrid optimization algorithms have been proposed in attempt to combine the benefits of distinguishable optimization algorithms. El Khamlichi et al. [23] proposed a hybrid deployment optimization mechanism combining simulated annealing and gradient algorithm to ensure high coverage under WSNs with a minimum number of sensor nodes. Simulated annealing algorithm can effectively maximize the coverage area of WSNs and the life of network devices. Similarly, Yin et al. [24] proposed a Yin–Yang pigeon-inspired optimization algorithm (Yin–YangPIO) that combines Yin-Yang-air optimization algorithm (YYPO) with pigeon-inspired optimization algorithm (PIO) to improve the optimal solution and assist node deployment to the optimal location.

(3) Solutions with improved ABC algorithms

Furthermore, many improved ABC algorithms have excellent optimization benefits on WSN coverage optimization. Some algorithms often use global optimal solution and the elite solution to guide algorithm optimization. Zhu et al. [25] proposed an improved algorithm

GABC based on the population's optimal individual guidance, including the Gbest term of the population optimal individual in the solution search equation and changing the search direction using Gbest's helpful information. However, the search direction of the new Gbest item is easily opposite to that of the random item, leading in oscillation of the overall direction and a reduction in the algorithm's search efficiency. As a result, Zhou et al. [26] proposed GBABC, an improved algorithm for Gaussian reduction that differs from GABC's direct addition of Gbest items. GBABC use Gaussian distribution to generate descendants in the search area defined by Gbest and the current individual, thus avoiding the phenomena of shock. Based on the concept of the elite, Kong et al. [27] proposed the improved algorithm ECABC directed by the elite group, which takes the center point of the elite group as the search starting point for solution search equation, so avoiding the precocious problem to some extent. However, because the elite group is built only on fitness, ECABC remains excessively greedy.

Others use adaptive search and global neighborhood search to optimize the problem depending on previous experience. Kiran et al. [28] proposed an enhanced variable search strategy algorithm ABCVSS, selected five various search equations to build a strategy candidate pool, and designed an adaptive strategy selection mechanism. In this mechanism, a counter is set for each strategy to record the number of nectar source that have been successfully updated, allowing the chance of each strategy being selected to be calculated. Peng et al. [29] proposed a modified neighborhood search algorithm NABC, which leverages the neighborhood search mechanism to conduct fine-grained searches in the neighborhood of the abandoned nectar source to enhance the probability of discovering the global optimal solution. Zhou et al. [17] presented an improved multi-neighborhood topology-based algorithm ABC-MNT. It may implement diverse capacities of spreading search information by taking into consideration the characteristics of different neighborhood topologies, which is favorable to better balancing exploration and development. Furthermore, Zhou et al. [30] proposed an improved algorithm FLABC based on fitness landscape by considering the structural characteristics of optimization problems.

However, while they have significantly improved WSN coverage, they are not the greatest solution. These algorithms are prone to falling into the problem's local optimization and converge prematurely in the optimization process, leading to a low accuracy of the final optimization results. The above work has cons and pros, prompting us to explore whether we can combine these advantages to create a new algorithm to satisfy the demands of high accuracy and fast convergence in solving coverage optimization problems.

3. Background and motivation

3.1. Background

3.1.1. Coverage of wireless sensor networks

Coverage is a basic problem that must be solved in the configuration of WSNs [33,34]. It depicts the variety of areas that WSNs can detect and monitor. The coverage optimization problem usually has some shortcomings, such as weak sensing range of sensor nodes, short working life, too dense node deployment, and many blind spots. Its common practical applications include area coverage and target coverage. Fig. 1 shows a simple area coverage diagram, where the square represents the detection area and the circle represents the sensing range of nodes. In brief, coverage optimization is the process of creating a superior deployment strategy. It deploys a limited number of sensor nodes to the area to be detected, which maximizes the area that can be detected by sensor nodes, that is, the coverage of the detection area.

In Fig. 1, the area within the square is the area to be covered, marked with blue shading. The goal of coverage optimization is to deploy these sensor nodes to maximize the coverage of this area and cover the blue shadow as much as possible. The blue shadow in Fig. 1 represents the coverage blind spots in WSN. Therefore, the goal of coverage optimization is to spread out sensor nodes as far as possible to cover the full detecting region while reducing coverage blind spots.

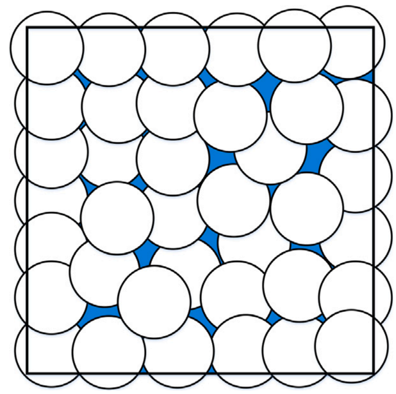


Fig. 1. Schematic diagram of WSN node coverage.

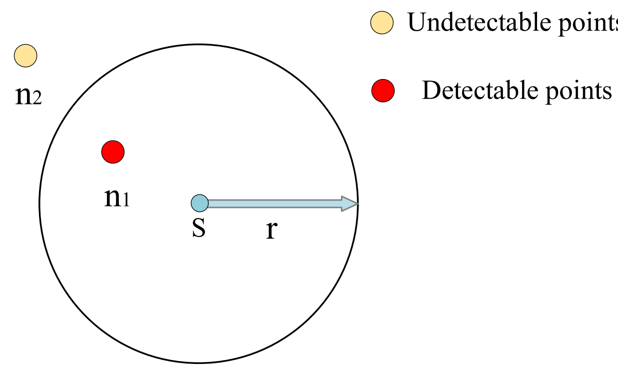


Fig. 2. Schematic diagram of binary sensing model.

Model building. The sensing model of sensor node is closely related to WSN coverage, which is the key to solve the problem of WSN coverage. The geometric relationship between sensor node and coverage area is determined by the calculation results of the sensing model. A precise sensing model can really simulate the actual working scene. The binary sensing model and the probability sensing model are the two fundamental sensor node awareness models used in the current study.

(1) Binary sensing model.

The binary sensing model is the most typical, simple and ideal sensing model in the current WSN coverage model [18]. This model refers to the circular monitoring area of sensor node S on a two-dimensional plane, with sensing radius r as the circular radius and sensor node S as the center. Fig. 2 illustrates two-dimensional plane diagram of the binary sensing model.

In Fig. 2, n1 and n2 are two random points in the plane region. Among them, n1 is within the monitoring area of this sensor, and the sensing probability is recorded as 1. While n2 exceeds the monitoring area of sensor S, the sensing probability of sensor S to point n2 is 0. For any given point $Q(x, y)$ of the two-dimensional plane, the mathematical equation of the Binary Sensing Model is as follows:

$$P_{S,Q} = \begin{cases} 1, & d(S,Q) \le r \\ 0, & d(S,Q) > r \end{cases} \tag{1}$$

where, $P_{S,Q}$ is the probability that sensor node S can detect monitoring point Q, r is the sensing radius of sensor node S. And $d(S,Q)$ is the Euclidean distance of monitoring point Q to sensor node S. The calculation method is as follows:

$$d(S,Q) = \sqrt{(S_x - Q_x)^2 + (S_y - Q_y)^2} \tag{2}$$

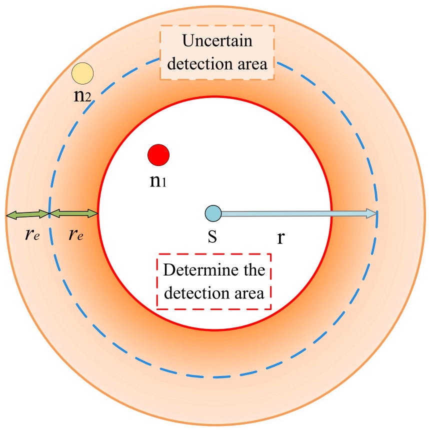


Fig. 3. Schematic diagram of probability sensing model.

where, S_x, S_y and Q_x, Q_y represents the horizontal and vertical coordinates of sensor node S and monitoring point Q in the two-dimensional plane respectively. The Binary Sensing Model can simplify the coverage problem and facilitate further research on the problem. As a result, this model was typically utilized in the early studies on the coverage issue in sensor networks. However, the binary sensing model does not account for environmental factors and so cannot accurately simulate actual application scenarios.

(2) Probability sensing model.

