



Research article

Petal-Spiral search optimizer: A novel optimization algorithm and its application to three-dimensional wireless sensor network coverage optimization

Xiaowei Wang and Jun Bao*

School of Tourism, Huangshan University, Huangshan, Anhui, China

* **Correspondence:** Email: 109047@hsu.edu.cn.

Abstract: Three-dimensional wireless sensor network deployment is a class of optimization problems that are high-dimensional, non-differentiable, and characterized by numerous locally competitive solutions. This paper proposes a novel petal-spiral search optimizer (PSSO). The method applies multi-lobe sign modulation to the step size via the radial component of a rose curve, forming the petal-bloom exploration operator. It employs the dimension-wise normalized modulus of an epitrochoid to construct the epitrochoid exploitation operator. A bloom weight, composed of a polynomial and trigonometric function, simultaneously governs the selection probability between the two operators and the internal fusion ratio of the exploration step size. A greedy acceptance strategy is further adopted to select and update the population. To examine the effective boundary of the proposed method, this study conducts 30 independent repeated experiments on four three-dimensional wireless sensor network coverage scenarios and 12 benchmark functions from CEC2022, along with a comparative analysis of six single-mechanism ablation variants. In the primary scenario, PSSO achieves an average coverage of 0.836504 with a standard deviation of 0.010121. Its worst result over 30 runs (0.814382) still surpasses the best historical results of artificial hummingbird algorithm (AHA), boomerang aerodynamic ellipse optimizer (BAEO), crayfish optimization algorithm (COA), electric eel foraging optimization (EEFO), and zebra optimization algorithm (ZOA). Compared with the suboptimal algorithm AHA, PSSO improves the average coverage by 5.91%. In the other three scenarios where the number of nodes, sensing radius, and region shape all vary, PSSO simultaneously ranks first in best, worst, and mean values, with relative improvements of 3.67%, 8.23%, and 9.21%, respectively. The total runtime of PSSO is the shortest, far lower than those of COA, EEFO, ZOA, and especially BAEO. The CEC2022 results show that PSSO achieves or ties for the optimal mean on six out of 12 functions. The ablation

experiment shows that the complete model achieves the highest search upper bound (0.860452) and outperforms all variants, further validating the effectiveness of the overall architecture of the proposed algorithm.

Keywords: metaheuristic optimization; geometric search operators; petal-spiral search optimizer; component ablation; three-dimensional wireless sensor networks; coverage optimization

Mathematics Subject Classification: 90C26, 90C59, 68W50

1. Introduction

1.1. Research background

Numerous decision-making problems in engineering design, production scheduling, machine learning, communication networks, and resource allocation reduce to complex optimization tasks. These problems frequently exhibit high dimensionality, non-convexity, multimodality, non-differentiability, and intricate constraints [1–3]. Such characteristics cause exact mathematical algorithms to suffer from excessive computational burdens while making gradient-dependent techniques difficult to apply directly.

Swarm intelligence and evolutionary computation offer gradient-free approximation paradigms for these challenges. Genetic algorithms and differential evolution drive search mechanisms via population-level mutation and selection [4,5]. Particle swarm optimization, grey wolf optimizer, and whale optimization algorithm shape search trajectories through information exchange among individuals [6–8]. Another family of methods constructs updating rules by mimicking physical phenomena or social behaviors [9–11]. Recently developed metaheuristics, such as the artificial hummingbird algorithm, snow ablation optimizer, and exponential distribution optimizer, have further enriched operator paradigms [12–14]. Nevertheless, the no-free-lunch theorem demonstrates that under its stated assumptions, no optimization algorithm can outperform all others when averaged across all unrestricted problem classes [15]. Consequently, performance assertions regarding newly proposed methods must be rigorously contextualized within specified problem domains and experimental conditions.

Optimization research is increasingly transitioning from purely metaphor-driven frameworks toward knowledge-guided and state-aware paradigms. Pan et al. introduced a knowledge-driven dual-population approach for distributed energy-efficient parallel machine scheduling by integrating co-evolution, differential evolution, non-dominated sorting, and knowledge-based local search [16]. Wang et al. developed a dual Q-learning adaptive scheduling framework to address assembly job shops under assembly time uncertainties [17]. Zhao et al. constructed a heterogeneous graph reinforcement learning framework utilizing problem-aware neighborhood aggregation and cross-operation prompt attention to solve dynamic flexible job shop scheduling [18]. In communication networks, hybrid metaheuristics have been applied to tasks, such as quality of service classification, as demonstrated by the BERPO framework combining Al-Biruni earth radius optimization with puma optimization [19]. Although these approaches differ from the geometric operator design investigated herein, they collectively underscore that operator structures merit explicit modeling and empirical validation.

1.2. Research gap and motivation

Candidate generation mechanisms in existing metaheuristics predominantly rely on Gaussian perturbations, population difference vectors, or attraction toward elite individuals. While effective, these mechanisms organize search space implicitly, making it difficult to ascertain whether an operator systematically allocates search directions or merely alters the marginal distribution of step sizes. Parametric curves provide a structured avenue to explicitly characterize the angular structure and radial scale of step sizes. The multi-petal oscillations of rose curves and the dense looping of epitrochoids inherently provide controllable structural information.

Transferring two-dimensional parametric curves into high-dimensional decision spaces presents technical challenges. Applying curve equations independently to each decision variable destroys original angular coupling relationships. Conversely, collapsing two spatial coordinates into a single scalar amplitude results in the loss of directional information. The compromise adopted in this study treats geometric properties of curves as structured multiplicative modulators rather than direct displacement vectors. The radial component $\cos(k\theta)$ of a rose curve supplies sign-alternating and amplitude-varying multiplicative modulators across dimensions. Meanwhile, the dimension-wise normalized coordinate modulus of an epitrochoid forms a bounded yet long-tailed fine-grained perturbation envelope, leaving directional randomness to be governed by Gaussian vectors. This strategy preserves non-independent and identically distributed properties stemming from geometric structures without sacrificing high-dimensional degrees of freedom via forced lower-dimensional embeddings.

Mathematical novelty does not guarantee empirical performance. This study does not presuppose that geometric operators inherently surpass Gaussian alternatives. Instead, it addresses this question through single-mechanism ablation to clarify the applicable operational boundaries of the method.

1.3. Main contributions and structure

The primary contribution of this work lies in formulating a structured high-dimensional petal exploration operator. By controlling the spatial frequency of multi-petal modulation via a blooming count that increases with iterations, a single update applies displacements of varying signs and magnitudes across dimensions, manifesting in three-dimensional deployment tasks as a coordinated spatial rearrangement of sensor nodes rather than independent point-wise random perturbations. A secondary contribution is the formulation of an epitrochoid exploitation operator that leverages a contracting fixed-circle radius, an oscillating rolling-circle radius, and a decreasing tracing-point offset to generate a coarse-to-fine perturbation envelope. Dimension-wise normalization ensures that perturbation magnitudes remain unaffected by absolute geometric parameter scales. Furthermore, a continuous composite blooming weight integrating polynomial and trigonometric functions is formulated to simultaneously govern operator selection probabilities and the internal blending ratio between geometric and differential terms, ensuring smooth transition management from exploration to exploitation. In addition, a low-frequency migration reset mechanism is introduced to restore population diversity at negligible computational cost without altering the primary evaluation budget per generation. Finally, comprehensive empirical diagnostics are conducted across four 3D coverage scenarios, the full CEC2022 benchmark suite, six single-mechanism ablation variants, and interval

estimations derived from repeated runs, providing mechanistic explanations for functions where the algorithm experiences performance degradation.

The remainder of this paper is organized as follows. Section 2 establishes the optimization model for three-dimensional wireless sensor network coverage and outlines its intrinsic computational challenges. Section 3 details the operator formulations, algorithmic execution flow, and computational complexity of petal-spiral search optimizer (PSSO). Section 4 presents empirical results regarding coverage optimization, multi-scenario generalization, and component ablation. Section 5 evaluates performance on CEC2022 functions and discusses operational boundaries. Section 6 concludes the paper.

2. Three-dimensional wireless sensor network coverage optimization model

2.1. Problem description

Let the target monitoring area be defined as a cuboid region $\Omega = [0, L] \times [0, W] \times [0, H]$, where L , W , and H denote side lengths along the x , y , and z axes, respectively. A total of M homogeneous sensor nodes are deployed inside Ω , where the spatial position of the j -th node is denoted as $q_j = (q_{j,x}, q_{j,y}, q_{j,z})^T \in \Omega$. All nodes share a common sensing radius r_s . The optimization objective is to determine the node coordinate set $\{q_1, q_2, \dots, q_M\}$ such that the network coverage over Ω is maximized. In algorithm encoding, a candidate deployment plan is flattened into a real-valued decision vector X of dimension $D = 3M$, where components from $3j - 2$ to $3j$ correspond to the three Cartesian coordinates of node q_j . Feasible intervals for each component are constrained by the side length of the corresponding spatial axis.

