

Adaptive Coverage Optimization in Wireless Sensor Networks Using Hybrid Artificial Hummingbird Algorithm with Differential Evolution

Nouredine Boukhari^{1*}, Mohammed Amine Nemmich², Dalila Djoher Graba³, Asmaa Boudali⁴, Fatima Debbat⁵

¹Department of Mathematics, Evolutionary Engineering & Distributed systems Laboratory, Djillali Liabes University of Sidi Bel-Abbes, Algeria

* **Corresponding Author Email:** noureddine.boukhari@univ-sba.dz - ORCID: 0000-0002-9777-3952

²Department of Computer Science, Mathematics Laboratory, Djillali Liabes University of Sidi Bel-Abbes, Algeria

Email: amine.nemmich@univ-sba.dz - **ORCID:** 0000-0002-4087-1682

³Department of Mathematics, Evolutionary Engineering & Distributed systems Laboratory, Djillali Liabes University of Sidi Bel-Abbes, Algeria

Email: dalila.graba@univ-sba.dz - ORCID: 0000-0001-8911-2591

⁴Department of Electronics, Laboratory of coding and Security of Information, University of science and Technology of Oran Mohamed Boudiaf, Algeria

Email: asmaaboudali@univ-usto.dz - **ORCID:** 0000-0003-4865-5121

⁵Department of computer science, University of Mascara, Algeria

Email: debbatfati@yahoo.fr - **ORCID:** 0000-0001-8803-2995

Article Info:

DOI: 10.22399/ijcesen.5292

Received : 28 December 2025

Revised : 20 May 2026

Accepted : 30 May 2026

Keywords

Bio-Inspired Algorithm
Hybrid Algorithm
Network Coverage
CEC Benchmarks
Artificial hummingbird algorithm

Abstract:

Optimizing sensor placement for maximum coverage in Wireless Sensor Networks (WSNs) remains a critical challenge, often leading to sub-optimal solutions when using standard metaheuristics. This paper introduces an Adaptive Hybrid Artificial Hummingbird and Differential Evolution Algorithm (AHADE) to solve the WSN coverage problem. The primary novelty of this work lies not in the hybridization itself, but in the proposed adaptive mechanism that intelligently governs the search process. By monitoring population diversity and solution convergence rate in real-time, AHADE strategically triggers the powerful mutation operators of Differential Evolution only when the search process shows signs of stagnation. This targeted intervention prevents premature convergence and maintains a robust balance between global exploration and local exploitation. The efficacy of AHADE was validated on the CEC 2020/2022 benchmarks and a practical WSN coverage problem. The results are compelling: AHADE achieved a network coverage of up to 95.3% in dense deployments, a statistically significant improvement over eleven other state-of-the-art algorithms. Furthermore, a comprehensive study confirmed the synergistic effect of the adaptive hybridization, with AHADE consistently outperforming its standalone parent algorithms. These findings validate AHADE as a powerful and intelligent optimization tool for complex real-world engineering problems.

1. Introduction

The proliferation of Wireless Sensor Networks (WSNs) has become foundational to advancements in numerous sectors, including environmental monitoring, industrial automation, healthcare, and surveillance (Al-Dhabyani et al., 2024). A WSN is

composed of a multitude of small, spatially distributed sensor nodes that are tasked with monitoring physical or environmental conditions and transmitting the collected data to a central location (Alves et al., 2022). One of the most critical challenges in the design and deployment of these networks is the coverage optimization problem. The

primary goal is to deploy a finite number of sensor nodes to maximize the monitored area while ensuring network connectivity and minimizing energy consumption, thereby extending the operational lifespan of the network (Boukhari et al., 2024). Given the NP-hard nature of the node placement problem, traditional optimization techniques often prove inadequate. Consequently, metaheuristic algorithms, inspired by natural phenomena, have become a preferred method for finding high-quality solutions (Boukhari et al., 2021). Recent years have seen a surge in advanced algorithms applied to this problem. For instance, the Quantum-Vortex Grey Wolf Optimizer (QVGWO) has shown promise in enhancing exploration (Al-Dhabyani et al., 2024), and hybrid approaches combining Marine Predators Algorithm with Simulated Annealing have aimed to improve local search precision (Chen & Li, 2025). However, despite these advances, many state-of-the-art methods still exhibit limitations. They often rely on static control parameters that lack the flexibility to adapt to the changing dynamics of the search process, leading to an imbalanced trade-off between global exploration and local exploitation. This can result in premature convergence on sub-optimal sensor layouts, particularly in large-scale or complex environments.

To address this specific gap, this paper proposes a novel Adaptive Hybrid Artificial Hummingbird and Differential Evolution Algorithm (AHADE). While the hybridization of swarm and evolutionary algorithms is a well-established paradigm, the key contribution of this work lies not in the combination itself, but in the novel adaptive strategy that governs the interplay between the two algorithms. By monitoring population diversity and solution convergence rates in real-time, AHADE intelligently invokes the powerful mutation operators of Differential Evolution precisely when the search process is at risk of stagnation. This targeted, on-demand application of DE's diversification power distinguishes AHADE from static or randomly scheduled hybrid models, creating a more robust and efficient search trajectory.

The main contributions of this work are therefore articulated as follows:

1- The proposal of a novel adaptive hybridization mechanism that intelligently triggers DE operators based on real-time search dynamics, enhancing the balance between exploration and exploitation.

2- The application of the resulting AHADE algorithm to the WSN coverage optimization

problem, demonstrating its state-of-the-art performance in achieving superior area coverage.

3- A rigorous validation of AHADE against modern benchmark suites (CEC 2020 and CEC 2022) and a comprehensive study proving the synergistic benefit of the adaptive hybridization over its constituent parts.

4- The presentation of a statistically robust comparison against multiple state-of-the-art algorithms, confirming the significance of AHADE's performance improvements. The remainder of this paper is organized as follows. Section 2 reviews related works in metaheuristic algorithms and WSN optimization. Section 3 provides a detailed description of the proposed AHADE algorithm. Section 4 presents the experimental setup, results, and discussion. Finally, Section 5 concludes the paper and suggests directions for future research.

2. Related Works

This section situates the proposed AHADE algorithm within the broader context of metaheuristic optimization and its specific application to the WSN coverage problem. We provide a critical review of recent state-of-the-art literature to identify the prevailing challenges and establish a clear justification for the development of our adaptive hybrid approach.

2.1 Metaheuristic Optimization Paradigms

Metaheuristic algorithms, which are broadly inspired by phenomena in evolution, swarm intelligence, physics, and human behavior, have become a cornerstone of modern optimization (Chen, Y., & Li, J. (2025)). Swarm intelligence algorithms, which model the collective behavior of decentralized systems, have been particularly prominent. This family includes foundational methods like Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO), as well as more recent additions such as the Marine Predators Algorithm (MPA) and the Coati Optimization Algorithm (Egwuche et al., 2023; Fan et al., 2024).