In practical application scenarios, the detecting range of sensor nodes is impacted by various unique environmental conditions. However, binary sensing model is too idealistic and does not consider the impact of these environmental factors. The results of experimental simulation are quite different from the actual results. Therefore, some scholars have modified the Binary Sensing Model. By combining the influence of environmental factors on sensors, a more realistic probability sensing model [19] is proposed. The model introduces the radius fluctuation value r_e. It is believed that the sensing range of sensor nodes can fluctuate with environmental factors. And its sensing probability can decline in a negative exponential trend with the increase of Euclidean distance between monitoring points and sensor nodes. The calculation equation of sensing probability of probability sensing model is as follows:

$$P_{S,Q} = \begin{cases} 1, & d(S,Q) \le r - r_e \\ e^{(-\alpha_1 \lambda_1^{\beta_2}/\lambda_2^{\beta_2} + \alpha_2)}, & r - r_e < d(S,Q) < r + r_e \\ 0, & d(S,Q) \ge r + r_e \end{cases} \tag{3}$$

where, r_e is the radius fluctuation value of the sensor node's uncertain detection capability, $\lambda_1 = r_e - r + d(S,Q)$, $\lambda_2 = r_e + r - d(S,Q)$, α_1, α_2, β_1, β_2 is the attenuation coefficient of the sensing probability, and is a fixed physical attribute of the sensor equipment. Fig. 3 illustrates the two-dimensional plane diagram of probability sensing model.

Fig. 3 shows that when a random point n1 in the monitoring area is located within the range $r - r_e$ from sensor node S, the probability that this monitoring point n1 can be monitored by sensor node S is 1. Therefore, the red area in the figure is the area that must be monitored by sensor node S. On the contrary, if the monitoring point is outside the range of $r + r_e$, the probability of being monitored is 0. In addition, in the area between $r - r_e$ and $r + r_e$, the probability of being monitored could decrease with the distance from sensor node S. In conclusion, the probability sensing model is more accurate for simulating and evaluating genuine industrial scenarios when integrated with the ambient aspects of sensor nodes.

Objective function construction. We choose the probability sensing model to calculate sensor node sensing probability in WSN and design an objective function to resolve WSN coverage in order to make the experimental results more accurate. Assume that D sensor nodes should be implemented in the monitoring region A with the size of $M \times N$. These sensor nodes have the same physical attributes, including the sensing radius r and the radius fluctuation value r_e. For the convenience of experimental calculation, all sensor nodes are regarded as particles whose volume and mass can be ignored. At the same time, the WSN monitoring area is simplified as a discrete two-dimensional plane area. It deploys the sensor node in the area A to be monitored, and $M \times N$ points are monitored. Therefore, an objective function can be constructed to calculate the coverage of area A to be monitored, and its function expression is as follows:

$$CR_A = \frac{|U_i{}^D P\{i\}|}{M \times N} \tag{4}$$

where, CR_A represents the coverage of monitoring area A, and $P\{i\}$ is the set of sensing probabilities of the points covered by sensor node i, $|U_i{}^D P\{i\}|$ indicates all points monitored by sensor nodes.

3.1.2. Artificial bee colony algorithm

Among various optimization algorithms, the artificial bee colony algorithm is a relatively innovative and effective optimization algorithm. Initially, ABC randomly generates nectar sources using Eq. (5):

$$x_{i,j} = x_j^{min} + rand(0,1) \cdot (x_j^{max} - x_j^{min}) \tag{5}$$

where, $x_{i,j} \in [x_j^{min}, x_j^{max}]$, x_j^{min}, x_j^{max} respectively represents the jth dimension boundary of the optimization problem, and $rand(0,1)$ is a uniform random number within the range of [0, 1]. The ABC optimization algorithm is divided into three phases: employed bee phase, onlooker bee phase and scout bee phase. The three phases are explained below.

Employed bee phase. The employed bees can explore for new nectar sources throughout the search space of optimization problem. Simultaneously, the position of the nectar source is updated by the search equation shown in Eq. (6).

$$v_{i,j} = x_{i,j} + \phi_{i,j} \cdot (x_{i,j} - x_{k,j}) \tag{6}$$

where, V_i is a new source of nectar, X_i is the original nectar source, $\phi_{i,j} \in [-1,1]$ is uniform random number, X_k is the nectar source randomly selected from the population, and $X_k \ne X_i$. Note that j is an arbitrarily selected dimension, and X_i and V_i is different only in this dimension. If V_i has more nectar than X_i, then V_i will replace X_i to enter the next iteration. Otherwise, X_i remains unchanged.

Onlooker bee phase. Onlooker bees will choose great nectar sources for exploitation based on the nectar quantity of the nectar source after obtaining the nectar source information shared by employed bees. It also searches for new nectar sources through using solution search equation shown in Eq. (6). The amount of nectar of the nectar source is the fitness value of the individual, which can be calculated according to the following Eq. (7):

$$fitness_i = \begin{cases} \frac{1}{1+f(X_i)}, & f(X_i) \ge 0 \\ 1 + |f(X_i)|, & f(X_i) < 0 \end{cases} \tag{7}$$

where, $fitness_i$ is the fitness value of the nectar source, $f(\cdot)$ is the objective function value. If the nectar amount of the nectar source is more, the probability of the nectar source being selected by the onlooker bees is higher. The selection probability can be calculated according to the following Eq. (8):

$$p_i = \frac{f(X_i)}{\sum_{j=1}^{SN} f(X_i)} \tag{8}$$

where, the selection probability of p_i as nectar source X_i. After obtaining the selection probability of all nectar sources, the onlooker bees will use the roulette mechanism to select.

Scout bee phase. When the nectar source associated with the employed bees is not updated after a set threshold $limit$, we think that the nectar source has been depleted. In this case, a new nectar source is randomly initialized through Eq. (5) to replace X_i.

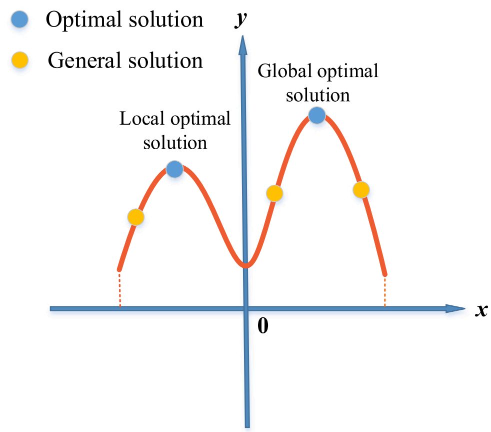


Fig. 4. Distribution of feasible solutions in optimization functions.

3.2. Motivation

In recent years, WSN has seen widely use as distributed environments like cloud computing and edge computing have grown. One of the most basic challenges of WSN is node coverage optimization, which is a NP-hard problem. Traditional deterministic techniques and algorithms cannot solve NP-hard problems in a reasonable computing time. In this case, it is better to use non-deterministic algorithms, such as meta-heuristic algorithms.

Some researches propose that PSO algorithm, ABC algorithm and other meta-heuristic algorithms have been proposed to address the WSN coverage optimization problem, which effectively optimizes the deployment mechanism of sensor nodes. Subsequently, some scholars proposed improved algorithms to further improve the coverage of sensors in WSN. Undoubtedly, these related works have played a positive role in the research of WSN coverage optimization. However, they all have many similar shortcomings.

Easy to fall into local optimal solution. The goal of WSN coverage optimization problem is to solve the optimal sensor node deployment mechanism to maximize the coverage in WSN, which is a maximum optimization problem. Nevertheless, coverage optimization problem is a non-convex optimization problem. The objective function has many maximum value, and the problem has many local optimal solutions. As a result, it might be challenging to achieve further optimization while tackling non-convex issues since it is possible to reach the problem's local optimum. It probably fall into the local optimal solution in the optimization solution and be unable to exit, leading to poor coverage of deployed WSN sensor nodes and numerous coverage blind spots. Fig. 4 shows the distribution of feasible solutions of a simple non-convex function, which has local optimal solutions. When the problem optimization solution reaches the local optimum, whether the search step size is too tiny or the search strategy is too aggressive, it is simple to approach the local optimum gradually. And then fall into the local optimal solution and it is difficult to jump out. When an algorithm is applied to an optimization problem and reaches the local optimum, it will experience problems including low accuracy and premature convergence.

Inadequate in-depth analysis of WSN coverage. Many improved algorithms to resolve the WSN coverage optimization problem, and they may certainly have an optimization consequence. However, this is only to verify the basis for its improved algorithms to solve practical problems. And these algorithms have no unique performance advantages in solving WSN coverage problems. In these related works, their main improvement perspective is to achieve better overall optimization performance on some standard test function sets, rather than specifically improving WSN coverage optimization. Therefore, it is difficult to design an optimization algorithm that conforms to the optimization problem without in-depth study and discussion of WSN coverage optimization problem and analysis of the characteristics of the optimization problem.

4. Our proposed SaMABC algorithm

In this section, we propose SaMABC, a self-adaptive multi-strategy artificial bee colony algorithm. The key to the design of SaMABC algorithm is to select an appropriate search strategy for WSN coverage optimization and build a strategies pool. At the same time, according to the principle of the working mechanism of ABC algorithm, the corresponding strategy selection mechanism is designed for different phases. In addition, WSN coverage optimization often has some shortcomings, such as easily falling into local optimal solution. As a result, we integrate a few pertinent optimization strategies to improve the algorithm's capacity to depart from the local optimum. The SaMABC algorithm's framework flow chart for solving optimization issues is shown in Fig. 5.