2.2. Three-dimensional boolean sensing model

This paper adopts the three-dimensional Boolean sensing model to characterize the sensing capability of sensor nodes. Under this model, the sensing probability of sensor node q_j for an arbitrary point $Z = (Z_x, Z_y, Z_z)$ in space is defined as Eqs 1 and 2.

$$P_s(Z, q_j) = \begin{cases} 1, & \text{if } \|Z - q_j\| \leq r_s \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

$$\|Z - q_j\| = \sqrt{(Z_x - q_{j,x})^2 + (Z_y - q_{j,y})^2 + (Z_z - q_{j,z})^2} \quad (2)$$

When multiple nodes operate collaboratively, target point Z is considered covered by the network if detected by at least one sensor node. The joint coverage indicator $C(Z)$ is given by Eq 3.

$$C(Z) = 1 - \prod_{j=1}^M (1 - P_s(Z, q_j)) \quad (3)$$

2.3. Coverage rate definition and optimization objective

To evaluate coverage rate computationally, the monitoring space is uniformly discretized along the three coordinate axes with a grid resolution of δ , yielding a target grid point set $\mathbb{Z}=\{Z_1, Z_2, \dots, Z_{N_g}\}$. The total number of spatial grid points N_g is calculated via Eq 4.

$$N_g = \left\lfloor \frac{L}{\delta} + 1 \right\rfloor \times \left\lfloor \frac{W}{\delta} + 1 \right\rfloor \times \left\lfloor \frac{H}{\delta} + 1 \right\rfloor \quad (4)$$

Smaller grid spacing produces a denser point distribution and higher approximation precision at the expense of greater computational cost per evaluation. The network coverage rate $\eta(X)$ over the domain is defined as the proportion of covered grid points, as shown in Eq 5.

$$\eta(X) = \frac{1}{N_g} \sum_{k=1}^{N_g} C(Z_k) \quad (5)$$

The coverage rate satisfies $\eta(X) \in [0,1]$, where a value of unity denotes that all target grid points are detected by at least one sensor. Because PSSO and baseline algorithms are structured for minimization, the actual fitness function $f(X)$ is defined as the negative coverage rate, expressed in Eq 6.

$$\min_X f(X) = -\eta(X) \quad (6)$$

where $X \in [0, L]^{3M}$.

2.4. Problem characteristics and algorithm suitability

The difficulty of solving this deployment problem stems from three primary characteristics. A deployment plan containing M nodes corresponds to $3M$ continuous variables, whereas grid-based Boolean coverage objectives are intrinsically non-differentiable. Small node displacements alter fitness values only when grid points cross sensing sphere boundaries, causing the fitness landscape to exhibit step-like plateau structures where local gradient or infinitesimal step adjustments fail to yield feedback. In addition, local competing solutions are abundant. Excessively close nodes induce redundant overlap, while overly sparse placements leave spatial blind spots. These contrasting configurations frequently produce nearly identical objective values, causing optimization trajectories to wander between plateau regions. Furthermore, nodes situated near domain boundaries project only portions of their sensing spheres inside the monitoring region, making corner regions difficult to cover and rendering optimal layouts sensitive to crucial boundary node positions.

These characteristics impose dual functional requirements on search operators. Operators must enable coordinated group displacements across plateaus to reshape global layout patterns, while also providing bounded fine-grained adjustments around high-quality configurations to fill localized boundary gaps. The multi-petal sign modulation of the proposed petal operator introduces structured directional differentiation across dimensions, satisfying global spatial rearrangement requirements. Concurrently, the contracting perturbation envelope of the epitrochoid operator fulfills localized boundary adjustment needs. This structural compatibility highlights why the method is conceptually

suitable to the problem, whereas empirical evidence regarding superiority over simplified operator alternatives is evaluated through ablation experiments in Section 4.4.

3. PSSO

3.1. Inspiration

The design inspiration for PSSO originates from morphological features observed in flowering plants. During blooming, flower petals radiate outward from the receptacle. In a corolla exhibiting k -fold symmetry, each petal defines a primary radial extension direction, forming a multi-directional spatial arrangement. As blooming progresses, resolvable petal orientation counts increase. Mapping this morphology to search behavior corresponds to scanning the decision space along multiple directional vectors simultaneously while refining directional resolution over iterations. During the seed-setting stage, central seeds align tightly along logarithmic spirals, achieving high packing density within bounded areas. This corresponds to generating dense, bounded perturbations within the neighborhood of promising candidate solutions. Guided by these insights, PSSO uses the radial component of rose curves and epitrochoid coordinates as geometric templates to construct exploration and exploitation operators, respectively, employing a blooming weight to regulate smooth transitions.

3.2. Initialization

Let the search space dimensionality be D , the population size be N , and the lower and upper bounds of the search space be lb and ub , respectively. The initial position $X_i^{(0)}$ of the i -th search agent is generated uniformly at random within the search space according to Eq 7:

$$X_{i,d}^{(0)} = lb_d + r_{i,d} \cdot (ub_d - lb_d) \quad (7)$$

where $i = 1, 2, \dots, N$, $d = 1, 2, \dots, D$, $r_{i,d} \sim U(0,1)$ is a uniformly distributed random number on $[0,1]$, and lb_d and ub_d are the lower and upper bounds of the d -th dimension, respectively. After initialization, the fitness $f(X_i^{(0)})$ of each agent is evaluated, the personal best position of each agent is set to $pbest_i = X_i^{(0)}$, and the global best position $gbest$ is determined.

PSSO maintains the following information structure: the personal best position $pbest_i$ of each agent and its associated fitness f_i^{pbest} ; the global best position $gbest$ of the population and its fitness f^{gbest} . In each iteration, PSSO probabilistically selects one of two strategies for every search agent: the petal bloom exploration strategy is executed with probability P_{bloom} , and the spiral convergence exploitation strategy is executed with probability $1 - P_{bloom}$. The probability P_{bloom} equals the bloom weight $w(\tau)$, the design of which is described in Section 3.3.

3.3. Adaptive bloom weight

Define the normalized iteration progress $\tau = t / T$, where t is the current iteration index and T is the maximum number of iterations. The bloom weight $w(\tau)$ quantifies the “degree of flower blooming” and serves as the core parameter governing the switching between the two search phases: when $w(\tau)$ approaches 1, the flower is in full bloom and the population primarily executes the petal bloom

exploration strategy; when $w(\tau)$ approaches the lower bound w_{min} , the flower has entered the seed-setting stage and the population predominantly executes the spiral convergence exploitation strategy. The bloom weight is defined by Eq 8 to 10:

$$\beta(\tau) = \beta_0 + \sin(2\pi\tau) \quad (8)$$

$$\gamma(\tau) = \gamma_0 + \gamma_1 \cos(\pi\tau) \quad (9)$$

$$w(\tau) = \max\left(w_{min}, (1 - \tau^{\beta(\tau)}) \cdot \left(\cos\left(\frac{\pi\tau}{2}\right)\right)^{\gamma(\tau)}\right) \quad (10)$$

where β_0 is the baseline exponent for the polynomial decay factor $(1-\tau^\beta)$, governing its overall decay rate; γ_0 and γ_1 are the baseline power and the oscillation amplitude of the trigonometric decay factor, respectively; and w_{min} is the lower bound of the bloom weight, ensuring that a minimal probability of executing the petal bloom strategy is retained even in the final iterations. Based on $w(\tau)$, the selection probabilities of the two strategies are:

$$P_{bloom}(\tau) = w(\tau) \quad (11)$$

$$P_{spiral}(\tau) = 1 - P_{bloom}(\tau) \quad (12)$$

3.4. Petal bloom exploration strategy

The petal bloom exploration strategy simulates the radial expansion of petals along multiple directions during blooming. It employs the radial component of the rose curve as a directional modulation factor for the step length, enabling search agents to perform multi-directional spatial scanning along structurally organized petal-like trajectories.

The petal count k is designed as a time-varying quantity that increases with bloom progress:

$$k(\tau) = k_0 + \lfloor \vartheta \cdot \tau \rfloor \quad (13)$$

where k_0 is the initial petal base count, ϑ is the petal growth rate parameter controlling the maximum increment in petal count over the entire search process, and $\lfloor \cdot \rfloor$ denotes the floor function.

