Knowledge-Based Systems

An Optimal Sensor Network Coverage Planning Based on Adaptive Crossover Mutation Differential Evolution

--Manuscript Draft--

Manuscript Number:	KNOSYS-D-23-03751
Article Type:	Full Length Article
Keywords:	Sensor networks; coverage planning; wireless sensor networks; optimization; adaptive crossover mutation differential evolution

Abstract:	Wireless sensor networks (WSNs) are crucial in various domains, including environmental monitoring, surveillance systems, and smart cities. Achieving optimal coverage in WSNs is a fundamental challenge in designing effective event detection and monitoring strategies. This research introduces an adaptive optimization approach called Adaptive Crossover Mutation Differential Evolution (ACMDE) to address the node coverage planning problem in WSNs. The proposed ACMDE method optimizes sensor placement to maximize coverage while minimizing the required sensors. It incorporates enhanced adaptive strategies for crossover, mutation, and reinitialization to improve coverage efficiency. The ACMDE approach is evaluated using test suite functions and the optimal sensor network coverage planning problem. It is compared against existing optimization strategies, utilizing various metrics for validation. Experimental results demonstrate the superior performance of the proposed approach in maximizing node coverage in WSNs compared to competing algorithms. The ACMDE approach presents a novel and effective solution for optimizing coverage in WSNs, with potential applications in practical scenarios.

An Optimal Sensor Network Coverage Planning Based on Adaptive Crossover Mutation Differential Evolution

Thi-Kien Dao [1,2,3], Trong-The Nguyen [1,2,3*], and Zhenyu Meng[1,2]

[1] Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350118, China
[2] School of Computer Science and Mathematics, Fujian University of Technology, Fuzhou 350118, China

[3] University of Information Technology, Ho Chi Minh City, Vietnam

[4] Vietnam National University, Ho Chi Minh City 700000, Vietnam

1101405123@nkust.edu.tw, thent@uit.edu.vn, 1100405110@nkust.edu.tw

Abstract:

Wireless sensor networks (WSNs) play a crucial role in various domains, including environmental monitoring, surveillance systems, and smart cities. Achieving optimal coverage in WSNs is a fundamental challenge in designing effective strategies for event detection and monitoring. This research introduces an adaptive optimization approach called Adaptive Crossover Mutation Differential Evolution (ACMDE) to address the node coverage planning problem in WSNs. The proposed ACMDE method aims to optimize sensor placement to maximize coverage while minimizing the number of required sensors. It incorporates enhanced adaptive strategies for crossover, mutation, and reinitialization to improve coverage efficiency. The ACMDE approach is evaluated using test suite functions and the optimal sensor network coverage planning problem. It is compared against existing optimization strategies, utilizing various metrics for validation. Experimental results demonstrate the superior performance of the proposed approach in maximizing node coverage in WSNs compared to competing algorithms. The ACMDE approach presents a novel and effective solution for optimizing coverage in WSNs, with potential applications in practical scenarios.

Keywords: Sensor networks, coverage planning, wireless sensor networks, optimization, adaptive crossover mutation differential evolution, node coverage, heuristic-based strategies.

1. Introduction

Wireless sensor networks (WSNs) have become a critical technology in various domains, including environmental monitoring, security systems, and smart cities [1][2]. These networks consist of autonomous sensor nodes that collaborate to collect and transmit data from their surroundings [3]. Achieving optimal coverage in WSNs is a key challenge in designing effective sensor networks, as it ensures the detection and monitoring of events or phenomena [4]. Strategic placement of sensor nodes is crucial to maximize coverage while minimizing the number of sensors required [5]. This task is essential for efficient utilization of network resources such as electricity, communication bandwidth, and processing power [6].

Effective coverage planning not only enhances network longevity but also enables prompt and accurate event detection while conserving resources. Probabilistic models calculate coverage based on sensor node distribution and sensing range [5] [6]. Metheuristic-based approaches utilize problem-specific heuristics to guide sensor placement. Coverage planning in sensor networks has been approached through various methods, including heuristic-based tactics, probabilistic models, and optimization algorithms. Recently, metaheuristic algorithms have been rapidly evolving and advancing over the past few decades. These algorithms have undergone significant improvements to enhance their efficiency, effectiveness, and applicability across various domains. Some notable metaheuristic algorithms that have gained prominence due to their fast development include: Genetic algorithms (GA) [7][8], Particle swarms optimization (PSO) [9], Ants lion optimizer (ALO) [10], Grey wolf optimizer (GWO) [11], Moth-flame optimization (MFO) [12], and Differential evolution (DE) [13] employ iterative search procedures to find the best solution.

The DE algorithm encompasses several improved variants developed to enhance performance and address specific challenges [13] [14]. The notably improved types of DE would be more and more effectively developed with the hot topic research of metaheuristic optimization algorithms. Several popular works are mentioned, for example, the Enhanced Differential Evolution (EDE) [15], which adaptive tuned the mutation and crossover rates; EDE enhances the algorithm's ability to explore the search space effectively and balance exploration and exploitation. The adaptive DE with an Optional external archive (JaDE) [16] was dynamically adjusted to the control parameters, and leveraging the archive, JaDE improves the algorithm's convergence speed and solution quality. The Self-Adaptive Differential Evolution (SaDE) [17] employed a stochastic learning strategy to adaptively update the parameter settings based on the information obtained from the previous generations. The adaptive approach improves the algorithm's robustness and effectiveness across optimization problems. The development of these improved DE types showcases the continuous efforts to refine and advance the capabilities of the DE algorithm in solving complex optimization problems [18].

While these methods have made significant contributions, they also have limitations. Optimization algorithms may face challenges like premature convergence or scalability issues in high-dimensional domains [19]. Probabilistic models often make idealized assumptions and may overlook real-world constraints or changing environmental factors [20]. Metaheuristic-based approaches heavily rely on problem-specific heuristics that may not be easily applicable to different scenarios or network configurations [21].

This study proposes an adaptive optimization technique called Adaptive Crossover Mutation Differential Evolution (ACMDE) for planning sensor network coverage in WSNs. The ACMDE method aims to maximize coverage while minimizing the number of sensors required by optimizing sensor placement. It incorporates improved adaptive strategies for crossover, mutation, and reinitialization to enhance coverage maximization. The paper covers the ACMDE algorithm, its application in designing sensor network coverage, and a comprehensive experimental analysis to evaluate its effectiveness. Experimental results demonstrate the ACMDE algorithm's potential for practical implementation in real-world sensor network applications by outperforming other optimization algorithms. The following sections of this paper will discuss related work, the proposed ACMDE approach, experimental research, and results analysis. The paper concludes with suggestions for future research and a discussion on how the ACMDE algorithm can advance sensor network design.

The main contributions of the proposed ACMDE algorithm for optimal sensor network coverage planning are summarized as follows:

- Suggested strategies for adjusting parameters, such as reinitialization of individual locations, mutation, and crossover variable adjustments, to find suitable solutions, overcome local optima drawbacks, and enhance population diversity.
- Developed WSN node coverage planning within the ACMDE approach, allowing for sensor placement optimization to maximize coverage while minimizing the number of sensors required.
- Evaluated the proposed method through testing and comparison with other algorithms found in the literature using test suite functions and WSN coverage planning. The analysis and discussion of experimental results demonstrate significant performance improvements in node coverage optimization with the proposed ACMDE strategies.

The remaining sections of this paper include a literature review in Section 2, which presents the problem statement and establishes the model for node coverage. Section 3 introduces the application strategies and mechanisms to enhance the DE algorithm (ACMDE), implements tests to verify its performance, and analyzes the results. Section 4 describes the ACMDE approach for optimal WSN node coverage planning. Finally, Section 5 provides a summary and conclusion of the study.

2. Related Work

This section provides an overview of the model for sensor network coverage planning with two streams of models—binary and probability over various optimization algorithms utilized in the context. We discuss the strengths and limitations of these methods, which leads us to emphasize the need for an adaptive optimization approach. The differential evolution (DE) algorithm is presented as the review in the subsection.

2.1 Wireless Sensor Network (WSN) Coverage and Connection Model

To address the optimization problem of maximizing coverage efficiency in WSNs, we establish a model that incorporates the concept of maximum coverage planning [22]. Each deployed sensor node within the sensing radius is defined to have the ability to sense and establish communication connections with other nodes within its range. In order to achieve optimal node coverage, it is crucial to understand the WSN coverage and connection model. During initial trials, sensor nodes are positioned arbitrarily to create an isomorphic WSN within their sensing radius, enabling monitoring of the target area. Previous studies have focused on defining the problem and developing models to enhance the efficiency of node coverage within WSNs [23]. The node coverage in WSNs ensures that each sensor node, with a fixed sensing radius, can effectively sense and communicate with other nodes within its range. To optimize WSN coverage, metaheuristic methods are employed to strategically position wireless sensor nodes [24]. Therefore, it is essential to strategically place each node within its limited sensing range to facilitate communication and achieve optimal coverage. The coverage problem involves identifying objects within the nodes' sensing radius to optimize their placement [25].

WSN coverage research can be categorized into two subcategories: area coverage and target coverage [23]. Area coverage studies consider all potential monitoring areas and evaluate the overall performance of the network [22]. On the other hand, target coverage focuses on specific locations within the sensing region. Mathematical fitness functions are constructed using two streams of models—binary and probability—to define network coverage models based on typical applications [1][26]. The binary model determines whether a sensor node completely covers an area or does not cover it at all, without considering any intermediate level of coverage [27]. In this model, the sensing range of a sensor node is represented as a fixed radius or distance [28]. Some studies have proposed energy-efficient target coverage schemes by adapting node sleep scheduling and sensing [29]. Additionally, a deterministic deployment strategy utilizing hexagonal grids has been developed to achieve uniform coverage [30].

On the other hand, the probabilistic model, by taking into account the likelihood that a sensor node will identify targets within its sensing range, the probabilistic model adopts a more realistic approach. It acknowledges that the sensing effectiveness may vary within the range due to factors such as signal attenuation, interference, or environmental conditions. For instance, a proposed coverage algorithm based on the probabilistic sensing model dynamically adjusts sensing ranges [31] [24] [32]. In the probabilistic model, the sensing range of a sensor node is represented as a probability distribution function [33]. This study focuses on the area coverage approach, where sensor nodes are deployed in a WSN using two patterns: random deployment and deterministic deployment. In some WSN applications, when the specifics of the sensing region are known in advance, a deterministic deployment of sensor nodes is preferred [26]. However, in certain scenarios, sensor nodes need to be placed randomly in uncharted or inaccessible locations, resembling battlegrounds [31]. As the effectiveness of a wireless sensor network depends on understanding sensor coverage, numerous recent studies have been conducted [32]. It is crucial for WSNs to consider the location of sensor nodes as it directly affects the network's sensing coverage [34]. Table 1 provides a summary comparing the binary and probability models of WSN node coverage.

Table 1. A summary of the binary and probability models employed in WSNs for node coverage

Models/Types	**Descriptions**	**Advantages**	**Applications**
Binary Model/ Deterministic	Represents WSN node coverage as binary decision variables (0 or 1) indicating presence or absence of coverage.	- Simple representation - Suitable for Boolean optimization problems - Efficient in handling binary constraints	WSN coverage optimization, target detection [26] [27] [28] [29] [30]
Probability Model/ Stochastic	Assigns a probability value (between 0 and 1) to each WSN node, representing the likelihood of coverage.	- Reflects uncertainty in coverage - Allows probabilistic optimization techniques - Flexible representation	Coverage planning, energy optimization, reliability analysis [31] [24] [32] [33] [34]

The WSN coverage planning with efficient and logical network deployment is a challenging task that is an NP-hard problem, as demonstrated by the difficulties encountered when deploying a large number of sensor nodes. The monitoring zone, typically in a two-dimensional space, is divided into *W.L* grids (*W*-width and *L*-length measures). Each grid contains a central monitoring point denoted as "*p*." The coordinates of each point are represented as *(x, y)*, where x and y are integers ranging from 1 to L and 1 to W, respectively. The number of sensor nodes denoted as "*n*" with n=$[n_1,\ n_2, .., \ n_i, .., \ n_s]$, with "s," is considered to be a collection of randomly distributed and isomorphic nodes within the monitoring region. Each sensor node in the set is identified by its two-dimensional coordinates $(x_i,\ y_i\ )$, where i represents the node index. The positions of the sensor nodes in the monitoring region are denoted by P, with P =$[(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), \dots, (x_s\ , y_s)]$ representing the positions of individual sensor nodes. The symbol R_s represents the radius of the sensor nodes. The sensing range of the sensor nodes is determined by the binary model and the probabilistic model, as described earlier. Both models utilize the following formula to calculate the coverage area based on the sensor node's sensing range radius and the Euclidean distance. The efficient and logical deployment of WSNs is an NP-hard problem, as evidenced by problems with large-scale sensor node deployment.

