

An Improved Cuckoo Search Algorithm for Maximizing the Coverage Range of Wireless Sensor Networks

Sen-Yu Yang^{id}, Yin-Hong Xiang^{id}, Di-Wen Kang^{id}, Kai-Qing Zhou*^{id}

School of Communication and Electronic Engineering, Jishou University, Jishou, 416000, Hunan, China.

*Corresponding Author.

ICAC2023: The 4th International Conference on Applied Computing 2023.

Received 27/09/2023, Revised 10/02/2024, Accepted 12/02/2024, Published 25/02/2024



© 2022 The Author(s). Published by College of Science for Women, University of Baghdad.

This is an Open Access article distributed under the terms of the [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract

The issue of increasing the range covered by a wireless sensor network with restricted sensors is addressed utilizing improved CS employing the PSO algorithm and opposition-based learning (ICS-PSO-OBL). At first, the iteration is carried out by updating the old solution dimension by dimension to achieve independent updating across the dimensions in the high-dimensional optimization problem. The PSO operator is then incorporated to lessen the preference random walk stage's imbalance between exploration and exploitation ability. Exceptional individuals are selected from the population using OBL to boost the chance of finding the optimal solution based on the fitness value. The ICS-PSO-OBL is used to maximize coverage in WSN by converting regional monitoring into point monitoring utilizing the discretization method in WSN. In the experiments, the ICS-PSO-OBL with the standard CS and three CS variants (MACS, ICS-2, and ICS) are utilized to execute the simulation experiment under different numbers of nodes (20 and 30, respectively). The experimental results reveal that the optimized coverage of ICS-PSO-OBL is 18.36%, 7.894%, 15%, and 9.02% higher than that of standard CS, MACS, ICS-2, and ICS when the number of nodes is 20. Moreover, it is 16.94%, 9.61%, 12.27%, and 7.75% higher when the quantity of nodes is 30, the convergence speed of ICS-PSO-OBL, and the distribution of nodes is superior to others.

Keywords: cuckoo search algorithm, dimension-by-dimension update, opposition-based learning, wireless sensor network, PSO operator.

Introduction

Wireless sensor network (WSN) comprises a self-organizing distributed system composed of numerous small sensor nodes, each equipped with wireless communication and computational capabilities¹. These nodes gather various data types within the monitored region, such as sound and temperature. Subsequently, this data is transmitted to a central management center for in-depth statistical analysis and processing². The effectiveness of WSN heavily relies on their

coverage, which plays an integral role in their overall performance. Consequently, addressing the coverage optimization challenge in WSN involves strategically deploying sensor nodes throughout the designated monitoring area while maintaining network connectivity³.

Hence, It has been a research focus to maximize coverage of the region with a small number of sensor nodes, and Swarm Intelligence (SI) Algorithms, which can find an approximate solution

to the optimization problem in an acceptable time ⁴, provide a means to address the issue of WSN coverage optimization. Currently, SI has produced beneficial achievements in the area of WSN coverage optimization ⁵. Duan et al. used the sparrow search algorithm (SSA) to optimize network coverage. Compared to the standard SSA, the upgraded SSA balances network coverage, energy use, and coverage redundancy ⁶. As a way to tackle the Maximum Coverage Set Scheduling (MCSS) problem, Mottaki et al. suggested a hybrid approach by combining genetic algorithm (GA) and tabu search. Experiments reveal that the new algorithm is more effective in locating the nearly ideal scheduling coverage set ⁷. He et al. introduced a novel model for optimizing the coverage of WSN. This model elevates the improved marine predator algorithm's efficacy by amalgamating a multi-echelon stochastic leadership approach with a dynamically adjustable inertia weight strategy ⁸.

There are various SI algorithms available for solving WSN coverage optimization problems. Musa et al. concludes his article by comparing the Cuckoo Search (CS) algorithm with various SI algorithms (e.g., GA, ACO, PSO, and so on), and concludes that CS performs better with respect to the relative ones ⁹. Therefore, it is a useful attempt to apply CS to the WSN coverage optimization problem.

The CS, a relatively recent swarm intelligence optimization technique, has gained widespread popularity in various domains, its broad adoption can be attributed to its robust global search capabilities, limited control parameters, and high applicability ^{10,11}. Scholars have made various improvements to the standard CS in terms of its parameter configuration and combining it with other algorithms in response to the problems of unbalanced exploration and exploitation capacity, readily prone to local optimal, and mutual interference between dimensions of overcoming challenges in high-dimensional optimization. Ye et al. proposed an innovative CS referred to as ICS-ABC-OBL. In this approach, they synergistically integrated an ABC algorithm with opposition-based learning (OBL), resulting in improved system performance and the effective

mitigation of the early convergence issue ¹². Nonlinear inertia weights and differential evolution (DE) are combined in Zhang et al.'s proposed improved CS (CSDE). The nonlinear inertia weights ensure that the algorithm makes it difficult to enter the local optimum, and the DE operator successfully enhances the information interchange between algorithm individuals. Through trials, the performance of CSDE is improved ¹³. Due to the results, Li et al.'s proposed improved cuckoo search method significantly increases the algorithm's convergence accuracy and speed. It is based on elite OBL and the golden sine operator ¹⁴. By making these changes, the conventional CS's shortcomings were improved and its ability to solve optimization issues will be enhanced, and all of them have achieved good results. Therefore, it is a valuable attempt to propose and apply an improved CS to WSN coverage optimization.

This study of this paper could be separated into three phases to fulfill this goal. First, an improved CS using PSO operator and OBL, namely ICS-PSO-OBL, is used to balance exploration and exploitation capabilities and enhance the susceptibility to local optima and interference among dimensions. Then, the coverage of WSN is modeled by transforming the area into point monitoring using a discretization. Finally, in terms of performance validation, ICS-PSO-OBL is compared with CS and CS variants through simulation experiments when the number of nodes is 20 and 30, and the experiments show that ICS-PSO-OBL can be more effectively applied to the network coverage problem to improve the quality of network node coverage, which demonstrates the practicality.

The remainder of this text is divided into the following sections. The key ideas discussed in this study, including WSN and the CS, are reviewed in Section 2. In Section 3, a thorough explanation of the planned ICS-OBL-OBL is given. In Section 4 and Section 5, the optimization of ICS-PSO-OBL is tested, and Section 5 applies it to the coverage optimization problem of WSN. A summary of Section 6's major conclusions and contributions comes at the end of the section.

Related notions

Sensor network node coverage model

The monitoring area of the WSN is characterized as a rectangular region with a length denoted as L and a width denoted as W , resulting in a total area of $L \times W$. Inside this monitoring region, n sensor nodes are distributed randomly. This n -node set is referred to as $A = \{A_i = (x_i, y_i) \mid i=1, 2, \dots, n\}$, where each node A_i is characterized by its coordinates (x_i, y_i) . The sensor nodes in the network employ a Boolean sensing model, meaning that they could detect events or targets within a certain distance and could not sense anything beyond that distance. Each node's sensing radius is represented by the letter R , indicating the radius of the circular region centered at the node within which it could sense events or targets. Furthermore, each node has a communication radius denoted as r , representing the maximum distance over which it could establish wireless communication with other nodes in the network.

To determine the network coverage, the monitoring area is discretized into $L \times W$ cells of equal size. The center point of each cell represents a monitoring point, and the set of these monitoring points is denoted as $B = \{B_j = (x_j, y_j) \mid j=1, 2, \dots, L \times W\}$.

As illustrated by Eq.1, the Euclidean distance serves as the metric for determining the separation between a monitoring point and a sensor node.

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

The probability of a monitoring point B_j is sensed by a sensor node A_i , given that the sensing radius of the nodes is R , is represented by Eq.2. In this equation, B_j is considered to be sensed by A_i if it falls within the circle centered at A_i with a radius of R .

$$P(A_i, B_j) = \begin{cases} 1 & \text{if } d(A_i, B_j) \leq R \\ 0 & \text{otherwise} \end{cases} \quad 2$$

Each monitoring point, denoted as B_j , has the potential to be detected by multiple sensor nodes, labeled as A_i . The collective sensing probability, quantifying the likelihood that B_j will be detected by at least one of the sensor nodes A_i , is precisely

defined in Eq.3.

$$C_p(A_{all}, B_j) = 1 - \prod_{i=1}^n (1 - p(A_i, B_j)) \quad 3$$

The coverage area is established as the summation of the joint sensing probabilities across all monitoring sites, with A_{all} representing the entire ensemble of sensor nodes within the region.