4.1. Multi-strategy pool

Each optimization problem includes different characteristics, such as unimodality and multimodality. Single-mode problems demand a strong local search capability, whereas multimodal functions call for both strong global and local search capabilities. Algorithms are often created with a balance between both local and global search capabilities, which correlate to the capacity for exploration and exploitation [35]. If the algorithm is optimized by a single strategy, it is challenging to have a strong exploration and exploitation capacity. However, multi-strategy mechanisms can use search strategies with various search capabilities. Through the complementarity of techniques, it establishes a balance between the algorithm's exploration and exploitation capabilities, enhancing algorithm performance.

When investigating the WSN coverage optimization problem, we find that it is a multimodal problem with many local optimal solutions. When these improved algorithms are optimized to the middle and late phase, there will be problems such as search stagnation and failure to further optimize the problem. The method has started to progressively converge in the middle and late phases of iteration, and it is challenging to leave the local optimum region. Therefore, this paper gives greater consideration to the search strategy that has higher global search capacity, when it designs which search strategy to pick to establish a multi-strategy pool.

The employed bee phase and the onlooker bee phase of the original ABC method update the solution using the same solution search Eq. (9). Where X_k is a random individual in the population, and $X_k \neq X_i$. Receive random individual X_k, the search direction of the algorithm has a strong randomness, leading to the algorithm has good exploration capabilities but poor exploitation flaws. In view of the strong exploration ability of random individuals in the search process, two random individuals X_k and X_t are introduced in Eqs. (10) and (11) at the same time. They broaden the diversity of individual learning and significantly improve the algorithm's capacity for global search [36].

$$V_i = X_i + \phi_{i,j} \cdot (X_i - X_k) \tag{9}$$

$$V_i = X_i + \phi_{i,j} \cdot (X_t - X_k) \tag{10}$$

$$V_i = X_k + \phi_{i,j} \cdot (X_t - X_k) \tag{11}$$

The above three solution search equations all have a wider search range by adding random individuals, so the algorithm has a very strong exploration capability. However, if the solution search strategy of the algorithm is biased towards exploration, it will lead to repeated cross jumps in the solution space, and it is difficult to find high-quality feasible solutions. Therefore, if the algorithm only has strong exploration

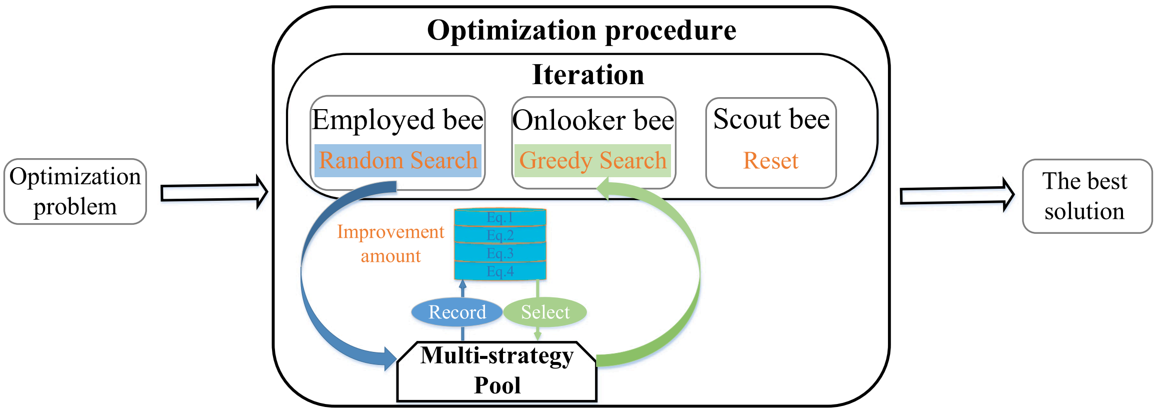


Fig. 5. Optimization frame diagram.

ability, but lacks the ability in exploitation, it is still challenging to obtain the optimal solution of the problem in multimodal functions.

The present optimal solution and elite solution are suggested to be used in many connected works in order to expand the exploitation capacity and improve the solution search equation. GBABC [24] proposes an improved Gaussian bare bones ABC algorithm. It uses Gaussian distribution to make use of the current optimal solution X_{best}, and update the solution in the search area formed by X_{best} and the current solution. The solution search equation effectively uses X_{best}, the algorithm can further search in the better region, and has good local search ability. The solution search equation proposed by GBABC is shown in Eq. (12).

$$V_i = gaussian(\frac{X_i + X_{best}}{2}, |X_i - X_{best}|) \tag{12}$$

where, $gaussian(\zeta_1, |\zeta_2|)$ is Gaussian distribution function, ζ_1 is the area center of Gaussian distribution, ζ_2 is the disturbance range.

We choose four alternative solution search equations to establish a multi-strategy pool in order to balance the algorithm's exploration and exploitation capacities, improve the search performance as a whole, and speed up the algorithm's convergence speed. In this multi-strategy pool, we select Eqs. (9), (10) and (11) solution search strategies with strong exploration capability and Eq. (12), which is more inclined to exploitation. In addition, we also designed the dynamic disturbance step parameter k to replace the original $\phi_{i,j}$. The multi-strategy pool built in this paper is as follows:

$$V_i = \begin{cases} X_i + k \cdot (X_i - X_k) \\ X_i + k \cdot (X_t - X_k) \\ X_k + k \cdot (X_t - X_k) \\ gaussian(\frac{X_i + X_{best}}{2}, |X_i - X_{best}|) \end{cases} \tag{13}$$

where, k is a variable coefficient that changes with the algorithm iteration, and $k \in [-1.5, 1.5]$, which will be described in detail in Section 4.3. And others follow the original literature.

4.2. Selection mechanism

In order to further balance the ABC algorithm's capabilities for exploration and exploitation, this paper proposes a variety of selection processes for the employed bee phase and onlooker bee phase in conjunction with its intrinsic mechanisms. To logically utilize the multi-strategy method and maximize the WSN coverage problem, this is done.

4.2.1. Random selection mechanism of employed bee phase

At employed bee phase, their main responsibility is to explore new nectar sources in the whole search space, and they should have better global search capabilities. For this reason, we designed a random selection method for this phase to use the above four solution search equations, that is, each employed bee randomly selects one solution search equation from the multi-strategy pool (13) and generate a new nectar source. This method can prevent the employed bees from over guiding a certain search strategy, balance the guiding role of many strategic mechanisms, and thus maintain the overall search ability of the employed bees.

4.2.2. Greedy selection mechanism of onlooker bee phase

At onlooker bee phase, the onlooker bees are mainly responsible for further exploit new nectar sources near better nectar sources, and should have better local search ability. Therefore, unlike the employed bee phase, the random selection method is no longer suitable for onlooker bees.We develop a greedy selection strategy for it, whereby onlooker bees are only permitted to apply the solution search equation that performs the best during the employed bee phase. To measure the optimization effect of search equations with various solutions, we utilize the index of fitness improvement, as given in Eq. (14):

$$\Delta_i = \begin{cases} fit(V_i) - fit(X_i), & fit(V_i) > fit(X_i) \\ 0, & fit(V_i) \le fit(X_i) \end{cases} \tag{14}$$

where, Δ_i represents the fitness improvement amount associated with the ith nectar source at employed bee phase, which is the particular optimization result of the associated solution search equation. The individual optimization impacts of each of the four solutions to the search equation can be determined, allowing the optimal solution to be chosen for the onlooker bee phase. This is done by adding up the fitness improvements amount of all nectar sources at employed bee phase. The greedy selection mechanism aids in maintaining a stronger local search capability throughout onlooker bee phase. It is worth noted that when the fitness improvement of all nectar sources is 0, none of the four options may make optimization effects. At this point, the onlooker bee will randomly select a solution search equation from multi-strategy pool and update the feasible solution.

4.3. Enhance the ability to jump from local optimum

Additionally, we propose using the concept of simulated annealing to aid the algorithm escape the local optimum in order to address the WSN coverage problem, which is simple to enter the local optimum. At the same time, variable search step size and dynamic threshold $limit$ are also established to improve the search performance of the algorithm.

The simulated annealing algorithm's basic idea is to randomly search for the objective function's global optimal solution in the solution space from a better position, paired with the probability jump properties. It will accept the viable solution that is worse than the present solution with a specific probability when it reaches the local optimum solution, and it may then leave the current local optimal zone.