In the monitoring region, which is typically two-dimensional, it is divided into $W{\cdot}L$ grids, with each grid represented by a monitoring point *p* at its center. The coordinates of each point *p* are denoted as *(x, y)*, where *x* and *y* are integers ranging from *1* to *L* (length) and 1 to *W* (width), respectively. The coordinates *(i, j)* represent *t* A set of sensor nodes is assumed isomorphic and randomly deployed in the monitoring area, which is indicated as $=[n_1,\ n_2,..,\ n_i,..,\ n_s]$, while *s* is the number of sensor nodes. The two-dimensional coordinators of any sensor node n_i in the set are denoted as The positions of the sensor nodes in the monitoring area are indicated as *P* =$[(x_1, y_1), (x_2, y_2), \dots, (x_i, y_i), \dots, (x_s, y_s)]$. The radius of sensor nodes is represented as R_s. As mentioned, two models, binary and probabilistic, calculate the sensor nodes' sensing range. One is the binary model, and the other is the probabilistic model. Both of them used the Euclidean distance to calculate the coverage area with the radius of the sensing range of a sensor node as follows.

$$d_{n_i} = \sqrt{(x_{n_i} - x)^2 + (y_{n_i} - y)^2}, \tag{1}$$

Where $(x_{n_i},\ y_{n_i})$ and *(x, y)* denote the coordinator of sensor node n_i and the coordinator of monitoring q point, and measured distance d_{n_i} is the distance between q and n_i.
The binary model, which assumes an environment without interferences or signal attenuation in the WSN, disregards the complexity of the actual network conditions. The widely used deterministic disk-based binary sensing model serves as the foundational approach. In this model, a sensor node S is considered to cover or detect a point or event *q* if it lies within S's sensing range R_s in the network field. The sensing region of S is represented by a disk centered at *S* with a radius equal to Rs. If the distance between S and q, denoted as *d(S, q)*, is less than or equal to *R*, the coverage function *Cr(S, q)* is set to 1. Otherwise, *Cr(S, q)* is set to 0.

$$Cr(S,q) = \begin{cases} 1, & if\ d(S,q) \le R_s \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

In the natural wireless transmission environment, wireless signal strength diminishes as the transmission distance increases, and interferences from other nodes and external noise sources occur. To account for the complexity of the wireless signal transmission environment, this research employs the probabilistic perception model. In this model, the probability that sensor node n_i covers the monitoring point q is represented as follows:

$$P_c(x,y,n_i) = \begin{cases} 1, & if\ d(n_i,q) \le R_s - r_e \\ e^{\left(-\alpha_1 \lambda_2^{\beta_2}\right)\lambda_2^{\beta_2} + \alpha_2} & if\ R_s - r_e < d(n_i,q) < R_s + r_e \\ 0, & if\ d(n_i,q) \ge R_s + r_e \end{cases}, \tag{3}$$

Where λ_1 and λ_2 are the influence coefficient; r_e is the sensor node reliability parameter; the constraint of measurement parameters: $0<r_e<R_s$; α_1, α_2, β_1, and β_2 are related to the characteristics of the sensor node r_e indicates the measurement related to the characteristics of the sensor node; λ_1 and λ_2 are given as following formula:

$$\begin{cases} \lambda_1 = r_e - R_s + d(n_i,q) \\ \lambda_2 = r_e - R_s + d(n_i,q) \end{cases}, \tag{4}$$

Let's assume that a monitoring point p is covered by multiple sensor nodes, denoted as C_r, where C_r belongs to the set C_s. The probability calculation formula for this event is expressed as follows:

$$P_c(C_r)_j = 1 - \prod_{n_i \in C_r} (1 - P_c(n_i, x, y)), \tag{5}$$

where $1 \le j \le (M \cdot N)$. To determine the coverage probability of each monitoring point, the coverage probability of every point is calculated. Subsequently, the coverage rate of the monitoring area is computed using the following formula:

$$R = \frac{\sum_{j=1}^{MxN} P_c(C_r)_j}{M{\cdot}N}, \tag{6}$$

Consider a two-dimensional WSN monitoring region network with an area of $M{\cdot}N$ square units. In this network, each sensor node has a communication radius denoted as r_e and a sensing radius denoted as R_s, both measured in meters, where r_e is greater than or equal to 2 times R_s (with $r_e \ge 2R_s$). The sensor nodes possess communication capabilities, sufficient power supply, and data accessibility. They share the same specifications, structure, and communication capacities. Additionally, the location of a mobile sensor node can be updated rapidly by calculating the coverage ratio for the probability of the target point $P_c(C_r)_j$, which is determined by the probability of the network deployed on the 2D surface of the WSN monitoring area. The two-dimensional location of a grid can be specified by its row (*i*) and column (*j*) within the monitoring area.

2.2 Differential Evolution (DE) Algorithm

The differential evolution (DE) [13] algorithm is a popular stochastic optimization technique that has been widely employed to address global optimization problems in various domains . This subsection provides a comprehensive review of the DE algorithm, highlighting its key concepts, working principles, and its relevance to the research objectives. DE algorithm is inspired by the principles of natural evolution and mimics the processes of mutation and selection observed in biological systems [14]. It operates by iteratively generating a new population of candidate solutions through a combination of mutation and crossover operations. The mutation operation introduces random perturbations to the current solutions, while the crossover operation combines selected solutions to produce new candidate solutions. This iterative process aims to explore the solution space and converge towards optimal or near-optimal solutions. One of the notable strengths of the DE algorithm is its ability to effectively handle global optimization problems, where the objective is to find the best solution across a large solution space. It is known for its robustness and ability to handle complex, multimodal, and non-linear optimization problems. The algorithm's stochastic nature allows it to escape local optima and explore diverse regions of the solution space, increasing the chances of finding better solutions.

The DE algorithm consists of several phases, including initialization, mutation, crossover, and selection. During the initialization phase, the population of candidate solutions is initialized, typically by randomly generating individuals with their respective parameter values. The mutation phase introduces random perturbations to the individuals, creating new potential solutions. The crossover phase combines selected individuals to create offspring, which inherit traits from their parents. Finally, the selection phase determines the best-performing individuals based on their fitness values, ensuring their survival and contribution to the next generation. While the DE algorithm has proven to be a powerful optimization technique, it is not without its limitations. The algorithm's performance heavily depends on the choice of its control parameters, such as population size, mutation strategy, and crossover rate. Poor parameter settings can lead to suboptimal results or slow convergence. Moreover, the computational complexity of the algorithm increases with the dimensionality of the problem, making it less efficient for high-dimensional optimization tasks.

In the context of this research, the DE algorithm is particularly relevant due to its capability to handle global optimization problems and its suitability for the specific objectives of the study. The algorithm's ability to explore and converge towards optimal solutions makes it a promising choice for optimizing the sensor network coverage planning. By leveraging the strengths of the DE algorithm and addressing its limitations through proper parameter tuning and problem-specific adaptations, this research aims to develop an adaptive optimization approach for sensor network coverage planning. At each iteration, the DE algorithm generates a new population of candidate solutions by combining and modifying the current solutions. The best-performing solutions are then selected to form the next generation of the population, based on their fitness values. The DE algorithm consists of several phases in the optimization process, including initialization, mutation, crossover, and selection of candidate solutions.

The **initialization** is a formula that is utilized to initialize the position information of the population. During the initialization phase, the population of candidate solutions is initialized. Each individual in the population represents a potential solution, and their position information determines the candidate solutions' characteristics. The initial positions of population members are randomly generated to ensure an even distribution across the D-dimensional search space, corresponding to the problem space of the optimization task. Typically, a random method is used to distribute the initial population positions, where N represents the population size, and the initial population position distribution is calculated using a formula.

$$x_{\mathrm{i,j}} = x_{\mathrm{i,j}}^{\mathrm{L}} + \mathrm{rand}\ ([0,1])(x_{\mathrm{i,j}}^{\mathrm{L}} - x_{\mathrm{i,j}}^{\mathrm{M}}) \tag{7}$$

The *i*-th individual in a population indicates a potential answer to an optimization issue, and the *j*-th decision variable of that individual is denoted by $x_{i,j}$. The i and j range from *1* to N_P and *1* to D, respectively. Meanwhile, the function rand (0, 1) creates a random integer with a uniform distribution within the range [0,1].
The **mutation** phase introduces random perturbations to the population. It is a crucial process for exploring new regions in the search space and creating new candidate solutions. The mutation operation involves calculating the

differences between two randomly selected individuals from the current population, resulting in new candidate solutions. Common mutation operations include various strategies.

Strategy 1 DE/rand/1/bin

$$V_i^{t+1} = x_{r_1}^t + M * (x_{r2}^t - x_{r_3}^t) \tag{8}$$

Strategy 2 DE/rand/2/bin

$$V_i^{t+1} = x_{r_1}^t + M * (x_2^t - x_{r_3}^t)+M * (x_{r_4}^t - x_{r_5}^t) \tag{9}$$

Strategy 3 DE/best/1/bin

$$V_i^{t+1} = x_{best}^t + M * (x_{r_1}^t - x_{r2}^t) \tag{10}$$

Strategy 4 DE/best/2/bin

$$V_i^{t+1} = x_{best}^t + M_1 * (x_{r_1}^t - x_{r2}^t) + M_2 * (x_{r_3}^t - x_{r_4}^t) \tag{11}$$

Strategy 5 DE/current-to-best/1/bin

$$V_i^{t+1} = x_i^t + M_1 * (x_{best}^t - x_i^t)+M_2 * (x_{r_1}^t - x_{r2}^t) \tag{12}$$

In the formula, $\mathrm{V_i^{t+1}}$ is the experimental individual i in the $t+1$ generation population after mutation, $i \in [1, N]$, and the population size is denoted by n; $x_{r_1}^t$,x_{r2}^t,$x_{r_3}^t$,$x_{r_4}^t$, and $x_{r_5}^t$ are the individuals randomly selected in the T generation population, and r_1,r_2,r_3,r_4 and r_5 represent the identification numbers of different individuals in the same generation population;x_{best}^t denotes the best person in the population's g-th generation. F is the variation probability and the value is between 0 and 1.

The **crossover** phase combines information from multiple candidate solutions to create new solutions with higher optimization potential. The binomial crossover operator is commonly used in the DE algorithm to perform the crossover operation, combining the values or information from different candidate solutions. The crossover operation in DE algorithm is expressed as follows.

$$U_{ij}^{t+1} = \begin{cases} v_{ij}^{t+1}, rand(j) \leq CR \text{ } or \text{ } j = rand(i) \\ x_{ij}^{t+1}, rand(j) > CR \text{ } or \text{ } j \neq rand(i) \end{cases}, \tag{13}$$

In the formula, U_{ij}^{t+1} represents an updated individual obtained by crossing the j gene of the test individual; $\mathrm{rand}(j)$ is a random integer with a homogeneous distribution, with a number ranging from 0 to 1, and j represents the j-th gene; CR is the cross probability, and its value is between 0 and 1 (in experiment we set value of CR in this paper is 0.9); $\mathrm{rand}(i)$ is the generated random integer, i takes the value in $[1, D]$, and d represents the D-dimensional parameter (number of decision variables); x_{ij}^t represents the individual of t generation population without mutation operation; v_{ij}^{t+1} represents the individuals of t generation population after mutation operation.

The **selection** process determines which candidate solutions will be carried over to the next generation. It involves selecting the best solutions based on their fitness values. The fitness of each trial as a vector or the target vector, will be retained and transferred to the next generation. The individuals following the crossover operation and others in the population are selected using a greedy algorithm. The individuals of the $t+1$ generation are selected by comparing their fitness to form a new population. Before selection, fitness should be determined for each trial vector using an objective function. The vector with inferior fitness will be eliminated, while the vector with superior fitness will be maintained. The following expression represents the selection operation.

$$x_{ij}^{t+1} = \begin{cases} u_i^{t+1}, f(u_i^{t+1}) \leq f(x_i^t) \\ x_i^t, \quad f(u_i^{t+1}) > f(x_i^t) \end{cases}, \tag{14}$$

Where $f(u_i^{t+1})$ represent that fitness of the test individual through crossover; $f(x_i^t)$ indicates the fitness of the target individual.

3. Adaptive Crossover Mutation Differential Evolution (ACMDE)

This section describes the adaptive methods for the ACMDE algorithm with strategies that include crossover for adaptive exploitation, mutation candidate solutions for adaptive exploration, and chaotic sequences for adaptive initialization. The optimization capabilities of the DE algorithm [13] need to be improved to overcome the issue of being vulnerable to local optima, and some modifying variables must be adjusted. The DE principles will then be integrated with adaptive crossover, mutation, and reinitialization procedures in the ACMDE algorithm. The

ACMDE algorithm reduces the number of necessary sensors while increasing node coverage in WSNs. Subsections of adaptive strategies and a thorough exploration of the algorithm's stages and parameters are given, and the experimental findings are presented as follows.In this section, we present the proposed method called Adaptive Crossover Mutation Differential Evolution (ACMDE) for optimal sensor network coverage planning. ACMDE combines the principles of the well-known Differential Evolution (DE) algorithm with adaptive optimization strategies to address the node coverage planning issue in Wireless Sensor Networks (WSNs).

3.1 Adaptive Improvement Strategies

Adaptive improvement strategies are carried out by adjusting and modifying parameters and variables to enhance the algorithm's optimization capability and overcome the trap local optima problem. The key parameters adapting consideration are the initialization population, mutation factor, and crossover probability. The adaptive strategies for improving algorithm performance are highlighted as follows.

Chaotic sequence initialization is one of the efficiencies of most current intelligent optimization algorithms that is greatly influenced by population initialization. The uniformly distributed population can appropriately broaden the algorithm's search scope, improving the algorithm's convergence speed and solution accuracy. Using chaos mapping to initialize the population, individuals can be distributed as evenly as possible in the search space. This feature can be used to improve the algorithm's performance. The primary concept is to map variables into the value range of the chaotic variable space using the properties of the variable c is set to constant. Then, convert the result into the ideal variable space linearly. There are currently many chaotic maps in the optimization field, most notably the Tent, messy, and logistic maps. In this case, we use the chaotic process to create seate starting population. The definition of a chaotic map is as follows.

$$x_{i+1} = mod(x_i + 0.2 - (\frac{0.5}{2\pi})sin(2\pi x_i),1), \quad (15)$$

Where x_{i+1} and x_i the locations of the current and previous iterations of the individual solutions. When generating the original population, the circular mapping technique produces a more uniform spread of population locations than randomly dispersed populations. The algorithm search space area adapts and broadens the population locations for distributing close target solutions that address the local optima and improve the algorithm's optimization efficiency.