The coverage rate (C_r) is then formally defined as the proportion of the coverage area in relation to the overall area of the region, as depicted in Eq.4.

$$C_r = \frac{\sum_{j=1}^{L \times W} C_p(A_{all}, B_j)}{L \times W} \quad 4$$

Standard CS

CS is an intelligent search algorithm that combines the brood parasitism behavior of cuckoos and the Lévy flight pattern observed in insects like fruit flies. The principle of CS is to map the cuckoo's brood parasitism behavior to the solutions in the algorithm's population space, using the quality of the cuckoo's nest as a measure of the solution's fitness. To establish a correlation between the brood parasitism behavior, three idealized rules are taken into account¹⁵:

Assumption 1: Only one egg is laid by each cuckoo at a time, and the nest chosen for incubation is at random.

Assumption 2: From a random selection of nests, the one situated in the best location is retained for the subsequent generation.

Assumption 3: The probability Pa denotes the possibility that a host bird will come into contact with a foreign egg, and the number of nests is constant. Upon such an encounter, the host bird can choose to abandon the foreign egg or initiate a new nest.

In accordance with these assumptions, the fundamental process of the standard CS algorithm unfolds as follows:

Step 1: Set the parameters for the algorithm and define the problem.

Step 2: Generate N nests in the D -dimensional

space using Eq.5 and evaluate the randomly generated nest positions. Retain the information of the best nest:

$$x = LB + \text{rand}(D) \times (UB - LB) \quad 5$$

where x is the set of randomly generated nests; The upper and lower boundaries of the search space are, respectively, UB and LB ; A D -dimensional random number, $\text{rand}(D)$, has values in each dimension ranging from 0 to 1.

Step 3: Lévy flight updates the nest positions. Assume that the position of the i th nest is x_i , $1 \leq i \leq N$, for the t th generation of nests x_i^t by Eq.6 to obtain the $t+1$ th generation of nests x_i^{t+1} :

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{Levy}(\beta), \quad i = 1, 2, \dots, n \quad 6$$

where α is the step control volume; \oplus is the point-to-point multiplication. $\text{Levy}(\beta)$ is the random search path, obeying the Lévy distribution: $\text{Levy}(\beta) \sim \mu = t^{-\beta}$, $1 < \beta \leq 3$, where μ is the random step size obtained from the Lévy flight. Due to the relative difficulty of performing integration operations on the Lévy distribution, Lévy flight is generally simulated by Eq.7.

$$\text{Levy}(\beta) = \frac{\mu}{|\nu|^{1/\beta}} \quad 7$$

where μ and ν are random numbers obeying normal distribution: $\mu \sim N(0, \sigma_u^2)$, $\nu \sim N(0, \sigma_v^2)$, generally $\sigma_v = 1, \beta = 1.5$, σ_u are calculated through Eq.8:

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta) \cdot \sin(\pi \beta / 2)}{\Gamma[(1+\beta)/2] \cdot \beta \cdot 2^{(\beta-1)/2}} \right\}^{1/\beta} \quad 8$$

Step 4: Preference random wandering. Following the removal of a portion of the solutions using a specific probability termed the discovery probability (P_a), the algorithm proceeds to generate an equal number of new solutions based on Eq.9. During this process, it evaluates the nest information subsequent to the update while

preserving the existing optimal nest information.

$$x_i^{t+1} = x_i^t + r(x_g^t - x_k^t) \quad 9$$

where r is a scaling factor obeying a (0,1) uniform distribution, and x_{gt} and x_{kt} are two randomly selected nests in the population of the t th generation.

Step 5: The algorithm's evaluation hinges on the fulfillment of the termination condition. The algorithm completes and outputs the optimal value if the termination condition is satisfied. Conversely, if the termination condition remains unsatisfied, the process advances to step 3, and iteration persists.

ICS-PSO-OBL algorithm

It is clear from examining the conventional CS's iterative approach that the optimization process is bifurcated into two distinct phases: global search and local search. The algorithm largely relies on Lévy flights to investigate and locate the best answer during the global search. The approach uses a differential operation between two randomly produced solutions during the local search phase to further focus the search.

Nonetheless, the CS faces a challenge in balancing exploration and exploitation. This predicament arises from the pronounced randomness associated with Lévy flights and the relatively limited local search capabilities stemming from the differential operation. Additionally, in high-dimensional optimization problems, there is interference between different dimensions, further complicating the search process.

The proposed improved CS using PSO operator and OBL (ICS-PSO-OBL) is used to address these challenges and better balance global and local search due to following three improvements.

Integration with PSO Operator

The PSO algorithm will retain two sets of information, the personal optimum and the global optimum, to guide the particle-in search during the run¹⁶. This search method has a stronger local search capability compared with the CS's double stochastic solution as a difference search method.

So in the preferred random wandering phase of CS, an adaptive inertia weight PSO operator is used to replace the differential operation between two randomly generated solutions to enhance the local optimization capability.

For the i_{th} individual in the t_{th} generation, its position is updated by the following formulas (10) and (11).

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (pbest_i^t - x_i^t) + c_2 r_2 (gbest^t - x_i^t) \quad 10$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad 11$$

Where, v_i^t , v_i^{t+1} represent the speed of the i th individual in the t th and $t+1$ th generation, respectively; x_i^t , x_i^{t+1} represent the position of the i_{th} individual in the t_{th} and $t+1_{th}$ generation, respectively; $pbest_i^t$ represents the optimal value of the i_{th} individual from the 1st generation to the t_{th} generation; $gbest^t$ represents the optimal value among all individuals in the t_{th} generation; c_1 , c_2 are learning factors, representing the "self-awareness" and "social awareness" of the individuals in the algorithm; r_1 and r_2 are random numbers between [0,1]; ω is the inertia weight, which determines how the speed of the next generation is influenced by the speed of the previous generation. The variance, i.e., the aggregation of individuals in the population, is updated adaptively. As the iterations continue, when the algorithm starts to converge, i.e., the variance decreases, the distribution of individuals in the population gradually concentrates and the population diversity decreases, ω increases accordingly at this time to avoid premature convergence of the algorithm, and ω is updated in the following way.

$$\omega_t = \omega_{min} + (\omega_{max} - \omega_{min}) \times D_t \quad 12$$

Where, ω_t represents the inertia weight ω in the t th generation; ω_{min} and ω_{max} represent the minimum maximum of ω , respectively; and D_t is the diversity function and calculated by Eq.13.

$$D_t = 1 - \frac{2}{\pi} \arctan(E) \quad 13$$

where E is the fitness variance for the entire

population as determined by Eq.14.

$$E = \frac{1}{N} \sum_{i=1}^N (f(x_i) - f_{ave})^2 \quad 14$$

Where, $f(x_i)$ is the adaptation value of the current i th individual; $f_{ave} = \frac{1}{N} \sum_{i=1}^N f(x_i)$ is the average adaptation value of all current individuals.

OBL Strategy

OBL is a search strategy that compares the separation between a point and a random point in its opposite position and the optimal solution, recursively halving the search interval to search for the optimal solution faster¹⁷. ICS-PSO-OBL incorporates the OBL strategy to broaden the search space exploration and enhance the population's diversity. To further solve the problem of substantial computational effort caused by the OBL strategy to form the reverse solution for all individuals in high-dimensional optimization problems, ICS-PSO-OBL divides the elite individuals by fitness value and only performs the reverse solution operation on the elite individuals to reduce the computational effort and improve the search efficiency.