Because of this, the simulated annealing process efficiently prevents slipping towards the local extreme value and eventually tends to the global ideal position. When updating the solution at employed bees and onlooker bees phase, we specify a certain probability p_m to accept a solution that is inferior to the existing solution. This is done using the principle of the simulated annealing method. Parameter p_m increases with the algorithm iteration, and its calculation equation is as follows:

$$p_m = 0.1 \cdot (\frac{FEs}{MaxFEs}) \tag{15}$$

where, FEs is the present fitness function's evaluation frequency and $MaxFEs$ is the fitness function's evaluation frequency at its maximum. In addition, the variable search step size k in the multi-strategy pool (15) also changes with the iteration of the algorithm, and its calculation equation is as follows:

$$k = \phi_s \cdot (\frac{FEs}{MaxFEs}) \tag{16}$$

where, ϕ_s is a random number with uniform distribution of [-1.5, 1.5].

At the scout bee phase, a nectar source is deemed to have been mined out at the scout bee phase if it has not been updated for a set amount of time $limit$. In order to prevent the algorithm from falling into search stagnation, reset is selected to break this possible search stagnation. However, it is extremely unreasonable to give up the experience of historical search without making effective use of it. In order to solve this problem, some related work has been proposed. MABC-NS [37] proposes a general reverse learning mechanism to generate the reverse solution of the abandoned nectar source, then selects the best from the reverse solution and the randomly generated new nectar source to retain. In order to increase the likelihood of discovering the global optimal solution, NABC [29] suggests a global neighborhood search method to carry out fine-grained search within the region of the abandoned nectar source. The global neighborhood search mechanism proposed by NABC is as follows:

$$TX_i = r_1 \cdot X_i + r_2 \cdot X_{best} + r_3 \cdot (X_j - X_k) \tag{17}$$

where, r_1, r_2 and r_3 is a random number in the interval [0, 1] and satisfies $r_1 + r_2 + r_3 = 1$. X_j and X_k is two random individuals in the population, $X_i \neq X_j \neq X_k$.

We adopt such excellent global search technology (17) to ensure population diversity, improve the algorithm's early exploration capability, and make sensible use of previous search experience. Generally, the preset threshold $limit$ for triggering this mechanism are 50 [38], 100 [39], and 200 [40]. Obviously, this fixed threshold $limit$ setting method is difficult to satisfy the demands of different optimization states to some extent. In the early phase of iteration, the algorithm should have a strong global search capability, and a small $limit$ setting can increase the trigger frequency of the mechanism. The local search ability of the problem should be favored in the middle and latter stages of the iteration, and the threshold $limit$ has to be set too high to lower the trigger frequency of the mechanism. To this end, this paper dynamically adjusts the parameter $limit$ to make it increase with iteration. The setting expression of the parameter $limit$ value is as follows:

$$limit = 200 \cdot (\frac{FEs}{MaxFEs}) \tag{18}$$

It is worth noting that the lower $limit$ cannot be less than 20. By doing so, the algorithm's ability to balance exploration and exploitation may be preserved, and its search performance can be improved.

4.4. Algorithm pseudocode

To more clearly explain the optimization process of SaMABC, algorithm 1 provides a pseudocode description. The number of nectar sources is SN, and the evaluation intervals of the consumed fitness function are FEs. And $MaxFEs$ is the preset fitness function's maximum evaluation period, which also serves as the algorithm's shutdown requirement. The best answer to the optimization problem's final output is denoted by the symbol P_{best}.

Algorithm 1: SaMABC algorithm

Input: The population size SN, the Maximum evaluation times $MaxFEs$
Output: Best solution P_{best}
Randomly generate SN food sources as the initial population according to the Eq.(5);
Evaluate each food source and set $FEs = SN$;
while $FEs < MaxFEs$ **do**
  /*Employed bee phase*/;
  for $i = 1$ *to* SN **do**
    Generate solution V_i by Randomly selecting from Eq. (13);
    Set $FEs = FEs + 1$;
    if $f(V_i) > f(X_i)$ **then**
      $X_i = V_i$ and set $trial_i = 0$;
      Record Δ_i for the selected solution search equation;
    else
      Accept inferior solutions with a certain probability p_m;
      Set $trial_i = trial_i + 1$;
    end
  end
  /*Onlooker bee phase*/;
  Keep the solution search equation with the best performance in the onlooker bee phase;
  for $i = 1$ *to* SN **do**
    Generate the new solution V_i for the old solution X_i based on the best performing solution search equation in Eq.(13);
    Set $FEs = FEs + 1$;
    if $f(V_i) > f(X_i)$ **then**
      $X_i = V_i$ and set $trial_i = 0$;
    else
      Accept inferior solution with a certain probability p_m;
      Set $trial_i = trial_i + 1$;
    end
  end
  /*Scout bee phase*/;
  for $i = 1$ *to* SN **do**
    if $trial_i > limit$ **then**
      Replace X_i with the new solution TX_i generated by the Eq.(17);
      Set $trial_i$=0 and $FEs = FEs + 1$;
    end
  end
end
return P_{best}

5. Performance evaluation and analysis

We choose five distinct WSN coverage situations for the simulation experiment in order to test the optimization performance of the algorithm SaMABC presented in this paper. The scenario parameter settings for each scenario are listed in Table 2. Additionally, the superior performance of the self-adaptive multi-strategy mechanism in SaMABC is confirmed by the effectiveness analysis of the algorithm's improvement points. In the experiment, we selected seven advanced optimization algorithms for comparative analysis with SaMABC, the descriptions of these algorithms are shown in Table 3.

The aforementioned algorithms used in the experimental comparison have excellent performance in solving optimization problems. Moreover, they have a certain correlation with the research content

Table 2
Parameter settings in WSN scenarios.

Parameter	40 × 40	50 × 50	100 × 100
D	30	40	50
r	4 (m)	5 (m)	10 (m)
r_e	2.5	2.5	5
α_1	1	1	1
α_2	0	0	0
β_1	1	1	1
β_2	1.5	1.5	1.5

Table 3
Description of optimization algorithm participating in experimental comparison.

Algorithm	Description
PSO [14]	Particle swarm optimization algorithm based on flight behavior of simulated birds
ABC [15]	Artificial bee colony algorithm based on simulated bee colony for intelligent nectar collection
GABC [25]	An improved ABC algorithm guided by global optimal solution
GBABC [26]	An improved ABC algorithm for gaussian bare bones
ECABC [27]	An improved ABC algorithm based on elite solution guidance
ABCVSS [28]	An improved multi-strategy ABC algorithm with variable search
NABC [29]	An improved ABC algorithm based on best neighbor and global neighborhood search
ABC-MNT [17]	Artificial bee colony algorithm based on multiple neighborhood topologies

Table 4
Optimized coverage of nine algorithms on 30 nodes.

Algorithm	Initial	Best	Worst	Mean	Std
PSO	65.13%	76.25%	70.00%	73.45%	0.0161
ABC	65.13%	79.56%	77.56%	78.35%	0.0045
GABC	64.63%	79.44%	77.69%	78.48%	0.0044
GBABC	**66.06%**	80.44%	77.81%	78.93%	0.0078
ECABC	64.75%	78.50%	76.50%	77.69%	0.0052
ABCVSS	64.69%	79.31%	77.75%	78.45%	0.0044
NABC	64.88%	79.38%	76.69%	77.54%	0.0056
ABC-MNT	65.68%	81.43%	79.37%	80.50%	0.0062
SaMABC	65.06%	**85.38%**	**84.81%**	**85.00%**	**0.0022**

of this paper. In order to prevent the unpredictability of experimental data and lessen the influence of random mistakes, all algorithms are independently run for 30 times to take their average results.

5.1. WSN coverage optimization in 40 m × 40 m scenario

In this experiment, the monitoring area is in a 40 m × 40 m square area. There are 30 sensors in all, and each one has a 4 m sensing radius r. Initially, 30 nodes are distributed at random in the detection region, and then the optimal deployment location of each sensor is solved using SaMABC and seven comparison algorithms. Table 4 displays the coverage optimization outcomes for the nine algorithms after 30 separate runs, with the ideal results shown in bold. As can be seen from Table 4, the initial coverage of the nine algorithms is basically the same, about 65%. However, the SaMABC finally achieves the maximum average coverage, increasing the average coverage from 65.06% to 85.25%. The coverage of the two original algorithms reached 77.31% and 78.37% respectively. The five revised ABC algorithms have nearly identical end coverage. The final coverage optimization of SaMABC is 5.3% higher than the best performance GBABC in the improved ABC algorithm, which shows that SaMABC has excellent performance in solving WSN coverage problems. At the same time, among the coverage

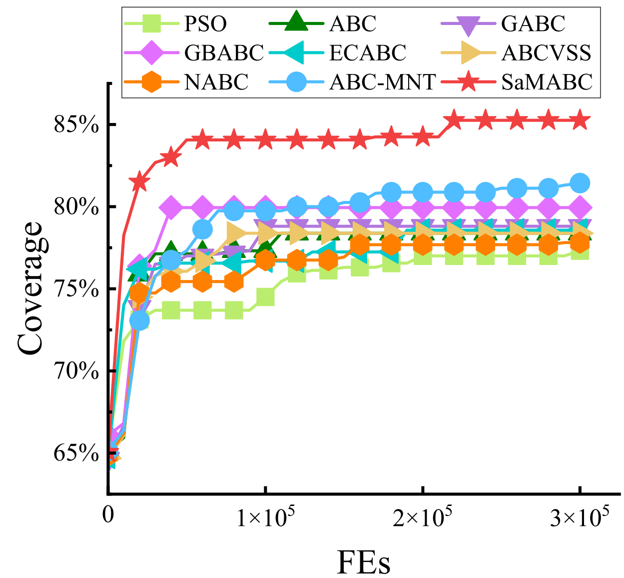


Fig. 6. Convergence curve of covering optimization of different algorithms on 30 nodes.

optimization results of the eight optimization algorithms, no matter the best result, worst result and variance, SaMABC is all the best.