Adaptive mutation strategy is carried out with improved *DE/rand/2/bin*; largely, mutation determines how well DE works. Using an adjusted mutation operator, *M* can increase good convergence performance in the later stages of the algorithm. A new mutation strategy can be proposed by using a dynamic adaptive factor to replace *M* with a new one to solve the problem of insufficient convergence performance in the later stages of the algorithm. The following is the adapting formula.

$$V_i^{t+1} = x_{r5}^t + \gamma * (x_{r_1}^t - x_{r2}^t) + M * (x_{r_3}^t - x_{r4}^t), \quad (16)$$

$$\gamma = (\gamma max - \gamma min) * ((T - t)/T) + \gamma min; \quad (17)$$

Where *T* represents the maximum iteration count, and *t* represents the current iteration count. The variation factors have upper and lower bounds, denoted by max and min respectively; $r_1 \neq r_2 \neq r_3 \neq r_4 \neq r_5$.

During the initial phase of evolution, the population explores a wider range of possibilities to discover the optimal solution, and a high value of *M* is preferable at this stage. As the evolution process advances, *M* should be gradually reduced to improve the population's ability to conduct accurate and focused local searches.

Adaptive crossover parameter as the operator is carried out with the value in the crossover operator that determines the proportion of individual information from parents in the new individual. A significant value favors mutants and improves convergence speed, while a small value favors parents and enhances global optimization. However, the standard DE algorithm uses a fixed value, which neglects the trade-off between global and local search. Therefore, an adaptive monotone-decreasing crossover operator is introduced to address this issue, and the formula is calculated as follows.

$$Cr^{t+1} = Cr^t * \exp(-2 * (t/T)), \quad (18)$$

Where t and T represent the present stage of the iteration process, and the maximum iteration, Cr^t is a crossover parameter that is set to 0.9. When the algorithm begins to operate, this operator takes a more considerable value, which can improve the algorithm's early convergence speed and decrease the value of the operator. This promotes the algorithm's global optimization and lessens the likelihood of focusing on local optimization later. Algorithm 1 shows the ACMDE pseudocode.

Algorithm 1. The ACMDE pseudo-code

Input: Set N_P- population size, D -dimension, T- Max_iter, and *ub,lb* -boundaries of function space.

Output: The best object for the entire population.

Initialization: **For** $i = 1$ to N_p **do**
Initiate the population using Eq.(15)
Endfor; circle mapping P^G.
t=1 and evaluate the fitness of each individual in the initial population.
While $t < T$ **do**
(1)Mutation operator
Random selection： r1 ≠ r2 ≠ r3 ≠ r4 ≠ r5;
γ=(γmax-γmin)((T-t)/T)+γmin;*
$V_i^t = X_{r5}^t + \gamma * (X_{r1}^t - X_{r2}^t) + M * (X_{r3}^t - X_{r4}^t)$)
(2)Crossover operator
$j_{rand} = rndint([1, D])$
For j=1 to D **do**
$C_r = C_r * exp(-2*(t/T))$;
If *rand([0,1])<CR* or $j = j_{rand}$
$U_i^t = V_i^t$
Else
$U_i^t = X_i^t$
End if
(3)Selection operator
If $f(U_i^t < f(X_i^t))$ **then**
$X_i^t = U_i^t$
End if
End for
$t = t + 1$
End While
Output: The best object for the entire population.

3.2 Experimental results on mathematic test functions

This subsection evaluates the performance of the proposed ACMDE algorithm by comparing it with popular algorithms. To assess the effectiveness and quality of the improvement algorithm, a set of twenty-eight benchmark functions from CEC2013 is utilized [35]. The benchmark functions encompass different types of modalities, including unimodal, multimodal, hybrid, and compound functions. These functions are identified by serial numbers ranging from CEC01 to CEC28. The unimodal functions are labeled as CEC01 to CEC05, multimodal functions as CEC06 to CEC15, hybrid functions as CEC16 to CEC21, and compound functions as CEC21 to CEC28. These benchmark functions provide diverse complexity and dimensionality settings, offering a comprehensive evaluation of the ACMDE algorithm's performance.

Table 2. Algorithm parameter settings

Algorithms	**Parameters settings**
ACMDE	$a = 2$ to $0, b = 1, l = [-1,1], u, v, g \in [0,1]$, $\beta = 1.5$, NP=60, Cr=0.6 F=0.6, Maxiter=1000
DE [13]	$a = 2$ to $0, b = 1, p \in [0,1]$, NP=60, Cr=0.6 F=0.6, Maxiter=1000
EDE [15]	$a = 2$ to $0, b = 1, p \in [0,1]$, NP=60, Cr=0.6 F=0.6, Maxiter=1000
JaDE [16]	$a = 2$ to $0, b = 1, p \in [0,1]$, NP=60, Cr=0.6 F=0.6, Maxiter=1000
SaDE [17]	$\beta \in [0,2], a = 2$ to $0, b = 1$, $p \in [0,1]$, $\mu, v \in [0,1]$, NP=60, Cr=0.6 F=0.6, Maxiter=1000
ALO [10]	$\omega \in [3\ to\ 6], r = 1\ or\ 0$, NP=60, Maxiter=1000
PSO [9]	$V_{max} = 10, V_{min} = -10, \omega \in [0.9, 0.4], c_1 = c_2 = 1.489$,P=60, Maxiter=1000
MFO [12]	$a = -1, b = 1$, NP=60, Maxiter=1000
GWO [11]	$\alpha \in [0, 2]$, C∈[0,2], $r_1, r_2, r_3 \in [0,1]$, NP=60, Maxiter=1000

The comparison tests with the original optimization algorithms are conducted with uniform parameter settings, including population size, number of iterations, and dimension. The test sets of comparison with the original optimization algorithms are implemented by uniformly setting the same population size, number of iterations, and dimension**,** e.g., 50D, and 100D [36]**.** The obtained results of the algorithms for the test are the global optimum presented in the form of tables and figures. The achieved optimal results of the ACMDE compared with the variant improved DE types of adjusted parameters and with the other algorithms in the literature. The set of improved defferential evolution types included the EDE [15], JADE [16], SaDE [17], and original DE [13] algorithms and the set of the other algorithms are such as ALO [10], GWO [11], MFO [12], and PSO [9] algorithms for the test function with different dimensions. Table 2 lists the selected popular algorithms' parameter settings for the benchmark testing functions.

Tables A1 and A2 (in the index section) show the obtained optimal results of the ACMDE against the EDE, JaDE, SaDE, and DE algorithms for the test function with 50D and 100D performances, respectively. The qualified performance of the suggested ACMDE algorithm is analyzed by comparing it with the selected popular improved DE algorithms. Table A3 shows the achieved optimal results of the ACMDE against the ALO, GWO, MFO, and PSO algorithms for the test function on 100D performance.

Table A3 and A4 show the other set of comparisons with the other algorithms in the literature that achieved optimal results of the ACMDE against the ALO, GWO, MFO, and PSO algorithms for the test function on 50D and 100D performance. The table's contents include two kinds of values of the best optimal 'BEST' and means of the average 'MEAN' of the best ones of 25 runs. The summarized statistics are set in the last of the table, e.g., 'Win,' 'Lose,' and 'Draw,' which means the suggested algorithm is better, less, or similar, respectively. The highlighted values are the best in comparing the tables' row.

The compared algorithms among the unimodal functions, except CEC4, the other seven functions achieved the best results, and the best values were found in cec1 and cec5. Among the 15 Basic Multimodal Functions, the ACMDE has achieved the best results in nine test functions: CEC6, 8, 10, 11, and CEC16~20, which shows that the ACMDE can effectively jump out of local optima. Among the eight Basic Multimodal Functions, CEC21, CEC2, 24, 25, 27, and 28 perform well, showing that the ACMDE can apply to multimodal functions and complex optimization problems. The tests also show that the ACMDE performs well in different dimensions, indicating that the ACMDE has good adaptability. It is seen that the number of wins belongs to the ACMDE algorithm.

The algorithm's convergence rate for the chosen functions, such as CEC1 through CEC5, CEC10 through CEC19, CEC21, and CEC22, is shown in Figure A1. Because the CEC1 and CEC5 unimodal functions are simple, the ACMDE can find the optimum value quickly. After leaving the local optimal, the ACMDE has the highest convergence accuracy and the fastest convergence speed. In the most complex composition functions, the convergence curve of the approach is superior to that of the comparison algorithm, demonstrating the ACMDE algorithm's excellent global optimization capability over other comparison methods. Specific algorithms slow down and enter local optimization in their later stages of evolution. The statistical outcome demonstrates that the ACMDE has more "better" numbers than the others, indicating that the ACMDE performs exceptionally well.

4. ACMDE for Node Coverage Planning

In this section, we delve into the strategy employed by the ACMDE algorithm to achieve optimal node coverage planning in Wireless Sensor Networks (WSNs). The algorithm incorporates several adjustments to facilitate the discovery of suitable solutions, mitigate the occurrence of local optima, and enhance population diversity. Its main objective is to optimize the placement of sensor nodes in WSNs, ensuring maximum coverage while minimizing the required number of sensors. The key factors considered in designing the objective function for optimal WSN coverage are the optimum probability ratio and the coverage target monitoring region ratio within the deployed WSN monitoring area. This section includes subsections on the objective function of the optimal coverage model, the planning of optimal node coverage using the ACMDE algorithm, and an analysis of experimental results.

4.1 Modeling WSN Node Coverage Planning for Objective Function

For the WSN to provide optimal coverage, each deployed node must be placed objectively. WSNs are frequently used in dynamic situations where the environment changes over time due to node failures and sensing area modifications. The ability of the system to autonomously detect and respond to these changes, optimizing node placement and configuration to maintain or improve coverage quality, is necessary to adapt the WSN coverage. Locating nodes in the best possible place might be compared to planets migrating toward a host star or region. The WSN coverage optimization with the ACMDE approach seeks to maximize coverage of the target monitoring region by employing a minimal number of carefully located sensor nodes throughout the target monitoring area.

The objective function aims to maximize the coverage of the target monitoring region while minimizing the number of needed sensor nodes. The coverage is determined by refer Eq.(3), which represents whether a particular position p within the monitoring region is covered or not. The objective *Fit()* is to maximize the summation of

Cov() for all positions p in the region, indicating the overall coverage achieved by the selected sensor nodes. The coverage ratio, which indicates the ideal probability ratio to the network deployed surface in the 2D WSN monitoring region, determines the objective function. This ratio can be maximized using the following formula:

$$Fit(x) = Maximize\ Cov_R = \frac{\sum_{j=1}^{M} P(S,T_j) \times cost(q)}{W \times L}, \tag{19}$$

where Cov_R is the coverage ratio of nodes with coverage distributed places. $P(S,T_j)$ is the probability of reaching target points *(q)* in sensing node of $W \times L$ area; *cost(q)* is the cost sensor node associated with, e.g., placing a sensor node at position q; $Fit(x)$ is the objective function as fitness of WSN nodes optimal coverage.

To ensure the optimization meets certain constraints and requirements, several conditions are imposed through additional constraints: (1). Σ(q ∈ P) $Cov(q) = N$: This constraint ensures that exactly N sensor nodes are deployed in the WSN. The variable P(S,T) is a variable that represents whether a sensor node is placed at position q. The summation of Cov(q) over all positions q in the potential placement set P should be equal to the desired number of sensor nodes N. (2). $Fit(q) \leq \Sigma(q' \in N(q))$ *Cov(q')*: This constraint ensures that a position q is considered covered if at least one sensor node is placed in its coverage neighborhood $N(p)$. N(q) represents the set of positions within the sensing range of position p. If any position p' in the neighborhood N(q) has a sensor node (*Cov(q') = 1*), the coverage variable P(S,T) should be set to 1. (3). $\Sigma(q \in P)\, Cov(q) \times cost(p) \leq Budget$: This constraint imposes a budget constraint on the total cost of sensor node deployment. The variable $cost(q)$ represents the cost associated with placing a sensor node at position q. The summation of $Cov(q)$ multiplied by the corresponding *cost(p)* over all positions q should be less than or equal to the specified budget. These constraints ensure that the deployment meets the desired number of sensor nodes, guarantees coverage within the sensing range of at least one node, and adheres to the budget limitations.

The proposed ACMDE algorithm incorporates several application strategies to enhance the performance of the DE algorithm. Application strategies in ACMDE include adaptive crossover, adaptive enhanced mutation, and reinitialization.The DE algorithm faces challenges when dealing with complex problems, such as difficulties in adapting dynamically, slow convergence, and susceptibility to local optima. Despite having certain advantages, such as simplicity in understanding and implementation, as well as the ability to perform local searches, it lacks balance in algorithm exploration and development, resulting in a population that lacks diversity. To address these limitations, the term "adapting" is used to emphasize the dynamic and automated nature of the system. This reflects its ability to respond dynamically to variations and maintain the desired level of coverage quality without manual intervention. The use of "adapting" instead of "enhancing" or "improving" highlights the concept of self-adjustment over iterations in the optimization process. The adapting strategies are outlined in detail below.

Improved adaptive crossover is a critical operation in the DE algorithm, and ACMDE introduces an enhanced adaptive crossover strategy as shown in Eq. (18). The strategy adjusts the crossover rate dynamically based on the current population's characteristics and the progress made during the optimization process. By adaptively controlling the crossover rate, ACMDE can strike a balance between exploration and exploitation, leading to improved convergence and better solutions.

Enhanced mutation as adapting mutation is another critical component of the DE algorithm, responsible for exploring new regions in the search space. The ACMDE incorporates an enhanced mutation strategy Eqs. (16) (17) that allows for more adequate space exploration solutions. This strategy adjusts the mutation parameters dynamically based on the population's diversity and convergence status, promoting a better exploration-exploitation trade-off.

Reinitialization of essential individual locations is to maintain population diversity and avoid premature convergence. The ACMDE incorporates a reinitialization strategy Eq. (15) for important particular locations that identify critical individuals within the population and reinitialize their positions to introduce new diversity. By preventing stagnation and encouraging exploration, this strategy improves the algorithm's ability to find better solutions.