Let the set $B_i = (b_{i,1}, b_{i,2}, \dots, b_{i,D})$ be the elite individuals in the current cuckoo population, and its corresponding inverse solution $\bar{B}_i = (\bar{b}_{i,1}, \bar{b}_{i,2}, \dots, \bar{b}_{i,D})$ is calculated using Eq.15:

$$\bar{B}_{i,j} = \varepsilon (x_j + y_j) - b_{i,j} \quad 15$$

Where: $\varepsilon \in (0,1)$ is the generalization factor; $b_{i,j} \in [x_j, y_j]$ is denoted as the dynamic boundary of the search space in the j th dimension, x_j , y_j are the upper and lower bounds respectively, which could be calculated by $x_j = \min(B_{i,j})$, $y_j = \max(B_{i,j})$.

Dimension-by-dimension Updating

When solving multidimensional optimization problems using the approach of updating old solutions in all dimensions, one commonly faces the issue of interferences among different dimensions¹⁸. This means that when the solution is updated, each dimension changes, which may deteriorate the solution's fitness. Consequently, the new solution might be discarded, neglecting the impact of

individual dimension changes on the solution's fitness. To address this problem, the strategy of evaluating dimensions one by one (the sequential update evaluation strategy) is introduced to effectively mitigate the coupling phenomenon among dimensions in issues with high-dimensional optimization. The specific strategy is as follows:

A new solution is created whenever a value in the t_{th} dimension of a solution is updated by combining it with the vectors of all other dimensions. If the quality of the new solution does not improve, the update in the current dimension is discarded, and the old value from the t_{th} dimension is retained. The evaluation then continues with the $t+1_{th}$ dimension. If the quality of the new solution improves, the new value from the t_{th} dimension is retained, and the evaluation proceeds to the $t+1_{th}$ dimension. In the CS, applying the sequential update evaluation strategy in the local search stage effectively utilizes the exceptional information from individual dimensions to guide the current solution in local search, resulting in higher-quality solutions.

The implementation steps for ICS-PSO-OBL are as follows.

Function optimization test experiment and analysis

Selection of benchmarking functions

Ten benchmarking functions are chosen in this part to evaluate the viability of the suggested ICS-PSO-OBL, and numerical comparisons of the optimization results are made with the standard CS¹⁰ and three other CS variants, MACS¹⁹, ICS-2²⁰,

Step 1: Initialize parameters.

Step 2: The process begins with creating a new population, and each individual's fitness is evaluated. Eq.5 stochastically generates the initial population within the search space.

Step 3: Update the positions of all nests using Eq.6-8 and greedily select nests with higher fitness to retain.

Step 4: For each nest, generate a random number in the range of 0 to 1, and compare it with the discovery probability (P_a) to determine the nests that need to be updated. Determine the inertia weight using Eq.12-14 and update the nest positions using the sequential update strategy in Eq.10-11. Greedily select nests with higher fitness to retain.

Step 5: After selecting the top 1/10 individuals based on fitness, generate reverse solutions using Eq.15 and update the current nest positions. Greedily select nests with higher fitness to retain.

Step 6: Output the best solution if the termination condition is satisfied after the current iteration; otherwise, return to Step 3.

ICS²¹. Each algorithm was optimized independently for each test function 30 times, and a maximum of 1000 iterations was established to assure the test's impartiality and reduce the impact of randomness in the algorithmic optimization process. Table 1 records the selected functions. Table 1 records the selected functions.

Table 1. Ten benchmark functions in the experiment.

No	Function	Formula	Dim	Interval	f(x*)
F1	Sphere	$F_1 = \sum_{i=1}^D x_i^2$	30	[-5.12,5.12]	0
F2	Step	$F_2 = \sum_{i=1}^D [x_i + 0.5]^2$	30	[-100,100]	0
F3	Schwefel 2.22	$F_3 = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	30	[-100,100]	0
F4	Rosenbrock	$F_4 = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[-30,30]	0
F5	Ackley	$F_5 = -20 \exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$	30	[-32,32]	0

F6	Griewank	$F_6 = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0
F7	Rastrigin	$F_7 = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	30	[-600,600]	0
F8	Bohachevsky	$F_8 = \sum_{i=1}^{D-1} \left[x_i^2 + 2x_{i+1}^2 - 0.3 \cos(3\pi x_i) \right] - 0.4 \cos(4\pi x_{i+1}) + 0.7$	30	[-100,100]	0
F9	Beale	$F_9 = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$	2	[-4.5,4.5]	0
F10	SchafferN.2	$F_{10} = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{\left[1 + 0.001(x_1^2 + x_2^2)\right]^2}$	2	[-100,100]	0

Parameter settings

based on references ^{6, 16, 17, 18}.

According to Table 2, the parameters was created

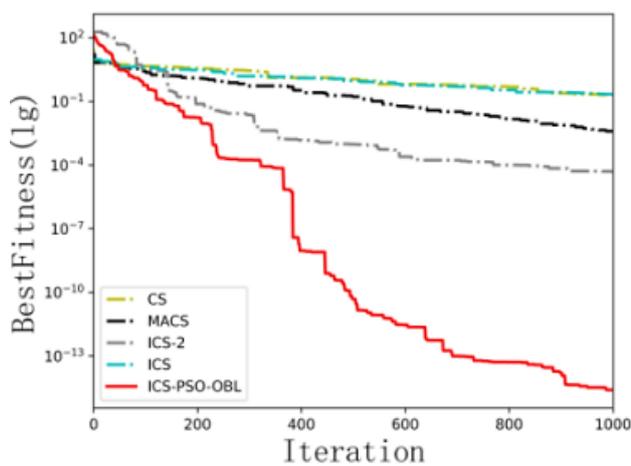
Table 2. Initialization parameters of all algorithms.

Initialization parameters for each algorithm	
CS	$\alpha_0 = 0.01, P_a = 0.25$
MACS	$\alpha_0 = 0.01, P_{a \max} = 0.25, P_{a \min} = 0.15$
ICS-2	$\alpha_0 = 0.01, P_{a \max} = 1, P_{a \min} = 0.3$
ICS	$\alpha_0 = 0.01, P_a = 0.25, P_b = 0.35$
ICS-PSO-OBL	$C_1 = C_2 = 1, \omega_{\max} = 0.6, \omega_{\min} = 0.05, V_{\max} = 1, V_{\min} = -1, \alpha_0 = 1.3, P_a = 0.25$

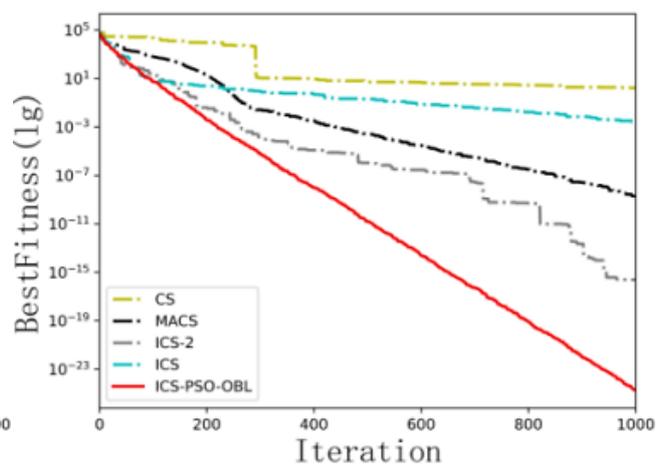
Optimization test results

Fig. 1 represents the average convergence curve of 5 algorithms for functions F1-F8, independently

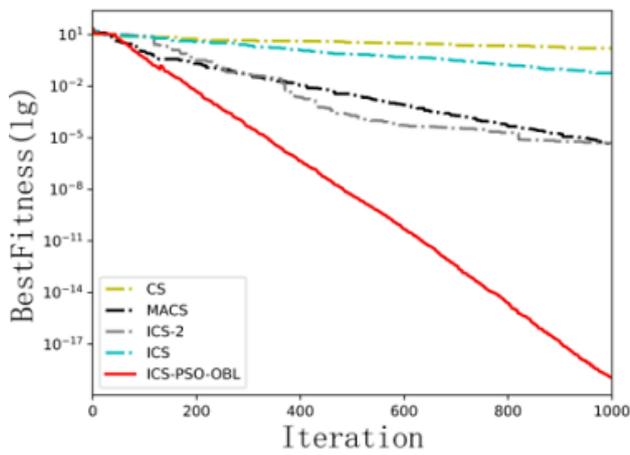
optimized 1000 times in 30-dimensional space. Fig. 2 represents the average convergence curve of 5 algorithms for function F9-F10 in 2-dimensional space for 1000 times of independent optimization.



