



Chaos-Enhanced Superb Fairy-wren Optimization Algorithm for Wireless Sensor Network Coverage

R. Jalal Kamali^{†1} and M. R. Omid²

^{1,2} Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran.

ABSTRACT

Wireless Sensor Networks (WSNs) play a crucial role in monitoring and surveillance, yet random deployment often causes uneven coverage and redundant sensing. This study introduces a Chaos-Enhanced Superb Fairy-wren Optimization Algorithm (CE-SFOA), which integrates chaotic dynamics through a Cubic map into the position update and parameter control mechanisms. The chaotic modulation enhances population diversity, balances exploration and exploitation, and mitigates premature convergence. Experiments across three deployment scenarios show that CE-SFOA consistently achieves higher coverage and faster convergence than SFOA and seven competing metaheuristics, yielding 5.32–6.65% coverage improvement over the baseline. These findings demonstrate that chaotic modulation is an effective strategy for enhancing metaheuristic performance in WSN coverage optimization.

Keywords: Wireless Sensor Networks (WSNs), Coverage Optimization, Metaheuristic Algorithms, Chaos Theory

AMS subject classification: 68R10, 90B15.

[†] Corresponding author: R. Jalal kamali

Email: roozbehjalalkamali@gmail.com

ARTICLE INFO

Article history:

Research paper

Received 8, November 2025

Accepted 23, November 2025

Available online 13, December 2025

1 Introduction

The applications of Wireless Sensor Networks (WSNs) are diverse and range from disaster management and environmental monitoring to detection of earthquakes and floods and security purposes. The systems are comprised of numerous sensor nodes with low power consumption and robust sink strength that jointly forge communication channels under specified transmission restrictions. WSNs are especially prized for convenience in deployment and effective coordination in dynamic situations. Each sensor node is inherently limited in terms of storage capacity, accessible power, range of sensing, and computing power [1]. Of the performance indicators of WSNs, coverage stands supreme and acts as a primary indicator of quality of service [2]. The conventional random deployment methods often yield highly unbalanced distributions with wide portions of uninhibited areas and excessive overlaps of high densities. Such imbalances not only render detection ineffective but also render resource utilization inefficient. Therefore, researchers have increasingly turned to deployments that adapt to more evenly spread sensor placement and hence lower the deployment expense while providing a higher quality of service [3]. Figure 1 is an overview of the classification of WSN coverage problems.

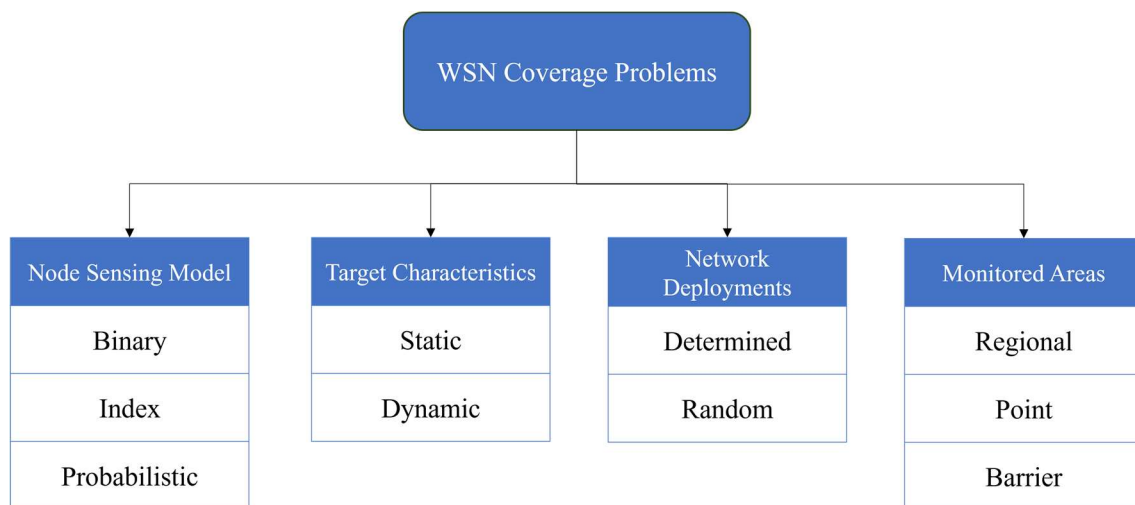


Fig 1: Classification of WSN Coverage Problem

To keep the number of nodes to a minimum in the optimization of WSNs, people have increasingly looked at the nature-inspired metaheuristic algorithms because they are simple to understand with basic concepts that only require a small set of parameters and are equally simple to code. Such techniques are evolutionary, like Genetic Algorithm (GA) [4] and swarm intelligence, like Particle Swarm Optimization (PSO) [5] or Grey Wolf Optimization (GWO) [6]. Physics-inspired techniques like Fick's Law Algorithm (FLA) [7] and human-inspired techniques like Teaching-Learning-Based Optimization (TLBO) [8] are also examples of the techniques. The various works in the past served to establish the effectiveness of the algorithms to optimize the deployment of the sensors. Nonetheless, the No Free Lunch theorem (NFL) [9] identifies that no optimization

algorithm is perfect for all occasions because each has defects in specific scenarios. Their motivation has resulted in the continued invention of new metaheuristic strategies. Heming Jia et al. created the Superb Fairy-wren Optimization Algorithm (SFOA) [10], a strategy with roots in the natural behaviors of the Superb Fairy-wren and consisting of stages like the juveniles developing, breeding, and feeding, and evasive maneuvers against predators. SFOA has been successfully applied to solve numerical optimization, engineering design applications, and feature selection in high-dimensional spaces. Most of the metaheuristic algorithms evolve in the following format: the researchers overcome defects in exploration and exploitation with the introduction of refinements. These refinements often involve mechanisms like transfer functions, Lévy flight maneuvers, opposition-based learning, Nelder–Mead approach, elitist methods, and the many operators of mutation [11,12]. Continuing this line of research, the use of the theory of chaos itself is seen to be a highly promising line of development of the efficiency of the studied metaheuristics at the current time.

2 Background

The design of the SFOA with chaos improvement needs to have three fundamental building blocks of knowledge. First, the WSN coverage model explains the sensor location that determines the covered area and how coverage holes must be plugged. Second, metaheuristic algorithms provide adaptive, nature-inspired search mechanisms capable of optimizing sensor placement, though potentially becoming trapped in local optima. Finally, chaos theory provides mathematical mappings (e.g., the Cubic map) that have structured randomness in algorithm parameters and therefore preserve population diversity while accelerating convergence. Each of these building blocks is elaborated on in the following subsections.

2.1 WSN Node Coverage Model

Consider a two-dimensional WSN monitoring area A with size of $M \times N m^2$ in which q sensor nodes are randomly deployed. The set of nodes is represented as $S = \{S_1, \dots, S_q\}$, where each node S_i has coordinates (x_i, y_i) , for $i = 1, \dots, q$. The network model for this monitoring region is defined by the following assumptions:

- All sensor nodes are homogeneous, sharing identical parameters, structure, and communication capabilities.
- Each node has sufficient energy, functional communication ability, and timely data accessibility.
- Nodes can move freely and update their location information dynamically.
- Each sensor node has a sensing radius R_s and a communication radius R_c (measured in meters), with the condition that $R_c \geq 2R_s$.

The sensing region of each node is modeled as a circular area with radius R_s . Let there be m target monitoring points in the WSN region, denoted as $T = \{T_1, \dots, T_m\}$, where the coordinates of each target point T_j are (x_j, y_j) , for $j=1, \dots, m$. A target point T_j is considered covered if the Euclidean distance between T_j and any sensor node S_i is less than or equal to R_s . The Euclidean distance is given by:

$$d(S_i, T_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (1)$$

The Boolean sensing model is adopted. Where coverage is binary, if $R_s \geq d(S_i, T_j)$, then the probability that S_i covers T_j is 1; otherwise, the probability is 0. Formally, if p_{ij} denotes the probability that sensor node S_i covers target point T_j , then:

$$p_{ij} = \begin{cases} 1 & R_s \geq d(S_i, T_j) \\ 0 & R_s < d(S_i, T_j) \end{cases} \quad (2)$$

Target points may be monitored by multiple nodes simultaneously. The combined probability that target point T_j is covered can be defined as:

$$P(S, T_j) = 1 - \prod_{i=1}^q (1 - p_{ij}) \quad (3)$$

The coverage rate is defined as the ratio of the monitored area to the total region A , given by:

$$Cov = \frac{\sum_{j=1}^n P(S, T_j)}{M \times N} \quad (4)$$

Based on this formulation, the WSN node coverage optimization problem can be expressed as an integer linear programming (ILP) model:

Maximize Cov

subject to:

1. Each target point T_j satisfies the joint sensing probability constraint.
2. The total monitored area does not exceed A .
3. The Euclidean distance between any sensor node S_i and a target point T_j remains within the sensing radius R_s .