By formulating the objective function and incorporating these constraints, the optimization algorithm, such as the ACMDE algorithm, can be applied to find the optimal solution that maximizes the coverage while satisfying the constraints. The algorithm will iteratively search for the placement of sensor nodes that achieve the highest coverage within the given constraints, leading to an optimal node coverage planning in WSNs.

4.2 ACMDE Algorithm with WSN Coverage Planning

The proposed strategy of ACMDE is assessed and validated using the objective function, which incorporates the optimum probability ratio (Eq. 3) and the coverage target monitoring region ratio within the deployed WSN monitoring area (Eq. 19). The implementation of the ACMDE algorithm for achieving optimal node coverage in WSN deployment involves several key steps, outlined below:

Step 1: Set the parameter inputs for the sensor network, including the dimensions of the deployed region ($W \times L$), communication range (*Rc*), sensing radius (*Rs*), and the number of nodes (*M*). Additionally, define the variables and algorithm parameters, such as the population size (*NP*), maximum number of iterations (maxIter), density factor, and randomly initialize the local position indexes ($r_1, r_2, .. r_5$).

Step 2: Initialize the population of candidate solutions randomly using the chaotic sequence approach (Eq. 15). This approach helps introduce diversity and exploration in the initial population.

Step 3: Evaluate Fitness: Calculate the fitness value for each candidate solution in the population by assessing the coverage quality of the WSN. The objective function is applied to validate the fitness values, representing the initial node coverage optima within the specified sorting range (Eq. 19).

Step 4: Termination Criteria: Check if the termination criteria have been met, such as reaching the maximum number of iterations or achieving the desired coverage level. If the criteria are satisfied, proceed to *Step 9*. Otherwise, continue to the next step.

Step 5: Mutation: Apply the mutation operation to the population, generating a mutant vector for each candidate solution. This operation introduces diversity by perturbing the candidate solutions.

Step 6: Crossover: Perform the crossover operation between the mutant vector and the target vector, creating a trial vector. This operation combines information from multiple solutions, promoting exploration and convergence towards better solutions.

Step 7: Selection: Compare the fitness values of the trial solution vector and the target solution vector. If the trial solution vector has a superior fitness value, replace the target solution vector with the trial solution vector. Otherwise, retain the target solution vector.

Step 8: Update: Update the population with the newly generated solution vectors. Proceed to Step 2 to continue the iterative process of mutation, crossover, and selection.

Step 9: Terminate the algorithm and output the best solution found for the WSN coverage planning. This solution represents the optimal placement of sensor nodes in the WSN, maximizing coverage while minimizing the required number of sensors.

By following these steps, the ACMDE algorithm systematically explores and refines candidate solutions, leading to improved node coverage planning in WSN deployment.

Algorithm 2. Pseudo-code of the ACMDE for WSN coverage planning

Input: Setting N_P-population, D-dimension, T-Max_iter, g-count set to 1, and input variables of involving sensor network: W×L-netwok deployed region, *M* -number nodes, *Rs*-sensing radius, and Rc-range of communication.
Output: The maximum coverage for the entire sensor netwok.

Initialization: Initiate the population Eq. (15) and evaluate the fitness value Eq.(19) for each candidate solution in the population based on the coverage quality of the WSN.

While $g < T$ **do**
(1) Mutation operator
Generate ($r_1, r_2, .. r_5$) randomly ∈[0,1]: r1 ≠ r2 ≠ r3 ≠ r4 ≠ r5;
Do adaptive mutation Eq. (17)
Do update velocity Eq. (16);
(2) Crossover operator
$j_{rand} = rndint([1, D])$
For j=1 to D **do**
Do adaptive crossover Eq.(18)
If rand([0,1])<CR or j=j_{rand}
$U_i(g) = V_i(g)$
Else

$U_i(g) = X_i(g)$
(3)Selection operator
If $f(U_i(g)) < f(X_i(g))$ **then**
$X_i(g) = U_i(g)$
End if
End for
$g = g + 1$
End While

Output: The maximum coverage for the entire sensor netwok.

The flowchart in Figure 2 illustrates the systematic process of the ACMDE algorithm for achieving optimal sensor network coverage planning in WSN deployment. The algorithm leverages direct screening or similar optimization methods to calculate fitness values, which represent the objective function and analyze the position of objects in the solution space. To guide the agents towards convergent outcomes and identify the best solution, each agent operates independently in the optimization space. The initial set of solutions is augmented by a modification strategy that generates new individuals for further refinement. These individuals introduce original and distinctive solutions to the population. The solutions within the population are then sorted based on their posterior probability, considering the objective function's challenges and requirements. This sorting process facilitates backward selection, allowing the algorithm to focus on promising positions during both the exploration and exploitation phases within the optimization problem space. By following this iterative approach, the ACMDE algorithm progressively refines the solutions and directs the search towards better coverage planning in WSN deployment. This strategy enables the algorithm to effectively navigate the solution space and converge towards the optimal fitness values, resulting in improved sensor network coverage.

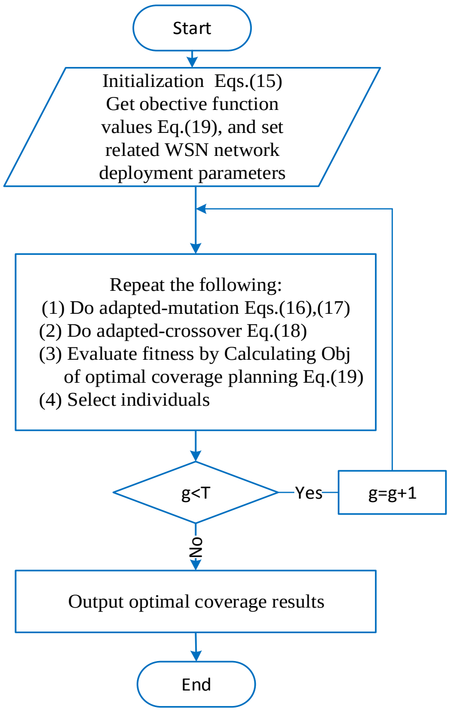


Figure 1: Flowchart of the ACMDE Algorithm for Optimal Sensor Network Coverage Planning during WSN Deployment.

Figure 1 illustrates the flowchart of the ACMDE algorithm, which is designed for achieving optimal coverage planning in WSN deployment. The algorithm follows an iterative process to generate and evaluate potential solutions to ensure adequate sensor network coverage. The algorithm begins by initializing a population of candidate solutions, representing possible sensor node placements. This initialization can be done randomly or using a predefined strategy, as Eq.(15) indicates. The population is then updated iteratively to improve the coverage performance. The mutation is performed by selecting three distinct solutions randomly from the people $(r_1, r_2, .. r_5)$ and perturbing the target solution using the differential operator, as defined in Eqs. (17) and (16). This adaptive mutation strategy helps explore the solution space effectively. Next, the crossover operator combines the previous solution with the target solution, resulting in a trial solution (Eq.(18)). This recombination process introduces new potential solutions for evaluation. The fitness evaluation is carried out to assess the quality of the trial solution in terms of coverage criteria, as defined by Eq. (19). The trial solution's coverage performance is compared with that of the target solution, determining if an improvement has been made. After evaluating and updating the target solutions, the algorithm checks if the termination criterion has been met or if convergence conditions have been achieved. If so, the algorithm stops iterating and proceeds to the next step. Finally, the best solution found during the iterations is returned as the optimal sensor network coverage plan. This solution represents the placement of sensor nodes that achieve maximum coverage according to the objectives of the algorithm.

4.3 Analysis and Discussion Results

The subsection presents an analysis, discussion, and comparison of the findings obtained from the experiment, focusing on various aspects of the experimental design and the proposed optimization strategies. The effectiveness of the performance criteria used to evaluate the schemes is investigated across different situations and settings that represent diverse contexts of WSN deployment.

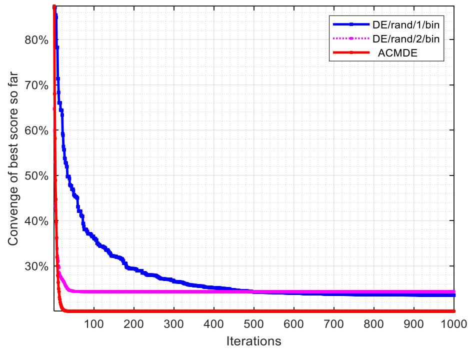
Experimental Setup: The experimental scenario is established by considering the deployment of sensor nodes in a monitoring area of varying dimensions, such as $W \times L$ (e.g., $50\times50m^2$, $90\times90m^2$, $110\times110m^2$, and $165\times165m^2$). In WSN deployments, it is common to encounter challenging environments with obstacles like high mountains or buildings, which result in certain regions being inaccessible to sensing nodes, referred to as "monitoring blind regions." The coverage ratio is calculated to determine the proportion of the monitoring area covered by the sensor network. A higher coverage ratio indicates better coverage quality, while the algorithm's computational efficiency is evaluated based on the time required to identify the optimal coverage solution. Applications that require prompt response favor faster execution speeds. Another important metric is the number of sensors needed to achieve the desired coverage, with a preference for a lower sensor count to minimize costs and energy consumption.

Table 3 provides the specification settings for the WSN node deployment areas as parameter configurations. It includes the sensing radius (*Rs*) of the sensor nodes set to 10.01 m, the communication radius (*Rc*) set to 20.01 m, and the number of sensor nodes (*M*) varying between 25, 40, 50, and 65. The maximum number of iterations (*MaxIter*) represents the maximum number of loop repetitions, which can be set to different values such as 500 and 1000, depending on the specific experiment.

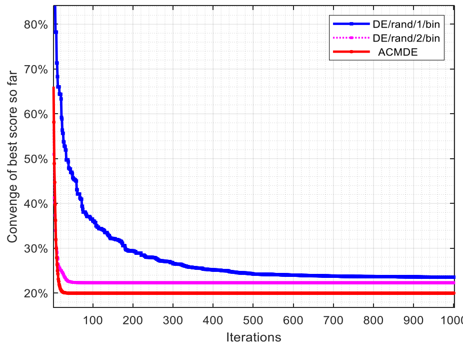
Table 3: Parameter settings for WSN deployment in experimental specification areas

Descriptions	**Parameters**	**Values**
The desired areas deployed WSN	$W \times L$	$50\times50m^2$, $90\times90m^2$, $110\times110m^2$, $165\times165m^2$
Number of required sensor nodes deploying	*M*	25, 40, 50, 65
Communication radius range	*Rc*	20.01 m
Sensing scanning radius range	*Rs*	10.01 m
Maximum generations in the loop	*MaxIter*	500, or 1000 generations

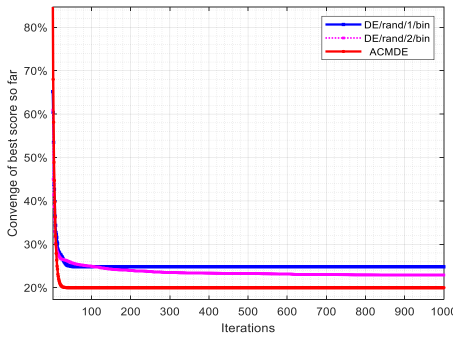
Comparative Analysis of Optimization Methods: In order to evaluate the effectiveness of the proposed optimization strategy, a comparison is conducted with various existing optimization methods and their variations. The evaluation is performed using different metrics, including the coverage ratio, sensor utilization, and computational efficiency. The ACMDE algorithm's optimal results are compared with other algorithms such as he ABC [31], PSO [24], GWO [32], ACO [33], and DE algorithms [37] [38] for the node optimal coverage planning in WSN deployment. This comparison aims to assess the performance of the proposed strategy against established approaches. The experimental specifications of WSN deployment areas, along with environment variables and parameter settings, are utilized to test and validate the performance and accuracy of the suggested approach under the same conditions as the compared algorithms. For instance, the initialization coverage planning phase for ACMDE considers different scenarios with *M* set to 25, 40, 50, and 65 nodes, respectively.