The polar equation of the rose curve, $r = \cos(k\theta)$, describes the petal shape, in which the radial value r exhibits petal-like pulse oscillations between positive values and zero as the polar angle θ varies. PSSO independently samples a petal angle θ_d for each agent i in each dimension d and computes the corresponding radial pulse value:

$$\theta_d \sim U(0, 2\pi), \quad d = 1, 2, \dots, D \quad (14)$$

$$\rho_d = \cos(k(\tau) \cdot \theta_d) \quad (15)$$

Each component of the vector $\rho = (\rho_1, \dots, \rho_D)^T$ takes values in $[-1, 1]$, with its sign and magnitude jointly determined by the angle and the petal count. This vector acts as a multiplicative directional modulation factor applied to the base step length, enabling step lengths across different dimensions to inherit the spatial symmetry characteristics of the petal bloom configuration. The radial pulse is fused with differential direction information pointing toward the global best position via a weighted combination governed by w , yielding the final petal bloom step:

$$s_i = w \cdot \alpha_p \cdot \rho \cdot r_1 \cdot (ub - lb) + (1 - w) \cdot (gbest - X_i^{(t)}) \cdot r_2 \quad (16)$$

where r_1 is a D -dimensional standard normal random vector, r_2 is a D -dimensional uniform random vector, and α_p is the petal step length scaling coefficient controlling the amplitude of the petal pulse component. The candidate updated position of the agent is:

$$X_i^{(t+1)} = X_i^{(t)} + s_i \quad (17)$$

3.5. Spiral convergence exploitation strategy

The spiral convergence strategy serves as the local exploitation engine of PSSO, simulating the morphology of flower-center seeds converging spirally toward the center. This strategy employs the dense spiral pattern of the epitrochoid as a perturbation template to generate high-density fine-grained search perturbations in the vicinity of known high-quality positions.

The epitrochoid is defined by three geometric parameters: the fixed circle radius R , the rolling circle radius r_e , and the tracing point offset d_e . PSSO designs these three parameters as time-varying quantities that evolve with iteration progress τ , simulating the progressive tightening of the spiral trajectory from loose and expansive to compact and dense:

$$R(\tau) = R_{max}(1 - \tau) + R_{min} \quad (18)$$

$$r_e(\tau) = r_0 + \varepsilon_r \sin(\omega_r \tau) \quad (19)$$

$$d_e(\tau) = d_{max}(1 - \tau) + d_{min} \quad (20)$$

where R_{max} and R_{min} are the initial and terminal values of the fixed circle radius, respectively; r_0 is the baseline rolling circle radius; ε_r is its oscillation amplitude; ω_r is the oscillation angular frequency; and d_{max} and d_{min} are the initial and terminal values of the tracing point offset, respectively.

An angle parameter φ_d is independently sampled for each agent i in each dimension d :

$$\varphi_d \sim U(0, \Phi_s), \quad d = 1, 2, \dots, D \quad (21)$$

where Φ_s is the angle sampling range parameter. The epitrochoid coordinate components in each dimension are then computed as:

$$\xi_d = (R + r_e) \cos \varphi_d - d_e \cos \left(\frac{R + r_e}{r_e} \varphi_d \right) \quad (22)$$

$$\eta_d = (R + r_e) \sin \varphi_d - d_e \sin \left(\frac{R + r_e}{r_e} \varphi_d \right) \quad (23)$$

The modulus is computed and normalized as:

$$m_d = \sqrt{\xi_d^2 + \eta_d^2} \quad (24)$$

$$\hat{m}_d = \frac{m_d}{\max m_d + \varepsilon} \quad (25)$$

where ε is a small positive constant to prevent division by zero and the normalized modulus satisfies $\hat{m}_d \in [0,1]$. The reference position for the spiral convergence strategy, conceptually acting as the flower center, is probabilistically selected from two candidates:

$$b_i = \begin{cases} pbest_i, & \text{if } rand < p_b \\ gbest, & \text{otherwise} \end{cases} \quad (26)$$

where p_b is the personal memory preference probability parameter. The normalized spiral modulus is combined with a normal random vector and a time-varying scaling factor to synthesize the final convergence perturbation:

$$\delta_i = \hat{m} \cdot r_3 \cdot (\alpha_s \cdot (1 - \tau)^2 \cdot (ub - lb)) \quad (27)$$

where r_3 is a D -dimensional standard normal random vector and α_s is the spiral step length scaling coefficient controlling the initial proportion of the perturbation amplitude relative to the search space span. The candidate updated position of the agent is:

$$X_i^{(t+1)} = b_i + \delta_i \quad (28)$$

3.6. Population migration reset

To prevent the population from losing its capacity to discover new regions due to excessive aggregation near the current optimal area, PSSO introduces a periodic population migration-reset mechanism, simulating the phenomenon of new seeds germinating at random distant locations after some plants wither in a natural garden. Whenever the iteration index t reaches an integer multiple of νN (where ν is the migration period factor), the individual with the worst fitness in the population is identified and uniformly randomly repositioned within the search space:

$$X_{worst}^{(t+1)} = lb + r \cdot (ub - lb), \quad r \sim U(0,1)^D \quad (29)$$

After resetting, the fitness of this individual is re-evaluated and its personal best record is updated accordingly.

3.7. Framework of the PSSO algorithm

The complete algorithmic procedure of PSSO is presented in Algorithm 1.

Algorithm 1. Pseudocode of the PSSO algorithm.

Input: Objective function f ; maximum iterations T ; population size N ; dimensionality D ; search bounds lb, ub .

Initialize population X_i ($i = 1, \dots, N$) randomly within $[lb, ub]$;

Evaluate fitness for all agents; set personal best $pbest_i$ and global best $gbest$;

1: **for** $t = 1$ to T **do**

2: Calculate progress ratio $\tau = t/T$;

3: Update control parameters: bloom weight w , petal count k , and spiral geometry parameters;

4: **for** each search agent X_i **do**

5: **if** $rand() < w$ **then**

```

6:      # Phase 1: Petal Bloom Exploration
7:      Generate directional vector guided by multi-lobed Rose curve;
8:      Compute step size using population differential information;
9:      Update candidate position  $X_i$  via petal-guided exploration;
10:     else
11:         # Phase 2: Spiral Convergence Exploitation
12:         Select target reference base by Eq (26);
13:         Compute dense perturbation using Epitrochoid spiral coordinates;
14:         Update candidate position  $X_i$  around the selected reference base;
15:     end if
16:     Enforce search boundary clipping on  $X_i$ ;
17:     Evaluate fitness  $f(X_i)$ ;
18:     Update  $X_i$ ,  $pbest_i$ , and  $gbest$  using greedy selection;
19: end for
20: if  $t$  reaches migration period then
21:     # Phase 3: Population Migration Reset
22:     Randomly re-initialize the worst-performing agent in  $[lb, ub]$ ;
23:     Update its fitness,  $pbest$ , and  $gbest$  if improved;
24: end if
25: end for
Output. Return global best solution  $gbest$  and its fitness  $f^{gbest}$ .

```

3.8. Computational complexity of PSSO

In each iteration of PSSO, N agents each execute one search strategy, involving $O(D)$ trigonometric and vector operations per agent, amounting to a total per-iteration cost of $O(N \cdot D)$. The objective function is evaluated N times per iteration, with a single evaluation cost of $O(C_f)$, yielding a total evaluation cost of $O(N \cdot C_f)$. Greedy selection and elite update involve $O(D)$ vector copying operations. The migration reset is triggered once every νN iterations, requiring an $O(N)$ search for the worst individual and one $O(C_f)$ function evaluation. Consequently, the overall time complexity of PSSO is $O(T \cdot N \cdot (D + C_f))$, and the number of function evaluations per iteration is consistently N plus a negligible number of migration evaluations, which is on par with most mainstream population-based optimization algorithms and introduces no additional evaluation overhead.

4. Experimental design and results analysis

4.1. Experimental setup

All experiments were implemented in Python 3 using the NumPy numerical computing library. To ensure statistical reliability, each experiment was conducted over 30 independent runs, with four statistical metrics reported: mean (Mean), standard deviation (Std), best value (Best), and worst value (Worst).

The parameter settings of PSSO are as follows: population size $N = 30$; bloom weight parameters $\beta_0 = 2$, $\gamma_0 = 1$, $\gamma_1 = 0.5$, $w_{\min} = 0.01$; initial petal count $k_0 = 2$, petal growth rate $\vartheta = 5$; petal step coefficient $\alpha p = 0.1$; spiral geometric parameters $R_{\max} = 5$, $R_{\min} = 0.5$, $r_0 = 2$, $\varepsilon r = 0.1$, $\omega r = 4\pi$, $d_{\max} = 3$, $d_{\min} = 0.2$; angle sampling range $\Phi s = 6\pi$; spiral step coefficient $\alpha s = 0.05$; personal memory preference probability $pb = 0.3$; and migration period factor $\nu = 2$.

The WSN experimental model parameters are specified as follows: spatial dimensions $L = W = H = 50$ m, number of sensor nodes $M = 40$, sensing radius $r_s = 10$ m, maximum number of iterations $T = 100$, search dimensionality $D = 120$, and the search range for each dimension is $[0, 50]$.

Five recently published metaheuristic algorithms were selected for comparison: the artificial hummingbird algorithm (AHA) [20], the boomerang aerodynamic ellipse optimizer (BAEO) [21], the crayfish optimization algorithm (COA) [22], the electric eel foraging optimization (EEFO) [23], and the zebra optimization algorithm (ZOA) [24]. The population size of all comparison algorithms is uniformly set to $N = 30$, and the remaining control parameters follow the recommended values reported in their respective original publications. To ensure a fair comparison, all algorithms share synchronized random seeds for population initialization in each independent run.