In order to further examine the performance variances of various algorithms, this paper also provides the convergence curve of the different algorithms shown in Fig. 6. It can be seen that SaMABC has achieved very high coverage in the early phase, and the convergence speed is also the fastest among all algorithms. Compared with GBABC, SaMABC displays a significantly improvement in coverage over time, demonstrating the algorithm's strong capacity to depart from local optimums. At the same time, ABCVSS is also an adaptive multi-strategy improvement algorithm. However, SaMABC, which effectively demonstrates the efficacy of the improvement points suggested in this paper, has a far better optimization impact than it does.

5.2. WSN coverage optimization in 50 m × 50 m scenario

In this experiment, the monitoring area is in a 50 m × 50 m square area. There are 40 sensors in all, and each one has a 5 m sensing radius r. Initially, 40 nodes are distributed at random in the detection region, and then the optimal deployment location of each sensor is solved using SaMABC and seven comparison algorithms. Table 5 displays the coverage optimization outcomes of the nine algorithms after 30 separate runs, with the best results highlighted in bold. As can be seen from Table 5, the initial coverage of the nine algorithms is basically the same, about 75%. Similarly, SaMABC finally achieved the maximum average coverage, increasing the average coverage from 74.64% to 95.46%, and the coverage of the monitoring area increased by 20.82%. At the same time, compared with the GBABC with the best optimization effect in the comparison algorithm, the coverage of SaMABC is 5.24% higher than that of the original ABC. At the same time, based on the variation of 30 runs in the experimental data, the optimization performance of SaMABC is quite consistent.

We also provide the convergence curve of the algorithm depicted in Fig. 7 to further examine the performance variances of various algorithms. It can be seen that the SaMABC algorithm has achieved very high coverage in the early phase, and the convergence speed is also the fastest among all algorithms. Similar improvements in coverage are also shown in the middle and late phases of SaMABC, further demonstrating the algorithm's potent capacity to break out of local optimization.

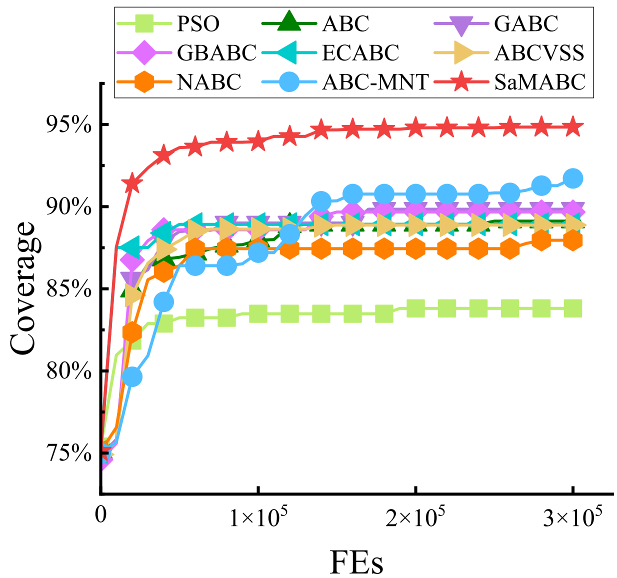


Fig. 7. Convergence curve of covering optimization of different algorithms on 40 nodes.

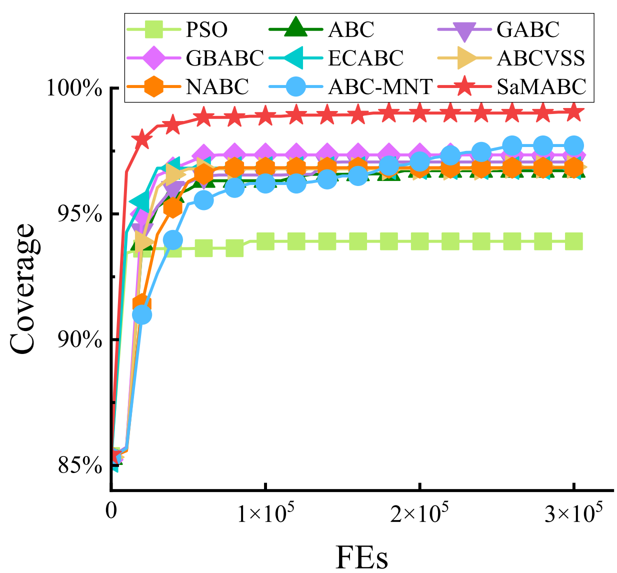


Fig. 8. Convergence curve of covering optimization of different algorithms on 50 nodes.

Table 5
Optimized coverage of nine algorithms on 40 nodes.

Algorithm	Initial	Best	Worst	Mean	Std
PSO	74.92%	86.92%	81.72%	84.52%	0.0131
ABC	74.96%	90.20%	88.60%	88.20%	0.0034
GABC	**75.44%**	90.12%	88.56%	89.25%	0.0043
GBABC	75.24%	90.64%	88.20%	89.27%	0.0057
ECABC	75.16%	89.40%	87.80%	88.64%	0.0043
ABCVSS	75.24%	89.72%	88.40%	89.11%	0.0033
NABC	75.04%	89.80%	87.40%	88.60%	0.0055
ABC-MNT	74.96%	91.72%	89.00%	90.49%	0.0064
SaMABC	74.64%	**95.48%**	**94.52%**	**94.98%**	**0.0027**

5.3. WSN coverage optimization in 100 m × 100 m scenario

In this experiment, the monitoring area is in a 100 m × 100 m square area. There are 50 sensors in all, and each one has a 10 m sensing radius r. As shown in Table 6, the initial coverage of the nine

Table 6
Optimized coverage of nine algorithms on 50 nodes.

Algorithm	Initial	Best	Worst	Mean	Std
PSO	85.37%	94.69%	91.63%	93.39%	0.0075
ABC	85.30%	97.31%	96.62%	96.96%	0.0019
GABC	85.34%	97.33%	96.66%	96.89%	0.0015
GBABC	85.23%	97.44%	96.51%	96.97%	0.0023
ECABC	85.17%	97.06%	96.24%	96.58%	0.0020
ABCVSS	85.33%	97.21%	96.64%	96.91%	0.0017
NABC	85.34%	97.11%	96.41%	96.65%	0.0016
ABC-MNT	85.38%	97.72%	95.67%	97.06%	0.0050
SaMABC	**85.42%**	**99.05%**	**98.90%**	**98.96%**	**0.0005**

Table 7
Parameter settings in two larger WSN scenarios.

Parameter	250 × 250	500 × 500
D	60	70
r	25 (m)	50 (m)
r_e	10	20
α_1	1	1
α_2	0	0
β_1	1	1
β_2	1.5	1.5

algorithms is basically the same, about 85%. The results indicate that SaMABC finally reaches the maximum average coverage rate, raising the average coverage rate from the original 85% to 99.05%, and that the coverage rate of the monitoring area grew by 14.05%.

To further compare the performance differences of different algorithms, the convergence curve of the algorithm shown in Fig. 8 is also given. It can be seen that the coverage optimization of the algorithm SaMABC has reached 99% in 50 000 evaluations, with the highest solution accuracy and the fastest convergence speed. In addition, Fig. 8 also displays the original sensor nodes deployment locations as well as the final deployment diagrams following the optimization of various algorithms. As can be seen from the deployment diagram in Fig. 9, SaMABC has more uniform node deployment and a greater detection coverage area than other comparative algorithms. Although the SaMABC has a small uncovered area, in fact, the nearby sensor nodes can perceive the area, showing the optimal deployment scheme for coverage. The results of three scenarios demonstrate that SaMABC is very competitive in WSN coverage and has excellent performance.

5.4. WSN coverage optimization in two larger scenarios

We simulated and analyzed the performance of the proposed algorithm SaMABC in three classic WSN scenarios in the preceding section, which effectively showed that SaMABC has excellent optimization performance in coverage optimization. We implemented extended simulation experiments in two large WSN scenarios, 250 m × 250 m and 500 m × 500 m, respectively, to demonstrate that SaMABC can also adapt to more complex WSN scenarios. Table 7 shows the parameter settings for two large sceneries, similarly as Table 2. Tables 7 and 8 display the coverage optimization results of nine algorithms after 30 runs in the 250 m × 250 m and 500 m × 500 m scenarios, respectively, with the optimal results highlighted in bold.