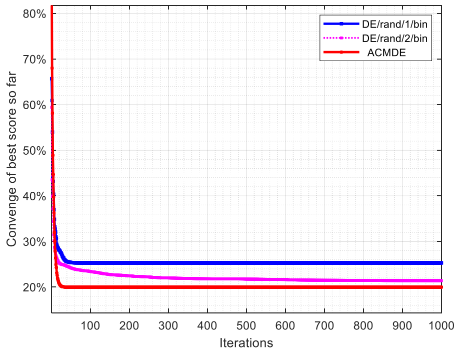
a) 25 nodes /50×50m^2



b) 40 nodes /90×90m^2



c) 50 nodes /110×110m^2

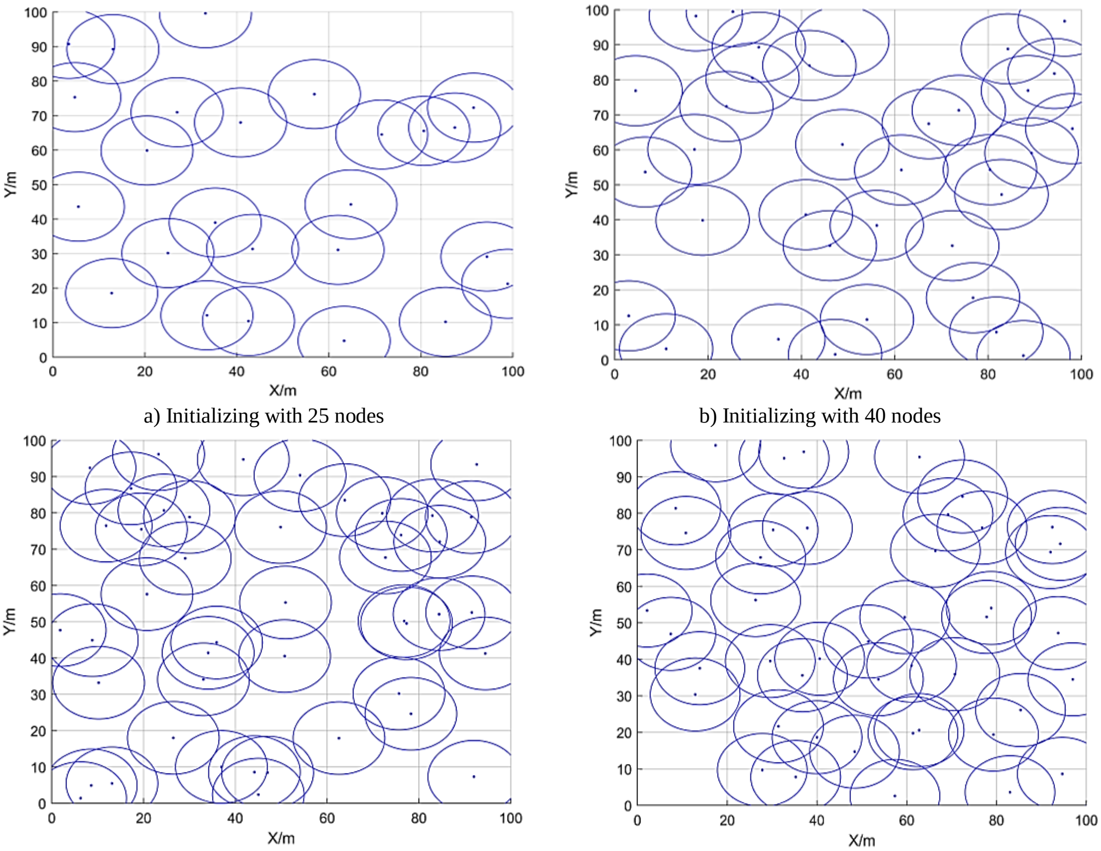


d) 65 node /165×165m^2

Figure 2: Graphical comparison of ACMDE's coverage diagram with original DE for statistical coverage optimization scheme in various node densities: a) 25 nodes /50x50 m^2, b) 40 nodes / 90x90 m^2, c) 50 nodes / 110x110 m^2, and d) 65 nodes / 165x165 m^2.

Comparing with Original DE Variants: Figure 2 presents a comparison between the graphical convergence diagrams of the ACMDE algorithm and the original DE variations [13] [14], specifically DE/rand/1bin (DE1) and DE/rand/2bin (DE2). The comparison is conducted for a statistical coverage optimization scheme with varying node densities: a) 25 nodes / 50x50 m², b) 40 nodes / 90x90 m², c) 50 nodes / 110x110 m², and d) 60 nodes / 165x165 m². In this comparison, the objective function is transformed into a minimization problem by multiplying the maximum objective function in Eq. (19) by -1 to facilitate comparison based on convergence speed. The values of the minimization problem are approximately -1 times those of the corresponding maximization problem.

Analyzing the convergence curves shown in Figure 2, it can be observed that the ACMDE algorithm generally achieves faster convergence compared to the DE/rand/1bin and DE/rand/2bin strategies. The graphical convergence curves of the ACMDE algorithm reach the global optimization point around 50 to 80 iterations (generations), whereas the DE algorithm requires 500 to 600 iterations to reach similar optimization results.
To evaluate the effectiveness of the ACMDE algorithm, it is compared against several popular optimization techniques commonly used for WSN node coverage planning. The initialization graphical nodes distribution for the statistical coverage optimization scheme is an essential aspect of the experimental cases. Figure 3 illustrates the initialization graphical coverage diagram of the ACMDE algorithm for the statistical coverage optimization scheme, with *M* set to 25, 40, 50, and 65, respectively.



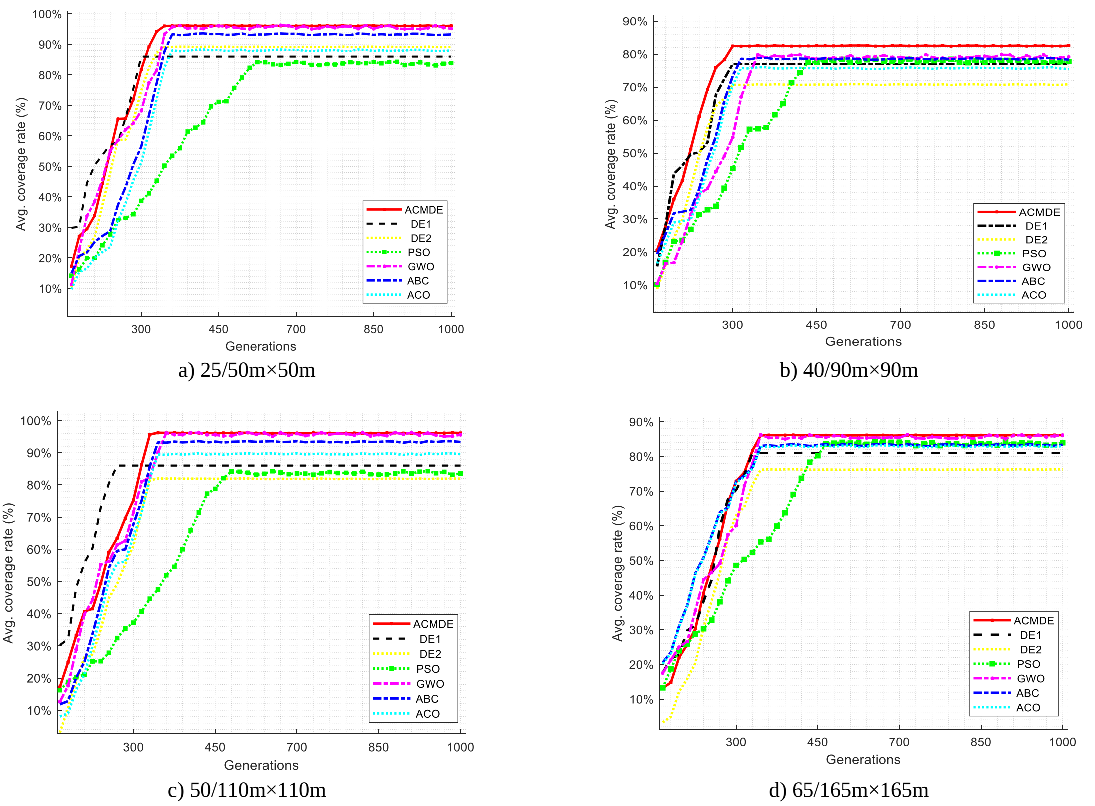
a) Initializing with 25 nodes b) Initializing with 40 nodes

c) Initializing with 50 nodes d) Initializing with 65 nodes

Figure 3: Initialization Graphical Coverage Diagram of ACMDE Algorithm for Statistical Coverage Optimization Scheme with Different m Sets of Sensor Nodes (25, 40, 50, and 65)

Figure 3 showcases the initialization graphical coverage diagram of the ACMDE algorithm for the statistical coverage optimization scheme. The diagram represents the sensor network deployment with different values of m, indicating the number of sensor nodes set to 25, 40, 50, and 65, respectively. The visualization illustrates the initial configuration of sensor nodes for the ACMDE algorithm before the optimization process begins.

The Different Metaheuristic Algorithms: various metaheuristic algorithms, including ACMDE, PSO [24], GWO [32], ABC [31], ACO [33], DE1 [37], and DE2 [38] are utilized to evaluate the WSN node deployment scenarios for achieving optimal coverage rates under the same density and environmental conditions. Figure 4 presents a graphical comparison of convergence among these algorithms, specifically ACO, ABC, PSO, GWO, DE1, DE2, and ACMDE, for WSN node deployment scenarios with different densities and environmental settings, such as (a) 25/50×50 m², (b) 40/90×90 m², (c) 50/110×110 m², and (d) 65/165×165 m². The experimental implementation scenarios in Figure 4 demonstrate the performance of the metaheuristic algorithms in terms of convergence speed for achieving optimal coverage rates in WSN deployment. ACMDE shows relatively high convergence rates, indicating its effectiveness in maximizing network coverage within the monitoring area. In certain cases, the proposed ACMDE approach outperforms other competing algorithms by achieving higher static coverage percentages.



a) 25/50m×50m

b) 40/90m×90m

c) 50/110m×110m

d) 65/165m×165m

Figure 4: Comparison of the ACMDE algorithm's best coverage rates with other algorithms for deploying WSN monitoring node areas of different sizes (a) 25/50×50m², (b) 40/90×90m², (c) 50/110×110m², and (d) 65/165×165m²

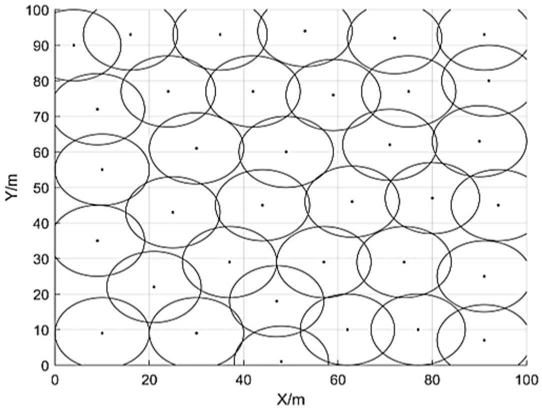
Table 4 presents a comparative analysis between the proposed ACMDE method and other techniques (ABC [31], PSO [24], GWO [32], ACO [33], DE variant [37] [38], and GA [39] algorithms) in various scenarios, including percentage coverage rate, running times, convergence iterations, and monitoring area sizes. The ACMDE scheme demonstrates a high coverage rate, achieving complete coverage of the node's space area due to its balanced exploration and exploitation search strategy. The results in Table 4 indicate that the ACMDE approach achieves a percentage coverage rate of 87% in 100m x 100m deployment scenarios, outperforming other methods such as GWO, which reached a maximum of 84% under the same conditions. Additionally, the ACMDE scheme exhibits faster execution times compared to alternative approaches, thanks to its adjusted initialization with stochastic reverse direction. Overall, the ACMDE method delivers superior performance and offers the best solutions in terms of coverage areas.

Table 4: Comparative analysis of results between the proposed ACMDE method and other techniques

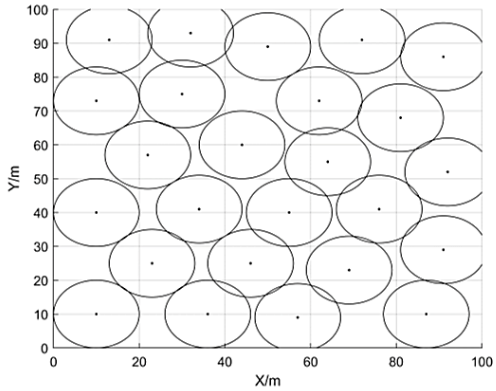
Approach	Factor variables	50×50 m²	90×90 m²	100×100 m²	165×165 m²
ACMDE	Obtained coverage rate (%)	**81%**	**83%**	**89%**	**82%**
	No. of deploying sensor nodes	25	40	50	65
	No. of generations reaching to convergence	**135**	**503**	**556**	**765**
	Executing time consumption (s)	**$2.75\times10^{+00}$**	**$6.15\times10^{+00}$**	**$6.57\times10^{+00}$**	**$8.31\times10^{+00}$**
DE1	Obtained coverage rate (%)	79%	79%	82%	77%
	No. of deploying sensor nodes	25	40	50	65
	No. of generations reaching to convergence	396	343	343	754
	Executing time consumption (s)	$2.78\times10^{+00}$	$6.22\times10^{+00}$	$6.65\times10^{+00}$	$8.41\times10^{+00}$
DE2	Obtained coverage rate (%)	80%	81%	83%	79%
	No. of deploying sensor nodes	25	40	50	65
	No. of generations reaching to convergence	665	333	563	954
	Executing time consumption (s)	$3.52\times10^{+00}$	$7.98\times10^{+00}$	$7.86\times10^{+00}$	$9.74\times10^{+00}$
GA	Obtained coverage rate (%)	76%	77%	80%	75%
	No. of deploying sensor nodes	25	40	50	65
	No. of generations reaching to convergence	445	555	665	876
	Executing time consumption (s)	$2.92\times10^{+00}$	$6.28\times10^{+00}$	$7.22\times10^{+00}$	$9.22\times10^{+00}$
PSO	Obtained coverage rate (%)	79%	81%	80%	79%

	No. of deploying sensor nodes	25	40	50	65
	No. of generations reaching to convergence	665	333	563	954
	Executing time consumption (s)	$3.12\times10^{+00}$	$6.98\times10^{+00}$	$7.46\times10^{+00}$	$9.44\times10^{+00}$
GWO	Obtained coverage rate (%)	80%	79%	83%	78%
	No. of deploying sensor nodes	25	40	50	65
	No. of generations reaching to convergence	451	555	675	876
	Executing time consumption (s)	$3.06\times10^{+00}$	$6.84\times10^{+00}$	$7.31\times10^{+00}$	$9.25\times10^{+00}$
ACO	Obtained coverage rate (%)	81%	82%	84%	78%
	No. of deploying sensor nodes	25	40	50	65
	No. of generations reaching to convergence	334	44	544	755
	Executing time consumption (s)	$3.12\times10^{+00}$	$6.98\times10^{+00}$	$7.46\times10^{+00}$	$9.44\times10^{+00}$
ABC	Obtained coverage rate (%)	80%	81%	81%	76%
	No. of deploying sensor nodes	25	40	50	65
	No. of generations reaching to convergence	145	256	234	844
	Executing time consumption (s)	$3.09\times10^{+00}$	$6.91\times10^{+00}$	$7.38\times10^{+00}$	$9.34\times10^{+00}$

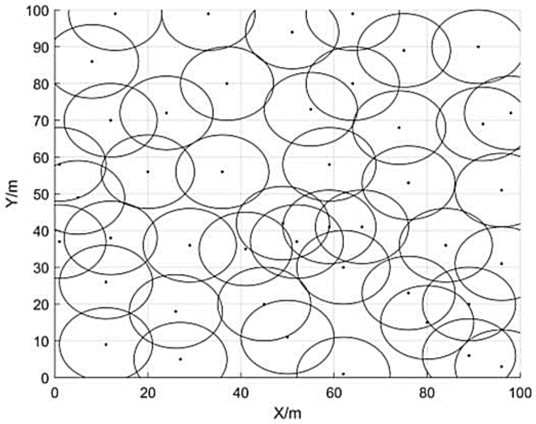
Figure 5 displays the graphical coverage of the selected different metaheuristic algorithms, e.g., the DE, ABC, PSO, GWO, ACO, and ACMDE approaches for the WSN node area deployment of the 40/90x90 case. It is seen that obtained coverage of the ACMDE scheme finding demonstrates that, under the identical test settings, the ACMDE technique offers a coverage rate that is quite a high fit, with less overlap and a better-changed layout of the sensor nodes.