4.2. WSN coverage optimization experimental results

The evaluation proceeds from visual inspection of deployment geometry, through convergence trajectory analysis, to statistical robustness assessment over 30 independent runs. Figures 1 and 2 present, respectively, the initial random deployment schemes, the best deployment schemes obtained across 30 runs, and the corresponding convergence curves for all six algorithms: PSSO, AHA, BAEO, ZOA, COA, and EEFO.

As observed from Figures 1 and 2, the overall coverage rate is relatively low under random initialization with 40 sensor nodes. After optimization, the best deployment scheme obtained by PSSO exhibits the most balanced spatial distribution: nodes form a three-dimensionally dispersed layout within the $50 \times 50 \times 50$ m monitoring volume, with a harmonious ratio between inter-node spacing and sensing radius that effectively minimizes coverage redundancy while filling the blind spots of the initial scheme—including boundary regions and corner positions. By comparison, the best deployment obtained by AHA shows slightly inferior overall spatial uniformity, with minor node clustering observable in localized regions. The optimized schemes of BAEO, EEFO, COA, and ZOA are noticeably inferior to PSSO in terms of uniform node distribution, exhibiting particularly insufficient coverage density at the edges of the monitoring region.

The right-hand panels of Figures 1 and 2 present the complete ensembles of coverage-rate trajectories generated by 30 independent runs of each algorithm. In Figure 1, the PSSO curves rise steeply within the first 20 to 30 iterations and thereafter remain tightly clustered at the highest coverage band, exhibiting only modest run-to-run scatter and no late-stage reversals or plateaus. The 30 trajectories of AHA follow a similar but slower initial climb and begin to flatten after the mid-search phase, while the BAEO trajectories display both greater vertical dispersion and systematically lower terminal values. Figure 2 likewise shows the 30-run collections for ZOA, COA, and EEFO. The EEFO paths initially parallel those of AHA yet stagnate earlier, whereas ZOA and especially COA exhibit the widest scatter, the longest mid-run plateaus, and the lowest clusters of final coverage, underscoring their greater sensitivity to random initialization and their tendency to become trapped in locally competitive node layouts.

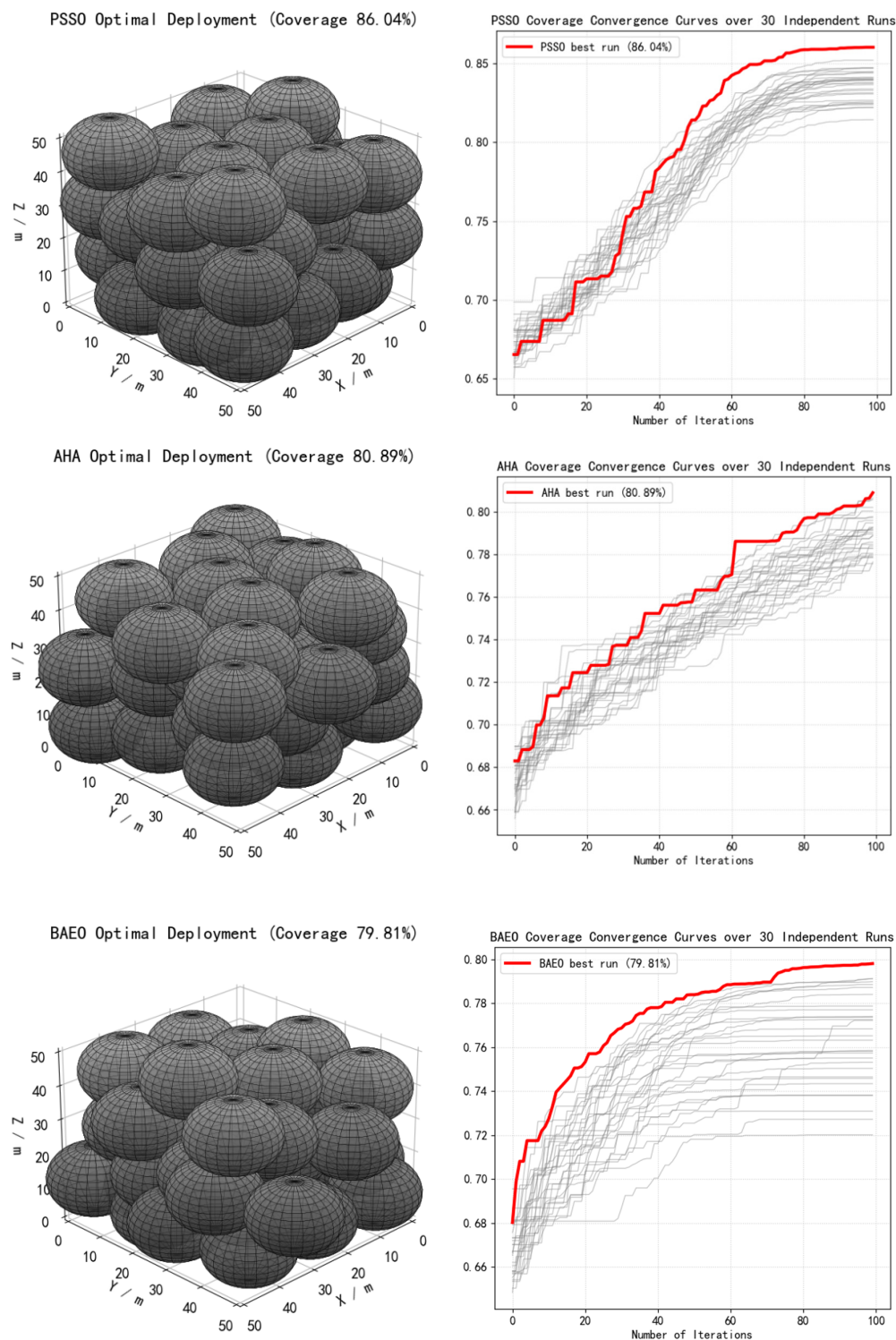


Figure 1. Initial schemes, best deployment schemes, and convergence curves for PSSO, AHA, and BAEO on the WSN coverage optimization problem.