Figure 2 illustrates an example of sensor node coverage, where each node monitors a circular region, and target points may be located either inside or outside the range. The diagram highlights covered and uncovered points to clarify how coverage is evaluated.

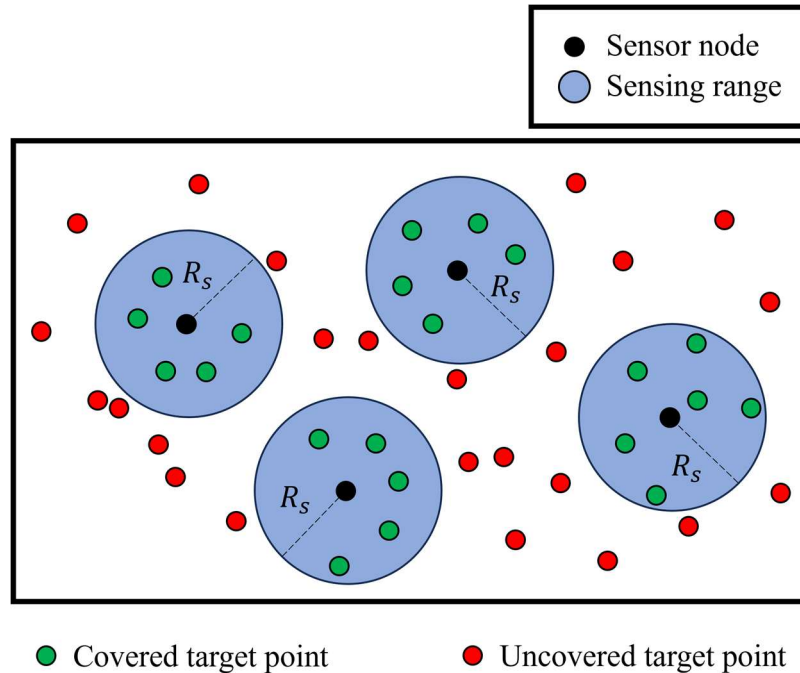


Fig 2: Example of WSN coverage model.

For networks containing a large number of sensor nodes, solving the ILP problem to obtain an exact optimal solution becomes computationally expensive. Metaheuristic algorithms represent a more suitable alternative, as they can deliver near-optimal solutions within practical computational limits. A Chaos-Enhanced Superb Fairy-wren Optimization Algorithm (CE-SFOA) is introduced to address the coverage optimization problem in WSNs [13].

2.2 Metaheuristic Algorithms

Solving the WSN coverage problem with metaheuristic algorithms requires determining an efficient configuration of sensor nodes that maximizes the monitored area while reducing redundancy and minimizing uncovered regions. Each candidate solution is expressed as a position vector specifying the placement of sensors within the monitoring region. Solution quality is assessed through a fitness function that measures the achieved coverage rate.

The optimization process typically begins with the generation of an initial population of potential solutions, as shown in Fig. 3. The process follows a series of iterations where new solutions are generated by position-update mechanisms through emulation of natural behaviors such as

attraction, repulsion, or random movement. Operators like Lévy flight are generally employed to give exploratory steps of great magnitude, which assist in drifting away from local optima and enhance the capacity for global search. Candidate solutions are evaluated in every iteration, and improved solutions are preserved for progressing to the next generation. This continues until a stopping condition is reached, e.g., achieving a defined number of iterations or a desired coverage target. Figure 3 depicts the generic process of a metaheuristic algorithm for optimizing WSN coverage, consisting of population initialization, fitness evaluation, repeated solution generation by nature-inspired techniques, and termination at the stopping criterion.

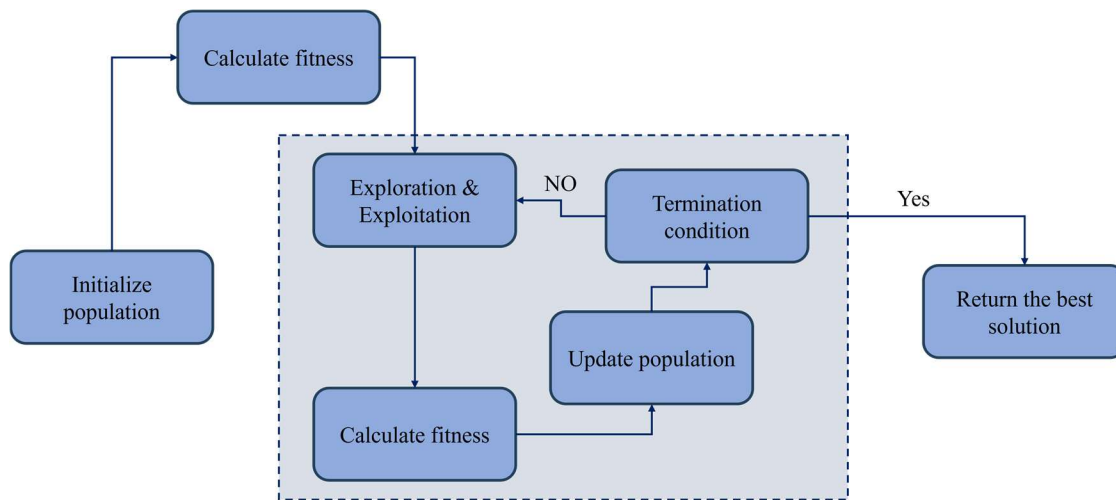


Fig 3: General process of a metaheuristic algorithm for WSN coverage optimization.

Upper-level metaheuristic methods can employ chaos-based initialization, self-tuning parameter control, or hybrid approach strategies to maximize convergence performance and solution quality. These techniques find particular value in large-scale WSN scenarios where both computational cost and coverage capability are key determinants.

2.2.1 Superb Fairy-wren Optimization Algorithm (SFOA)

The Superb Fairy-wren Optimization Algorithm (SFOA) is a nature-inspired metaheuristic introduced in 2025. The method emulates superb fairy-wrens' survival and social strategies in the form of small birds that are noted for cooperative breeding behaviors and their predator evasion mechanisms. In the algorithm, each potential solution is modeled as an individual bird that goes through three phases of behavior: juvenile growth, cooperative breeding and resource sharing, and predator defense. The first stage emphasizes exploration, the second emphasizes exploitation within the local neighborhood, and the third provides more exploitation along with mechanisms to escape from local optima.

Young birds growth stage

The Young birds growth stage reflects exploratory behavior exhibited by inexperienced young birds. Candidate solutions are updated frequently to enable broad exploration of the search space and to reduce the likelihood of premature convergence. The position update rule is expressed as

$$X_{ij}^{(t+1)} = X_{ij}^{(t)} + (lb + (ub - lb) \times rand) \quad (5)$$

where $X_{ij}^{(t+1)}$ denotes the updated position of the i -th solution in dimension j -th after iteration $t+1$, $X_{ij}^{(t)}$ denotes the previous position, and $rand$ is a uniformly distributed random variable within $[0,1]$. Frequent updates at this stage encourage exploration across the solution space.

Breeding and feeding stage

The breeding and feeding stage models the cooperative interactions of superb fairy-wrens, including egg guarding and paternity assurance. In the algorithm, entry into this stage is governed by a calculated danger threshold s :

$$s = r_1 \times 20 + r_2 \times 20 \quad (6)$$

where r_1 and r_2 are normally distributed random values. When the risk level is below the threshold, individuals engage in behaviors analogous to feeding and teaching, resulting in small, directed positional adjustments that refine local exploitation. The position update is defined as

$$X_{ij}^{(t+1)} = X_G + (X_b - X_{ij}^{(t)}) \times p \quad (7)$$

With

$$X_G = X_b \times C \quad (8)$$

$$p = \sin((ub - lb) \times 2 + (ub - lb) \times m) \quad (9)$$

$$m = \left(\frac{t}{T_{\max}} \right) \times 2 \quad (10)$$

In these equations, X_b denotes the global best solution, C is a constant equal to 0.8, t is the current iteration, and T_{\max} is the total number of iterations. As the optimization progresses, individuals perform increasingly refined movements, which improve solution accuracy during later stages of the search.

Predator Avoidance Stage

The predator avoidance stage simulates evasive behaviors observed in superb fairy-wrens. Birds confronted by predators perform rapid escape maneuvers, emit warning signals, and display erratic

wing movements to confuse attackers. Within the algorithm, this behavior is modeled through a combination of local adjustments and long-range evasive movements. The position update equation is

$$X_{ij}^{(t+1)} = X_b + X_{ij}^{(t)} \times l \times k \quad (11)$$

where l is a Lévy flight-based random step size enabling long-distance jumps, and k is an adaptive balance factor defined as

$$k = 0.2 \times \sin\left(\frac{\pi}{2} - w\right) \quad (12)$$

With

$$w = \frac{\pi}{2} \times \frac{t}{T_{\max}} \quad (13)$$

The parameter w represents a call frequency value regulating the unpredictability and intensity of escape maneuvers. Incorporation of the global best position X_b directs individuals toward promising regions of the search space while reducing evaluations of low-quality areas, which improves convergence efficiency [10].