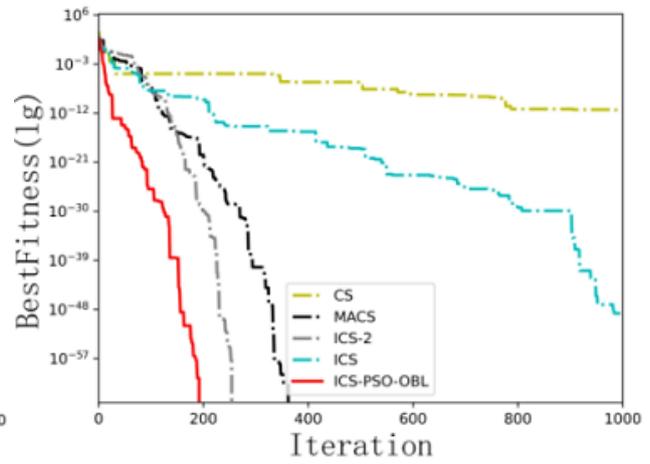
a. Average convergence of F1



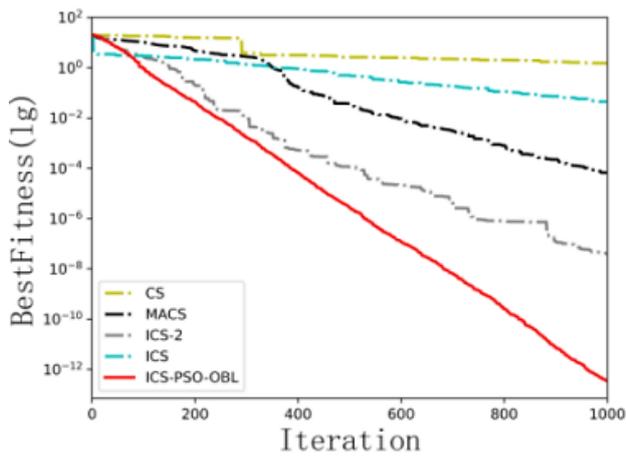
b. Average convergence of F2



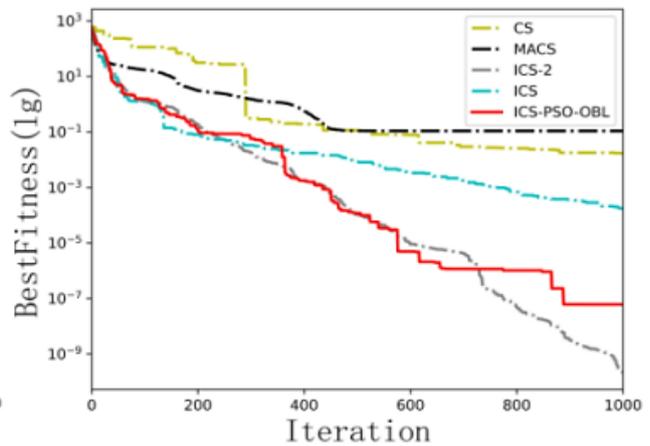
c. Average convergence of F3



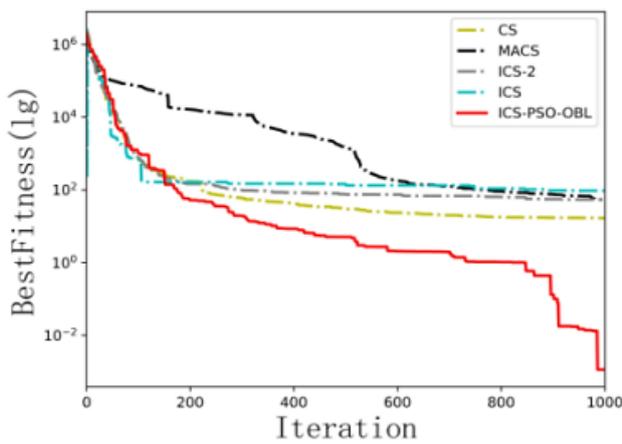
d. Average convergence of F4



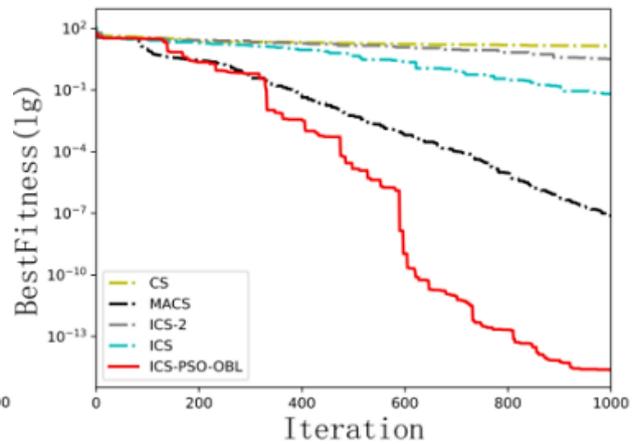
e. Average convergence of F5



f. Average convergence of F6



g. Average convergence of F7



h. Average convergence of F8

Figure 1. Optimization of test functions in 30 dimension.

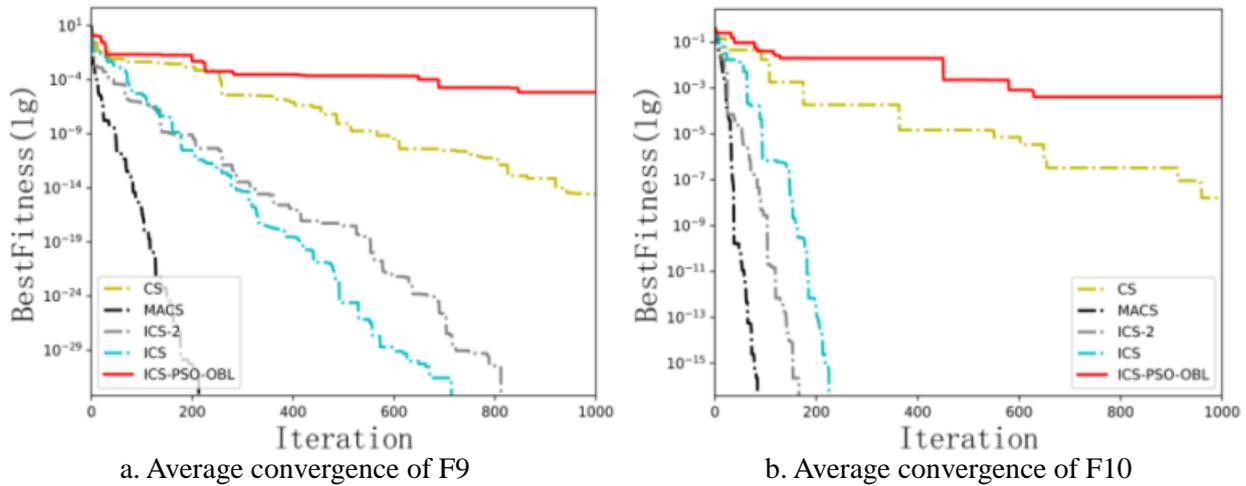


Figure 2. Optimization of test functions in 2 dimension.

Table 3 lists the five methods' outcomes of optimization in a 30-dimensional space for the functions F1-F8 after 1000 iterations, and Table 4 lists the 5-algorithm optimization outcomes in 2-dimensional space for the functions F9-F10 after 1000 iterations. *min* denotes the optimal value of

the algorithms in 30 optimization attempts, *mean* represents the average value of the algorithms' outcomes over 30 different optimization efforts, while *std* represents the algorithms' standard deviation.