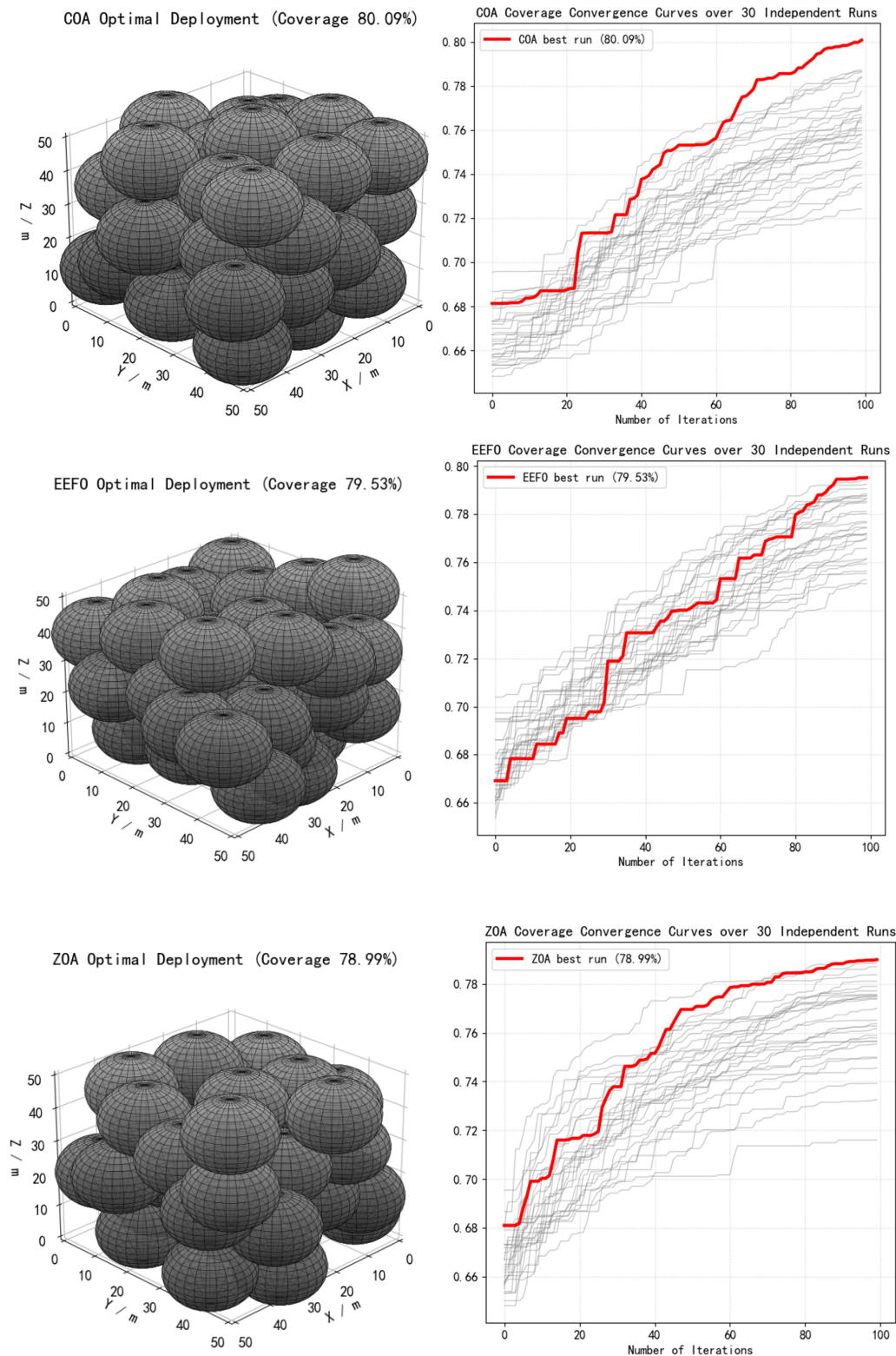


Figure 2. Initial schemes, best deployment schemes, and convergence curves for ZOA, COA, and EEFO on the WSN coverage optimization problem.

Table 1 summarizes the statistical results of all six algorithms on the three-dimensional WSN coverage optimization problem over 30 independent runs, reporting five metrics: best value, worst value, mean, standard deviation, and total running time.

Table 1. Coverage optimization experimental results for six algorithms.

	best	worst	mean	std	time
PSSO	0.8604	0.8143	0.8365	0.0101	1505.2
AHA	0.8089	0.7756	0.7898	0.0091	2449.3
BAEO	0.7981	0.7201	0.7632	0.0207	6717.1
COA	0.8009	0.7243	0.7620	0.0178	2974.0
EEFO	0.7953	0.7512	0.7754	0.0128	2692.2
ZOA	0.7899	0.7161	0.7655	0.0174	4789.5

Based on the experimental results summarized in Table 1, three principal conclusions can be drawn. First, PSSO attains the highest mean coverage rate of 0.8365, surpassing the second-best algorithm AHA by 0.0467, which represents a relative improvement of 5.91 percent. The 95 percent confidence interval for the mean coverage of PSSO is from 0.8329 to 0.8401, which is entirely above that of AHA from 0.7865 to 0.7931, with a gap of approximately 0.0398 between the lower bound of PSSO and the upper bound of AHA. Second, and more importantly, the worst-case coverage achieved by PSSO over 30 independent runs is 0.8143, which is still higher than the best results obtained by all competing algorithms, including AHA at 0.8089, EEFO at 0.7953, COA at 0.8009, ZOA at 0.7899, and BAEO at 0.7981. Third, this implies that every single execution of PSSO outperforms the best records of the other five methods across their 30 runs.

With regard to computational cost, PSSO requires a total runtime of 1505.2 seconds, which is the shortest among all six algorithms. This runtime is substantially shorter than that of AHA at 2449.3 seconds (approximately 38.5 percent less), COA at 2974.0 seconds, EEFO at 2692.2 seconds, ZOA at 4789.5 seconds, and especially BAEO at 6717.1 seconds. Under identical evaluation budgets, the additional computational burden of PSSO per generation arises from the dimension-wise evaluation of trigonometric functions and epitrochoid coordinates, whereas AHA relies primarily on basic vector operations. In contrast, the higher runtime of COA (and especially BAEO) is attributable to its evaluation of multiple candidate solutions per individual in each iteration, which substantially increases the total number of function evaluations.

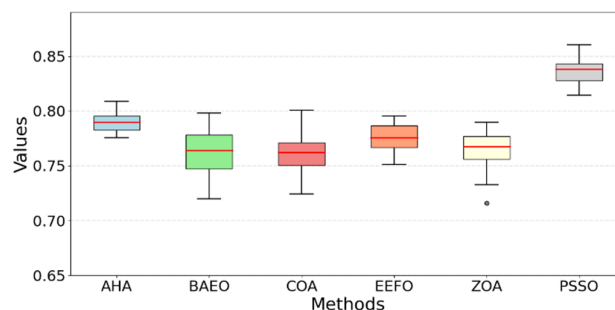


Figure 3. Boxplot comparison of different algorithms.

To visually compare the central tendency and dispersion of results across algorithms, Figure 3 presents box plots of the final coverage rates over 30 independent runs.

As shown in the boxplot in Figure 3, PSSO exhibits the highest box position, with a median of approximately 0.837, a narrow interquartile range, short whiskers, and very few outliers, confirming its ability to converge stably to high coverage levels. AHA occupies the second-highest box position, with a median roughly 4.7 percentage points below that of PSSO and a comparably narrow interquartile range, reflecting good execution consistency. EEFO yields a median close to 0.776 with a moderately wider spread. ZOA and BAE0 display broader interquartile ranges and extended lower whiskers reaching 0.716 and 0.720, respectively, indicating susceptibility to degraded solutions in individual runs. COA registers the lowest box position, with a median near 0.753 and whiskers extending to 0.719, showing high variability and frequent entrapment in poor local optima. These boxplot distributions are in good agreement with the numerical results presented in Table 1.

Figure 4 displays the complete convergence trajectories of best coverage rate over 100 iterations for all six algorithms.

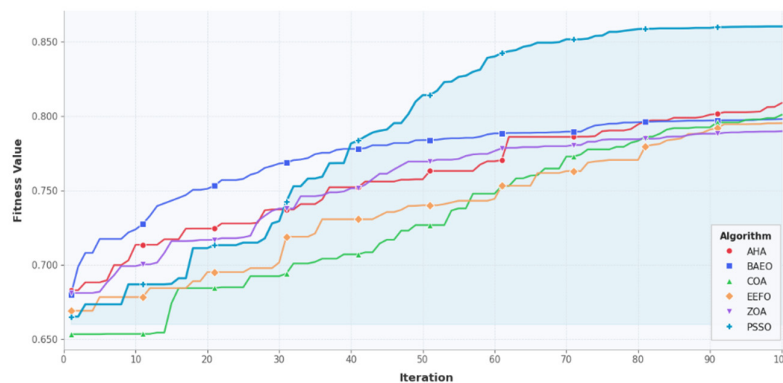


Figure 4. Convergence curves of best coverage rate for all six optimization algorithms over 100 iterations.

Figure 4 isolates the single best-performing run of each algorithm and traces its coverage-rate evolution over the full 100 iterations. The optimal execution of PSSO produces the steepest early gain, overtaking all competitors by approximately iteration 30, and thereafter continues a smooth, monotonic ascent free of both prolonged plateaus and oscillatory reversals. The best trajectory of AHA is qualitatively similar but levels off after iteration 60. The optimal run of EEFO tracks closely with AHA in the early stage but enters a plateau around iteration 50. Even the most successful executions of BAE0 and ZOA exhibit an extended mid-search stall from roughly iteration 30 to 60, a pattern characteristic of temporary entrapment by local attractors. Although COA starts from the lowest position, it accelerates in the subsequent stages and gradually narrows the gap formed in the early iterations.

4.3. Multi-scenario WSN coverage optimization results and analysis

To evaluate the robustness of the proposed method under variations in network scale, sensing parameters, and regional geometry, this study constructs three additional test scenarios. Scenario S1

involves the deployment of 30 sensor nodes within a cubic region of dimensions 50 by 50 by 50 meters, with a sensing radius of 10 meters and a search dimensionality of 90. This scenario is designed to examine algorithmic performance under sparse node conditions. Scenario S3 deploys 50 nodes within a region of identical size but with the sensing radius reduced to 9 meters, resulting in a search dimensionality of 150. This configuration is intended to assess the collaborative capability of the algorithm when the number of nodes increases while the per-node sensing capacity declines. Scenario S4 considers a non-cubic region of dimensions 60 by 50 by 40 meters, in which 40 nodes are deployed with a sensing radius of 9 meters, yielding a search dimensionality of 120. This scenario is used to investigate algorithmic adaptability to unequal bounds across dimensions. For ease of reference, the primary scenario presented in Section 4.2 is hereafter referred to as Scenario S2. For all algorithms and across all scenarios, the population size is uniformly set to 30, the maximum number of iterations is 100, and each algorithmic configuration is independently executed 30 times. The statistical results for Scenarios S1, S3, and S4 are summarized in Table 2.

Table 2. Coverage optimization performance across six algorithms in extended scenarios S1, S3, and S4.