Tables 8 and 9 show that the initial coverage of the nine algorithms is roughly the same, at about 85.5% and 90.1%, respectively. SaMABC finally achieved maximum average coverage, improving average coverage from 85.76% to 99.80% and 90.12% to 99.80%. Overall, it demonstrates that SaMABC continues to overperform in solving large WSN coverage issues. In addition, SaMABC outperforms the other eight optimization algorithms in terms of coverage optimization results, regardless of the best, worst, or variance.

Figs. 10 and 11 demonstrate the convergence curves optimized by different algorithms in two large scenarios to further explore the performance differences of various algorithms. It is worth noting that

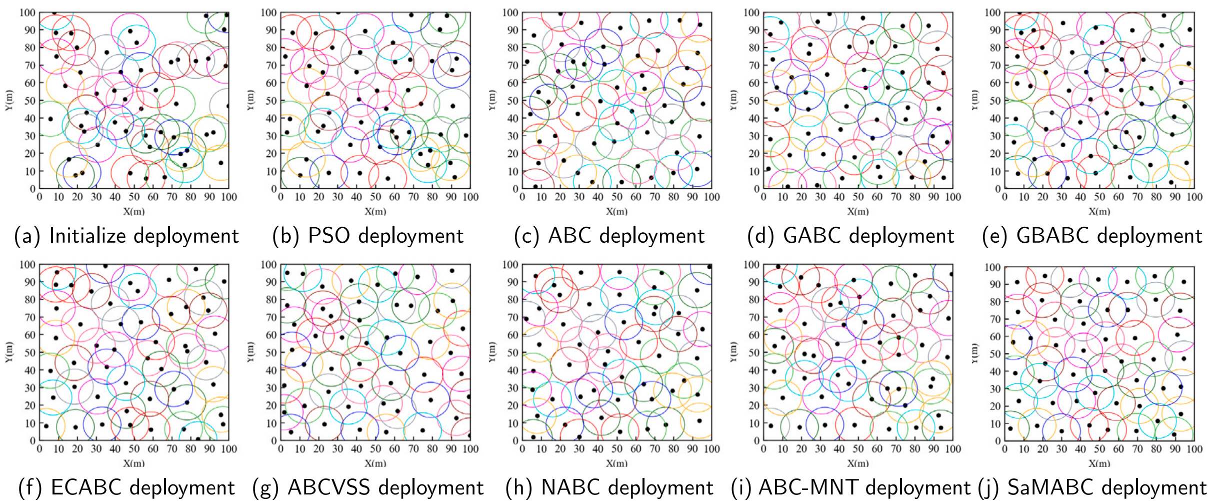


Fig. 9. Sensor node deployment diagram of 50 nodes optimized by different algorithms.

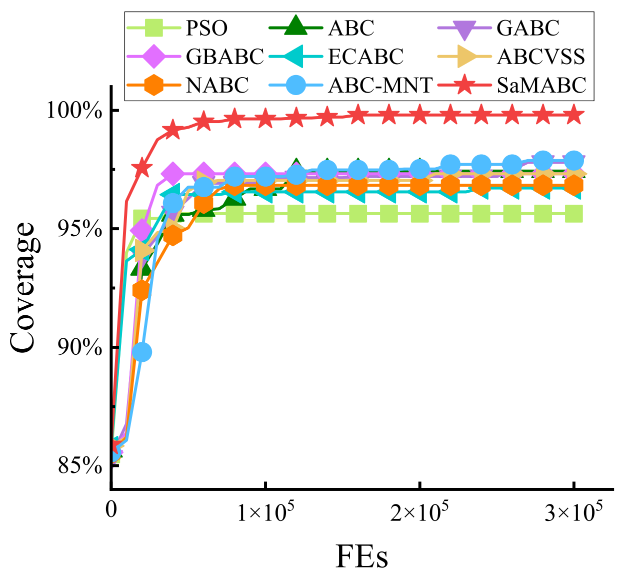


Fig. 10. Convergence curve of covering optimization of different algorithms on 60 nodes.

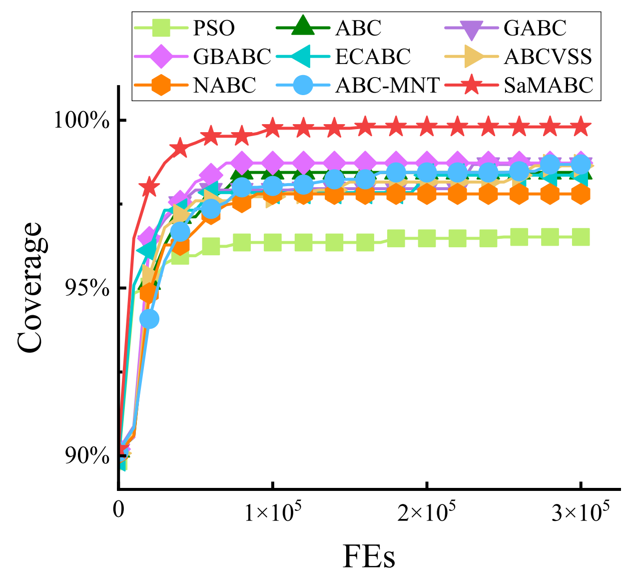


Fig. 11. Convergence curve of covering optimization of different algorithms on 70 nodes.

in the WSN coverage optimization problem, the number of sensors D corresponds to the dimension of the optimization problem. As a consequence, in these two large-scale WSN scenarios, coverage optimization has become a high-dimensional optimization problem, making the optimization algorithm extremely difficult to solve. However, as shown in Figs. 10 and 11, SaMABC may still reach very high coverage in the early stages, and its convergence rate is the fastest of all algorithms. In summary, even as the problem dimension grows, SaMABC can maintain excellent optimization performance, high solution accuracy, and quick convergence rate.

Table 8
Optimized coverage of nine algorithms on 60 nodes.

Algorithm	Initial	Best	Worst	Mean	Std
PSO	85.44%	95.64%	90.52%	93.08%	0.0149
ABC	85.68%	97.44%	96.60%	97.06%	0.0028
GABC	85.64%	97.80%	96.84%	97.27%	0.0035
GBABC	**85.96%**	97.32%	96.60%	96.85%	0.0025
ECABC	85.80%	96.72%	96.32%	96.58%	0.0012
ABCVSS	85.64%	97.32%	96.76%	97.12%	0.0180
NABC	85.52%	96.84%	95.84%	96.44%	0.0035
ABC-MNT	85.52%	97.88%	96.76%	97.46%	0.0035
SaMABC	85.76%	**99.80%**	**99.56%**	**99.71%**	**0.0006**

5.5. Algorithm verification

In order to validate the efficacy of SaMABC, the section conducts ablation experiments on the multi-strategy mechanism of SaMABC under the scenario of 40 sensor nodes. Four comparison algorithms are designed using four solution search equations of the multi-strategy pool (13), which are ABC-1, ABC-2, ABC-3 and ABC-4. The experimental results for SaMABC and the four comparator algorithms are displayed in Table 5. Additionally, the bolded results are the best ones.

From the experimental results in Table 10, four separate search strategy algorithms have the same performance in solving the WSN coverage problem, and the final coverage optimization results are about 89%. In order to better demonstrate the importance of various search strategies in the optimization process, we provide the number of successful improvements and the proportion of successful improvements of each search strategy in resolving the WSN coverage problem. Fig. 12

Table 9
Optimized coverage of nine algorithms on 70 nodes.

Algorithm	Initial	Best	Worst	Mean	Std
PSO	90.04%	96.52%	94.44%	95.20%	0.0059
ABC	**90.56%**	98.44%	97.72%	98.09%	0.0024
GABC	90.16%	98.68%	97.88%	98.18%	0.0022
GBABC	90.20%	98.72%	97.76%	98.20%	0.0031
ECABC	90.16%	98.36%	97.56%	97.94%	0.0022
ABCVSS	90.12%	98.64%	97.96%	98.19%	0.0018
NABC	90.08%	97.80%	96.92%	97.37%	0.0026
ABC-MNT	90.08%	98.68%	90.92	96.62%	0.0301
SaMABC	90.12%	**99.80%**	**99.72%**	**99.75%**	**0.0002**

Table 10
Optimization comparison of five algorithms on WSN coverage.