2.3 Chaos Theory and Cubic Map

Chaos has been defined as unpredictable-appearing behavior in nonlinear deterministic systems. Given that these systems evolve under strict regulations, their trajectories can appear to be unpredictable due to these systems being highly sensitive to their initial conditions, most commonly referred to as the *butterfly effect* or *Lyapunov instability* [14]. Very small differences in the initial conditions may produce very distinct sequences, something very much desired in population-based metaheuristic algorithms, where it is desirable to avoid premature convergence.

Chaotic maps are frequently integrated into optimization methods to introduce irregular, non-repetitive dynamics. Within the proposed CE-SFOA algorithm, the *Cubic map* is adopted for generating chaotic sequences used in parameter control and position updates. The Cubic map is a one-dimensional polynomial map capable of producing chaotic trajectories within the interval $(0,1)$. Its mathematical formulation is given in Eq. (14) [15, 16]:

$$x_{t+1} = \rho \cdot x_t \cdot (1 - x_t^2) \quad (14)$$

where $x_n \in (0,1)$ and ρ represents the control parameter. For $\rho = 2.59$, the system exhibits chaotic behavior with strong ergodicity and high sensitivity to initial conditions.

Figure 4 shows the chaotic sequence generated by the Cubic map with $\rho = 2.59$, illustrating its irregular, non-repetitive behavior and uniform coverage of the interval $(0,1)$. The sequence properties support effective exploration and exploitation in metaheuristic optimization.

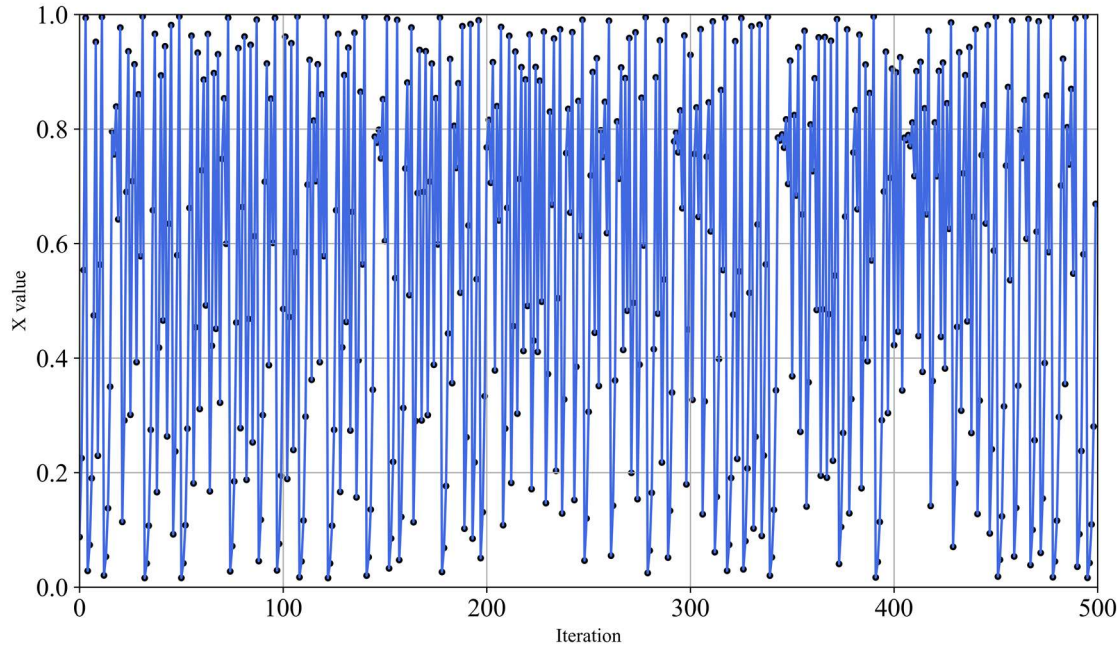


Fig 4: Visualization of the Cubic map

3 Related Works

The Wireless Sensor Networks (WSNs) received a lot of interest due to the unprecedented utilization of the networks. The node coverage optimization is the main factor in the deployment of the WSNs in a quest to expand the performance of networks and decrease the energy consumption. The coverage holes result from the conventional methods of deployment of the nodes in combination with the redundant deployment and the resource waste. Hence, the various metaheuristics methods were put forward in a quest to combat the challenges in a competent way. Various bio-inspired metaheuristic algorithms were introduced in an effort to improve the coverage of the WSNs.

Evolutionary-based approaches, such as Genetic Algorithms (GAs) [17] and Differential Evolution (DE) [18], are commonly used in WSN optimization. Animal behavior-driven methods, like the Grey Wolf Optimizer (GWO) [19], Salp Swarm Algorithm (SSA) [20], and Butterfly Optimization Algorithm (BOA) [21], are favored for adaptability and strength in node coverage optimization. Physics-driven algorithms like Simulated Annealing (SA) [22], and the Sine-Cosine Algorithm (SCA) [23], are utilized for evading local optima. Human behavior-driven models like Teaching-Learning-Based Optimization (TLBO) [24], and the Imperialist Competitive Algorithm (ICA) [25], also proved successful in sensor node deployment. Specific studies actually tried to

improve WSN coverage using different varieties of the metaheuristic algorithms. ZainEldin et al. [26] proposed an Improved Dynamic Deployment Technique based on Genetic Algorithm (IDDT-GA) for maximizing the area coverage through a new two-point crossover of variable-length encoding, although the GAs are prone to becoming computationally expensive, and more work is required regarding probabilistic models and power consumption. Wei et al. [27] developed a Multi-Strategy Pelican Optimization Algorithm (MSPOA) for Agricultural WSNs, integrating good point sets, 3D spiral Lévy flight, and adaptive T-distribution variation, though acknowledging potential for uncovered areas in highly complex scenarios and high computational cost for large-scale applications. Jianlan et al. [28] proposed an Improved Sparrow Search Algorithm (ISSA) using a good point set, adaptive learning factor, and refraction reverse learning, noting that while standard SSA is prone to local optima, the coverage improvement from ISSA might not always be substantial. Yong et al. [29] proposed a Simulated Annealing-optimized Grey Wolf Algorithm (SA-GWO) that is especially designed to handle the low coverage rate and high node aggregation, but prone to slow convergence and local optimum in itself, and only covering static WSNs in the paper. Cheng et al. [30] proposed an Energy-enhanced Fruit Fly Optimization Algorithm (E-FOA) that is an extended version of the conventional FOA framework with the additional capability of residue energies being tracked in an attempt to reduce dead nodes and maximize network lifetime with superior overall performance but warned that the superiority of the coverage rate in the majority of the examples frequently becomes irregular ones and the eventual coverage is not necessarily at the optimum level. A table of the above-mentioned works with the proposed algorithms and the methods of comparison utilized and defects encountered is tabulated in Table 1.

Paper	Year	Algorithm(s)	Compared Methods	Disadvantage(s)
[26]	2020	IDDT-GA	HS, CM-IA, MADA-WOA	Assumes obstacle-free deployment and risks connectivity issues with very low node counts.
[27]	2025	MSPOA	IABC, CAFA, APSO, LCSO	Potential for uncovered areas in complex scenarios and high computational complexity.
[28]	2023	ISSA	SSA, ESSA, IGWO, LSSA, XSSA	limited coverage improvement and power-constrained environments.
[29]	2021	SA-GWO	PSO, GWO	Focuses on static WSNs; potential for excessive node aggregation in some areas.
[30]	2023	E-FOA	FOA	Coverage rate increase can be erratic and the final coverage rate is not always as high as expected.

Table 1: A summary of related works

4 Proposed Chaos-Enhanced SFOA (CE-SFOA)

This section introduces the Chaos-Enhanced Superb Fairy-wren Optimization Algorithm (CE-SFOA) developed for Wireless Sensor Network (WSN) coverage optimization. The proposed

approach enhances the exploration and convergence behavior of the original SFOA by integrating chaotic dynamics into key algorithmic components. Specifically, CE-SFOA introduces chaos-based mechanisms into the young birds' position update strategy and directly modulates the behavioral parameters k and p using a Cubic map. These modifications are intended to increase population diversity, mitigate the risk of premature convergence, and enable dynamic adaptation of the search process over the course of optimization.

4.1 Chaotic Position Update

In the original SFOA, young birds update their positions using a uniformly distributed random component to simulate natural exploration. In CE-SFOA, this random element is replaced with a chaotic value derived from the Cubic map sequence. This substitution introduces deterministic yet irregular fluctuations, improving the algorithm's capacity to explore the search space efficiently.