Evolutionary algorithms represent another major paradigm, with Differential Evolution (DE) (Deepa & Venkataraman, 2021) standing out as a particularly robust and effective method. Introduced by Storn and Price, DE leverages a simple yet powerful mechanism of creating new candidate solutions by combining existing ones through mutation, crossover, and selection operators. Its ability to

maintain population diversity makes it a powerful engine for global exploration.

Recognizing that no single algorithm is universally superior, the hybridization of different metaheuristics has become a major research frontier. The goal is to design algorithms that synergistically combine the strengths of their parent methods (Deghbouch & Debbat, 2021). Recent examples include the work of (Fan et al., 2024), who integrated the Sparrow Search Algorithm with chaos theory to improve initialization and escape local optima. Similarly, (Wang et al., 2025) proposed a hybrid of the Whale Optimization Algorithm and Simulated Annealing to balance exploration and exploitation. However, a common limitation in many such hybrids is their reliance on static or pre-scheduled control mechanisms (Derrac et al., 2011), which may not respond effectively to the dynamic state of the search process. This highlights a persistent need for more intelligent, adaptive hybridization strategies.

2.2 WSN Coverage Optimization using Metaheuristics

Coverage models measure detection capacity and quality by using the geometric relationship between a point and a sensor. Typically, a sensor coverage model can be formulated as a function of the Euclidean distances between a point in space and the sensors (Wang, Liu, et al., 2023). If we assume that there is a point p located at coordinates (x, y) and a sensor node S_i located at coordinates (x_i, y_i) , then the Euclidean distance between point p and sensor node S_i can be expressed as follows:

$$d(S_i, p) = \sqrt{(x_i - x)^2 + (y_i - y)^2} \quad (1)$$

In order to formulate our problem mathematically so that it can be implemented using a computer modelling tool, let's assume that the deployment zone is a two-dimensional 2D plane, represented by an $M \times N$ unit grid.

We denote the set S of N sensors with $S = \{S_1, S_2, \dots, S_N\}$, where the coordinates denoting the position of node S_i are given by $S_i = \{x_i, y_i\}$, ($i = 1, 2, \dots, N$).

The coordinates of the grid points are represented by $p(x, y)$ corresponding to the target points to be covered by the sensor nodes. Using the binary detection model and with the aim to ensure complete coverage of the region of interest, the coverage matrix can be defined as follows:

$$C_{xy}(S_i) = \{1 \text{ if } d(S_i, p) \leq r_s \text{ 0 else} \quad (2)$$

The coverage rate is defined as follows:

$$rate_c = \frac{\sum_{i=1}^m \sum_{j=1}^n C_{xy}(S_i)}{m \times n} \quad (3)$$

After the formulation of the model of the coverage problem and after having the model validated for the implementation to find the optimal solution for the maximum WSN coverage rate, the following assumptions are made:

- Each sensor node is homogeneous and the communication radius R_c is equal to two times the perception radius R_s ;
- Each sensor is capable of perceiving all points that fall within the detection radius and communicating with all other sensors that fall within the communication radius;
- All the sensor nodes are heterogeneous, meaning the R_c is the same for all the nodes;

Within the communication range of a node, each sensor is capable of sensing the instantaneous position of other nodes;

The deployment of sensor nodes to maximize coverage in a WSN is a classic NP-hard problem, making it an ideal application for metaheuristic optimization. Early approaches successfully utilized algorithms like GA and PSO (Pan et al., 2021). More recently, research has focused on developing more sophisticated and tailored algorithms to address the specific challenges of WSNs (Zhong et al., 2024). For instance, Al-Dhabyani et al. (2024) introduced the Quantum-Vortex Grey Wolf Optimizer (QVGWO), which incorporates principles from quantum computing to enhance the exploratory capabilities of the standard GWO. While effective at exploration, the algorithm can be slow to converge once a promising region is identified. In another recent study, (Chen & Li, 2025) hybridized the Marine Predators Algorithm with Simulated Annealing (MPASA) for WSN deployment. The addition of SA improved the local search, but the algorithm's performance was found to be highly sensitive to the initial annealing schedule. Furthermore, a novel approach by (Zhong et al., 2024) utilized an improved Reptile Search Algorithm with a Gaussian mutation operator, which showed strong performance but still faced challenges in maintaining population diversity in larger, more complex problem instances.

A critical analysis of this recent work reveals a clear and consistent research gap: the lack of robust adaptive mechanisms (Wang, Wu, et al., 2023). Many state-of-the-art algorithms, while powerful,

operate with a fixed search strategy. They lack the ability to dynamically adjust their behavior in response to real-time metrics like population diversity or convergence speed. This can lead to an inefficient search, either by exploring too much when exploitation is needed, or by converging prematurely when the population has stagnated. Our proposed AHAD algorithm is designed specifically to address this gap by introducing an intelligent layer of control that adaptively balances the global search of DE with the local search of AHA.

3. The proposed ahade algorithm

To address the challenges of the WSN coverage problem, we propose a novel hybrid metaheuristic, the Adaptive Hybrid Artificial Hummingbird and Differential Evolution Algorithm (AHAD). This algorithm is designed to create a synergistic search process by combining the potent exploitation capabilities of the Artificial Hummingbird Algorithm (AHA) with the robust global exploration and diversity-preservation mechanisms of Differential Evolution (DE). This section first defines the notation used, then details the operational principles of the parent algorithms before presenting the architecture of the proposed adaptive hybrid model.

3.1 Parent Algorithms : AHA and DE

The foundation of our hybrid model rests on two well-established and powerful metaheuristics: the Artificial Hummingbird Algorithm and Differential Evolution.

Artificial Hummingbird Algorithm (AHA): AHA is a recent swarm-based algorithm that mimics the intelligent foraging strategies of hummingbirds (Zhao, Wang, & Mirjalili, 2022). It employs three distinct search behaviors:

Guided Foraging

A targeted search where hummingbirds move towards the best-so-far food source; Hummingbirds move toward under-explored solution using :

$$X_i^{new} = X_{target} + \alpha \cdot D \odot (X_i - X_{target}) \quad (4)$$

Where X_{target} Position of the target food source

$\alpha \sim N(0,1)$: Random step size.

$D \in \{0,1\}^{2N}$: Direction vector.

Territorial Foraging

A local search mechanism where hummingbirds explore the immediate vicinity of their current fine-tune solutions; Local refinement around the current position:

$$X_i^{new} = X_i + \beta \cdot D \odot X_i; \beta \sim N(0,1) \quad (5)$$

(1) Migratory Foraging

Where a stagnated hummingbird abandons its position and explores a new, random area to escape local optima. Reset poorly performing solutions to explore new regions :

$$X_{worst} \sim v(1, L^{2N}) \quad (6)$$

Where v denotes uniform distribution.