Algorithm	Initial	Best	Worst	Mean	Std
ABC-1	**75.00%**	89.84%	88.60%	89.13%	0.0037
ABC-2	74.76%	90.08%	88.56%	89.23%	0.0043
ABC-3	74.96%	89.60%	88.44%	89.04%	0.0028
ABC-4	**75.00%**	90.08%	88.56%	89.41%	0.0034
SaMABC	74.64%	**95.48%**	**94.52%**	**94.98%**	**0.0027**

displays the contribution points for the four search techniques. The improvement times of the four search algorithms account for around 25%, with minimal variation, as shown in Fig. 12(a). The search strategy Eq. (2) makes twice as big of a contribution to issue optimization as the other three search strategies, as shown in Fig. 12(b). We find that the optimization effect of Eq. (2) is not as good as that of Eq. (1) in a single strategy, but Eq. (2) shows excellent performance in a multi-strategy mechanism. This is mostly due to the adaptive multi-strategy mechanism presented in this research, which enhances algorithmic performance by balancing the advantages and disadvantages of each search strategy.

Fig. 13 illustrates the SaMABC algorithm, which shows the best optimization performance and the fastest convergence speed. The search strategies of ABC-1, ABC-2, and ABC-3 all have strong global search capabilities, but the final node deployment coverage is less than 90%. This is because they are still trapped in the local optimization of the WSN coverage problem and cannot get out of the way to get a better deployment scheme. Similarly, although ABC-4 has strong local search capability, its coverage optimization effect is also poor. SaMABC complements the benefits and drawbacks of the four search techniques to combine their good performance and provide effective optimization performance. In general, SaMABC shows excellent performance in solving WSN coverage optimization problems.

5.6. Discussion

According to the comparison results with the eight related improved ABC, SaMABC's performance is very competitive, not only in terms of convergence accuracy, but also in terms of convergence speed. Furthermore, we evaluate SaMABC's running time, recording its real CPU running time (in seconds) in the 50 m × 50 m scenario, and comparing it to the eight related improved ABC outlined above. To ensure a fair comparison, each algorithm is run 30 times independently, with the average CPU running time being the final result. The algorithm operating platform's setup information is as follows: CPU: Intel (R) Core i7-9750H, RAM: 16 GB, OS: Microsoft Windows 10 Professional, and programming language: Java.

Table 11 displays the algorithm's CPU running time. The second column shows the algorithm's overall average CPU running time for 30 times, and the last column shows the variation of the CPU running time for 30 times. As can be seen, SaMABC has the shortest average running time. Although ABC-MNT has the shortest completion time,

Table 11
Comparison result of CPU running time with relevant improved ABC algorithm (unit: second)

Algorithm	Best	Worst	Mean	Std
PSO	239.3	270.4	252.9	10.1289
ABC	236.1	273.0	254.4	10.5257
GABC	226.7	277.9	254.6	11.2831
GBABC	225.7	278.2	254.4	11.4595
ECABC	218.4	277.4	244.2	9.2379
NABC	225.3	273.1	234.847	8.7868
ABC-MNT	**182.6**	276.1	225.173	31.2147
SaMABC	185.5	**199.1**	**189.6**	**3.5402**

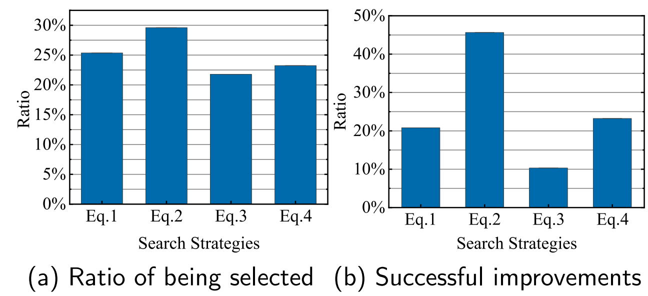


Fig. 12. Contribution of search strategy in SaMABC.

the variation indicates that the ABC-MNT optimization technique is not stable enough. SaMABC not only has the shortest average completion time, but also the smallest variance. However, it should be noted that when the number of evaluation times FEs of SaMABC in Fig. 8 reaches 50 000, the coverage rate has reached 99%, and its optimization value is very near to the final optimization result of 99.05%, as shown in the convergence diagram of algorithm optimization. We set a big number of evaluation times to see if the algorithm may be further optimized later on. Figs. 6 and 7 show that SaMABC can continue to generate optimization in the middle and later stages of the iteration, due largely to the newly designed simulated annealing strategy and global neighborhood search mechanism in SaMABC, which effectively helps the algorithm jump out of local optimization, resulting in a higher quality feasible solution. SaMABC illustrates clear advantages in convergence and solution quality, and its overall performance remains competitive.

6. Conclusion

The problem of wireless sensor coverage is easy to slip into local optimization and has poor coverage optimization impact. We provide SaMABC, a revolutionary self-adaptive multi-strategy artificial bee colony method, as a solution to this issue. In this algorithm, we choose the appropriate search strategy to build the multi-strategy pool according to the characteristics of the wireless sensor coverage problem. And we design different strategy selection mechanisms according to the working principle of ABC algorithm. At the same time, SaMABC combines the idea of simulated annealing and dynamic parameters that vary with the degree of optimization to help the algorithm escape the local optimization. We assess the performance of the enhanced strategy in the experimental simulation and compare our algorithm to the conventional PSO, ABC, and six superbly upgraded ABC methods. The results show that the coverage of wireless sensor networks in SaMABC achieves around 99.1% and outperforms the initial coverage by up to 14.1%. In the larger WSN scenarios, SaMABC still maintains excellent solution accuracy and convergence speed. In future work, we will combine other optimization techniques to improve the performance of the algorithm.

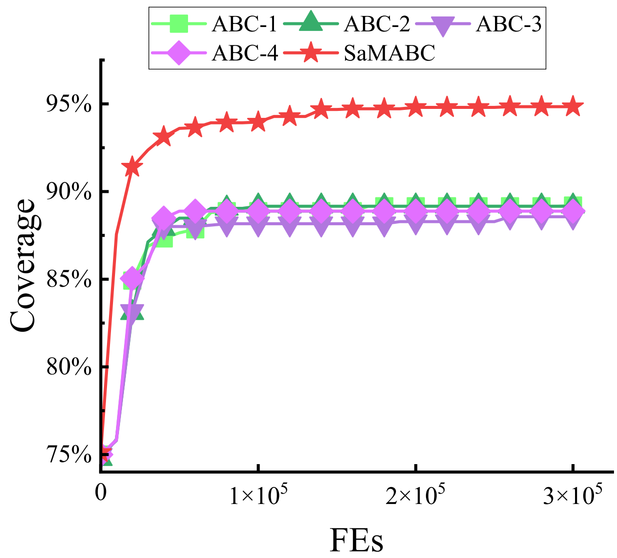


Fig. 13. Convergence curves of coverage optimization of five algorithms on 40 nodes.

CRediT authorship contribution statement

Jin Wang: Analysis and manuscript preparation. **Ying Liu:** Experiment, Data analyses, Wrote the manuscript. **Shuying Rao:** Performed the algorithm design. **Xinyu Zhou:** Provided the code of the experiment, Contributed to the conception of the study. **Jinbin Hu:** Helped perform the analysis with constructive discussions.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jinbin Hu reports financial support was provided by the National Natural Science Foundation of China (62102046, 62072056), and the Natural Science Foundation of Hunan Province (2022JJ30618, 2020JJ2029). Jinbin Hu reports a relationship with by the National Natural Science Foundation of China (62102046, 62072056), and the Natural Science Foundation of Hunan Province (2022JJ30618, 2020JJ2029). that includes: funding grants.

Data availability

No data was used for the research described in the article.

References

[1] S. Qu, L. Zhao, Z. Xiong, Cross-layer congestion control of wireless sensor networks based on fuzzy sliding mode control, Neural Comput. Appl. 32 (17) (2020) 13505–13520.

[2] D.D.K. Rathinam, D. Surendran, A. Shilpa, A.S. Grace, J. Sherin, Modern agriculture using wireless sensor network (WSN), in: 2019 5th International Conference on Advanced Computing & Communication Systems, ICACCS, 2019, pp. 515–519.

[3] B.D. Deebak, F. Al-Turjman, A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks, Ad Hoc Netw. 97 (2020) 102022.

[4] A. Singh, D. Kumar, J. Hötzel, IoT Based information and communication system for enhancing underground mines safety and productivity: Genesis, taxonomy and open issues, Ad Hoc Netw. 78 (2018) 115–129.

[5] A. Boukerche, P. Sun, Connectivity and coverage based protocols for wireless sensor networks, Ad Hoc Netw. 80 (2018) 54–69.

[6] H. Ahmadi, F. Viani, R. Bouallegue, An accurate prediction method for moving target localization and tracking in wireless sensor networks, Ad Hoc Netw. 70 (2018) 14–22.

[7] G. Sun, Y. Liu, M. Yang, A. Wang, S. Liang, Y. Zhang, Coverage optimization of VLC in smart homes based on improved cuckoo search algorithm, Comput. Netw. 116 (2017) 63–78.

[8] J. Wang, Y. Gao, X. Yin, F. Li, H. Kim, An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks, Wirel. Commun. Mob. Comput. 2018 (2018) 9472075.