Scenario	Algorithm	Best	Worst	Mean	Std
WSN-S1	PSSO	0.748748	0.699647	0.727775	0.009924
WSN-S1	BAEO	0.697542	0.597861	0.648506	0.027631
WSN-S1	AHA	0.680815	0.640589	0.660380	0.011389
WSN-S1	COA	0.679904	0.615043	0.651128	0.015307
WSN-S1	EEFO	0.733955	0.589611	0.702010	0.024467
WSN-S1	ZOA	0.640077	0.570380	0.611931	0.018310
WSN-S3	PSSO	0.810423	0.758819	0.791769	0.010397
WSN-S3	BAEO	0.743400	0.656293	0.683923	0.021244
WSN-S3	AHA	0.749203	0.697315	0.717440	0.011029
WSN-S3	COA	0.728835	0.687358	0.707730	0.010175
WSN-S3	EEFO	0.790680	0.631998	0.731581	0.049425
WSN-S3	ZOA	0.691682	0.642922	0.664683	0.013192
WSN-S4	PSSO	0.742172	0.697152	0.720991	0.011512
WSN-S4	BAEO	0.684568	0.601087	0.624761	0.018777
WSN-S4	AHA	0.672752	0.629859	0.654685	0.009110
WSN-S4	COA	0.667848	0.625901	0.643353	0.011134
WSN-S4	EEFO	0.715054	0.567529	0.660191	0.052236
WSN-S4	ZOA	0.636299	0.579168	0.607446	0.016076

The results in Table 2 demonstrate that PSSO achieves the highest best, worst, and mean coverage rates in all four scenarios. Thus, its advantage is not limited to the principal WSN-S2 configuration but remains evident when the number of sensors, sensing radius, and region shape are changed.

In the sparsest scenario, WSN-S1, with only 30 nodes deployed, PSSO achieves a mean coverage of 0.727775, significantly outperforming EEFO at 0.702010, with an absolute improvement of 0.025765 (a relative improvement of approximately 3.67%). It also attains the highest best coverage (0.748748) and the highest worst coverage (0.699647). Its standard deviation is only 0.009924, the lowest among all algorithms, far below those of BAEO (0.027631) and EEFO (0.024467), indicating that PSSO can still distribute limited sensors stably and uniformly under extreme node scarcity.

When the number of nodes increases to 50 and the sensing radius shrinks to 9 m (WSN-S3), PSSO attains a mean coverage of 0.791769, which is 0.060188 higher than the second-ranked EEFO (0.731581), representing an improvement of about 8.23%. Its best value of 0.810423 and worst value of 0.758819 are both excellent, with its worst already surpassing the best values of BAEO, AHA, COA, and ZOA. Although COA has the lowest standard deviation (0.010175), PSSO's 0.010397 is nearly identical, yet it outperforms COA in mean coverage by 0.084039. Meanwhile, although EEFO achieves a relatively high best value, its standard deviation is as large as 0.049425, indicating significantly greater variability. This demonstrates that PSSO can effectively adapt to reduced sensing radii and fully utilize the additional nodes to construct complementary three-dimensional coverage structures.

In the rectangular anisotropic space known as WSN-S4, PSSO achieves a mean coverage of 0.720991, far exceeding EEFO's 0.660191, with an absolute improvement of 0.060800 and a relative improvement of about 9.21 percent, which is the largest among the four scenarios. It also records the highest best coverage at 0.742172 and the highest worst coverage at 0.697152. Although EEFO's best reaches 0.715054, its worst drops sharply to 0.567529, with a standard deviation as high as 0.052236, indicating high sensitivity to initialization. In contrast, PSSO's standard deviation is only 0.011512, demonstrating that its search operators can exploit dimension-independent upper and lower bound constraints, effectively adapting to asymmetric spatial boundaries with unequal lengths along the x, y, and z axes, while maintaining robust and superior coverage performance.

More importantly, PSSO ranks first not only in terms of the best solution found but also in terms of the worst result across 30 runs in every scenario. This indicates that its performance advantage is not produced by a small number of exceptionally successful runs. Instead, PSSO maintains a consistently higher coverage level throughout repeated independent executions.

Overall, these results demonstrate that the proposed petal–spiral search framework provides a robust balance between global exploration and local exploitation. The rose-curve-guided exploration strategy enables the population to examine diverse deployment patterns, whereas epitrochoid-based exploitation refines promising node configurations at progressively smaller scales. Together with the migration-reset mechanism, these components enable PSSO to maintain high coverage accuracy, low cross-run variability, and competitive computational efficiency across heterogeneous three-dimensional WSN coverage problems.

4.4. Component-ablation analysis

To evaluate individual component contributions, six ablation variants were evaluated against the full model. Variant PSSO-NoPetal replaces rose curve radial modulation in Eq 15 with normalized Gaussian vectors to test the impact of multi-petal sign modulation. Variant PSSO-NoSpiral replaces epitrochoid normalization envelopes in Eq 25 with normalized Gaussian vectors to evaluate epitrochoid geometry contributions. Variant PSSO-LinearWeight replaces composite weight formulation in Eq 10 with a linear decay schedule from 1 to w_{min} . Variant PSSO-NoMigration disables migration resets. Variant PSSO-NoPbest forces reference base selection in Eq 26 to select global memory exclusively. Variant PSSO-FixedPetals fixes petal count to k_0 . Ablation variants executed 30 independent runs in the primary scenario under identical random seed sequences and evaluation budgets. Statistical outcomes are presented in Table 3.

Table 3. Ablation evaluation results across algorithm variants in the primary scenario.

Variant	Best	Worst	Mean	Std
PSSO-Full	0.860452	0.814382	0.836504	0.010121
PSSO-NoPetal	0.851104	0.810822	0.836421	0.009869
PSSO-NoSpiral	0.857590	0.824135	0.840728	0.008714
PSSO-LinearWeight	0.849795	0.821290	0.838848	0.007856
PSSO-NoMigration	0.857078	0.816113	0.838554	0.009598
PSSO-NoPbest	0.859752	0.829540	0.848069	0.008405
PSSO-FixedPetals	0.852128	0.816170	0.836721	0.010323

As shown in Table 3, the complete model achieves the highest search upper bound, with its best value of 0.860452 surpassing all variants. The closest competitor among the variants is PSSO-NoPbest with a value of 0.859752, while PSSO-LinearWeight and PSSO-NoPetal decline to 0.849795 and 0.851104, respectively. This indicates that the multi-petal modulation of the rose curve and the composite weight scheduling jointly determine the optimal level that the algorithm can reach. Replacing either of these components causes the best record over 30 runs to drop by approximately 0.01, a magnitude comparable to the standard deviation of the algorithm itself. The dynamic adjustment of petal count primarily affects consistency. The variant with fixed petal count exhibits the highest standard deviation among all variants at 0.010323, while its mean value of 0.836721 is nearly identical to that of the complete model. This suggests that this mechanism serves to suppress fluctuation across runs rather than to enhance the average performance.

Taken together, the most robust conclusion drawn from the ablation results is that the advantage of PSSO over the baseline methods stems mainly from the architectural level. After replacing the rose curve or the epitrochoid with a Gaussian direction, the mean coverage of the variants remains above 0.836, far higher than the 0.789847 achieved by the strongest baseline AHA in Table 1. This confirms the framework alone offers adequate search capability for the problem.

5. Performance evaluation on the CEC2022 benchmark functions

To evaluate the general numerical optimization capability of PSSO beyond coverage problems, this study conducts experiments on all 12 functions of the CEC2022 test suite. The dimensionality is set to 20, the population size to 30, and the maximum number of iterations to 500, resulting in an evaluation budget of 15,000 function evaluations per run. Each function is independently executed 30 times. PSSO is compared with AHA, BAE0, COA, EEFO, and ZOA under identical budgets, and the statistical results are presented in Table 4.

The results in Table 4 demonstrate that PSSO provides competitive performance across the CEC2022 benchmark suite. Based on the reported mean values, PSSO obtains the uniquely best result on F2, F8, and F12. It also reaches the same reported mean level as the leading algorithms on F1, F3, and F11. Therefore, PSSO achieves or shares the best mean result on six of the 12 functions.

With respect to the best values, PSSO uniquely ranks first on F2, F10, and F12 and shares the leading result on F1, F3, and F11. These results indicate that PSSO possesses a strong ability to locate high-quality regions of the search space. Its advantage is particularly evident on F2, F8, and F12, where the algorithm combines a favorable best value with the lowest mean objective value.

Table 4. Statistical results obtained for the multimodal functions in CEC2022 test set.