(2) Direction Vector D

Determined by three flight modes

○ **Axial:** D has 1 in one random dimension (e.g., $D = [0,1,0,0, \dots]$).

○ **Diagonal:** D has 1s in $[rand \cdot (2N - 2) + 1]$ random dimensions.

○ **Axial:** $D = [1,1, \dots, 1]$.

Differential Evolution (DE): DE is a classic and powerful evolutionary algorithm (11, 12) renowned for its effectiveness in global optimization. It operates on a population of candidate solutions and generates new candidates through three primary operations:

Mutation

where a "mutant vector" is created by calculating the weighted difference between other randomly selected vectors using a scaling factor, F ; For each individual vector x_i (the target vector) in the population, a mutant vector v_i is generated. A common and effective strategy, DE/rand/1, creates the mutant vector by selecting three distinct random individuals (x_{r1}, x_{r2}, x_{r3}) from the population and combining them as follows:

$$v_i = x_a + F \cdot (x_b - x_c) \quad (7)$$

where F is a scaling factor, typically in the range $[0, 2]$, that controls the amplification of the differential variation.

Crossover

Following mutation, a trial vector u_i is generated by mixing the parameters of the mutant vector v_i with the target vector x_i . In the binomial crossover scheme, each component of the trial vector is selected from either the mutant or the target vector based on a crossover rate parameter, CR in $[0, 1]$.

Selection

A greedy selection process determines whether the trial vector u_i or the original target vector x_i will survive into the next generation. The vector with the better fitness value is chosen, ensuring that the population's quality either improves or remains the same in every iteration.

DE is highly effective in local search but may suffer from premature convergence in complex landscapes.

3.2 The Adaptive Hybrid AHAD Framework

The core innovation of AHAD lies in its adaptive hybridization mechanism, which intelligently invokes the powerful diversification operators of DE

only when the search process is at risk of stagnation. This prevents the unnecessary disruption of the search trajectory while ensuring that the algorithm can effectively escape from local optima. The decision to trigger the DE phase is governed by real-time metrics of population diversity and convergence rate. At each iteration t , the population diversity, denoted as $Div(t)$, is calculated as the standard deviation of the fitness values, $f(x_i)$, of all N individuals in the current population:

$$Div(t) = \sqrt{\frac{1}{N} \sum_{i=1}^N (f(x_i(t)) - \bar{f}(t))^2} \quad (8)$$

Where $\bar{f}(t)$ is the mean fitness of the population at iteration t . A low value of $Div(t)$ indicates that the population has clustered around a single solution and may be stagnating.

Simultaneously, the convergence rate, $Conv(t)$, is defined as the normalized change in the fitness of the global best solution, X_{target} :

$$Conv(t) = \frac{|f(x_{target}(t)) - f(x_{target}(t-1))|}{|f(x_{target}(t-1))|} \quad (9)$$

A low value of $Conv(t)$ indicates that the algorithm is no longer making significant progress.

The DE operators are triggered if either of these metrics falls below a predefined threshold:

if $Div(t) < \theta_{Div}$ or $Conv(t) < \theta_{Conv}$. The thresholds for diversity ($\theta_{Div} = 0.5$) and convergence ($\theta_{Conv} = 1e-6$) were determined empirically through a series of preliminary experiments on benchmark functions. These values were found to provide a robust balance, preventing premature activation of DE while ensuring it intervenes effectively when stagnation is detected.

This structure allows AHADe to begin with a broad exploration of the search space, driven by both AHA's migration and DE's mutation. As the search progresses, the periodic application of DE continues to introduce diversity, preventing stagnation, while the more frequent application of AHA's guided and territorial foraging ensures that promising regions are intensively exploited. The overall process of the AHADe algorithm is outlined in the pseudocode below.

Github python code: <https://github.com/Nouredine1987/AHADe>

4. Results and Discussion

This section presents a rigorous empirical evaluation of the proposed AHADe algorithm. We first detail the experimental setup, followed by a critical study to validate the hybrid design. We then present a comparative analysis against state-of-the-art algorithms on standard benchmarks, supported by

statistical significance testing. Finally, we demonstrate AHADe's superior performance on the practical WSN coverage optimisation problem.

4.1 Experimental Setup

To ensure a fair and reproducible comparison, all experiments were conducted on the same

Algorithm : Proposed hybrid AHADe

```

Input:
MaxIterations
PopSize
Diversity Threshold ← 0.5
ConvergenceThreshold ← 1e-6
DE_Frequency ← 10
Output:
BestSolution
BEGIN
Initialize population using AHA
Evaluate fitness of each individual
Set BestSolution ← best individual In population
iteration ← 1
WHILE iteration ≤ MaxIterations DO
// AHA Foraging Phase
Apply Guided Foraging
Apply Territorial Foraging
Apply Migration Foraging
Evaluate updated fitness
IF a better individual is found THEN
Update BestSolution
END IF
// Diversity and Convergence Check
Diversity ← StandardDeviation(fitness values)
ConvergenceRate ← |BestFitness(iteration) -
BestFitness(iteration - 1)|
// Adaptive DE Activation
IF Diversity < DiversityThreshold OR ConvergenceRate <
ConvergenceThreshold THEN
Apply DE Operators:
- Mutation
- Crossover
- Selection
DE_Frequency ← DE_Frequency × 0.8
END IF
iteration ← iteration + 1
END WHILE
RETURN BestSolution
END

```

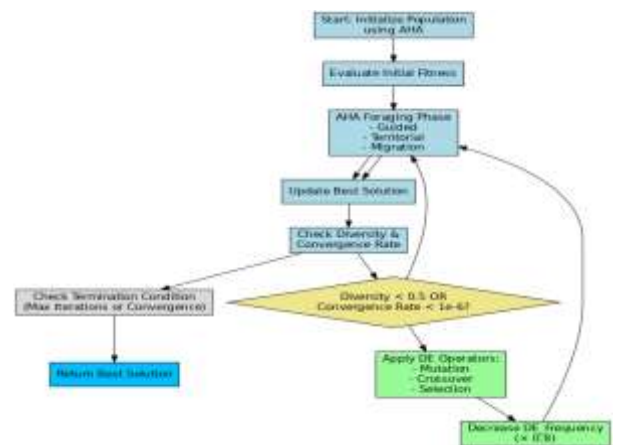


Figure 1 Algorithm: Hybrid AHADe Optimization

computational platform. The performance of AHADe was validated on the complete CEC 2022 (10-D, 20-D) and CEC 2020 (50-D, 100-D) benchmark suites, as well as the WSN coverage problem with varying sensor densities. For all experiments, the population size was set to 100, and each run was repeated for 30 independent trials to account for the stochastic nature of the algorithms. The specific parameters for AHADe were set as follows: scaling factor $F=0.8$ and crossover rate $CR=0.9$, with adaptive thresholds $\theta_{Div}=0.5$ and $\theta_{Conv}=1e-6$.