[9] J. Hu, J. Huang, W. Lv, Y. Zhou, J. Wang, T. He, CAPS: Coding-based adaptive packet spraying to reduce flow completion time in data center, IEEE/ACM Trans. Netw. 27 (6) (2019) 2338–2353.

[10] J. Hu, J. Huang, W. Lv, W. Li, Z. Li, W. Jiang, J. Wang, T. He, Adjusting switching granularity of load balancing for heterogeneous datacenter traffic, IEEE/ACM Trans. Netw. 29 (5) (2021) 2367–2384.

[11] J. Wang, H. Han, H. Li, S. He, P. Kumar Sharma, L. Chen, Multiple strategies differential privacy on sparse tensor factorization for network traffic analysis in 5G, IEEE Trans. Ind. Inform. 18 (3) (2022) 1939–1948.

[12] J. Hu, J. Huang, Z. Li, J. Wang, T. He, A receiver-driven transport protocol with high link utilization using anti-ECN marking in data center networks, IEEE Trans. Netw. Serv. Manag. 20 (2) (2023) 1898–1912.

[13] A.K. Idrees, W.L. Al-Yaseen, Distributed genetic algorithm for lifetime coverage optimisation in wireless sensor networks, Int. J. Adv. Intell. Paradig. 18 (1) (2021) 3–24.

[14] J. Wang, Y. Cao, B. Li, H. Kim, S. Lee, Particle swarm optimization based clustering algorithm with mobile sink for WSNs, Future Gener. Comput. Syst. 76 (2017) 452–457.

[15] X. Zhou, Y. Wu, M. Zhong, M. Wang, Artificial bee colony algorithm based on adaptive neighborhood topologies, Inform. Sci. 610 (2022) 1078–1101.

[16] Y. Zhang, L. Cao, Y. Yue, Y. Cai, B. Hang, A novel coverage optimization strategy based on grey wolf algorithm optimized by simulated annealing for wireless sensor networks, Comput. Intell. Neurosci. 2021 (2021) 1–14.

[17] X. Zhou, Y. Wu, M. Zhong, M. Wang, Artificial bee colony algorithm based on multiple neighborhood topologies, Appl. Soft Comput. 111 (2021) 107697.

[18] P. Soreanu, Z. Volkovich, New sensing model for wireless sensor networks, Int. J. Adv. Netw. Serv. 2 (4) (2009) 261–272.

[19] Y. Wang, Y. Zhang, J. Liu, R. Bhandari, Coverage, connectivity, and deployment in wireless sensor networks, in: Recent Development in Wireless Sensor and Ad-Hoc Networks, 2014, pp. 25–44.

[20] M. Cardei, D. MacCallum, M.X. Cheng, M. Min, X. Jia, D. Li, D.Z. Du, Wireless sensor networks with energy efficient organization, J. Interconnect. Netw. 3 (4) (2002) 213–229.

[21] Y. Wang, Y. Peng, L. Chen, Y. Duan, J. Li, WSN node coverage optimization algorithm based on global and neighborhood difference DE, China Commun. 19 (3) (2022) 215–229.

[22] C. Zeng, T. Qin, W. Tan, C. Lin, Z. Zhu, J. Yang, S. Yuan, Coverage optimization of heterogeneous wireless sensor network based on improved wild horse optimizer, Biomimetics 8 (1) (2023) 70.

[23] Y.E. Khamlichi, A. Tahiri, A. Abtoy, I. Medina-Bulo, F. Palomo-Lozano, A hybrid algorithm for optimal wireless sensor network deployment with the minimum number of sensor nodes, Algorithms 10 (3) (2017) 80.

[24] J. Yin, N. Deng, J. Zhang, Wireless Sensor Network coverage optimization based on Yin-Yang pigeon-inspired optimization algorithm for Internet of Things, Internet Things 19 (2022) 100546.

[25] G. Zhu, S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, Appl. Math. Comput. 217 (7) (2010) 3166–3173.

[26] X. Zhou, Z. Wu, H. Wang, S. Rahnamayan, Gaussian bare-bones artificial bee colony algorithm, Soft Comput. 20 (3) (2016) 907–924.

[27] D. Kong, T. Chang, W. Dai, Q. Wang, H. Sun, An improved artificial bee colony algorithm based on elite group guidance and combined breadth-depth search strategy, Inform. Sci. 442–443 (2018) 54–71.

[28] M.S. Kiran, H. Hakli, M. Gunduz, H. Uguz, Artificial bee colony algorithm with variable search strategy for continuous optimization, Inform. Sci. 300 (2015) 140–157.

[29] H. Peng, C. Deng, Z. Wu, Best neighbor-guided artificial bee colony algorithm for continuous optimization problems, Soft Comput. (Berlin Germany) 23 (18) (2019) 8723–8740.

[30] X. Zhou, J. Song, S. Wu, M. Wang, Artificial bee colony algorithm based on online fitness landscape analysis, Inform. Sci. 619 (2023) 603–629.

[31] A. Hossain, P.K. Biswas, S. Chakrabarti, Sensing models and its impact on network coverage in wireless sensor network, in: 2008 IEEE Region 10 and the Third International Conference on Industrial and Information Systems, 2008, pp. 1–5.

[32] S. Kumar, D.K. Lobiyal, Sensing coverage prediction for wireless sensor networks in shadowed and multipath environment, Sci. World J. 2013 (2013) 565419.

[33] J. Wang, Y. Gao, C. Zhou, S. Sherratt, L. Wang, Optimal coverage multi-Path scheduling scheme with multiple mobile sinks for WSNs, CMC-Comput. Mater. Contin. 62 (2) (2020) 695–711.

[34] J. Wang, C. Ju, Y. Gao, A.K. Sangaiah, G.J. Kim, A PSO based energy efficient coverage control algorithm for wireless sensor networks, Comput. Mater. Contin. 56 (3) (2018) 433–446.

[35] Z. Du, G. Liu, D. Han, X. Yu, J. Jia, Artificial bee colony algorithm with global and unbiased search strategy, Acta Electron. Sin. 46 (2) (2018) 308–314.

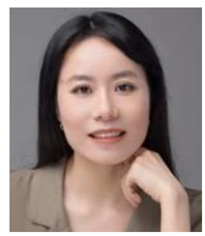
[36] W. Gao, S. Liu, L. Huang, A novel artificial bee colony algorithm based on modified search equation and orthogonal learning, IEEE Trans. Cybern. 43 (3) (2013) 1011–1024.
[37] X. Zhou, J. Wu, C. Deng, H. Peng, Neighbor-hood search-based artificial bee colony algorithm, J. Cent. South Univ. (Sci. Technol.) 46 (2) (2015) 534–546.
[38] T.K. Sharma, M. Pant, Shuffled artificial bee colony algorithm, Soft Comput. (Berlin Germany) 21 (20) (2017) 6085–6104.
[39] H. Wang, W. Wang, S. Xiao, Z. Cui, M. Xu, X. Zhou, Improving artificial bee colony algorithm using a new neighborhood selection mechanism, Inform. Sci. 527 (2020) 227–240.
[40] X. Gao, S. Liu, L. Huang, A novel artificial bee colony algorithm based on modified search equation and orthogonal learning, IEEE Trans. Cybern. 43 (3) (2013) 1011–1024.

Jin Wang received the M.S. degree from Nanjing University of Posts and Telecommunications, China in 2005. He received Ph.D. degree from Kyung Hee University Korea in 2010. Now, he is a professor at Changsha University of Science and Technology. He has published more than 400 international journal and conference papers. His research interests mainly include wireless ad hoc and sensor network, network performance analysis and optimization etc. He is a senior member of the IEEE and a Fellow of IET.

Ying Liu is currently pursuing the M.E. degree in the School of Computer and Communication Engineering at Changsha University of Science and Technology, China. His research interests include datacenter networks and evolutionary computation.

Shuying Rao is currently pursuing the M.E. degree in the School of Computer and Communication Engineering at Changsha University of Science and Technology, China. Her research interests include datacenter networks and network security.

Xinyu Zhou received the B.E. and M.E. degrees from Jiangxi University of Science and Technology, China, in 2008 and 2011, respectively. He received Ph.D. degree in Computer Software and Theory from Wuhan University, China, in 2014. Now, he is an associated professor with the School of Computer and Information Engineering, Jiangxi Normal University. His current research interests include evolutionary computation, swarm intelligence-based optimization algorithms, GPU-based parallel computing, and their applications to practical engineering projects.



Jinbin Hu received the B.E. and M.E. degrees from Beijing Jiao Tong University, China, in 2008 and 2011, respectively, and the Ph.D. degree in computer science from Central South University, China, in 2020. She is currently a Post-Doc in the Department of Computer Science and Engineering, Hong Kong University of Science and Technology, and working in the School of Computer and Communication Engineering, Changsha University of Science and Technology, China. Her current research interests are in the area of datacenter networks, RDMA networking and learning-based network systems.