Fun.	Metric	PSSO	AHA	BAEO	COA	EEFO	ZOA
F1	Best	3.0000E+02	3.0000E+02	3.0000E+02	3.0000E+02	3.0000E+02	8.3246E+02
	Mean	3.0000E+02	3.0000E+02	3.0000E+02	3.0008E+02	3.0000E+02	3.8089E+03
	Std	2.1915E-07	1.2258E-03	6.5858E-11	5.8746E-02	2.6852E-03	2.2618E+03
F2	Best	4.0555E+02	4.4491E+02	4.4490E+02	4.4522E+02	4.4911E+02	4.7444E+02
	Mean	4.5343E+02	4.6468E+02	4.5440E+02	4.6145E+02	4.6038E+02	5.7412E+02
	Std	1.8177E+01	1.3075E+01	1.0099E+01	1.8184E+01	1.1345E+01	6.3103E+01
F3	Best	6.0000E+02	6.0000E+02	6.0000E+02	6.0000E+02	6.0000E+02	6.0000E+02
	Mean	6.0000E+02	6.0000E+02	6.0000E+02	6.0000E+02	6.0000E+02	6.0013E+02
	Std	1.0182E-11	4.5299E-08	3.2819E-14	7.3678E-12	5.2562E-09	1.0511E-01
F4	Best	8.0042E+02	8.0025E+02	8.0035E+02	8.0044E+02	8.0042E+02	8.0037E+02
	Mean	8.0072E+02	8.0048E+02	8.0072E+02	8.0084E+02	8.0101E+02	8.0050E+02
	Std	2.6906E-01	1.6640E-01	3.7925E-01	2.3311E-01	3.3598E-01	1.1989E-01
F5	Best	9.0045E+02	9.0018E+02	9.0009E+02	9.0163E+02	9.0018E+02	9.0123E+02
	Mean	9.0253E+02	9.0155E+02	9.0297E+02	9.0239E+02	9.0107E+02	9.0286E+02
	Std	1.9721E+00	8.7865E-01	2.0104E+00	6.7711E-01	8.8057E-01	1.3808E+00
F6	Best	4.1210E+04	1.8371E+04	2.4414E+04	3.7115E+04	2.1605E+04	2.4765E+04
	Mean	7.5310E+04	3.6836E+04	4.1980E+04	7.3947E+04	5.4888E+04	2.8155E+06
	Std	2.1504E+04	9.7030E+03	1.1996E+04	2.4669E+04	3.0015E+04	9.0779E+06
F7	Best	2.0232E+03	2.0220E+03	2.0274E+03	2.0391E+03	2.0238E+03	2.0788E+03
	Mean	2.0396E+03	2.0266E+03	2.0702E+03	2.0916E+03	2.0370E+03	2.3891E+03
	Std	1.6681E+01	3.8913E+00	6.3630E+01	4.6555E+01	1.3016E+01	1.9385E+02
F8	Best	2.2288E+03	2.2257E+03	2.2574E+03	2.2414E+03	2.2383E+03	2.3412E+03
	Mean	2.2488E+03	2.2751E+03	2.4005E+03	2.8749E+03	2.2864E+03	3.4543E+03
	Std	1.9028E+01	7.4974E+01	1.1320E+02	8.4785E+02	4.9209E+01	5.9481E+02
F9	Best	2.6361E+03	2.6375E+03	2.3000E+03	2.6361E+03	2.6356E+03	2.6691E+03
	Mean	2.6372E+03	2.6454E+03	2.5837E+03	2.6388E+03	2.6359E+03	2.8260E+03
	Std	1.0349E+00	5.1135E+00	1.2690E+02	3.8376E+00	5.8581E-01	1.0670E+02
F10	Best	2.7598E+03	2.7622E+03	2.7611E+03	2.7660E+03	2.7608E+03	2.7931E+03
	Mean	3.3946E+03	3.2200E+03	3.9133E+03	3.5835E+03	3.3243E+03	3.0961E+03
	Std	7.4977E+02	7.9373E+02	7.0362E+02	1.3469E+03	6.6628E+02	4.0005E+02
F11	Best	2.6000E+03	2.6000E+03	2.6000E+03	2.6000E+03	2.6000E+03	2.6407E+03
	Mean	2.6000E+03	2.6000E+03	2.6000E+03	2.6067E+03	2.6028E+03	2.7455E+03
	Std	2.7699E-05	3.4021E-02	7.5915E-07	6.4956E+00	5.8828E+00	1.0481E+02
F12	Best	2.9402E+03	2.9473E+03	3.1193E+03	2.9448E+03	2.9449E+03	3.3012E+03
	Mean	2.9568E+03	2.9791E+03	3.2016E+03	2.9936E+03	2.9625E+03	3.4690E+03
	Std	1.2674E+01	2.6934E+01	8.7143E+01	3.7594E+01	1.4045E+01	1.1883E+02

PSSO demonstrates high convergence accuracy and strong global search capability on F1–F3. On F1, it reaches the best and mean value of 3.0000×10^2 with a standard deviation of only 2.1915×10^{-7} , indicating highly repeatable results. On F2, PSSO achieves the lowest best value of 4.0555×10^2 , representing an improvement of approximately 8.84% over the strongest competing algorithm, while also attaining the lowest mean value of 4.5343×10^2 . On F3, PSSO again reaches the optimal level with a standard deviation of only 1.0182×10^{-11} , exhibiting nearly deterministic convergence. Overall, PSSO achieves the reported best mean level on all three functions in this group, validating the

effectiveness of combining petal exploration with spiral exploitation. On F4, PSSO attains a mean of 8.0072×10^2 with a standard deviation lower than those of BAEO and EEFO, remaining competitive. On F5, its best value of 9.0045×10^2 differs from the optimum by only 0.36, though its mean ranks fourth, indicating some run-to-run variation. F6 is the most challenging function for PSSO, with a mean of 7.5310×10^4 , yet it still significantly outperforms ZOA (mean 2.8155×10^6). This limitation may stem from excessive late-stage concentration on the global optimum, leading to reduced diversity, which could be mitigated by adaptively adjusting the migration frequency or introducing multiple elite centers.

On F7, PSSO achieves a mean of 2.0396×10^3 , ranking among the top three and significantly outperforming BAEO, COA, and ZOA. On F8, PSSO obtains the lowest mean of 2.2488×10^3 and the lowest standard deviation of 1.9028×10^2 , with standard deviations 74.6% lower than AHA and 61.3% lower than EEFO, demonstrating both high accuracy and strong stability. On F9, PSSO attains a mean of 2.6372×10^3 with a standard deviation of only 1.0349, indicating highly concentrated results, whereas BAEO, despite a slightly lower mean, exhibits a standard deviation as high as 1.2690×10^2 , reflecting substantial volatility. On F10, PSSO achieves the lowest best value of 2.7598×10^3 , showcasing strong peak-search capability, but its mean of 3.3946×10^3 and standard deviation of 7.4977×10^2 are relatively large, indicating room for improved consistency. On F11, PSSO reaches the optimal level of 2.6000×10^3 with a standard deviation of only 2.7699×10^{-5} , demonstrating extremely high convergence precision. On F12, PSSO ranks first across all three metrics (best value 2.9402×10^3 , mean 2.9568×10^3 , and standard deviation 1.2674×10^1) achieving comprehensive superiority. In summary, PSSO exhibits competitive accuracy (F7), combined high accuracy and stability (F8–F9), strong peak-discovery capability (F10), high repeatability precision (F11), and comprehensive balance (F12) across F7–F12. The results on F8 and F12 strongly support the effectiveness of its geometric search framework, while the insufficient consistency on F10 points toward directions for future improvement.

Moreover, AHA, BAEO, EEFO, COA, and ZOA each retain advantages on specific functions, which is consistent with the no free lunch theorem. Overall, the CEC2022 results show that PSSO is not limited to three-dimensional WSN coverage applications; its structured geometric search operators provide competitive numerical optimization performance across heterogeneous landscapes, with particularly notable strengths in convergence accuracy and repeated-run reliability.

6. Conclusion

This paper proposes and systematically analyzes the PSSO, whose core components include an exploration operator that applies multi-petal sign modulation to the step size via the radial component of a rose curve, an exploitation operator that constructs a perturbation envelope using the normalized modulus of an epitrochoid, a bloom weight governed by a polynomial-trigonometric composite curve that simultaneously regulates the operator selection probability and the internal fusion ratio of exploration, as well as supporting mechanisms consisting of individual memory, global memory, greedy acceptance, and low-frequency migration resets. The study takes three-dimensional wireless sensor network coverage optimization as the primary application background, with the CEC2022 test suite and component ablation serving as tools for examining the methodological boundaries.

In the primary scenario, PSSO achieves a mean coverage of 0.836504 with a standard deviation of 0.010121. The worst result over 30 runs, 0.814382, exceeds the best results of all five competing

algorithms. The mean coverage improvement over the second-best method is 5.91 percent, and the confidence interval of the mean does not overlap with those of any competing algorithm. In the other three scenarios where the number of nodes, sensing radius, and region shape vary, PSSO simultaneously ranks first in best, worst, and mean values, with relative improvements of 3.67 percent, 8.23 percent, and 9.21 percent respectively. The standard deviation consistently remains within the lowest range. PSSO has the shortest total runtime, far below that of AHA, COA, and especially the slowest, BAE0. These results demonstrate that the proposed method exhibits stable and reproducible advantages on axis-aligned box-constrained spatial deployment problems.