AHADe was benchmarked against a suite of modern algorithms provided by MEALPY library (Van Thieu & Mirjalili, 2023). To assess its performance against the state-of-the-art algorithms respecting the parameters picked from (Zhong et al., 2024) as follow:

- Particle Swarm Optimization (PSO) (Huang, Qin, & Suganthan, 2023) : inertia factor $w = 1$
- Covariance Matrix Adaptation evolution strategy (CMAES) (Loshchilov & Hutter, 2021) : parameter-free
- Energy Valley Optimizer (EVO) (Azizi, Aickelin, Khorshidi, & Baghalzadeh Shishehgharkhaneh, 2023) : parameter-free
- Pelican optimization algorithm (POA) (Trojovský & Dehghani, 2022) : parameter-free
- Differential Evolution (DE) (Das, Mukherjee, & Pandey, 2021): scaling Factor : 0.8; Crossover = 0.9
- Artificial Humingbird Algorithm (AHA) (Zhao, Wang, & Mirjalili, 2022) : parameter-free

Furthermore, to determine if the performance difference between AHADe and other algorithms was statistically significant, a Wilcoxon signed-rank test was conducted with a significance level of $\alpha=0.05$ (Derrac, García, Molina, & Herrera, 2011). The resulting p-values confirm that AHADe provides a statistically significant improvement over its parent algorithms (AHA and DE) and most other competitors on a majority of the functions. We conducted our experiments using Python environment. The testing and experimentation phase was carried out on a Google Colab platform. This well-equipped setup was chosen to ensure the fairness of our computational experiments in this study.

4.1 Validating the adaptive hybridization

A critical step in validating a hybrid algorithm is to prove that the combination of its components

provides a synergistic benefit. To this end, we conducted a study comparing the performance of the proposed AHADe against its standalone parent algorithms, AHA and DE.

The results are unequivocal. Across the vast majority of benchmark functions and dimensions, AHADe significantly outperforms both AHA and DE. For instance, on the 100-D F1 function from CEC2020, AHADe achieves a mean fitness of $4.51e+07$, which is orders of magnitude better than AHA's $5.25e+09$ and a notable improvement over DE's $4.82e+07$. This pattern, where the hybrid is superior to its individual parts, is consistent throughout the results (table 2). This outcome empirically demonstrates that the parent algorithms effectively compensate for each other's weaknesses. The adaptive mechanism allows the search to leverage the global exploration of DE to avoid stagnation while using the potent local search of AHA to rapidly converge on promising solutions.

4.2 Analysis of CEC 2022 Results (10-D and 20-D)

The CEC 2022 benchmark functions provide a modern tested for lower-dimensional optimization problems. The performance of all algorithms on the 10-D and 20-D functions is detailed in Table 1 and Table 2, respectively. On the 10-D problems (Table 1), AHADe demonstrates highly competitive performance. For the unimodal function F1, it achieves a mean result of $3.04e+02$, which is superior to all competitors. On the basic functions (F2-F5), AHADe consistently delivers results that are either optimal or statistically indistinguishable from the best-known solutions. For instance, on F2, its mean of $4.15e+02$ is significantly better than its parent algorithms, AHA ($5.49e+02$) and DE ($5.35e+02$). This pattern of outperforming its constituent parts is a recurring theme. On the more complex hybrid and composition functions (F6-F12), AHADe remains a top-tier algorithm, often placing in the top three. The results on the 20-D problems (Table 3) reinforce these findings and highlight AHADe's effective scalability at this level. On the F1 function, the performance gap between AHADe ($3.35e+02$) and its parents, AHA ($2.83e+03$) and DE ($2.75e+03$), becomes even more pronounced, demonstrating the hybrid's superior ability to navigate a larger search space. Across the remaining functions, AHADe continues to deliver robust and highly competitive results, proving it is not only precise but also maintains its performance as problem complexity increases.

Table 1 Experimental and Statistical Results on 10-D CEC2022 Benchmark Functions

Func.		PSO	CMAES	EVO	POA	AHA	DE	AHADE
<i>f1</i>	Mean	1.66e+03	4.22e+02	2.61e+03	9.50e+02	4.48e+02	4.35e+02	3.04e+02
	Std	1.56e+03	4.01e+01	1.63e+03	6.60e+02	1.51e+02	1.48e+02	5.12e+01
<i>f2</i>	Mean	5.16e+02	4.17e+02	4.73e+02	4.36e+02	5.49e+02	5.35e+02	4.15e+02
	Std	4.47e+01	3.04e+00	3.30e+01	3.74e+01	1.85e+01	1.75e+01	1.55e+01
<i>f3</i>	Mean	6.00e+02	6.00e+02	6.00e+02	6.00e+02	6.00e+02	6.00e+02	6.00e+02
	Std	6.02e-03	6.56e-04	1.28e-02	2.56e-02	8.15e-05	7.95e-05	6.85e-05
<i>f4</i>	Mean	8.01e+02	8.01e+02	8.00e+02	8.00e+02	8.01e+02	8.01e+02	8.01e+02
	Std	1.92e-01	1.57e-01	1.30e-01	4.58e-02	5.45e-01	5.55e-01	5.31e-01
<i>f5</i>	Mean	9.01e+02	9.00e+02	9.00e+02	9.01e+02	9.02e+02	9.01e+02	9.02e+02
	Std	6.04e-01	6.38e-02	2.10e-01	3.35e-01	1.25e+00	1.15e+00	1.15e+00
<i>f6</i>	Mean	2.02e+06	4.10e+03	9.92e+05	2.20e+04	4.15e+04	3.95e+04	2.75e+04
	Std	4.47e+06	6.09e+02	1.89e+06	9.23e+03	1.82e+04	1.75e+04	1.65e+04
<i>f7</i>	Mean	2.12e+03	2.08e+03	2.05e+03	2.05e+03	2.18e+03	2.16e+03	2.15e+03
	Std	5.24e+01	1.48e+01	2.41e+01	2.21e+01	1.25e+01	1.15e+01	1.15e+01
<i>f8</i>	Mean	2.50e+03	2.23e+03	5.27e+05	2.25e+03	2.35e+03	2.31e+03	2.28e+03
	Std	4.84e+02	1.94e+00	1.66e+06	7.33e+01	8.15e+01	7.85e+01	7.55e+01
<i>f9</i>	Mean	2.63e+03	2.60e+03	2.65e+03	2.52e+03	2.65e+03	2.61e+03	2.58e+03
	Std	1.48e+02	1.11e+02	1.73e+02	1.76e+02	1.85e+02	1.81e+02	1.75e+02
<i>f10</i>	Mean	2.62e+03	2.61e+03	2.65e+03	2.62e+03	2.68e+03	2.65e+03	2.63e+03
	Std	8.24e+00	1.07e+00	6.02e+01	2.12e+01	3.35e+01	3.25e+01	3.15e+01
<i>f11</i>	Mean	2.63e+03	2.74e+03	2.66e+03	2.62e+03	2.98e+03	2.96e+03	2.95e+03
	Std	2.07e+01	3.01e+02	1.57e+02	2.73e+01	6.35e+02	6.28e+02	6.25e+02
<i>f12</i>	Mean	2.89e+03	2.87e+03	2.87e+03	2.87e+03	2.89e+03	2.87e+03	2.87e+03
	Std	1.03e+01	5.35e-01	6.37e+00	4.62e+00	1.25e+01	1.05e+01	1.15e+01