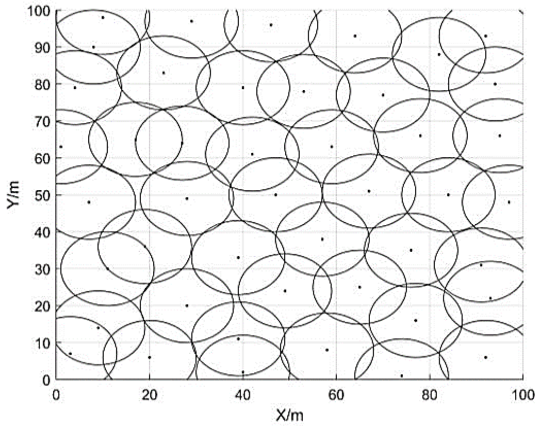
a) ACMDE



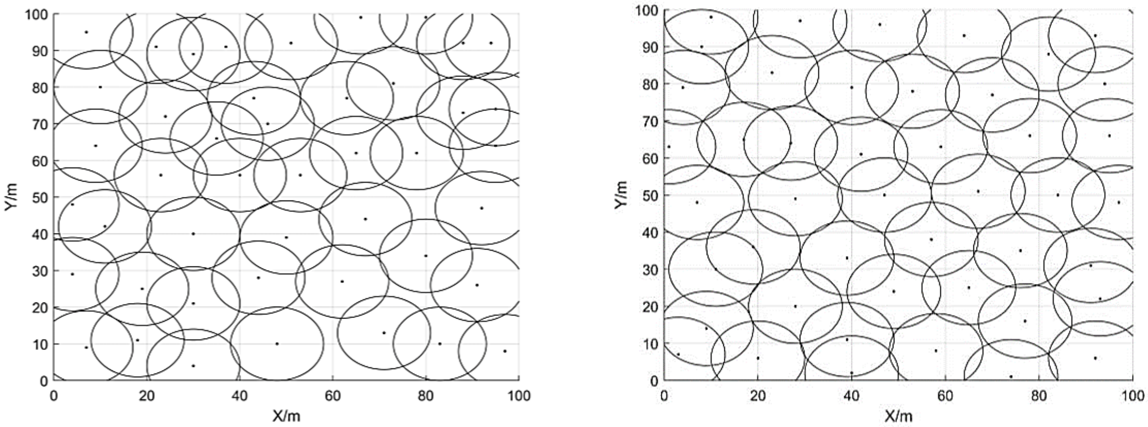
b) DE2



c) PSO



d) GWO



e) ABC f) ACO

Figure 5: Graphical coverage comparison of different metaheuristic algorithms (DE, ABC, PSO, GWO, ACO, and ACMDE) for WSN node area deployment.

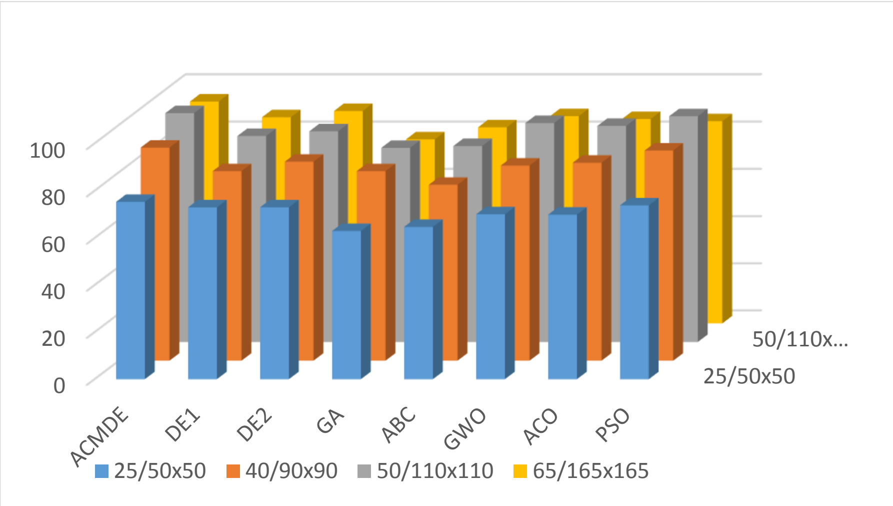


Figure 6: Comparison of coverage rates for different sensor node counts deployed on W×L area monitoring scenarios.

Figure 6 illustrates the statistical deployment coverage rate of sensor nodes achieved through the ACMDE optimization approach, compared to the ABC, PSO, GWO, ACO, DE1,2 and GA algorithms, across different 2D monitoring areas. The results clearly demonstrate that the ACMDE algorithm consistently achieves a high coverage rate in the network's monitoring area. Notably, the ACMDE technique outperforms the other algorithms in terms of coverage rate, exhibiting reduced overlap and a more optimized layout of sensor nodes.

Table 5 compares the outcomes obtained from the proposed ACMDE method and other existing approaches. The ACMDE method outperforms the alternative algorithms regarding percentage coverage rate, running times, iterations to convergence, and monitoring area sizes. The table summarizes the comparison between the ACMDE method and various approaches, including GA, PSO, Grey GWO, ACO, DE (DE1, DE2) and GA algorithms. It examines factors such as percentage coverage rate, running times, iterations to convergence, and monitoring area sizes. The ACMDE method achieves a high percentage coverage rate, indicating its ability to cover a significant portion of the target area effectively. This demonstrates its proficiency in optimizing multiple dimensions simultaneously and improving overall coverage. These results highlight the effectiveness and superiority of the ACMDE method in addressing multi-dimensional optimization problems and its potential for diverse applications in monitoring, optimization, and decision-making.

Table 5: Comparison of Outcomes between Proposed ACMDE Method and Existing Approaches

Approach	**% Coverage Rate**	**Running Times**	**Convergence**	**Monitoring Area**
ACMDE	**High**	**Low**	**Fast**	**Large**
DE1	Low	High	Slow	Small

DE2	Moderate	Low	Moderate	Moderate
GA	Low	High	Slow	Small
PSO	Low	High	Slow	Moderate
GWO	Moderate	Moderate	Moderate	Moderate
ACO	Moderate	Moderate	Slow	Moderate
ABC	Moderate	High	Moderate	Moderate

Regarding running times, the ACMDE method exhibits low execution times, showcasing its computational efficiency and suitability for time-sensitive applications and resource-constrained environments. Additionally, the ACMDE method demonstrates fast convergence, requiring fewer iterations to reach convergence compared to the other algorithms. This indicates its capability to approach optimal solutions quickly and reduces the overall optimization time. Furthermore, the ACMDE method can effectively cover extensive monitoring areas, as evidenced by the associated monitoring area scale. This attribute is precious in applications requiring broad surveillance or monitoring capabilities. The ACMDE method presents good performance characteristics, making it a robust optimization approach for various practical scenarios.

The analysis and discussion of the experimental results strongly support the effectiveness of the proposed ACMDE algorithm for optimal sensor network coverage planning in WSNs. The algorithm surpasses alternative optimization strategies, delivering superior coverage ratios while maintaining competitive computational efficiency. These findings highlight the ACMDE algorithm's potential as a valuable tool for enhancing WSN coverage and optimizing network performance.

5. Conclusion

This study introduced the ACMDE, an adaptive optimization technique, as a solution for sensor network coverage planning in WSNs. The ACMDE algorithm effectively addressed issues related to uneven node distribution and limited coverage in random deployments by employing enhanced adaptive strategies for crossover, mutation, and reinitialization. By adapting and updating equations for crossover, modification, and reinitialization, the ACMDE algorithm overcomes the limitations of slow convergence speed and local extrema convergence observed in the original DE algorithm. The mathematical modeling of the objective function considers the sensing radius and communication capabilities of sensor nodes during deployment, enabling adequate WSN node coverage. The experimental results validate the efficiency of the ACMDE method and establish its superiority over other optimization techniques. Comparative analyses with existing approaches consistently demonstrate that the ACMDE methodology provides the best options for WSN coverage. The algorithm's performance can be further enhanced in future research, and its applicability can be extended to diverse sensor networks. The ACMDE approach holds promise for facilitating the effective and reliable deployment of WSNs, thereby enhancing the overall quality of service in various application domains. Continued advancements and refinements in the ACMDE algorithm can contribute to optimizing WSN coverage planning and foster improvements in network performance and efficiency.

References

[1] R. Elhabyan, W. Shi, and M. St-Hilaire, “Coverage protocols for wireless sensor networks: Review and future directions,” *Journal of Communications and Networks*, vol. 21, no. 1, pp. 45–60, 2019.

[2] T.-T. Nguyen, T.-K. Dao, M.-F. Horng, and C.-S. Shieh, “An Energy-based Cluster Head Selection Algorithm to Support Long-lifetime in Wireless Sensor Networks,” *Journal of Network Intelligence*, vol. 01, no. 01, pp. 23–37, 2016.

[3] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008, doi: 10.1016/j.comnet.2008.04.002.

[4] A. Singh, S. Sharma, and J. Singh, “Nature-inspired algorithms for Wireless Sensor Networks: A comprehensive survey,” *Computer Science Review*, vol. 39, p. 100342, 2021, doi: https://doi.org/10.1016/j.cosrev.2020.100342.

[5] W. Wu, Z. Zhang, W. Lee, and D. Du, *Optimal coverage in wireless sensor networks*. Springer, 2020.

[6] A. Ghosh and S. K. Das, “Coverage and connectivity issues in wireless sensor networks: A survey,”

Pervasive and Mobile Computing, vol. 4, no. 3, pp. 303–334, 2008.

[7] J. H. Holland, *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975.

[8] M. Srinivas and L. M. Patnaik, "Genetic Algorithms: A Survey," *Computer*, vol. 27, no. 6, pp. 17–26, 1994, doi: 10.1109/2.294849.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, vol. 6(4), pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[10] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015, doi: 10.1016/j.advengsoft.2015.01.010.

[11] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[12] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-based systems*, vol. 89, pp. 228–249, 2015.

[13] K. V Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution. A Practical Approach to Global Optimization*. Springer Science & Business Media., 2005.

[14] J.-S. Pan, Z. Meng, S.-C. Chu, and J. F. Roddick, "QUATRE Algorithm with Sort Strategy for Global Optimization in Comparison with DE and PSO Variants BT - Proceedings of the Fourth Euro-China Conference on Intelligent Data Analysis and Applications," 2018, pp. 314–323.

[15] W. Gong, Z. Cai, C. X. Ling, and H. Li, "Enhanced Differential Evolution With Adaptive Strategies for Numerical Optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 2, pp. 397–413, 2011, doi: 10.1109/TSMCB.2010.2056367.

[16] J. Zhang and A. C. Sanderson, "JADE: adaptive differential evolution with optional external archive," *IEEE Transactions on evolutionary computation*, vol. 13, no. 5, pp. 945–958, 2009.

[17] A. K. Qin and P. N. Suganthan, "Self-adaptive differential evolution algorithm for numerical optimization," in *2005 IEEE Congress on Evolutionary Computation*, 2005, vol. 2, pp. 1785-1791 Vol. 2, doi: 10.1109/CEC.2005.1554904.

[18] Z. Meng and J. Pan, "HARD-DE: Hierarchical ARchive Based Mutation Strategy With Depth Information of Evolution for the Enhancement of Differential Evolution on Numerical Optimization," *IEEE Access*, vol. 7, pp. 12832–12854, 2019, doi: 10.1109/ACCESS.2019.2893292.

[19] A. H. Gandomi, X. S. Yang, S. Talatahari, and A. H. Alavi, "Metaheuristic Algorithms in Modeling and Optimization," in *Metaheuristic Applications in Structures and Infrastructures*, 2013, pp. 1–24.

[20] N. Liu, J.-S. Pan, J. Wang, and T.-T. Nguyen, "An adaptation multi-group quasi-affine transformation evolutionary algorithm for global optimization and its application in node localization in wireless sensor networks," *Sensors (Switzerland)*, vol. 19, no. 19, 2019, doi: 10.3390/s19194112.

[21] T.-T. Nguyen *et al.*, "A hybridized parallel bats algorithm for combinatorial problem of traveling salesman," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 5, pp. 5811–5820, Feb. 2020, doi: 10.3233/jifs-179668.

[22] T.-K. Dao, S.-C. Chi, T.-T. Nguyen, T.-D. Nguyen, and V.-T. Nguyen, "An Optimal WSN Node Coverage Based on Enhanced Archimedes Optimization Algorithm," *Entropy*, vol. 24, no. 8. 2022, doi: 10.3390/e24081018.

[23] B. Wang, "Coverage problems in sensor networks: A survey," *ACM Computing Surveys (CSUR)*, vol. 43, no. 4, pp. 1–53, 2011.