Table 3. Test function optimization results under 30 dimension.

No	Function	Indicator	CS	MACS	ICS-2	ICS	ICS-PSO-OBL
F1	Sphere	Best	1.88E-02	4.24E-11	1.07E-05	6.30E-03	1.00E-20
		Mean	1.73E-01	1.57E-09	4.93E-05	1.23E-01	2.15E-16
		Std	5.83E-02	1.65E-09	3.36E-05	6.47E-02	6.58E-16
F2	Step	Best	7.78E-02	4.93E-11	1.08E-18	3.17E-06	1.94E-25
		Mean	7.09E-01	2.72E-09	1.38E-12	3.91E-03	3.77E-24
		Std	2.01E-01	3.75E-09	7.46E-12	3.24E-03	4.45E-24
F3	Schwefel 2.22	Best	4.47E-01	2.25E-06	2.44E-20	1.25E-04	2.44E-20
		Mean	1.65E+00	6.44E-06	1.26E-19	4.57E-02	1.16E-19
		Std	3.46E-01	3.78E-06	8.81E-20	2.33E-02	6.87E-20
F4	Rosenbrock	Best	1.22E-20	0.00E+00	0.00E+00	0.00E+00	0.00E+00
		Mean	5.39E-13	0.00E+00	0.00E+00	1.66E-31	0.00E+00
		Std	1.01E-12	0.00E+00	0.00E+00	9.10E-31	0.00E+00
F5	Ackley	Best	3.24E-01	1.00E-05	1.98E-09	2.43E-04	1.53E-13
		Mean	9.79E-01	9.19E-05	1.05E-07	2.54E-02	8.52E-13
		Std	3.09E-01	2.21E-04	1.66E-07	1.45E-02	7.68E-13
F6	Griewank	Best	4.87E-04	5.14E-09	0.00E+00	5.90E-05	2.42E-10
		Mean	5.91E-03	3.10E-01	6.67E-05	2.03E-04	1.08E-02
		Std	2.41E-03	3.76E-01	1.92E-04	9.96E-05	1.51E-02
F7	Rastrigin	Best	1.11E+01	5.14E+01	3.29E+01	7.37E+01	5.12E-10
		Mean	1.47E+01	8.32E+01	5.77E+01	1.19E+02	3.85E-02
		Std	1.89E+00	1.60E+01	2.78E+01	1.84E+01	1.79E-01
F8	Bohachevsky	Best	4.88E-15	6.74E-09	1.40E-02	1.28E-11	0.00E+00
		Mean	6.15E-09	1.77E-01	4.72E+00	1.95E-01	5.35E-10
		Std	1.11E-08	2.83E-01	2.76E+00	2.19E-01	2.87E-09

Table 4. Test function optimization results under 2 dimension.

No	Function	Indicator	CS	MACS	ICS-2	ICS	ICS-PSO-OBL
F9	Beale	Best	1.02E-18	0.00E+00	0.00E+00	0.00E+00	5.92E-04
		Mean	7.38E-14	0.00E+00	0.00E+00	0.00E+00	9.33E-05
		Std	3.94E-13	0.00E+00	0.00E+00	0.00E+00	1.33E-04
F10	SchafferN.2	Best	5.18E-11	0.00E+00	0.00E+00	0.00E+00	8.28E-07
		Mean	3.23E-08	0.00E+00	0.00E+00	0.00E+00	2.21E-03
		Std	7.99E-08	0.00E+00	0.00E+00	0.00E+00	3.09E-03

By studying Figs 1-2 and Tables 3-4, the following conclusions may be made.

For F1, for this kind of continuous single-peaked spherical function with only a unique global minimum, the other algorithms except ICS-PSO-OBL have a gentle downward trend. At the same time, the convergence curve of ICS-PSO-OBL is sometimes horizontal. Sometimes, there is a jump, and the overall shape is a ladder. In addition, ICS has a similar iteration trend compared with the standard CS, which does not show a significant advantage. In contrast, MACS and ICS-2 show significant differences in the iteration trend with the standard CS, which has a faster convergence speed.

For F2, because the values of such functions are distributed in each dimension in a step-like manner and are not derivable at the step boundaries, it is more difficult to find the correct direction toward the global optimal solution. All the functions cannot converge stably to the global optimum, in which all four algorithms significantly outperform the standard CS. The ICS-PSO-OBL shows a clear advantage when facing this kind of function, with an iterative tendency that is better than the other three CS variants and a higher search accuracy for the optimum.

For F3, the solution space of such functions will show intense complexity at higher dimensions. However, in the 30-dimensional search space, all four CS variants can be optimized efficiently, among which, ICS-2 decreases faster in the early stage while maintaining similar convergence accuracy with MACS in the later stage; the iterative trend of ICS-PSO-OBL has a slanting trend, which indicates that the dimension-by-dimension updating strategy allows the current optimal solution of ICS-PSO-OBL to be updated efficiently at each

iteration.

For F4, MACS, ICS-2, and ICS-PSO-OBL converge to the theoretical optimum within 400 iterations. However, ICS-PSO-OBL combined with the PSO operator significantly outperforms the other two CS variants regarding convergence speed. ICS does not converge to the global optimum, but its convergence trend is still better than standard CS.

For F5, for this kind of test function with multiple extreme points, MACS and ICS-2 with adaptive discovery probability can balance their own local and global search ability, which makes them have strong optimization ability in the face of multiple extreme points, whereas ICS has added the current optimal solution as a guide in the local search phase, which does not show a good effect in the face of this kind of function with multiple extreme points; and ICS-PSO-OBL is not converged to the global optimal value, but it has the optimal iterative tendency.

For F6 and F7, the two test functions are similar, with ample search space and many local minima. However, the performance of ICS-2 is diametrically opposite in F6 and F7: ICS-2 achieves the optimal convergence accuracy in F6. At the same time, the iterative tendency of ICS-2 is even worse than that of the standard CS in F7. The improvement strategies of ICS-2 and MACS may not be as effective as those of the standard CS, suggesting that the improvement strategies of ICS-2 and MACS may not achieve better results when facing such functions with ample search space. Meanwhile, MACS is not as good as standard CS, indicating that the improved strategies of ICS-2 and MACS may not achieve good results when facing this kind of function with ample search space. ICS-PSO-OBL shows a stepwise decrease, indicating that the

algorithm has fallen into the local optimum many times, possibly due to the dimension-by-dimension updating strategy to ensure that each dimension has been updated. However, the final combination of PSO ICS-PSO-OBL, which combines the PSO operator, dimension-by-dimension updating, and the opposition-based learning strategy, still achieves the best optimization results. ICS has the lowest standard deviation in function F6, which indicates that it has the highest stability. At the same time, it takes 200 iterations to enter a local optimum in function F7, which makes it ineffective for optimization.

For F8, ICS-2, ICS, and standard CS show similar iterative trends, with no significant difference in convergence accuracy and convergence speed compared to standard CS, and standard CS has the lowest standard deviation and higher stability compared to ICS-2 and ICS. The convergence accuracy and speed of MACS are far higher than those of the CS but still insufficient compared to ICS-PSO-OBL. ICS is similar to ICS-PSO-OBL, which combines PSO operators. Both add the current global optimal solution as a guide during the local search process. However, the addition of the other search strategies of ICS-PSO-OBL leads to the final optimization result of ICS being inferior to

that of ICS-PSO-OBL. Inferior to ICS-PSO-OBL.

For F9 and F10, for such low-dimensional test functions, MACS, ICS-2, and ICS all converge stably to the optimal value, compared to which the iterative trend of ICS-PSO-OBL is even worse than that of the standard CS, which suggests that the improvement strategy of ICS-PSO-OBL fails to achieve a better result when facing such low-dimensional test functions.

Overall, in the optimization experiments on 10 test functions, ICS-PSO-OBL has a better iteration trend in the face of high-dimensional test functions and has the best iteration trend, convergence accuracy, and algorithmic stability, except for function F6; however, its improvement strategy is ineffective in the face of low-dimensional test functions and even reduces the convergence accuracy and speed. The above results show that ICS-PSO-OBL, which integrates PSO operator, dimension-by-dimension updating, and opposition-based learning strategy, has more vital optimization ability and stability than the standard CS and the rest of the three CS variants when facing high-dimensional functions, which also verifies the effectiveness of ICS-PSO-OBL from the side.