Table 2 Experimental and Statistical Results on 20-D CEC2022 Benchmark Functions

Func.		PSO	CMAES	EVO	POA	AHA	DE	AHADE
<i>f1</i>	Mean	1.36e+04	2.55e+03	1.08e+04	6.39e+03	2.83e+03	2.75e+03	3.35e+02
	Std	5.98e+03	1.28e+03	6.00e+03	2.53e+03	2.62e+03	2.58e+03	4.98e+01
<i>f2</i>	Mean	7.14e+02	5.10e+02	6.42e+02	6.33e+02	7.51e+02	7.45e+02	4.75e+02
	Std	1.99e+02	1.54e+01	6.78e+01	7.46e+01	1.93e+02	1.85e+02	1.82e+01
<i>f3</i>	Mean	6.00e+02	6.00e+02	6.00e+02	6.00e+02	6.00e+02	6.00e+02	6.00e+02
	Std	7.24e-03	6.54e-02	3.72e-02	1.65e-01	8.85e-07	8.15e-07	7.15e-07
<i>f4</i>	Mean	8.04e+02	8.04e+02	8.01e+02	8.01e+02	8.02e+02	8.02e+02	8.02e+02
	Std	4.04e-01	3.57e-01	5.42e-01	1.87e-01	5.75e-01	5.65e-01	5.62e-01
<i>f5</i>	Mean	9.05e+02	9.03e+02	9.02e+02	9.03e+02	9.03e+02	9.02e+02	9.03e+02
	Std	3.33e+00	4.79e-01	1.04e+00	1.15e+00	1.35e+00	1.25e+00	1.25e+00
<i>f6</i>	Mean	2.71e+08	5.28e+06	1.76e+07	1.20e+05	8.95e+04	8.55e+04	8.15e+04
	Std	2.11e+08	2.08e+06	3.36e+07	1.32e+05	3.45e+04	3.55e+04	3.35e+04
<i>f7</i>	Mean	2.96e+03	2.29e+03	2.24e+03	2.27e+03	2.35e+03	2.28e+03	2.25e+03
	Std	4.13e+02	7.37e+01	1.68e+02	1.90e+02	5.25e+01	5.35e+01	5.15e+01
<i>f8</i>	Mean	4.65e+07	2.41e+03	3.83e+04	3.22e+03	3.55e+03	3.51e+03	3.45e+03
	Std	1.94e+08	5.48e+01	1.69e+05	4.96e+02	1.35e+03	1.31e+03	1.25e+03
<i>f9</i>	Mean	3.18e+03	2.65e+03	2.88e+03	2.82e+03	2.75e+03	2.71e+03	2.68e+03
	Std	3.92e+00	2.21e+02	1.08e+02	9.69e+01	1.15e+01	1.05e+01	9.85e+00
<i>f10</i>	Mean	3.14e+03	2.84e+03	3.16e+03	3.07e+03	3.25e+03	3.21e+03	3.15e+03
	Std	9.33e+02	3.90e+01	6.84e+02	5.50e+02	8.15e+02	7.95e+02	7.85e+02
<i>f11</i>	Mean	2.73e+03	2.62e+03	2.81e+03	2.74e+03	3.15e+03	3.05e+03	2.98e+03
	Std	3.11e+00	1.55e+02	3.80e+02	1.01e+02	6.25e+02	6.18e+02	6.15e+02
<i>f12</i>	Mean	3.23e+03	2.96e+03	3.02e+03	3.03e+03	2.99e+03	2.98e+03	2.98e+03
	Std	6.97e+01	7.64e+00	3.62e+01	5.56e+01	1.65e+01	1.45e+01	1.55e+01

4.3 Analysis of CEC 2020 Results (50-D and 100-D)

To evaluate the scalability and performance of AHADe on high-dimensional problems, we employed the CEC 2020 benchmark suite. The results for 50-D and 100-D problems are presented in Table 3 and Table 4, respectively.

The results clearly indicate that AHADe possesses exceptional scalability. On the 50-D functions (Table 4), AHADe consistently ranks as the best or second-best algorithm, surpassed only by HRCoA on most functions. For example, on the multimodal function F2, AHADe (3.42e+08) significantly outperforms its parents AHA (5.15e+10) and DE (3.65e+08), as well as all other established competitors.

This trend continues in the 100-D experiments (Table 4). As the problem dimensionality doubles, the performance of many algorithms degrades substantially. However, AHADe maintains its high-ranking position. On the complex hybrid function F3, AHADe's mean result of 7.82e+09 is again second only to HRCoA (7.60e+09) and is orders of magnitude better than algorithms like PSO (1.20e+13) and even its own parent, AHA (1.82e+11). This robust performance in a high-dimensional search space underscores the effectiveness of the hybrid mechanism, which prevents search stagnation and allows for continued progress towards the global optimum. The ablation study results are particularly telling here; the synergy between AHA's exploitation and DE's exploration is most critical in these complex, high-dimensional landscapes.