[24] N. A. B. A. Aziz, A. W. Mohemmed, and M. Y. Alias, "A wireless sensor network coverage optimization algorithm based on particle swarm optimization and Voronoi diagram," in *2009 International Conference on Networking, Sensing and Control*, 2009, pp. 602–607, doi: 10.1109/ICNSC.2009.4919346.

[25] T.-T. Nguyen, J.-S. Pan, T.-Y. Wu, T.-K. Dao, and T.-D. Nguyen, "Node coverage optimization strategy based on ions motion optimization," *Journal of Network Intelligence*, vol. 4, no. 1, 2019.

[26] X. Guo, C. Zhao, X. Yang, and C. Sun, “A Deterministic Sensor Node Deployment Method with Target Coverage and Node Connectivity BT - Artificial Intelligence and Computational Intelligence,” 2011, pp. 201–207.

[27] A. Tripathi, H. P. Gupta, T. Dutta, R. Mishra, K. K. Shukla, and S. Jit, “Coverage and Connectivity in WSNs: A Survey, Research Issues and Challenges,” *IEEE Access*, vol. 6, pp. 26971–26992, 2018, doi: 10.1109/ACCESS.2018.2833632.

[28] J. Tian, M. Gao, and G. Ge, “Wireless sensor network node optimal coverage based on improved genetic algorithm and binary ant colony algorithm,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1–11, 2016.

[29] A. Shan, X. Xu, and Z. Cheng, “Target Coverage in Wireless Sensor Networks with Probabilistic Sensors,” *Sensors*, vol. 16, no. 9. 2016, doi: 10.3390/s16091372.

[30] H. M. Ammari, “Connected k-coverage in two-dimensional wireless sensor networks using hexagonal slicing and area stretching,” *Journal of Parallel and Distributed Computing*, vol. 153, pp. 89–109, 2021, doi: https://doi.org/10.1016/j.jpdc.2020.12.008.

[31] X. Xu, G. Zhou, and T. Chen, “Research on coverage optimisation of wireless sensor networks based on an artificial bee colony algorithm,” *International Journal of Wireless and Mobile Computing*, vol. 9, no. 2, pp. 199–204, 2015.

[32] Z. Wang, H. Xie, Z. Hu, D. Li, J. Wang, and W. Liang, “Node coverage optimization algorithm for wireless sensor networks based on improved grey wolf optimizer,” *Journal of Algorithms & Computational Technology*, vol. 13, p. 1748302619889498, 2019.

[33] P. Huang, F. Lin, C. Liu, J. Gao, and J. Zhou, “ACO-Based Sweep Coverage Scheme in Wireless Sensor Networks,” *Journal of Sensors*, vol. 2015, p. 484902, 2015, doi: 10.1155/2015/484902.

[34] S. G. Shiva Prasad Yadav and A. Chitra, “Wireless Sensor Networks - Architectures , Protocols , Simulators and Applications : a Survey,” *International Journal of Electronics and Computer Science Engineering*, vol. 1, no. 4, pp. 1941–1953, 2012.

[35] J. J. Liang, B. Y. Qu, and P. N. Suganthan, “Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization,” *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore*, vol. 635, 2013.

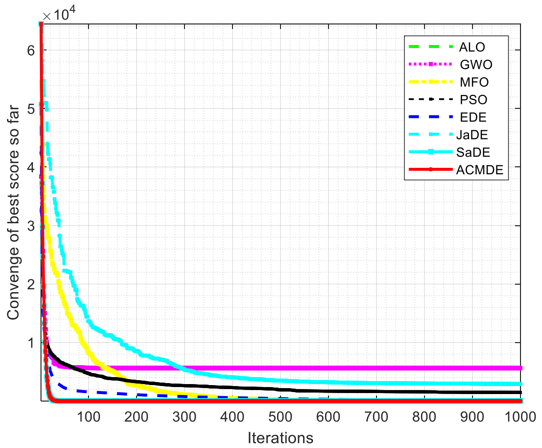
[36] J. J. Liang, B. Y. Qu, D. W. Gong, and C. T. Yue, “Problem definitions and evaluation criteria for the CEC 2019 special session on multimodal multiobjective optimization,” 2019.

[37] N. Qin and J. Chen, “An area coverage algorithm for wireless sensor networks based on differential evolution,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 8, p. 1550147718796734, 2018.

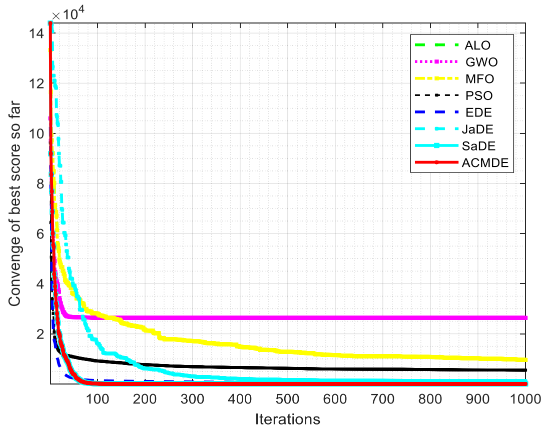
[38] C. Naik and D. Pushparaj Shetty, “A novel meta-heuristic differential evolution algorithm for optimal target coverage in wireless sensor networks,” in *Innovations in Bio-Inspired Computing and Applications: Proceedings of the 9th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2018) held in Kochi, India during December 17-19, 2018* 9, 2019, pp. 83–92.

[39] H. ZainEldin, M. Badawy, M. Elhosseini, H. Arafat, and A. Abraham, “An improved dynamic deployment technique based-on genetic algorithm (IDDT-GA) for maximizing coverage in wireless sensor networks,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 4177–4194, 2020.

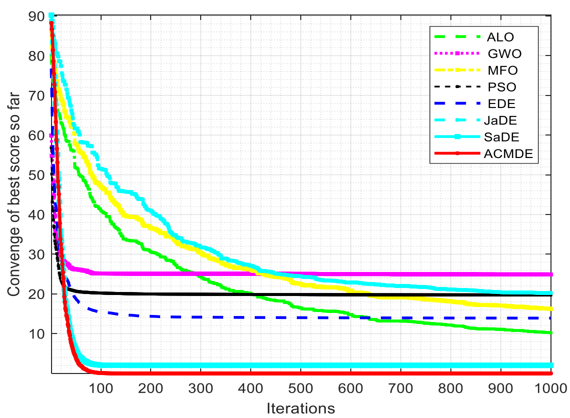
Appendix A



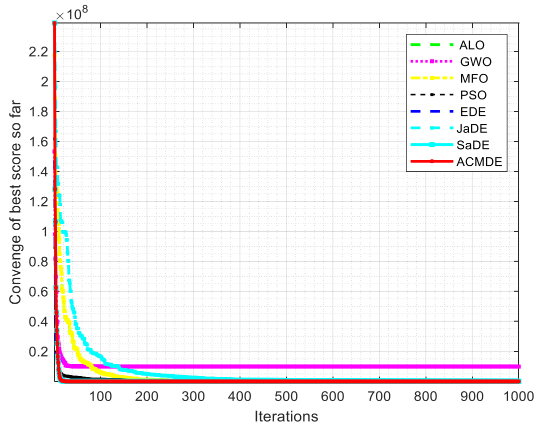
a) CEC1



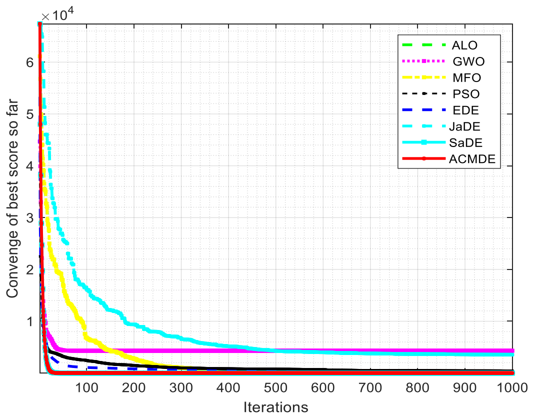
b) CEC3



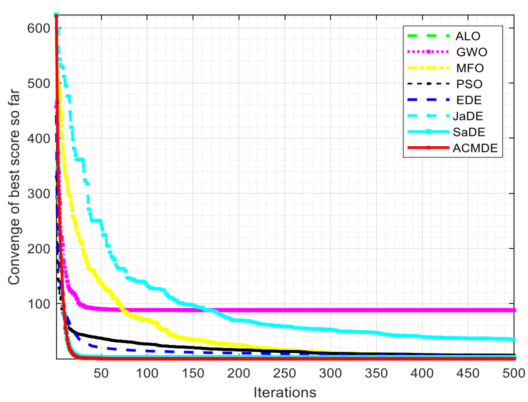
c) CEC5



d) CEC7



e) CEC13



f) CEC22

Figure A1. The comparison of convergence curve outcomes of the ACMDE as achieved optimal results with the SaDE, JaDE, EDE, DE, PSO, MFO, GWO, and ALO algorithms for the selected test functions with 50D.

Table A1. The achieved optimal results of the ACMDE compares with the DE, EDE, JaDE, and SaDE algorithms for the test function on 50D performance.

50D	DE		EDE		JaDE		SaDE		**ACMDE**	
	BEST	MEAN	BEST	MEAN	BEST	MEAN	BEST	MEAN	BEST	MEAN
CEC1	-1.25×10^{-03}	1.87×10^{-02}	$-1.37\times10^{+03}$	$2.12\times10^{+02}$	$\mathbf{-1.40\times10^{-03}}$	$\mathbf{-1.40\times10^{-03}}$	7.07×10^{-01}	1.39×10^{-01}	$\mathbf{-1.40\times10^{-03}}$	$-1.39\times10^{+04}$
CEC2	$4.32\times10^{+06}$	$2.46\times10^{+07}$	$1.04\times10^{+07}$	$3.74\times10^{+07}$	$1.24\times10^{+08}$	$1.71\times10^{+08}$	$8.28\times10^{+07}$	$1.14\times10^{+08}$	$\mathbf{1.86\times10^{-06}}$	$\mathbf{7.02\times10^{-06}}$
CEC3	$1.57\times10^{+09}$	$4.74\times10^{+09}$	$1.65\times10^{+10}$	$3.01\times10^{+03}$	$1.74\times10^{+09}$	$3.14\times10^{+09}$	$3.70\times10^{+10}$	$5.47\times10^{+10}$	$\mathbf{5.68\times10^{-07}}$	$\mathbf{2.35\times10^{-05}}$
CEC4	$2.13\times10^{+04}$	$\mathbf{3.86\times10^{+04}}$	$\mathbf{1.12\times10^{+04}}$	$3.98\times10^{+04}$	$7.22\times10^{+04}$	$9.00\times10^{+04}$	$6.42\times10^{+04}$	$9.62\times10^{+04}$	$5.87\times10^{+04}$	$1.11\times10^{+05}$