WSN Coverage Optimization Experiment and Analysis

Parameter setting

To verify the CS's efficiency in solving the WSN coverage issue, the standard CS is used to optimize the problem with the coverage rate (Cr) (Eq.4) as the fitness value. In addition, the MACS, ICS-2, ICS, and ICS-PSO-OBL algorithm are also included

for comparison in the same environment to demonstrate the superiority of the ICS-PSO-OBL over the other algorithms. According to reference 3, the experimental parameter settings are created as presented in Table 5.

Table 5. Parameter setting of WSN coverage experiment.

Parameters	Value
Monitoring area	100m × 100m
Number of nodes	20/30
Node perception radius r	12m
Number of iterations T	200
Number of Pixel Dots	100 × 100

The comparison algorithm is parameterized based on the Table 2.

Optimize coverage results

Table 6 lists the coverage rate values obtained by CS, MACS, ICS-2, ICS, and ICS-PSO-OBL to

optimize the WSN coverage problem under different scenarios with node counts of 20 and 30.

Table 6. Comparison of coverage results.

Algorithm/Number of Nodes	Coverage by algorithm	
	20	30
CS	64.22%	80.50%
MACS	74.69%	87.83%
ICS-2	67.58%	85.17%
ICS	73.56%	89.69%
ICS-PSO-OBL	82.58%	97.44%

From Table 6, it could be observed that in the case of 20 nodes, the coverage rate of the ICS-PSO-OBL is 18.36% higher than that of the CS, 7.89% higher than that of the MACS, 15% higher than that of the ICS-2, and 9.02% higher than that of the ICS. Similarly, in the case of 30 nodes, the coverage rate of the ICS-PSO-OBL is 16.94% higher than that of the CS, 9.61% higher than that of the MACS, 12.27% higher than that of the ICS-2, and 7.75% higher than that of the ICS. Therefore, the ICS-PSO-OBL

consistently achieves higher coverage rates compared to the CS and CS variants, regardless of the number of nodes.

Sensor node distribution map

Fig. 3-7 depict the node distribution for optimizing WSN coverage using the CS, MACS, ICS-2, ICS, and ICS-PSO-OBL under scenarios with 20 and 30 nodes.

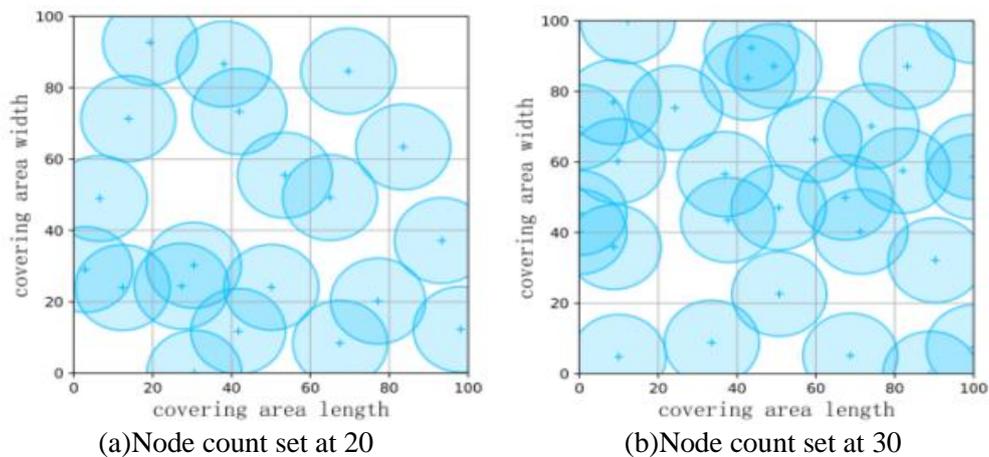


Figure 3. Optimize coverage of sensor networks using CS.

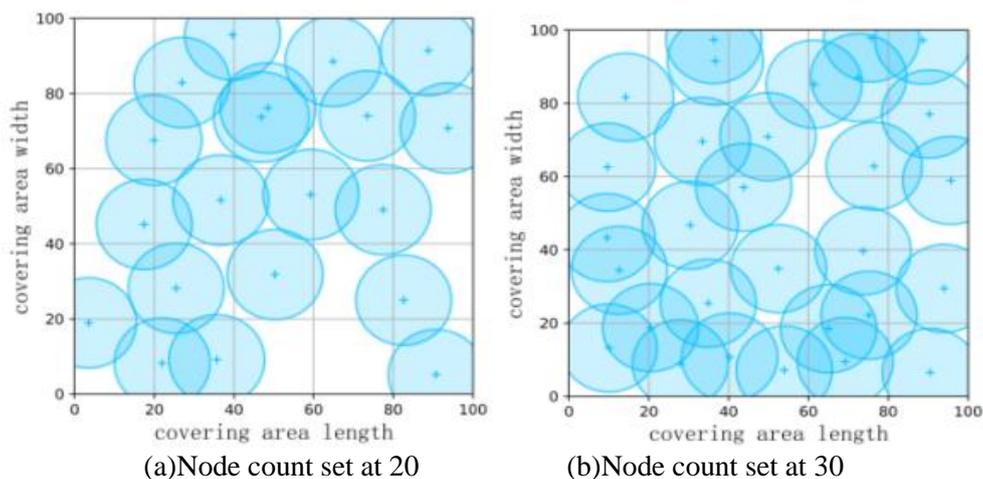


Figure 4. Optimize coverage of sensor networks using MACS.

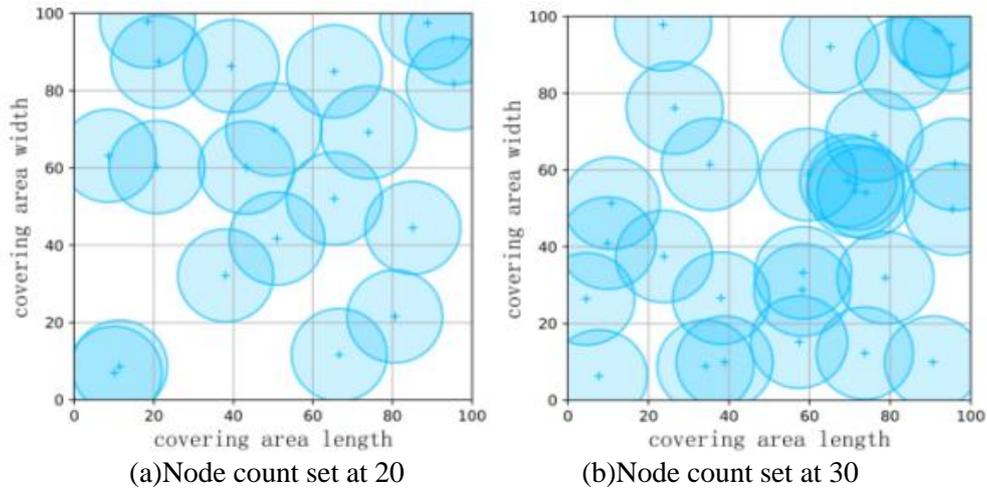


Figure 5. Optimize coverage of sensor networks using ICS-2.