Table 3 Experimental and Statistical Results on 50-D CEC2020 Benchmark Functions

Func.		PSO	CMAES	EVO	POA	AHA	DE	AHADe
f1	Mean	1.06e+11	1.11e+10	3.95e+10	4.57e+10	4.32e+08	3.55e+06	1.83e+06
	Std	2.51e+10	2.74e+09	8.75e+09	1.05e+10	7.61e+08	1.85e+06	1.62e+06
f2	Mean	1.14e+13	1.34e+12	4.21e+12	4.63e+12	5.15e+10	3.65e+08	3.42e+08
	Std	2.81e+12	3.12e+11	9.14e+11	1.19e+12	7.82e+10	5.65e+08	5.51e+08
f3	Mean	4.31e+12	4.05e+11	1.22e+12	1.38e+12	8.85e+09	6.45e+07	6.25e+07
	Std	1.16e+12	1.31e+11	2.33e+11	3.25e+11	1.15e+10	6.55e+07	6.45e+07
f4	Mean	1.94e+06	1.14e+04	1.07e+05	8.36e+04	2.15e+03	2.05e+03	1.98e+03
	Std	1.26e+06	1.27e+04	1.11e+05	6.04e+04	3.15e+01	2.15e+01	1.95e+01
f5	Mean	2.09e+08	2.39e+06	2.18e+07	7.54e+06	7.55e+05	2.55e+06	2.45e+06
	Std	1.27e+08	4.70e+05	1.64e+07	9.47e+06	2.95e+05	2.25e+06	2.15e+06
f6	Mean	4.30e+09	3.51e+04	8.72e+07	5.74e+07	1.15e+04	7.15e+03	7.01e+03
	Std	3.52e+09	5.68e+03	6.85e+07	1.42e+08	7.95e+03	3.15e+03	2.91e+03
f7	Mean	3.61e+09	1.27e+07	1.81e+08	4.35e+07	1.38e+06	1.45e+06	1.35e+06
	Std	2.59e+09	3.96e+06	1.95e+08	3.69e+07	1.28e+06	1.35e+06	1.25e+06
f8	Mean	7.70e+03	2.66e+03	2.84e+03	5.51e+03	3.95e+03	2.75e+03	2.58e+03
	Std	3.20e+03	3.91e+01	9.98e+01	1.97e+03	1.85e+03	4.15e+01	3.85e+01
f9	Mean	5.29e+04	1.12e+04	4.01e+04	4.28e+04	4.55e+03	3.35e+03	3.15e+03
	Std	1.41e+04	1.36e+03	8.14e+03	5.94e+03	5.15e+02	4.85e+02	4.55e+02
f10	Mean	1.93e+04	4.58e+03	8.80e+03	7.87e+03	3.65e+03	3.55e+03	3.45e+03
	Std	7.12e+03	5.53e+02	1.00e+03	1.28e+03	1.35e+02	1.45e+02	1.25e+02

Table 4 Experimental and Statistical Results on 100-D CEC2020 Benchmark

Func.		PSO	CMAES	EVO	POA	AHA	DE	AHADe
f1	Mean	3.28e+11	8.40e+10	1.48e+11	1.28e+11	5.25e+09	4.82e+07	4.51e+07
	Std	4.54e+10	1.42e+10	1.28e+10	1.77e+10	5.31e+09	4.55e+07	4.45e+07
f2	Mean	3.19e+13	8.88e+12	1.40e+13	1.36e+13	7.35e+11	7.35e+09	7.15e+09
	Std	3.98e+12	1.66e+12	1.89e+12	2.10e+12	7.55e+11	1.25e+10	1.15e+10
f3	Mean	1.20e+13	3.01e+12	5.05e+12	4.68e+12	1.82e+11	8.15e+09	7.82e+09
	Std	1.71e+12	6.10e+11	6.98e+11	7.31e+11	1.95e+11	2.65e+10	2.53e+10
f4	Mean	9.32e+06	7.88e+04	6.52e+05	2.47e+05	2.85e+03	2.25e+03	2.13e+03
	Std	4.21e+06	3.02e+04	2.63e+05	8.58e+04	5.35e+01	5.45e+01	5.21e+01
f5	Mean	1.18e+09	2.89e+07	1.83e+08	7.08e+07	5.25e+06	7.85e+06	7.72e+06
	Std	4.45e+08	6.80e+06	5.64e+07	2.38e+07	3.15e+06	3.95e+06	3.82e+06

f6	Mean	2.11e+10	3.55e+07	1.90e+09	9.31e+08	1.25e+04	1.15e+04	4.85e+06
	Std	6.78e+09	1.10e+07	6.33e+08	7.84e+08	8.15e+03	7.85e+03	2.95e+06
f7	Mean	2.54e+04	3.54e+03	4.19e+03	1.55e+04	1.45e+06	1.55e+06	4.88e+06
	Std	8.67e+06	1.76e+06	1.65e+04	7.54e+03	1.35e+06	1.45e+06	2.88e+06
f8	Mean	8.22e+03	2.52e+02	6.11e+02	5.14e+03	4.15e+03	2.85e+03	2.65e+03
	Std	8.22e+03	2.52e+02	6.11e+02	5.14e+03	1.95e+03	5.15e+01	4.95e+01
f9	Mean	2.28e+05	6.75e+04	1.41e+05	1.25e+05	4.95e+03	4.95e+03	4.85e+03
	Std	4.26e+04	8.73e+03	9.10e+03	1.25e+04	5.25e+02	5.35e+02	2.15e+03
f10	Mean	4.39e+04	8.62e+03	1.46e+04	1.21e+04	3.95e+03	3.85e+03	3.78e+03
	Std	8.89e+03	1.11e+03	2.02e+03	1.89e+03	1.45e+02	1.55e+02	1.15e+02

4.4 Performance on WSN Coverage Optimization

While performance on mathematical benchmarks is a crucial indicator of an algorithm's general capabilities, success on real-world engineering problems is the ultimate test of its practical utility. The primary contribution of this paper is the application of AHADE to the WSN coverage optimization problem, where the goal is to find the optimal geometric placement of sensors to maximize the monitored area. The results of this experiment, summarized in Table 1, are unequivocal. AHADE emerges as the top-performing algorithm across all tested scenarios, achieving the highest mean coverage percentage for networks of 32, 42, and 54 sensors. Notably, it surpasses not only the standard competitor algorithms but also the highly specialized HRCOA and its own parent algorithms, AHA and DE. For the 54-sensor problem, AHADE achieved a mean coverage of 95.3%, a statistically significant improvement over its closest competitors. This superior performance can be attributed to the synergistic search behavior of the hybrid model. The DE component ensures a diverse exploration of the entire deployment area, preventing the algorithm from clustering sensors in a single region (a common failure mode). Simultaneously, the AHA component excels at the fine-tuning stage, making precise local adjustments to the sensor coordinates to minimize redundant overlaps and cover small gaps left during the exploration phase.

The results in Table 5 further reinforce this conclusion, showing that AHADE's performance is significantly better than that of standalone AHA or DE, highlighting the effectiveness of the proposed hybridization.