Component ablation experiments quantitatively assess the contribution of each search strategy. The complete model yields the highest search upper bound of 0.860452, while the dynamic petal count primarily contributes to reducing inter-run variability. The mean coverage values of all variants span a range of merely 0.0116, whereas the variant that relies exclusively on global memory attains a higher mean of 0.848069. This suggests that the advantage over baseline methods is primarily supported by the overall architecture formed by geometric modulation, population differential-based adaptive scaling, dual memory structures, and greedy selection, while a genuine trade-off exists between individual memory preference in elevating the upper bound and in improving the average performance. The CEC2022 results further delimit the applicable scope of the method. PSSO achieves or ties for the optimal mean on 6 out of 12 functions, with an average rank of 2.75, placing it in the first tier together with the strongest baselines. However, its competitiveness on rotated and composite functions is limited, attributable to the coordinate dependence arising from dimension-wise sampling.

Future work will systematically extend the coverage model to encompass more engineering-oriented scenarios, including probabilistic sensing, communication connectivity, energy consumption and lifetime, obstacle occlusion, heterogeneous node deployment, localization errors, node failures, and deployment costs. Subsequently, the applicability of the proposed framework will be re-assessed within a multi-objective optimization setting.

Data Availability

Data sharing not applicable because no datasets are analyzed or generated in this article.

Author contributions

Xiaowei Wang contributed to conceptualization, methodology, software, validation, formal analysis, investigation, visualization, writing of the original draft, and review and editing of the manuscript. Jun Bao contributed to conceptualization, supervision, project administration, funding acquisition, and review and editing of the manuscript. All authors have read and approved the final version of the manuscript.

Use of Generative-AI tools declaration

Not applicable.

Acknowledgements

This work was supported by the Ministry of Education of the People's Republic of China, Humanities and Social Sciences Research Planning Fund Project, under Grant No. 25YJA630001. The project is entitled "Human-Machine Collaborative Job Crafting in the Context of Digital Transformation: Scale Development, Antecedent Configurations, and Impact Mechanisms."

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. L. Abualigah, M. A. Elaziz, A. M. Khasawneh, M. Alshinwan, R. A. Ibrahim, M. A. A. Al-qaness, et al., Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: a comprehensive survey, applications, comparative analysis, and results, *Neural Comput. Applic.*, **34** (2022), 4081–4110. <https://doi.org/10.1007/s00521-021-06747-4>
2. L. Wang, Y. Huang, A review of research progress and multi-field applications of Zeroing neural networks in optimization problems, *Cluster Comput.*, **29** (2026), 135. <https://doi.org/10.1007/s10586-026-05956-7>
3. W. Lu, J. Chen, K. Yang, D. Liu, X. Li, Y. Fu, et al., MOSBOA: A novel multi-objective optimization via secretary bird algorithm with reference point guidance, *Cluster Comput.*, **29** (2026), 222. <https://doi.org/10.1007/s10586-026-05981-6>
4. Z. Meng, Z. Song, X. Shao, J. Zhang, H. Xu, FD-DE: Differential Evolution with fitness deviation based adaptation in parameter control, *ISA Trans.*, **139** (2023), 272–290. <https://doi.org/10.1016/j.isatra.2023.05.005>
5. A. Aguilar-Rivera, The unscented genetic algorithm for fast solution of GA-hard optimization problems, *Appl. Soft Comput.*, **139** (2023), 110260. <https://doi.org/10.1016/j.asoc.2023.110260>
6. C. L. Camacho-Villalon, M. Dorigo, T. Stutzle, PSO-X: A Component-Based Framework for the Automatic Design of Particle Swarm Optimization Algorithms, *IEEE Trans. Evol. Computat.*, **26** (2022), 402–416. <https://doi.org/10.1109/TEVC.2021.3102863>
7. A. N. Soumana Hamadou, C. Wa Maina, M. Moindze Soidridine, A Grey Wolf Optimiser (GWO)-Based RIS Phase Shift Design for a RIS-Assisted Multi-Antenna System, *IEEE Commun. Lett.*, **28** (2024), 2171–2175. <https://doi.org/10.1109/LCOMM.2024.3423013>
8. J. Liu, J. Shi, F. Hao, M. Dai, A novel enhanced global exploration whale optimization algorithm based on Lévy flights and judgment mechanism for global continuous optimization problems, *Eng. Comput.*, **39** (2023), 2433–2461. <https://doi.org/10.1007/s00366-022-01638-1>
9. B. Khurshid, S. Maqsood, A hybrid evolution strategies-simulated annealing algorithm for job shop scheduling problems, *Eng. Appl. Artif. Intell.*, **133** (2024), 108016. <https://doi.org/10.1016/j.engappai.2024.108016>
10. Z. Qian, Y. Xie, S. Xie, MAR-GSA: Mixed attraction and repulsion based gravitational search algorithm, *Inf. Sci.*, **662** (2024), 120250. <https://doi.org/10.1016/j.ins.2024.120250>
11. Z. Zeng, H. Dong, Y. Xu, W. Zhang, H. Yu, X. Li, Teaching-learning-based optimization algorithm with dynamic neighborhood and crossover search mechanism for numerical optimization, *Appl. Soft Comput.*, **154** (2024), 111332. <https://doi.org/10.1016/j.asoc.2024.111332>

12. B. Sasmal, A. Das, K. G. Dhal, R. Saha, R. Rai, T. Bharasa, et al., Artificial hummingbird algorithm: Theory, variants, analysis, applications, and performance evaluation, *Comput. Sci. Re.*, **56** (2025), 100727. <https://doi.org/10.1016/j.cosrev.2025.100727>
13. L. Ding, Y. Bai, M. Fan, W. Song, H. Ren, Using a snow ablation optimizer in an autonomous echo state network for the model-free prediction of chaotic systems, *Nonlinear Dyn.*, **112** (2024), 11483–11500. <https://doi.org/10.1007/s11071-024-09656-y>
14. M. Abdel-Basset, D. El-Shahat, M. Jameel, M. Abouhawwash, Exponential distribution optimizer (EDO): a novel math-inspired algorithm for global optimization and engineering problems, *Artif. Intell. Rev.*, **56** (2023), 9329–9400. <https://doi.org/10.1007/s10462-023-10403-9>
15. D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Comput.*, **1** (1997), 67–82. <https://doi.org/10.1109/4235.585893>
16. Z. Pan, D. Lei, L. Wang, A Knowledge-Based Two-Population Optimization Algorithm for Distributed Energy-Efficient Parallel Machines Scheduling, *IEEE Trans. Cybern.*, **52** (2022), 5051–5063. <https://doi.org/10.1109/TCYB.2020.3026571>
17. H. Wang, B. R. Sarker, J. Li, J. Li, Adaptive scheduling for assembly job shop with uncertain assembly times based on dual Q-learning, *Int. J. Prod. Res.*, **59** (2021), 5867–5883. <https://doi.org/10.1080/00207543.2020.1794075>
18. F. Zhao, Z. Fu, L. Wang, H. Sang, A Heterogeneous Graph Reinforcement Learning Framework With Question-Aware Neighborhood Aggregation and Interoption Prompt Attention for Dynamic Flexible Job Shop Scheduling Problem, *IEEE Trans. Ind. Inform.*, **22** (2026), 2863–2874. <https://doi.org/10.1109/TII.2025.3646962>
19. A. A. Alhussan, E. M. El-Kenawy, M. M. Eid, N. Khodadadi, Hybrid AI-Biruni and Puma Optimization (BERPO) for boosting the classification of Quality-of-Service (QoS) in 5G networks, *J. Net. Comput. Appl.*, **250** (2026), 104462. <https://doi.org/10.1016/j.jnca.2026.104462>
20. W. Zhao, L. Wang, S. Mirjalili, Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications, *Comput. Methods Appl. Mech. Eng.*, **388** (2022), 114194. <https://doi.org/10.1016/j.cma.2021.114194>
21. S. Zhao, F. Meng, L. Cai, R. Yang, Boomerang aerodynamic ellipse optimizer: A human game-inspired optimization technique for numerical optimization and multilevel thresholding image segmentation, *Math. Comput. Simul.*, **238** (2025), 604–636. <https://doi.org/10.1016/j.matcom.2025.07.006>
22. H. Jia, H. Rao, C. Wen, S. Mirjalili, Crayfish optimization algorithm, *Artif. Intell. Rev.*, **56** (2023), 1919–1979. <https://doi.org/10.1007/s10462-023-10567-4>
23. W. Zhao, L. Wang, Z. Zhang, H. Fan, J. Zhang, S. Mirjalili, et al., Electric eel foraging optimization: A new bio-inspired optimizer for engineering applications, *Expert Syst. Appl.*, **238** (2024), 122200. <https://doi.org/10.1016/j.eswa.2023.122200>
24. Q. Ren, F. Feng, PID parameter tuning optimization based on multi-strategy fusion improved zebra optimization algorithm, *J. Supercomput.*, **81** (2025), 266. <https://doi.org/10.1007/s11227-024-06548-1>



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)