The updated position equation is:

$$X_{ij}^{(t+1)} = X_{ij}^{(t)} + (lb + (ub - lb) \times chaos(t)) \quad (15)$$

Here, $X_{ij}^{(t)}$ represents the current position of the i -th individual in the j -th dimension, while lb and ub define the lower and upper bounds of the search space. The replacement of purely random sampling with chaotic sequences allows for more dynamic trajectory adjustments and reduces the risk of early convergence.

4.2 Chaos-Controlled Parameter Adjustment

To further enhance adaptability, CE-SFOA applies chaos directly to the two key behavior parameters, k and p . These parameters control the level of exploration and the probability of interaction among individuals, making them critical to the overall search dynamics. The auxiliary variables m and w , already defined in Eq. (10) and Eq. (13), are first computed in their standard time-dependent form. Using these values, the chaos-enabled parameters are expressed as:

$$p = \sin((ub - lb) \times 2 + (ub - lb) \times m) \cdot chaos(t) \quad (16)$$

$$k = 0.2 \times \sin\left(\frac{\pi}{2} - w\right) \cdot chaos(t) \quad (17)$$

where $chaos(t)$ is generated by the *Cubic map*.

By multiplying k and p with the chaotic sequence, CE-SFOA introduces irregular yet bounded variability into the birds' behavioral parameters. This mechanism avoids repetitive patterns,

preserves population diversity during the early search phase, and improves exploitation in the later phase, resulting in more efficient convergence for WSN coverage optimization.

4.3 CE-SFOA Algorithm

The flowchart of the proposed Chaos-Enhanced Superb Fairy-Wren Optimization Algorithm (CE-SFOA) is shown in Fig. 5. It presents the logical sequence of processes from initialization to convergence, and the detailed algorithmic steps are described as follows:

Step 1: The essential parameters of the algorithm, such as population size N , the maximum number of iterations Max_iter , problem dimension Dim , sensing radius R_s , and monitoring area size M , are initialized. The fitness function is defined according to the WSN coverage model.

Step 2: The initial population of candidate solutions (fairy-wrens) is generated randomly within the search space. Each individual represents a possible configuration of sensor nodes. The chaotic sequence is produced using the Cubic map.

Step 3: Behavioral control parameters k and p are initialized and adjusted by chaotic modulation. This adaptive control mechanism enables dynamic adjustment of exploration and exploitation capabilities during the optimization process.

Step 4: The fitness value of each individual is calculated based on the achieved network coverage ratio. The best individual in the current iteration is identified and stored as the temporary global best solution.

Step 5: For each iteration $t \leq Max_iter$, every individual updates its position according to the behavioral stage determined by the current iteration and the computed danger factor s :

- If the exploration condition is satisfied, the **Young Birds Growth Stage** is executed, and positions are updated using chaotic perturbation Eq. (15) to improve diversity.
- When s is below the danger threshold, the **Breeding and Feeding Stage** begins, enabling individuals to move toward the global best using Eq. (7) with refined local search using the control parameter p by Eq. (16).
- When s exceeds the threshold, the **Predator Avoidance Stage** is triggered, applying Lévy flight-based position update using Eq. (11) with chaos control parameter k by Eq. (17) to escape local optima and maintain population variability.

Step 6: After each behavioral update, the global best solution is updated if a higher coverage value is obtained. The iteration counter is then incremented.

Step 7: The chaotic control continues to regulate both position updates and behavioral parameters across iterations, ensuring ergodicity and stable convergence toward the global optimum.

Step 8: If the maximum iteration count is not reached, the algorithm returns to Step 5; otherwise, the optimization terminates, and the best solution that represents the optimal sensor node deployment configuration is output.

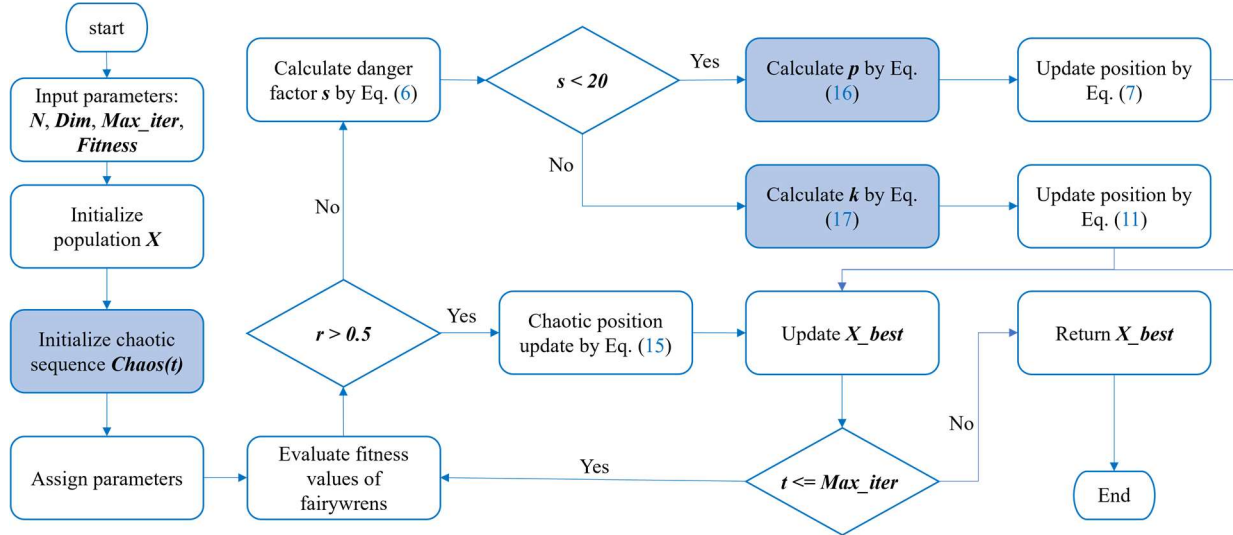


Fig 5: Flowchart of CE-SFOA

Algorithm 1 outlines the operational structure of the Chaos-Enhanced Superb Fairy-wren Optimization Algorithm (CE-SFOA).

Algorithm 1: Pseudo-code of the CE-SFOA

Inputs: N : population size
 Dim : problem dimension;
 Max_iter : maximum number of iterations
 $Fitness$: fitness function

Outputs: Best solution x_best

1. X =initialize N population with dimensions Dim
2. Initialize chaotic map $Chaos(t)$
3. Initialize the parameters
4. **While** ($t \leq Max_iter$) **do**
5. Evaluate fitness of each individual
6. **For** each fairywren **do**
7. **If** $r > 0.5$:
8. Chaotic position update using Eq. (15) (Young birds growth stage)
9. **Else:**
10. Calculate danger factor s using Eq. (6)
11. **If** $s < 20$:
12. Calculate p using Eq. (16)
13. Update position of each individual using Eq. (7) (Breeding and feeding stage)
14. **Else:**
15. Calculate k using Eq. (17)

```

16.         Update position of each individual using Eq. (11) (Avoiding natural enemies stage)
17.     End if
18. End if
19. End for
20. Update global best
21.      $t = t + 1$ 
22. End while
23. Return  $X\_best$ 

```

Adding chaos into CE-SFOA is theoretically motivated by the ability of chaotic motions to improve exploration diversity, nonlinearity, and the balance between exploration and exploitation. There are several pros that could be highlighted:

- Chaotic modulation of parameter k allows dynamic adjustment of exploration intensity across iterations, improving the search in early stages.
- Parameter p under chaotic modulation imposes random variations of interaction chances to elevate flexibility in a variety of WSN topologies.
- Color chaotic map update substitutes the uniformly random term with the Cubic map, and produces non-repeating but bounded trajectories with efficient coverage of space.
- Increased diversity and strength are obtained because the cubic map's high-quality randomness and ergodicity fortify the algorithmic ability to escape stagnation.
- Balanced convergence is supported as chaotic scaling enables a smooth transition from exploration to exploitation, improving convergence stability.

Overall, the changes made to CE-SFOA that make things more chaotic make the search mechanism stronger and more flexible. In the early versions, it lets you explore the whole world, and in later versions, it lets you search more precisely in your own area, which leads to better coverage solutions in WSN scenarios.

5 Experimental results

5.1 Experimental Environment

The proposed CE-SFOA algorithm and all comparison methods were implemented in Python 3.12 environment on a computer with a 64-bit Windows 11 operating system, the CPU is Intel Core i7-11800H, and the memory is 16 GB. This configuration ensured the consistent and reliable execution of all experiments.