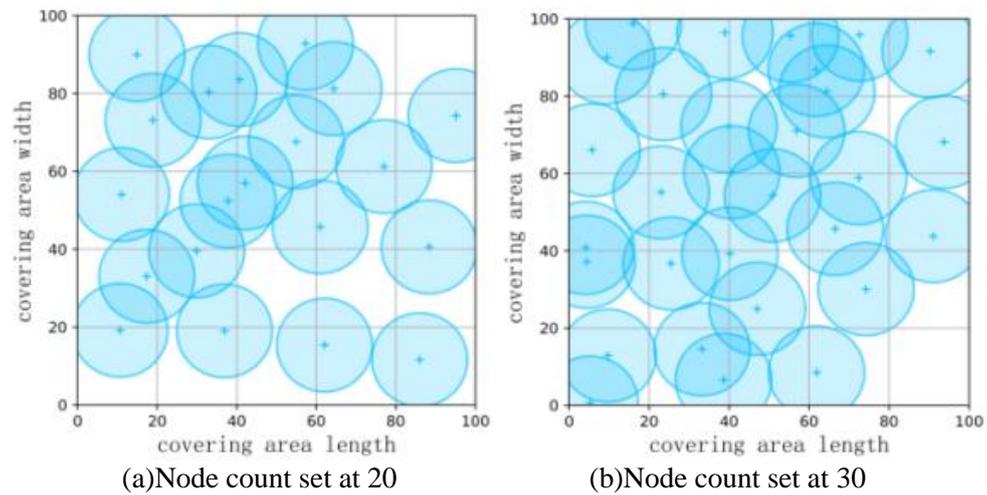


Figure 6. Optimize coverage of sensor networks using ICS.

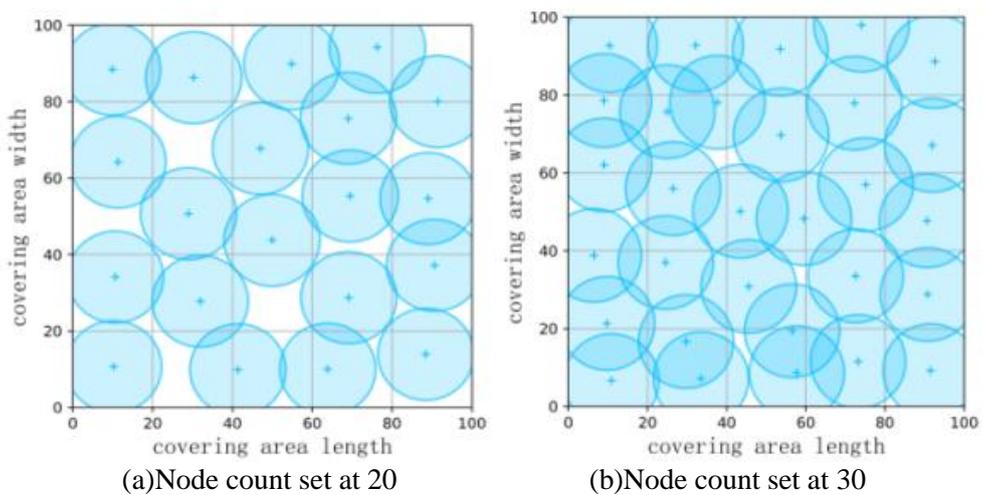


Figure 7. Optimize coverage of sensor networks using ICS-PSO-OBL.

Fig. 3-7 show that, whether with 20 or 30 sensor nodes, the optimization results of CS, MACS, ICS-2, and ICS algorithms are inferior to that of the ICS-PSO-OBL algorithm. The four algorithms

exhibit significant coverage holes and high redundancy in the monitored area. In contrast, the ICS-PSO-OBL achieves a more uniform node distribution, reduced redundancy, and a more

comprehensive coverage range in the wireless network.

Coverage convergence graph

Fig. 8 represents the convergence curve of coverage

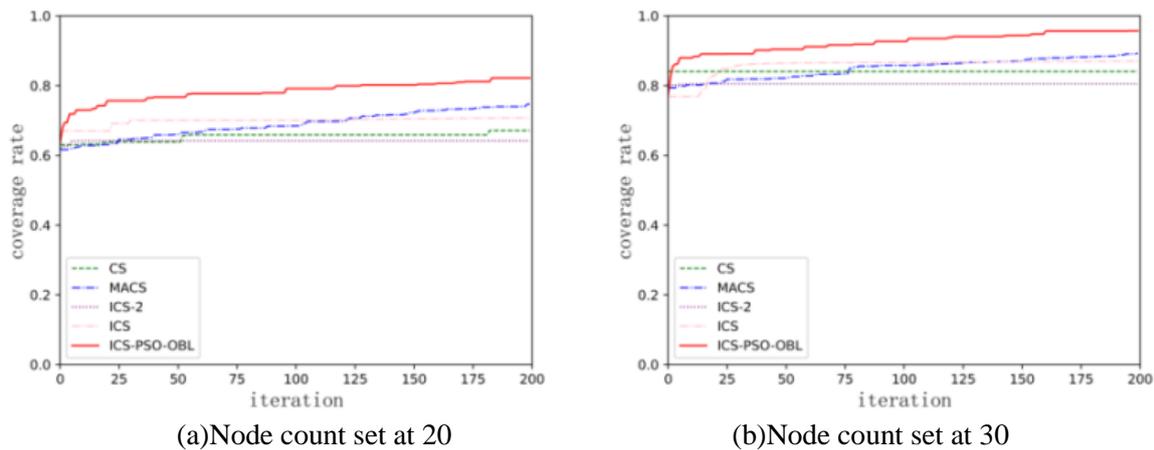


Figure 8. Convergence curves.

Fig. 8 shows that in the optimization of WSN coverage, the standard CS enters a stagnation state too early and gets trapped in local optima for both 20 and 30 sensor nodes. While MACS, ICS-2, and ICS algorithms show effectiveness in optimization, their convergence speed and accuracy are inferior to

rates for CS, MACS, ICS-2, ICS, and ICS-PSO-OBL algorithms during 200 iterations in the optimization of WSN coverage for both 20 and 30 sensor nodes.

ICS-PSO-OBL, regardless of the number of nodes being 20 or 30. Therefore, compared to CS, MACS, ICS-2, and ICS algorithms, the ICS-PSO-OBL demonstrates more substantial competitiveness regarding search accuracy, convergence speed, and stability in optimizing WSN coverage.

Conclusion

The ICS-PSO-OBL is the foundation of the network coverage approach proposed in this paper. ICS-PSO-OBL effectively balances the algorithm's capability for local and global search, and reduces the interference between dimensions in high-dimensional optimization problems by combining the PSO operator with the dimension-by-dimension update local search method. By choosing elite individuals based on their fitness scores, the algorithm performs reverse solution operations to broaden the search area and increase optimization precision, and its advantages

over the other three CS variants are examined by optimization tests with ten benchmark functions.

Finally, the application of ICS-PSO-OBL is extended to address optimization challenges in WSN coverage. Through simulation experiments, it is found that ICS-PSO-OBL could significantly improve the WSN coverage compared to the standard CS and its three variants. It achieves a more uniform distribution of sensor nodes. Moreover, applying ICS-PSO-OBL reduces the number of nodes and lowers deployment costs.

Acknowledgments

Project supported by the National Natural Science Foundation of China (No. 62066016), the Research Foundation of Education Bureau of Hunan Province, China (No. 22B0549, 22B1046), the Postgraduate

Scientific Research Innovation Project of Hunan Province (CX20231088), the Xiangxi State Science Program Project-Collaborative control of UAV swarm system based on improved cuckoo algorithm

(State Financial Education Index [2022] No.5), the Postgraduate Scientific Research Innovation Project

of Hunan Province (CX20231088).

Authors' Declaration

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Furthermore, any Figures and images, that are not ours, have been

- included with the necessary permission for re-publication, which is attached to the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in Jishou University.

Authors' Contribution Statement

S. Y. and K.Z. designed and directed the project. Y. X. and D. K. performed the experiments. All authors

contributed to the final manuscript.