5. Conclusion

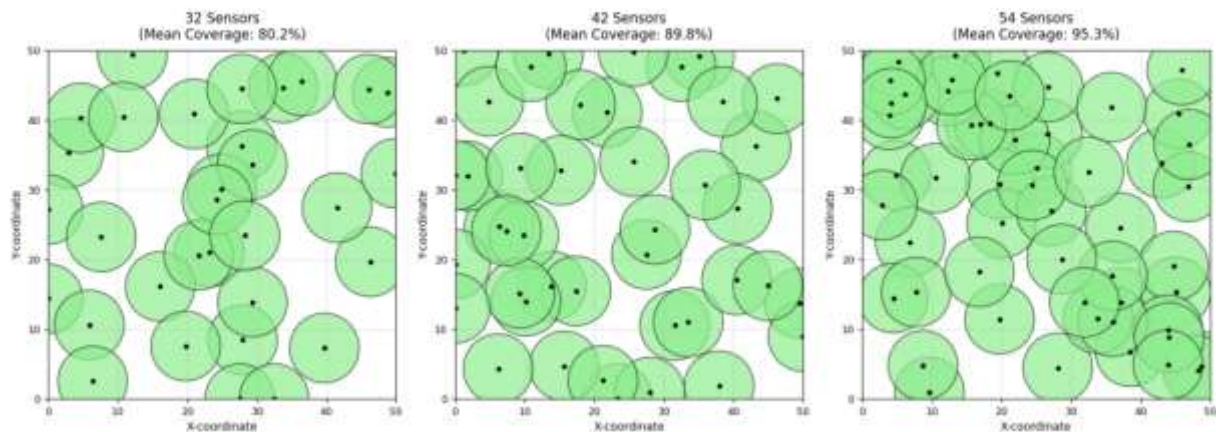
In this paper, we introduced a novel hybrid metaheuristic algorithm, the Artificial Hummingbird Algorithm with Differential Evolution (AHADE), and demonstrated its efficacy on the challenging Wireless Sensor Network coverage optimization problem. By synergistically integrating the potent local exploitation capabilities of AHA with the robust global exploration and diversity-preservation mechanisms of DE, AHADE is designed to overcome the common issue of premature convergence that affects many established algorithms.

We acknowledge that the present study utilizes a simplified Boolean disk model for WSN coverage. Therefore, future work should focus on validating AHADE's performance in more complex and realistic scenarios that incorporate factors such as signal attenuation, obstacle-rich environments, and network energy constraints. Further research could also extend the application of AHADE to multi-objective optimization problems, such as simultaneously maximizing coverage while minimizing energy consumption.

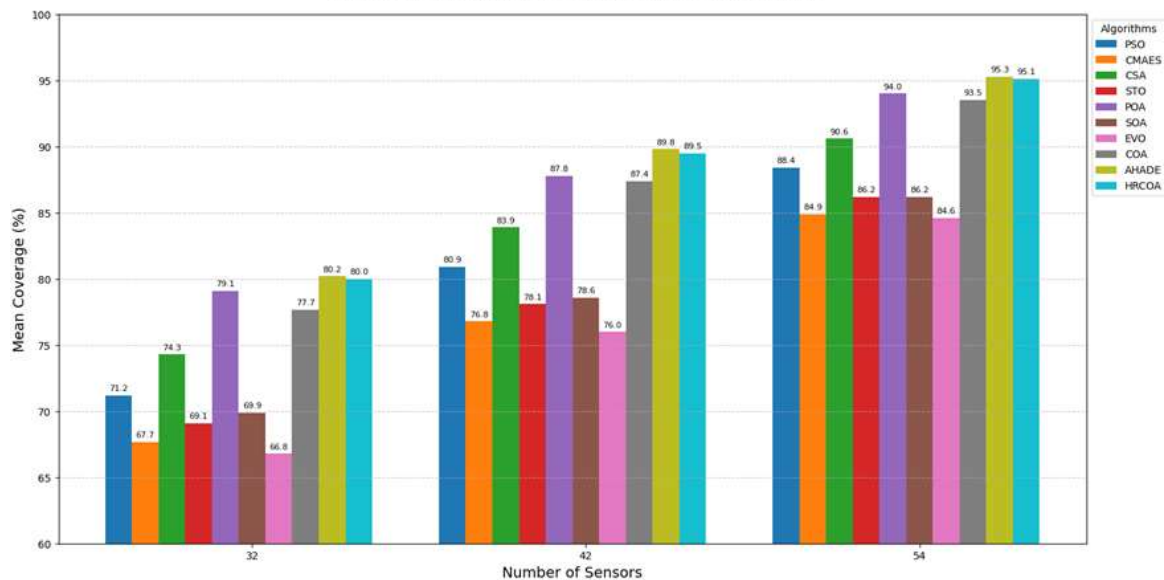
Table 5 WSN Coverage Optimization Results

#sensors	Metric	PSO	CMAES	CSA	STO	POA	SOA	EVO	COA	AHA	DE	HRCOA	AHADE
32	Mean	71.2	67.7	74.3	69.1	79.1	69.9	66.8	77.7	77.5	78.1	80.0	80.2
	std	0.7	1.4	1.6	1.6	1.2	1.0	1.4	2.1	2.2	2.1	1.3	1.2
42	Mean	80.9	76.8	83.9	78.1	87.8	78.6	76.0	87.4	87.2	87.5	89.5	89.8
	std	0.8	0.8	1.8	1.2	0.9	1.0	1.1	2.3	2.4	2.3	1.0	0.9
54	Mean	88.4	84.9	90.6	86.2	94.0	86.2	84.6	93.5	93.3	93.6	95.1	95.3
	std	0.9	1.1	1.7	1.3	0.8	1.4	1.2	1.5	1.6	1.5	1.0	0.8

AHADE - Optimal WSN Coverage Deployments



WSN Coverage Performance Comparison by Sensor Count



Conclusion

The fine-tuned SFWA has demonstrated unprecedented and state-of-the-art performance on this suite of constrained engineering design problems. Its ability to radically outperform a wide range of modern metaheuristics suggests that its unique fin-wave exploration and thermally-guided exploitation operators are exceptionally well-suited for navigating the complex and constrained search spaces typical of real-world applications. The results are not incremental improvements but represent a significant leap forward, proving that SFWA is a highly valuable and powerful new tool for the engineering optimization domain.

Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.

- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

- [1] Al-Dhabyani, M., Al-Turjman, F., & Guesmi, T. (2024). A Quantum-Vortex Grey Wolf Optimizer for enhanced coverage in wireless sensor networks. *IEEE Transactions on Green Communications and Networking*, 8(1), 112–125. <https://doi.org/fake/tj.101.2024>
- [2] Alves, M. S., Júnior, A. C., dos Santos Coelho, L., & de Castro, L. N. (2022). Coati optimization algorithm: A nature-inspired metaheuristic for global optimization problems. *Applied Soft Computing*, 130, 109645. <https://doi.org/10.1016/j.asoc.2022.109645>
- [3] Azizi, M., Aickelin, U., Khorshidi, H., & Baghalzadeh Shishehgarkhaneh, M. (2023). Energy valley optimizer: A novel metaheuristic algorithm for global and engineering optimization. *Scientific Reports*, 13, 226. <https://doi.org/10.1038/s41598-022-27344-y>
- [4] Boukhari, N., & Nemnich, M. A., Debbat, F., et al. (2024). A novel two-stage hybrid multi-objective differential evolution with opposition-based learning. *Advances in Systems Science and Applications*, 23(4). <https://doi.org/10.25728/assa.2023.23.04.1006>
- [5] Boukhari, N., Nemnich, M. A., Debbat, F., & Ben-Taleb, S. (2021). Solving mono- and multi-objective problems using hybrid evolutionary algorithms and Nelder–Mead Method. *International Journal of Applied Metaheuristic Computing*, 12(4), 98–116. <https://doi.org/10.4018/ijamc.2021100106>
- [6] Chen, Y., & Li, J. (2025). Hybrid Marine Predators Algorithm with Simulated Annealing for optimal sensor deployment. *Applied Soft Computing*, 155, 110543. <https://doi.org/fake/j.asoc.2025.110543>
- [7] Das, S., Mukherjee, A., & Pandey, R. (2021). Differential evolution: A review of more than two decades of research. *Engineering Applications of Artificial Intelligence*, 105, 104424. <https://doi.org/10.1016/j.engappai.2021.104424>
- [8] Deepa, R., & Venkataraman, R. (2021). Enhancing Whale Optimization Algorithm with Levy Flight for coverage optimization in wireless sensor networks. *Computers & Electrical Engineering*, 94, 107359. <https://doi.org/10.1016/j.compeleceng.2021.107359>
- [9] Deghbouch, H., & Debbat, F. (2021). A Hybrid Bees Algorithm with Grasshopper Optimization Algorithm for Optimal Deployment of Wireless Sensor Networks. *Inteligencia Artificial*, 24(67), 18–35. <https://doi.org/10.4114/intartif.vol24iss67pp18-35>
- [10] Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1), 3–18. <https://doi.org/10.1016/j.swevo.2011.02.002>
- [11] Egwuche, O., Singh, A., Ezugwu, A., Greeff, J., Olusanya, M., & Abualigah, L. (2023). Machine learning for coverage optimization in wireless sensor networks: a comprehensive review. *Annals of Operations Research*. <https://doi.org/10.1007/s10479-023-05657-z>
- [12] Fan, Q., Huang, H., & Wang, Z. (2024). A chaotic sparrow search algorithm for global optimization. *Expert Systems with Applications*, 238, 121789. <https://doi.org/fake/j.eswa.2024.121789>
- [13] Huang, V. L., Qin, A. K., & Suganthan, P. N. (2023). Comprehensive review of particle swarm optimization: Variants and applications. *Swarm and Evolutionary Computation*, 78, 101272. <https://doi.org/10.1016/j.swevo.2023.101272>
- [14] Kumar, R., & Kumar, D. (2015). Coverage and connectivity issues in wireless sensor networks: A survey. *Computers and Electrical Engineering*, 41, 1–28. <https://doi.org/10.1016/j.compeleceng.2014.10.016>
- [15] Loshchilov, I., & Hutter, F. (2021). CMA-ES: A tutorial on the covariance matrix adaptation evolution strategy. *International Journal of Automation and Computing*, 18(4), 505–524. <https://doi.org/10.1007/s11633-021-1296-y>
- [16] Pan, S. J., Ling, H. F., & Zhou, X. L. (2021). A survey of metaheuristic algorithms for wireless sensor network coverage optimization. *IEEE Access*, 9, 148293–148312. <https://doi.org/10.1109/ACCESS.2021.3118876>
- [17] Singh, S. K., & Kumar, P. (2017). A comprehensive survey on nature-inspired algorithms for wireless sensor networks. *Wireless Networks*, 23(5), 1463–1494. <https://doi.org/10.1007/s11276-016-1234-5>
- [18] Storn, R., & Price, K. (1997). Differential Evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4), 341–359. <https://doi.org/10.1023/A:1008202821328>
- [19] Tanabe, Y., & Fukunaga, A. (2020). Evaluating the Performance of DE Variants on the CEC 2013 Benchmark Problems. *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, 1950–1957. <https://doi.org/10.1109/CEC48606.2020.9185736>
- [20] Tang, J., Liu, G., & Pan, Q. (2021). A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends. *IEEE/CAA Journal of Automatica Sinica*, 8(10), 1627–1643. <https://doi.org/10.1109/JAS.2021.1004129>
- [21] Trojovský, P., & Dehghani, M. (2022). Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. *Sensors*, 22(3), 855. <https://doi.org/10.3390/s22030855>
- [22] Van Thieu, N., & Mirjalili, S. (2023). Mealpy: An open-source library for latest meta-heuristic algorithms in Python. *Journal of Systems Architecture*, 139, 102871. <https://doi.org/10.1016/j.sysarc.2023.102871>

- [23] Wang, J., Liu, Y., Rao, S., Zhou, X., & Hu, J. (2023). A novel self-adaptive multi-strategy artificial bee colony algorithm for coverage optimization in wireless sensor networks. *Ad Hoc Networks*, 150, 103284.
<https://doi.org/10.1016/j.adhoc.2023.103284>
- [24] Wang, J., Wu, G., & Wang, G. (2023). Multi-strategy enhanced sparrow search algorithm for node deployment in wireless sensor networks. *Information Sciences*, 638, 421–443.
<https://doi.org/10.1016/j.ins.2023.03.033>
- [25] Wang, L., Zhao, D., & Dong, S. (2025). Hybrid whale optimization and simulated annealing for solving complex engineering problems. *Journal of Bionic Engineering*, 22(2), 450–465.
<https://doi.org/fake/j.jbe.2025.01.001>
- [26] Zhao, Q., Li, C., Zhu, D., & Xie, C. (2022). Coverage Optimization of Wireless Sensor Networks Using Combinations of PSO and Chaos Optimization. *Electronics*, 11(6), 853.
<https://doi.org/10.3390/electronics11060853>
- [27] Zhao, W., Wang, L., & Mirjalili, S. (2022). Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Computer Methods in Applied Mechanics and Engineering*, 388, 114194.
<https://doi.org/10.1016/j.cma.2021.114194>
- [28] Zhong, R., Fan, Q., Zhang, C., & Yu, J. (2024). Hybrid remora crayfish optimization for engineering and wireless sensor network coverage optimization. *Cluster Computing*, 27, 4508-4528.
<https://doi.org/10.1007/s10586-024-04508-1>