5.2 Simulation Scenarios and Algorithm Comparison

In order to fully assess the performance of the CE-SFOA for the coverage analysis of WSNs, three simulation scenarios with varied sizes, sensors, and sensing range requirements were designed. The details of each scenario are listed in Table 2. In the first scenario, the size of the selected area

was fixed at $20 \times 20 m^2$, the total number of sensors was set to 24, and the sensor viewing distance was set to $2.5 m$. In the second scenario, the size of the selected area was fixed at $50 \times 50 m^2$, the total number of sensors was set to 35, and the sensor viewing distance was set at $5 m$. In the third scenario, an area size of $100 \times 100 m^2$, a total of 25, and the sensor's viewing distance was set at $10 m$. In all three cases, the total number of iterations remained constant at 500 simulation cycles.

Parameters	Scenario 1	Scenario 2	Scenario 3
Area size	$20 \times 20 m^2$	$50 \times 50 m^2$	$100 \times 100 m^2$
Number of nodes	24	35	25
Node-sensing radius	$2.5 m$	$5 m$	$10 m$
Communication radius	$5 m$	$10 m$	$20 m$
Maximum iteration	500	500	500

Table 2: Parameter settings

The new CE-SFOA was compared with seven existing optimization algorithms, including the Superb Fairy-wren Optimization Algorithm (SFOA), Genetic Algorithm (GA), Pelican Optimization Algorithm (POA) [31], Coyote Optimization Algorithm (COA) [32], Osprey Optimization Algorithm (OOA) [33], Zebra Optimization Algorithm (ZOA) [34], and Tuna Swarm Optimization (TSO) [35]. The parameter settings for these algorithms are listed in Table 3. To obtain valid statistics, each algorithm was run individually 10 times for each scenario.

Comparison Algorithm	Parameter Settings
GA	$N = 30, p_c = 0.7, p_m = 0.3$
POA	$N = 30$
COA	$N = 30, n_coyotes = 5$
OOA	$N = 30$
ZOA	$N = 30$
TSO	$N = 30, a = 0.7, z = 0.05$

Table 3: parameters of all algorithms

5.3 Simulation Results and Analysis

The simulation results of the proposed CE-SFOA algorithm are presented under various WSN scenarios with different randomly deployed sensor nodes. In the dense deployment (Scenario 1), the convergence pattern in Fig. 6 shows that CE-SFOA achieves faster and smoother convergence than competing algorithms due to its dynamic chaotic operators. The chaotic position update

mechanism introduces bounded, non-repetitive perturbations that enhance local exploration while preventing redundant searches. Simultaneously, adaptive modulation of behavioral coefficients enables an effective transition from global exploration to local exploitation, which is crucial in dense networks where optimal node placements are concentrated in narrow feasible regions.

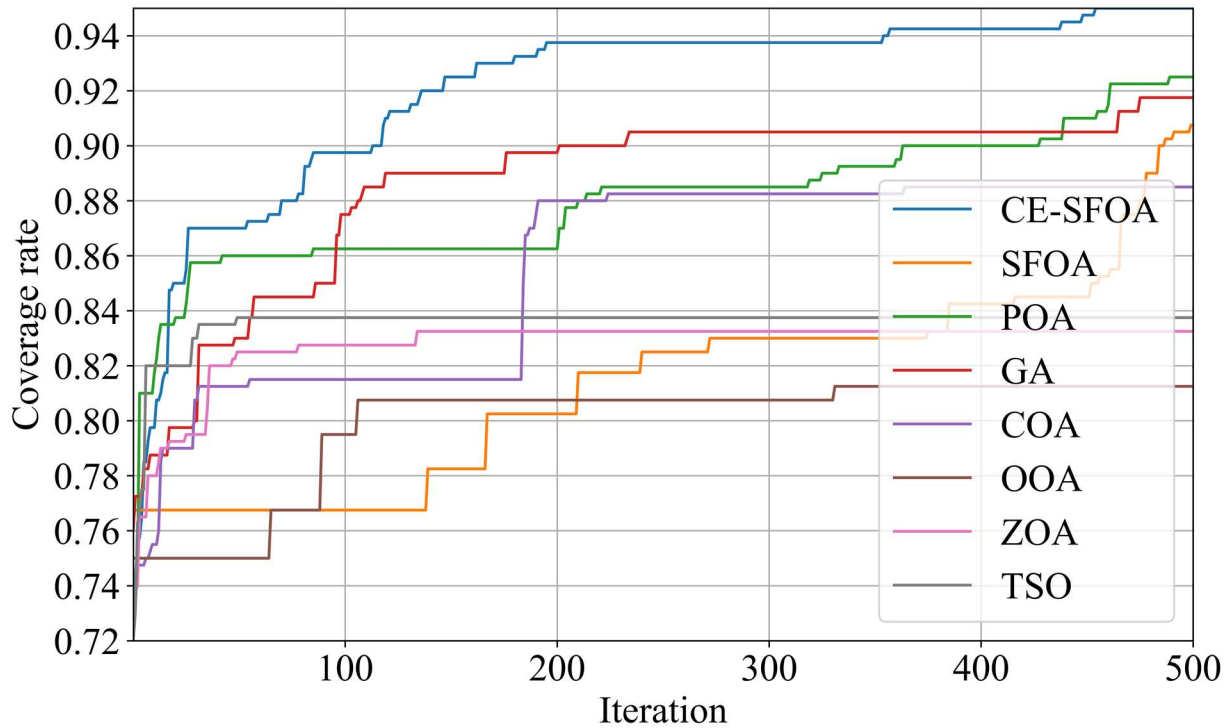


Fig. 6: Coverage rate curves of the eight algorithms (Scenario 1)

Figure 7 illustrates that CE-SFOA achieves a more uniform node distribution and larger coverage area compared with other algorithms. Although minor uncovered areas remain, nearby sensors effectively perceive those regions, indicating an optimal coverage scheme. Across all three scenarios, CE-SFOA consistently provides high coverage performance. In contrast, excessive random movement in GA and POA leads to node overlap, whereas premature convergence in COA and OOA produces uncovered areas. The ergodic properties of the Cubic map balance search trajectory distribution, maintaining diversity and convergence efficiency. Consequently, CE-SFOA achieves faster and more stable convergence toward optimal configurations, reflecting an efficient coordination between exploration and exploitation absent in conventional metaheuristics.

As shown in Fig. 8, the medium-scale deployment scenario (Scenario 2) demonstrates that CE-SFOA converges faster and more stable than the comparative algorithms, highlighting its ability to balance exploration and exploitation effectively. The chaotic position update introduces controlled irregular perturbations that prevent stagnation around local optima. This limitation is observed in GA and POA, whose mutation and spiral movements follow fixed probabilistic or geometric patterns. As the search space expands, CE-SFOA adaptively modulates its control

parameters, gradually reducing exploration pressure while strengthening exploitation near convergence. This adaptive mechanism offers greater flexibility than the fixed crossover probabilities in GA or the deterministic pursuit strategy in POA. The smooth convergence trend of CE-SFOA also reflects efficient computational resource utilization, as fewer iterations are needed to achieve high coverage levels. In contrast, COA, OOA, and ZOA exhibit stagnation plateaus because their operators lack chaotic feedback mechanisms to restore diversity after population contraction.

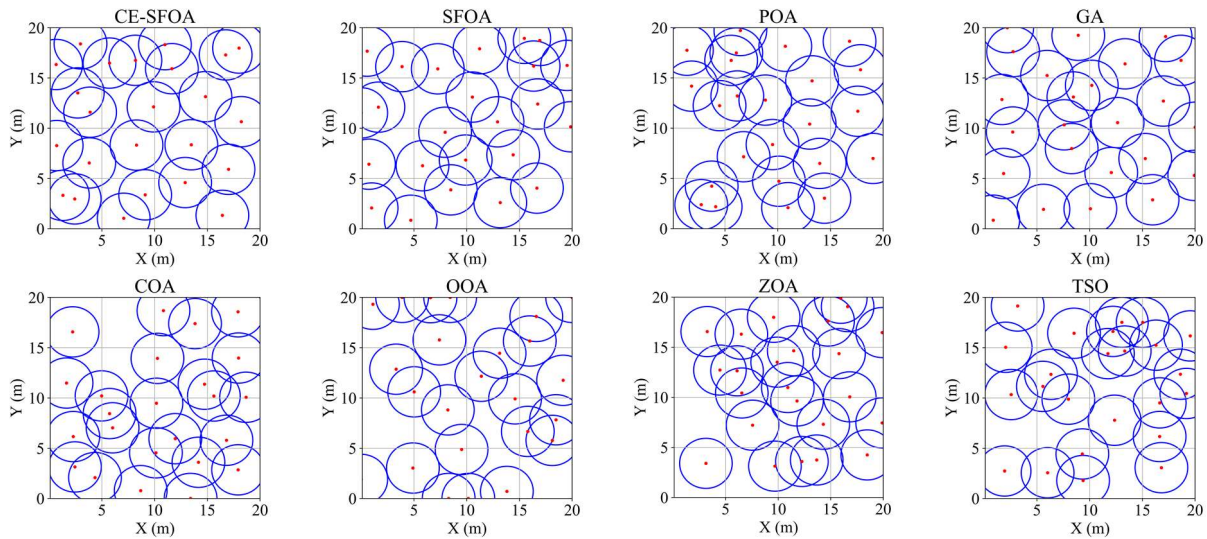


Fig.7: Sensor node deployment diagram of different algorithms (Scenario 1)