References

1. Elhabyan R, Shi W, St-Hilaire M. Coverage protocols for wireless sensor networks: Review and future directions. *J.Commn.Net.* 2019 Feb;21(1):45-60.<https://doi.org/10.1169/JCN.2019.00005>
2. Saeedi ID, Al-Qurabat AK. Perceptually important points-based data aggregation method for wireless sensor networks. *Baghdad Sci. J.* 2022 Aug 1;19(4):0875.<https://doi.org/10.21123/bsj.2022.19.4.0875>
3. Maheshwari A, Chand N. A survey on wireless sensor networks coverage problems. In *Proceedings of 2nd International Conference on Communication, 6 Chandigarh, India .2019; (pp. 153-164).* Springer Singapore. https://doi.org/10.1007/978-981-13-1217-5_16
4. Yasear SA, Ku-Mahamud KR. Taxonomy of memory usage in swarm intelligence-based metaheuristics. *Baghdad Sci. J.* 2019;16(2):0445.[http://dx.doi.org/10.21123/bsj.2019.16.2\(SI\).0445](http://dx.doi.org/10.21123/bsj.2019.16.2(SI).0445)
5. Singh A, Sharma S, Singh J. Nature-inspired algorithms for wireless sensor networks: A comprehensive survey. *Comput. Sci. Rev.* . 2021 Feb 1;39:100342.<https://doi.org/10.1016/j.cosrev.2020.100342>
6. Duan J, Yao AN, Wang ZT, Yu L. An Improved Sparrow Search Algorithm Optimizes Coverage in Wireless Sensor Networks. *Eng. Technol. Ed.* . [cited 2023 Feb 21];1 - 11. <http://kns.cnki.net/kcms/detail/22.1341.T.20220726.1034.003.html>.
7. Mottaki NA, Motameni H, Mohamadi H. An effective hybrid genetic algorithm and tabu search for maximizing network lifetime using coverage sets scheduling in wireless sensor networks. *J. Supercomput.* 2023 Feb;79(3):3277-97.<https://doi.org/10.1007/s11227-022-04710-1>
8. He Q, Lan Z, Zhang D, Yang L, Luo S. Improved marine predator algorithm for wireless sensor network coverage optimization problem. *Sustainability.* 2022 Aug 11;14(16):9944.<https://doi.org/10.3390/su14169944>
9. Musa JA, Romli R, Yusoff N. An analysis on the applicability of meta-heuristic searching techniques for automated test data generation in automatic programming assessment. *Open Access Baghdad Sci J.* 2019;16(SI):515-33.[https://doi.org/10.21123/bsj.2019.16.2\(SI\).0515](https://doi.org/10.21123/bsj.2019.16.2(SI).0515)
10. Yang XS, Deb S. Engineering optimisation by cuckoo search. *IJMMNO.* 2010 Jan 1;1(4):330-43.<https://doi.org/10.1504/IJMMNO.2010.03543>
11. Mohamad AB, Zain AM, Nazira Bazin NE. Cuckoo search algorithm for optimization problems—a literature review and its applications. *Appl. Artif. Intell.* . 2014 May 28;28(5):419-48.<https://doi.org/10.1080/08839514.2014.904599>
12. Ye SQ, Wang FL, Ou Y, Zhang CX, Zhou KQ. An improved cuckoo search combing artificial bee colony operator with opposition-based learning. In *2021 China Automation Congress (CAC).* 2021 Oct 22; pp. 1199-1204. *IEEE.* <https://doi.org/10.1109/CAC53003.2021.9727912>
13. Zhang CX, Zhou KQ, Ye SQ, Zain AM. An improved cuckoo search algorithm utilizing nonlinear inertia weight and differential evolution for function optimization problem. *IEEE Access.* 2021 Nov 25;9:161352-73.<https://doi.org/10.1109/ACCESS.2021.3130640>

14. Li PC, Zhang XY, Zain AM, Zhou KQ. An Improved Cuckoo Search Algorithm Using Elite Opposition-Based Learning and Golden Sine Operator. In International Conference on Adaptive and Intelligent Systems. Cham: Springer International Publishing. 2022 Jul 4 ;(pp. 276-288). https://doi.org/10.1007/978-3-031-06794-5_23
15. Yang XS, Deb S. Cuckoo search via Lévy flights. NaBIC. 2009 Dec 9; pp. 210-214. Ieee. <https://doi.org/10.1109/NABIC.2009.5393690>
16. Iqbal Z, Ilyas R, Chan HY, Ahmed N. Effective Solution of University Course Timetabling using Particle Swarm Optimizer based Hyper Heuristic approach. Baghdad Sci. J. . 2021 Dec 20;18(4 (Suppl.)):1465-. [https://doi.org/10.21123/bsj.2021.18_4\(Suppl.\).1465](https://doi.org/10.21123/bsj.2021.18_4(Suppl.).1465)
17. Tizhoosh HR. Opposition-based learning: a new scheme for machine intelligence. In International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06). 2005 Nov 28; Vol. 1: pp. 695-701). IEEE. <https://doi.org/10.1109/CIMCA.2005.1631345>
18. Kamaruzaman AF, Zain AM, Yusuf SM, Udin A. Levy flight algorithm for optimization problems-a literature review. Appl. Mech. Mater. . 2013 Dec 12;421:496-501. <https://doi.org/10.4028/www.scientific.net/AMM.421.496>
19. Zhang ZZ, He XS, Yu QL, Yang XS. Cuckoo Algorithm for Multi-stage Dynamic Disturbance and Dynamic Inertia Weight. Comput. Appl. Eng. Educ. . 2021 Apr 29;58(01):79-88. https://doi.org/10.3778/j.issn.1002-8331.2012-02_81
20. Sun M, Wei H. An improved adaptive inertial weight Cuckoo algorithm. Journal of Yangtze University(Natural Science Edition). 2019 Jul 17;16(07):81-87. <https://doi.org/10.16772/j.cnki.1673-1409.2019.07.016>
21. Zheng HQ, Feng WJ. An improved Cuckoo Search algorithm for Constrained Optimization Problems. Chinese Journal of Engineering Mathematics. 2023 Feb 15;40(01);135-146. <https://doi.org/10.3969/j.issn.1005-3085.2023.01.010>

خوارزمية بحث الوقواق المحسنة لزيادة نطاق التغطية لشبكات الاستشعار اللاسلكية

سين-يو يانغ، بين-هونغ شيانغ، دي-وين كانغ، كاي-تشينغ تشو

مدرسة الاتصالات والهندسة الإلكترونية، جامعة جيشو، جيشو، 416000، هونان، الصين.

الخلاصة

تمت معالجة مسألة زيادة النطاق الذي تغطيه شبكة أجهزة الاستشعار اللاسلكية باستخدام أجهزة الاستشعار المقيدة باستخدام CS المحسنة التي تستخدم خوارزمية PSO والتعلم القائم على المعارضة (ICS-PSO-OBL) في البداية، يتم تنفيذ التكرار عن طريق تحديث بعد الحل القديم نلو الآخر لتحقيق تحديث مستقل عبر الأبعاد في مشكلة التحسين عالية الأبعاد. يتم بعد ذلك دمج مشغل PSO لتقليل تفضيل مرحلة المشي العشوائي بين القدرة على الاستكشاف والاستغلال. يتم اختيار أفراد استثنائيين من بين السكان باستخدام OBL لتعزيز فرصة العثور على الحل الأمثل بناءً على قيمة اللياقة البدنية. يتم استخدام ICS-PSO-OBL لتعظيم التغطية في WSN عن طريق تحويل المراقبة الإقليمية إلى مراقبة النقاط باستخدام طريقة التمييز في WSN. في التجارب، يتم استخدام ICS-PSO-OBL مع CS القياسي وثلاثة متغيرات CS (MACS، ICS-2، و ICS)، و ICS للتنفيذ تجربة المحاكاة تحت أعداد مختلفة من العقد (20 و 30، على التوالي). تكشف النتائج التجريبية أن التغطية المحسنة لـ ICS-PSO-OBL أعلى بنسبة 18.36% و 7.894% و 15% و 9.02% من التغطية القياسية لـ CS و MACS و ICS-2 و ICS عندما يكون عدد العقد 20. علاوة على ذلك، فهي أعلى بنسبة 16.94% و 9.61% و 12.27% و 7.75% عندما تكون كمية العقد 30، وسرعة تقارب ICS-PSO-OBL، وتوزيع العقد متفوق على الآخرين.

الكلمات المفتاحية: خوارزمية بحث الوقواق، تحديث البعد نلو الآخر، التعلم القائم على المعارضة، شبكة استشعار لاسلكي، مشغل PSO.