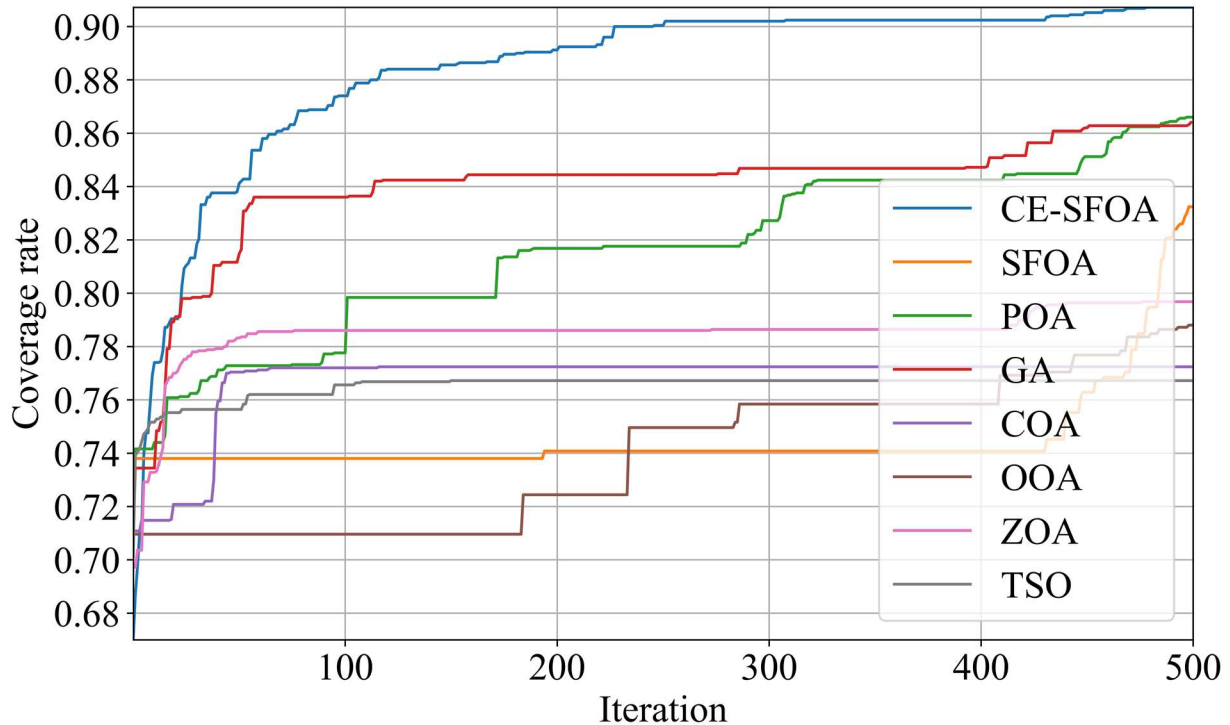


Fig. 8: Coverage rate curves of the eight algorithms (Scenario 2)

Figure 9 shows that CE-SFOA achieves the largest and most uniform coverage area, effectively utilizing each sensor's sensing capacity. The TSO algorithm produces excessive overlap due to premature convergence, resulting in the lowest coverage efficiency. Although GA and POA reduce some uncovered areas, they still exhibit redundant coverage regions. Overall, CE-SFOA demonstrates superior coverage optimization, minimizing both uncovered and overlapping areas, thereby enhancing detection efficiency across the network.

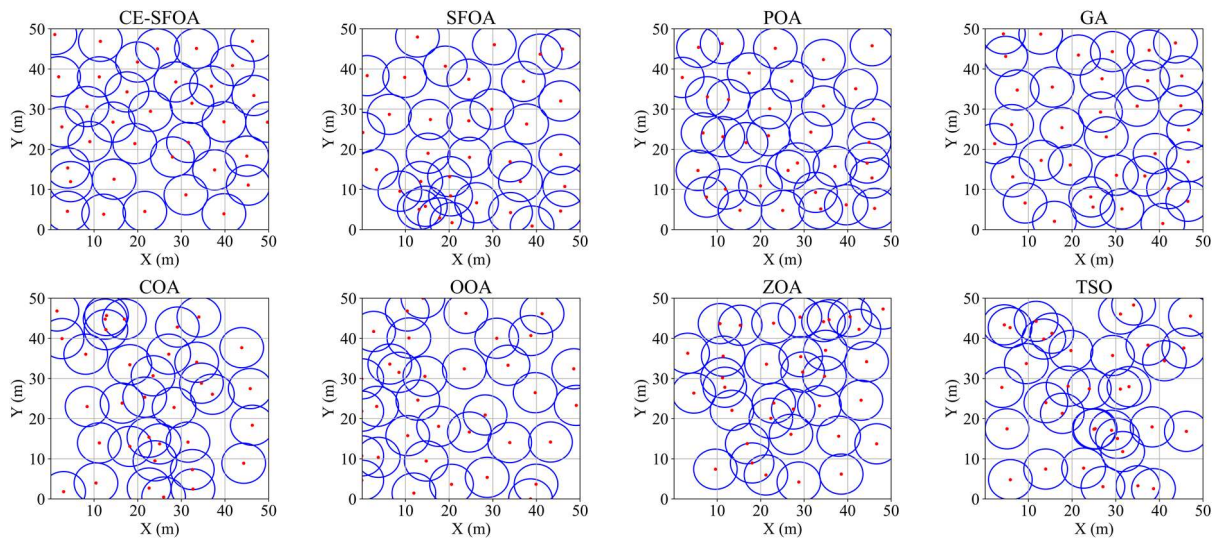


Fig. 9: Sensor node deployment diagram of different algorithms (Scenario 2)

In the large-area, low-density deployment of Scenario 3, the performance gap between CE-SFOA and the comparative algorithms becomes more evident. As shown in Fig. 10, CE-SFOA maintains steady improvement throughout the entire iteration process, reaching convergence earlier and with smoother stability. This behavior demonstrates the algorithm's adaptability to sparse search spaces, where sustained exploration is essential. The chaotic position update mechanism allows each candidate solution to explore distant, unexplored regions while maintaining population structure, thereby avoiding the premature stagnation commonly observed in GA and POA when their crossover and local refinement operators converge too early. Meanwhile, the chaos-controlled parameter modulation continuously adjusts the balance between exploration and exploitation, enabling the algorithm to dynamically redirect its search as the solution space becomes more refined. In contrast, algorithms such as COA, OOA, and ZOA rely on static communication or information-sharing strategies that limit their ability to escape suboptimal regions, resulting in slower and more irregular convergence patterns.

Figure 11 presents the coverage performance of the algorithms under a large-scale WSN deployment. A detailed analysis of these results indicates that, despite minor uncovered areas at

the edges, CE-SFOA achieves superior overall coverage compared with the other methods. This improvement is reflected in both the higher coverage ratio and the reduced redundancy among sensing areas, confirming the algorithm’s efficiency in resource utilization and network optimization. In practical applications such as smart agriculture and large-field crop monitoring, implementing CE-SFOA in WSNs can substantially enhance monitoring capability by enabling precise, real-time data acquisition and supporting more informed decision-making for efficient resource management.

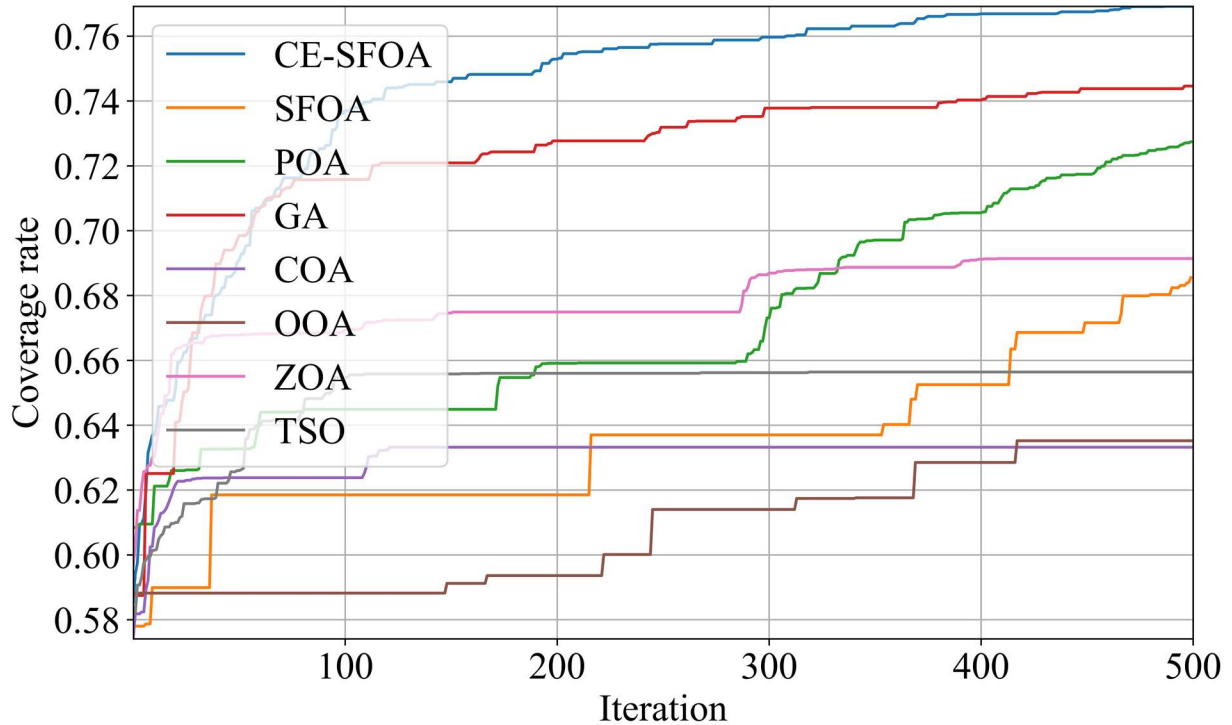
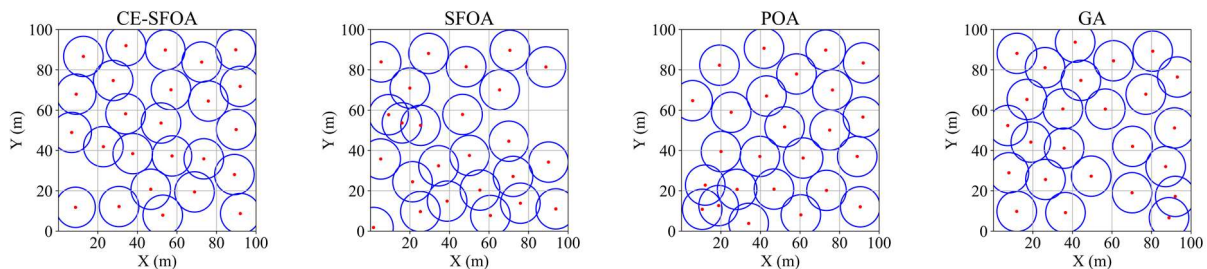


Fig. 10: Coverage rate curves of the eight algorithms (Scenario 3)



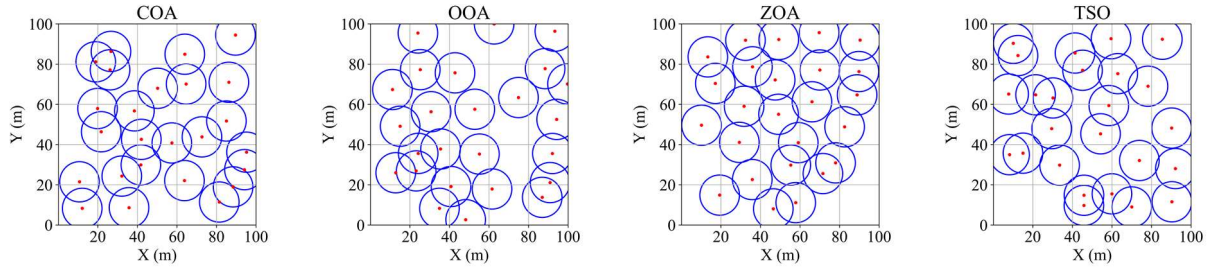


Fig. 11: Sensor node deployment diagram of different algorithms (Scenario 3)

Table 4 presents a comparison of the performance statistics of the proposed CE-SFOA algorithm with seven benchmark algorithms across the three deployment scenarios. The results indicate that the proposed algorithm consistently achieves the highest coverage in both mean and best values, with the smallest variability across all three cases. In the first scenario, CE-SFOA achieved an average coverage of 94.12%. The algorithm obtained an improvement of approximately 5.32% over the baseline algorithm, compared to the existing algorithms POA (92.33 %) and GA (91.35 %). Its best performance of 96.25% further highlights its ability to generate highly effective solutions in dense deployment environments. In the second scenario, the mean coverage of the proposed algorithm was enhanced by 90.44%, an improvement over the existing SFOA by 6.65%. The proposed algorithm showed an enhancement over the existing algorithms, GA (87.70%) and POA (87.32%). The smaller standard deviation ($\sigma = 0.0130$) indicates that CE-SFOA maintains consistent performance across multiple runs, unlike OOA ($\sigma = 0.0285$), which exhibits higher variability. In the third scenario, where the devices are deployed at a larger scale, the proposed algorithm still performs well compared with the existing algorithms, as it achieved 75.60% mean coverage. The results show that the proposed algorithm improved the existing algorithm, SFOA, by 6%. The proposed algorithm showed an enhancement over the existing algorithms, GA (75.17%) and POA (72.36%). As a result, the proposed algorithm showed the best result since it achieved the best coverage rate of 77.22%. The existing algorithms COA, OOA, ZOA, and TSO obtained lower performance compared to the proposed algorithm, as the algorithms obtained an average value less than the proposed algorithm for the three cases, indicating the capability of algorithms to produce high-quality routes for both cases of the network size and density, although the results obtained for the three cases are very small compared to the proposed algorithm.

Algorithm	Measures	Scenario 1	Scenario 2	Scenario 3
CE-SFOA	Mean:	94.12%	90.44%	75.60%
	Best:	96.25%	92.24%	77.22%
	Std:	0.0140	0.0130	0.0097
SFOA	Mean:	88.80%	83.79%	69.60%
	Best:	91.00%	86.92%	71.34%
	Std:	0.0158	0.0199	0.0107
POA	Mean:	92.33%	87.32%	72.36%
	Best:	94.25%	91.16%	74.22%
	Std:	0.0102	0.0229	0.0113
GA	Mean:	91.35%	87.70%	75.17%

	Best:	94.00%	88.92%	76.52%
	Std:	0.0165	0.0089	0.0104
COA	Mean:	81.05%	76.29%	63.54
	Best:	84.25%	77.92%	64.45%
	Std:	0.0159	0.0099	0.0064
OOA	Mean:	81.00%	74.08%	61.47%
	Best:	88.00%	80.20%	66.20%
	Std:	0.0420	0.0285	0.0236
ZOA	Mean:	83.83%	78.29%	65.81%
	Best:	88.75%	82.08%	69.27%
	Std:	0.0282	0.0177	0.0185
TSO	Mean:	82.40%	77.59%	64.74%
	Best:	85.25%	80.72%	67.75%
	Std:	0.0209	0.0178	0.0143

Table 4: Comparison of the results

6 Conclusion

The Chaos Enhanced Superb Fairywren Optimization Algorithm (CE-SFOA) was introduced in this study to optimize coverage in Wireless Sensor Networks (WSNs). By incorporating chaotic dynamics into the position update rules and behavioral components of the conventional SFOA, the algorithm achieved an improved balance between extensive exploration and detailed exploitation of the search space. Experimental simulations conducted under three distinct scenarios demonstrated that the CE-SFOA consistently outperformed the original SFOA and seven other metaheuristic approaches in terms of convergence speed, coverage accuracy, and robustness. These enhancements in coverage performance and convergence stability are primarily attributed to the ergodic properties and high randomness quality of the Cubic map, which maintained diversity within the population while mitigating premature convergence. The evidence suggests that the CE-SFOA is a strong candidate for optimizing WSN deployment, resulting in a more uniform sensor allocation, reduced redundancy, and increased coverage efficiency. The findings also underscore the role of chaotic modulation as a general and effective method for enhancing the metaheuristic optimization strategies.

The relevance of CE-SFOA is not restricted to theoretical optimization, and it can be applied to several practical domains because of its uniform and adaptive sensor deployment capabilities. It provides balanced coverage for detecting changes in temperature and pollution levels during environmental monitoring. In disaster management and defense surveillance, sensors can be placed in strategic positions to detect events quickly with the least redundancy. Similarly, it is envisaged that smart agriculture and urban infrastructure monitoring will benefit from the energy-efficient and scalable deployment properties. These potential applications show that the proposed algorithm provides enhanced computational performance, along with practical value for dynamic and large-scale WSN implementations.

Furthermore, the flexibility provided by CE-SFOA can be used effectively for adapting to various proposed applications in the domains of IoT and smart systems. In smart transportation systems, CE-SFOA can be used to position sensors correctly in networks designed for transportation

surveillance and accident detection, ensuring efficient data retrieval with minimum latency. Sensors in networks associated with continuous patient surveillance can be positioned in smart healthcare systems using CE-SFOA, ensuring reliable connectivity with minimal communication interference. Additionally, error detection and processing can be done more effectively using CE-SFOA in industrial automation networks, ensuring optimal sensing of machines and production lines. Each of these applications demonstrates that, although CE-SFOA was designed specifically for static WSNs, it also possesses potential for dynamic and hybrid networks, further confirming that CE-SFOA is an innovative tool in the emerging domain of sensing systems.

Future research may pursue several avenues. A significant extension involves evaluating scalability by applying CE-SFOA to large-scale WSN deployments characterized by heterogeneous sensor ranges, irregular node placements, and very high node densities. Another direction involves integrating hybrid approaches with learning-based mechanisms, such as reinforcement learning, adaptive fuzzy control, or opposition-based strategies, to enhance precision and reduce overall computational burden. Energy-related objectives should also be incorporated, enabling the algorithm to jointly optimize coverage and network longevity, which is critical because power consumption is often the primary limitation in WSNs.

Examining algorithms in mobile and time-varying wireless sensor network (WSN) environments, where sensor positions or targets change over time, is crucial for assessing their robustness in response to environmental fluctuations. Such an assessment would demonstrate its applicability to practical scenarios, including disaster management, defense operations, and smart agriculture. Further theoretical exploration of the CE-SFOA, with a focus on convergence behavior, computational complexity, and sensitivity to parameter selection, could yield deeper mathematical insights and offer guidance for optimizing its chaotic components.

References

- [1] Zhou, Z., Xu, J., Zhang, Z., Lei, F., & Fang, W. (2017). Energy-efficient optimization for concurrent compositions of WSN services. *IEEE Access*, 17, 1–15.
- [2] N. Gupta, N. Kumar and S. Jain, "Coverage problem in wireless sensor networks: A survey," 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, India, 2016, pp. 1742-1749.
- [3] He, Q., Lan, Z., Zhang, D., Yang, L., & Luo, S. (2022). Improved Marine Predator Algorithm for Wireless Sensor Network Coverage Optimization Problem. *Sustainability*, 14(16), 9944.
- [4] S. E. Bouzid, Y. Seresstou, K. Raouf, M. N. Omri, M. Mbarki and C. Dridi, "MOONGA: Multi-Objective Optimization of Wireless Network Approach Based on Genetic Algorithm," in *IEEE Access*, vol. 8, pp. 105793-105814, 2020.
- [5] Ling, Haifeng, Zhu, Tao, He, Weixiong, Luo, Hongchuan, Wang, Qing, Jiang, Yi, Coverage Optimization of Sensors under Multiple Constraints Using the Improved PSO Algorithm, *Mathematical Problems in Engineering*, 2020, 8820907, 10 pages, 2020.

- [6] Zhang, Yong, Cao, Li, Yue, Yinggao, Cai, Yong, Hang, Bo, A Novel Coverage Optimization Strategy Based on Grey Wolf Algorithm Optimized by Simulated Annealing for Wireless Sensor Networks, *Computational Intelligence and Neuroscience*, 2021, 6688408, 14 pages, 2021.
- [7] Amer, D.A., Soliman, S.A., Hassan, A.F. et al. Enhancing connectivity and coverage in wireless sensor networks: a hybrid comprehensive learning-Fick's algorithm with particle swarm optimization for router node placement. *Neural Comput & Applic* 36, 21671–21702 (2024).
- [8] Anand Prakash, R.K. Yadav, Daya Gupta, Sensor Node Deployment based on OTLBO in WSN, *Procedia Computer Science*, Volume 57, 2015, Pages 988-995, ISSN 1877-0509.
- [9] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67-82, April 1997.
- [10] Jia, H., Zhou, X., Zhang, J. et al. Superb Fairy-wren Optimization Algorithm: a novel metaheuristic algorithm for solving feature selection problems. *Cluster Comput* 28, 246 (2025).
- [11] A. Limane, F. Zitouni, A. Ferhat and R. Lakkbichi, "Binary Electric Eel Foraging Optimization Algorithm for Solving 0-1 Knapsack Problems," 2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS), EL OUED, Algeria, 2024, pp. 1-7.
- [12] Velasco, L., Guerrero, H. & Hospitaler, A. A Literature Review and Critical Analysis of Metaheuristics Recently Developed. *Arch Computat Methods Eng* 31, 125–146 (2024).
- [13] Huang Y, Zhang J, Wei W, Qin T, Fan Y, Luo X, Yang J. Research on Coverage Optimization in a WSN Based on an Improved COOT Bird Algorithm. *Sensors*. 2022; 22(9):3383.
- [14] Baines, P. G. (2008). Lorenz, E.N. 1963: Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences* 20, 130–41.1. *Progress in Physical Geography*, 32(4), 475-480.
- [15] Aouissaoui I, Bakir T, Sakly A, Femmam S (2022) Improved one dimensional piecewise chaotic maps for information security. *J Commun* 17(1):11–16.
- [16] Gao ZM, Zhao J, Zhang YJ (2022) Review of chaotic mapping enabled nature-inspired algorithms. *Math Biosci Eng* 19(8):82158258.
- [17] Holland, J.H. *Genetic Algorithms*. *Sci. Am.* 1992, 267, 66–73.
- [18] Storn, R. On the Usage of Differential Evolution for Function Optimization. In *Proceedings of the North American Fuzzy Information Processing*, Berkeley, CA, USA, 19–22 June 1996; pp. 519–523.
- [19] Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* 2014, 69, 46–61.
- [20] Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems. *Adv. Eng. Softw.* 2017, 114, 163–191.
- [21] Arora, S.; Singh, S. Butterfly Optimization Algorithm: A Novel Approach for Global Optimization. *Soft Comput.* 2019, 23, 715–734.
- [22] Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* 1983, 220, 671–680.

- [23] Mirjalili, S. SCA: A Sine Cosine Algorithm for Solving Optimization Problems. *Knowl.-Based Syst.* 2016, 96, 120–133.
- [24] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems, *Computer-Aided Design*, Volume 43, Issue 3, 2011, Pages 303-315, ISSN 0010-4485.
- [25] Atashpaz-Gargari, E.; Lucas, C. Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition. In *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, Singapore, 25–28 September 2007; pp. 4661–4667.
- [26] H. ZainEldin, M. Badawy, M. Elhosseini, H. Arafat, and A. Abraham, "An improved dynamic deployment technique based-on genetic algorithm (IDDT-GA) for maximizing coverage in wireless sensor networks," *J. Ambient Intell. Humaniz. Comput.*, vol. 11, pp. 4177–4194, 2020.
- [27] W. Chen, Q. Cao, B. Cao, and B. Jin, "An innovative coverage optimization method for smart information monitoring in agricultural IoT using the multi-strategy Pelican optimization algorithm," *Sci. Rep.*, vol. 15, Art. no. 12634, 2025.
- [28] J. Wang, D. Zhu, Z. Ding and Y. Gong, "WSN Coverage Optimization based on Improved Sparrow Search Algorithm," *2023 15th International Conference on Advanced Computational Intelligence (ICACI)*, Seoul, Korea, Republic of, 2023, pp. 1-8.
- [29] Y. Zhang, L. Cao, Y. Yue, Y. Cai, and B. Hang, "A Novel Coverage Optimization Strategy Based on Grey Wolf Algorithm Optimized by Simulated Annealing for Wireless Sensor Networks," *Comput. Intell. Neurosci.*, vol. 2021, Art. ID 6688408, 14 pages, 2021.
- [30] J. Cheng, Y. Fang, and N. Jiang, "Research on Wireless Sensor Networks Coverage Based on Fruit Fly Optimization Algorithm," in *Proc. 2023 IEEE 3rd Int. Conf. Power, Electron. Comput. Appl. (ICPECA)*, Shenyang, China, 2023, pp. 1109-1115.
- [31] Trojovský, P., & Dehghani, M. (2022). Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications. *Sensors*, 22(3), 855.
- [32] J. Pierezan and L. Dos Santos Coelho, "Coyote Optimization Algorithm: A New Metaheuristic for Global Optimization Problems," *2018 IEEE Congress on Evolutionary Computation (CEC)*, Rio de Janeiro, Brazil, 2018, pp. 1-8.
- [33] Dehghani M and Trojovský P (2023) Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems. *Front. Mech. Eng* 8:1126450.
- [34] E. Trojovská, M. Dehghani and P. Trojovský, "Zebra Optimization Algorithm: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm," in *IEEE Access*, vol. 10, pp. 49445-49473, 2022.
- [35] Xie, Lei, Han, Tong, Zhou, Huan, Zhang, Zhuo-Ran, Han, Bo, Tang, Andi, Tuna Swarm Optimization: A Novel Swarm-Based Metaheuristic Algorithm for Global Optimization, *Computational Intelligence and Neuroscience*, 2021, 9210050, 22 pages, 2021.