CEC5	$-8.61\times10^{+02}$	$-1.93\times10^{+02}$	$-8.51\times10^{+02}$	$7.72\times10^{+02}$	$-9.99\times10^{+02}$	$-9.98\times10^{+02}$	$5.20\times10^{+02}$	$2.89\times10^{+03}$	**$-1.00\times10^{+03}$**	**$-1.00\times10^{+03}$**
CEC6	$-8.10\times10^{+02}$	$-7.73\times10^{+02}$	$-8.48\times10^{+02}$	$-6.56\times10^{+02}$	$-8.54\times10^{+02}$	$-8.49\times10^{+02}$	$-5.39\times10^{+02}$	$-1.05\times10^{+02}$	**$-8.84\times10^{+02}$**	**$-8.58\times10^{+02}$**
CEC7	$-6.56\times10^{+02}$	$-7.31\times10^{+02}$	$-6.55\times10^{+02}$	$-3.76\times10^{+02}$	$-7.51\times10^{+02}$	**$-7.03\times10^{+02}$**	$-6.52\times10^{+02}$	$-6.17\times10^{+02}$	**$-7.58\times10^{+02}$**	$-7.14\times10^{+02}$
CEC8	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**
CEC9	**$-5.87\times10^{+02}$**	**$-5.80\times10^{+02}$**	$-5.71\times10^{+02}$	$-5.65\times10^{+02}$	$-5.63\times10^{+02}$	$-5.60\times10^{+02}$	$-5.71\times10^{+02}$	$-5.67\times10^{+02}$	$-5.82\times10^{+02}$	$-5.73\times10^{+02}$
CEC10	$-4.16\times10^{+02}$	$-2.19\times10^{+02}$	$-4.85\times10^{+02}$	$-1.03\times10^{+02}$	$-4.06\times10^{+02}$	$-3.60\times10^{+02}$	$6.66\times10^{+02}$	$1.35\times10^{+03}$	**$-5.00\times10^{+02}$**	**$-4.68\times10^{+02}$**
CEC11	$-3.49\times10^{+02}$	$-3.05\times10^{+02}$	$-2.10\times10^{+02}$	$-1.02\times10^{+02}$	$-2.24\times10^{+02}$	$-2.00\times10^{+02}$	-4.48×10^{-01}	$5.98\times10^{+01}$	**$-3.61\times10^{+02}$**	**$-3.39\times10^{+02}$**
CEC12	$-2.25\times10^{+02}$	$-1.59\times10^{+02}$	$-9.88\times10^{+01}$	$2.87\times10^{+00}$	$-9.16\times10^{+01}$	$-6.89\times10^{+01}$	$1.09\times10^{+02}$	$1.67\times10^{+02}$	**$-2.66\times10^{+02}$**	**$-2.00\times10^{+02}$**
CEC13	**$-8.90\times10^{+01}$**	$1.95\times10^{+01}$	$4.31\times10^{+01}$	$1.70\times10^{+02}$	$-8.62\times10^{+00}$	$3.01\times10^{+01}$	$2.12\times10^{+02}$	$2.72\times10^{+02}$	$-8.79\times10^{+01}$	**$-1.82\times10^{+01}$**
CEC14	**$1.80\times10^{+03}$**	**$3.58\times10^{+03}$**	$2.89\times10^{+03}$	$4.73\times10^{+03}$	$6.13\times10^{+03}$	$6.86\times10^{+03}$	$6.35\times10^{+03}$	$7.15\times10^{+03}$	$2.07\times10^{+03}$	$3.77\times10^{+03}$
CEC15	**$2.69\times10^{+03}$**	$5.20\times10^{+03}$	$3.36\times10^{+03}$	**$5.12\times10^{+03}$**	$6.85\times10^{+03}$	$7.72\times10^{+03}$	$5.61\times10^{+03}$	$6.77\times10^{+03}$	$4.33\times10^{+03}$	$7.15\times10^{+03}$
CEC16	$2.02\times10^{+02}$	**$2.03\times10^{+02}$**	**$2.01\times10^{+02}$**	**$2.03\times10^{+02}$**	$2.02\times10^{+02}$	**$2.03\times10^{+02}$**	$2.02\times10^{+02}$	**$2.03\times10^{+02}$**	$2.02\times10^{+02}$	$2.04\times10^{+02}$
CEC17	$4.03\times10^{+02}$	$4.77\times10^{+02}$	$5.21\times10^{+02}$	$6.73\times10^{+02}$	$5.38\times10^{+02}$	$5.61\times10^{+02}$	$8.76\times10^{+02}$	$9.33\times10^{+02}$	**$3.68\times10^{+02}$**	**$4.35\times10^{+02}$**
CEC18	$6.34\times10^{+02}$	$6.67\times10^{+02}$	$6.58\times10^{+02}$	$7.57\times10^{+02}$	$6.53\times10^{+02}$	$6.70\times10^{+02}$	$9.77\times10^{+02}$	$1.04\times10^{+03}$	**$5.03\times10^{+02}$**	**$6.09\times10^{+02}$**
CEC19	$5.05\times10^{+02}$	$6.00\times10^{+02}$	$5.19\times10^{+02}$	$1.01\times10^{+03}$	$5.21\times10^{+02}$	**$5.23\times10^{+02}$**	$2.35\times10^{+03}$	$7.14\times10^{+03}$	**$5.04\times10^{+02}$**	$5.24\times10^{+02}$
CEC20	**$6.12\times10^{+02}$**	**$6.13\times10^{+02}$**	**6.12**$\times10^{+02}$	$6.14\times10^{+02}$	$6.13\times10^{+02}$	**$6.13\times10^{+02}$**	$6.14\times10^{+02}$	$6.14\times10^{+02}$	**$6.12\times10^{+02}$**	**$6.13\times10^{+02}$**
CEC21	$1.14\times10^{+03}$	$1.67\times10^{+03}$	$9.43\times10^{+02}$	$1.13\times10^{+03}$	$1.01\times10^{+03}$	$1.03\times10^{+03}$	$2.65\times10^{+03}$	$2.79\times10^{+03}$	**$9.00\times10^{+02}$**	$9.82\times10^{+02}$
CEC22	**$2.29\times10^{+03}$**	$3.95\times10^{+03}$	$5.00\times10^{+03}$	$6.18\times10^{+03}$	$7.54\times10^{+03}$	$8.37\times10^{+03}$	$7.48\times10^{+03}$	$8.38\times10^{+03}$	$2.90\times10^{+03}$	**$3.84\times10^{+03}$**
CEC23	**$3.35\times10^{+03}$**	**$5.68\times10^{+03}$**	$5.53\times10^{+03}$	$7.02\times10^{+03}$	$8.29\times10^{+03}$	$8.92\times10^{+03}$	$6.79\times10^{+03}$	$8.05\times10^{+03}$	$5.33\times10^{+03}$	$8.08\times10^{+03}$
CEC24	$1.24\times10^{+03}$	$1.27\times10^{+03}$	$1.30\times10^{+03}$	$1.34\times10^{+03}$	$1.30\times10^{+03}$	$1.30\times10^{+03}$	$1.28\times10^{+03}$	$1.30\times10^{+03}$	$1.25\times10^{+03}$	**$1.26\times10^{+03}$**
CEC25	**$1.36\times10^{+03}$**	$1.39\times10^{+03}$	$1.40\times10^{+03}$	$1.44\times10^{+03}$	$1.40\times10^{+03}$	$1.41\times10^{+03}$	$1.41\times10^{+03}$	$1.43\times10^{+03}$	**$1.36\times10^{+03}$**	**$1.38\times10^{+03}$**
CEC26	$1.40\times10^{+03}$	$1.54\times10^{+03}$	$1.58\times10^{+03}$	$1.60\times10^{+03}$	$1.42\times10^{+03}$	**$1.43\times10^{+03}$**	**1.40**$\times10^{+03}$	**$1.43\times10^{+03}$**	**$1.40\times10^{+03}$**	$1.55\times10^{+03}$
CEC27	$2.01\times10^{+03}$	**$2.14\times10^{+03}$**	$2.43\times10^{+03}$	$2.67\times10^{+03}$	$2.56\times10^{+03}$	$2.61\times10^{+03}$	$2.43\times10^{+03}$	$2.52\times10^{+03}$	**$2.04\times10^{+03}$**	$2.23\times10^{+03}$
CEC28	$2.10\times10^{+03}$	$2.63\times10^{+03}$	$2.44\times10^{+03}$	$4.21\times10^{+03}$	$1.71\times10^{+03}$	**$1.74\times10^{+03}$**	$4.12\times10^{+03}$	$4.51\times10^{+03}$	**$1.70\times10^{+03}$**	$1.96\times10^{+03}$
win	8	7	8	8	9	7	8	7	-	-
lose	17	18	18	17	17	20	18	18	-	-
draw	4	3	2	3	2	1	3	4	-	-

Table A2. Optimal results comparison of ACMDE with DE, EDE, JaDE, and SaDE algorithms for 100D test function performance.

	DE		EDE		JaDE		SaDE		**ACMDE**	
100D	BEST	MEAN	BEST	MEAN	BEST	MEAN	BEST	MEAN	BEST	MEAN
CEC1	$1.74\times10^{+02}$	$2.35\times10^{+03}$	$-9.36\times10^{+02}$	$5.45\times10^{+02}$	$-1.25\times10^{+03}$	$-1.17\times10^{+03}$	$2.64\times10^{+04}$	$3.73\times10^{+04}$	**$-1.40\times10^{+01}$**	**$-1.31\times10^{+01}$**
CEC2	$2.06\times10^{+03}$	$5.17\times10^{+03}$	$2.30\times10^{+03}$	$9.17\times10^{+03}$	$4.87\times10^{+03}$	$6.25\times10^{+03}$	$1.94\times10^{+03}$	$2.97\times10^{+03}$	**$1.09\times10^{+01}$**	**$1.85\times10^{+01}$**
CEC3	$8.07\times10^{+03}$	$2.01\times10^{+4}$	$3.20\times10^{+4}$	$7.18\times10^{+5}$	$5.34\times10^{+4}$	$6.88\times10^{+5}$	$7.50\times10^{+5}$	$1.04\times10^{+5}$	**$3.17\times10^{+09}$**	**$1.68\times10^{+01}$**
CEC4	$3.99\times10^{+04}$	**$5.64\times10^{+04}$**	**$3.84\times10^{+04}$**	$7.30\times10^{+04}$	$1.41\times10^{+05}$	$1.65\times10^{+05}$	$1.34\times10^{+05}$	$1.58\times10^{+05}$	$9.85\times10^{+01}$	$2.01\times10^{+02}$
CEC5	$-4.79\times10^{+02}$	$1.01\times10^{+02}$	$-7.36\times10^{+02}$	$3.42\times10^{+02}$	$-9.15\times10^{+02}$	$-8.78\times10^{+02}$	$1.61\times10^{+03}$	$2.99\times10^{+03}$	**$-1.00\times10^{+03}$**	**$3.66\times10^{+01}$**
CEC6	$-7.29\times10^{+02}$	$-5.87\times10^{+02}$	$-7.87\times10^{+02}$	$-6.16\times10^{+02}$	$-8.19\times10^{+02}$	$-7.79\times10^{+02}$	$9.24\times10^{+02}$	$1.32\times10^{+03}$	**$-8.56\times10^{+02}$**	**$-8.23\times10^{+02}$**
CEC7	**$-7.46\times10^{+02}$**	**$-7.22\times10^{+02}$**	$-6.69\times10^{+02}$	$-3.91\times10^{+02}$	$-6.31\times10^{+02}$	$-6.13\times10^{+02}$	$-6.38\times10^{+02}$	$-5.96\times10^{+02}$	$-7.09\times10^{+02}$	$-6.48\times10^{+02}$
CEC8	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**-6.79×10**$^{+02}$	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**
CEC9	**$-5.64\times10^{+02}$**	**$-5.58\times10^{+02}$**	$-5.44\times10^{+02}$	$-5.34\times10^{+02}$	$-5.29\times10^{+02}$	$-5.26\times10^{+02}$	$-5.45\times10^{+02}$	$-5.35\times10^{+02}$	$-5.63\times10^{+02}$	$-5.49\times10^{+02}$
CEC10	$3.81\times10^{+01}$	$3.27\times10^{+02}$	$-8.44\times10^{+01}$	$3.33\times10^{+02}$	$8.72\times10^{+02}$	$1.80\times10^{+03}$	$2.17\times10^{+03}$	$3.22\times10^{+03}$	**$-4.97\times10^{+02}$**	**$-4.74\times10^{+02}$**
CEC11	$-2.16\times10^{+02}$	$-1.38\times10^{+02}$	$1.14\times10^{+02}$	$3.42\times10^{+02}$	$8.47\times10^{+00}$	$5.53\times10^{+01}$	$3.41\times10^{+02}$	$3.95\times10^{+02}$	**$-3.17\times10^{+02}$**	**$-2.17\times10^{+02}$**
CEC12	$-1.16\times10^{+02}$	$2.40\times10^{+01}$	$2.28\times10^{+02}$	$3.89\times10^{+02}$	$1.38\times10^{+02}$	$2.20\times10^{+02}$	$4.88\times10^{+02}$	$5.88\times10^{+02}$	**$-1.40\times10^{+02}$**	**$-8.56\times10^{+00}$**
CEC13	**$8.81\times10^{+01}$**	**$2.15\times10^{+02}$**	$4.43\times10^{+02}$	$5.95\times10^{+02}$	$2.91\times10^{+02}$	$3.30\times10^{+02}$	$5.23\times10^{+02}$	$6.63\times10^{+02}$	$9.26\times10^{+01}$	$2.56\times10^{+02}$
CEC14	**$4.22\times10^{+03}$**	**$6.13\times10^{+03}$**	$6.34\times10^{+03}$	$9.49\times10^{+03}$	$1.23\times10^{+04}$	$1.32\times10^{+04}$	$1.34\times10^{+04}$	$1.40\times10^{+04}$	$4.52\times10^{+03}$	$6.08\times10^{+03}$
CEC15	**$5.37\times10^{+03}$**	**$8.40\times10^{+03}$**	$8.56\times10^{+03}$	$1.08\times10^{+04}$	$1.39\times10^{+04}$	$1.47\times10^{+04}$	$1.22\times10^{+04}$	$1.36\times10^{+04}$	$7.87\times10^{+03}$	$1.34\times10^{+04}$
CEC16	$2.03\times10^{+02}$	$2.04\times10^{+02}$	**$2.01\times10^{+02}$**	**$2.02\times10^{+02}$**	$2.03\times10^{+02}$	$2.04\times10^{+02}$	$2.03\times10^{+02}$	$2.04\times10^{+02}$	$2.03\times10^{+02}$	$2.04\times10^{+02}$
CEC17	$5.65\times10^{+02}$	$7.06\times10^{+02}$	$1.05\times10^{+03}$	$1.21\times10^{+03}$	$8.26\times10^{+02}$	$8.91\times10^{+02}$	$1.36\times10^{+03}$	$1.48\times10^{+03}$	**$4.53\times10^{+02}$**	**$5.49\times10^{+02}$**
CEC18	$7.79\times10^{+02}$	$9.66\times10^{+02}$	$1.08\times10^{+03}$	$1.29\times10^{+03}$	$9.58\times10^{+02}$	$1.01\times10^{+03}$	$1.50\times10^{+03}$	$1.59\times10^{+03}$	**$7.23\times10^{+02}$**	**$9.34\times10^{+02}$**
CEC19	$5.55\times10^{+02}$	**$1.61\times10^{+03}$**	$6.54\times10^{+02}$	$1.79\times10^{+03}$	$2.75\times10^{+03}$	$5.56\times10^{+03}$	$9.76\times10^{+03}$	$1.63\times10^{+04}$	**$5.10\times10^{+02}$**	$5.68\times10^{+03}$
CEC20	**$6.20\times10^{+02}$**	**$6.22\times10^{+02}$**	$6.22\times10^{+02}$	$6.24\times10^{+02}$	$6.23\times10^{+02}$	$6.23\times10^{+02}$	$6.23\times10^{+02}$	$6.24\times10^{+02}$	$6.22\times10^{+02}$	$6.23\times10^{+02}$
CEC21	$1.86\times10^{+03}$	$2.87\times10^{+03}$	$1.56\times10^{+03}$	$2.06\times10^{+03}$	$1.58\times10^{+03}$	$2.57\times10^{+03}$	$4.73\times10^{+03}$	$4.88\times10^{+03}$	**$9.00\times10^{+02}$**	**$1.60\times10^{+03}$**
CEC22	$6.41\times10^{+03}$	$8.43\times10^{+0}3$	$9.32\times10^{+03}$	$1.24\times10^{+04}$	$1.40\times10^{+04}$	$1.47\times10^{+04}$	$1.49\times10^{+04}$	$1.60\times10^{+04}$	**$5.36\times10^{+03}$**	**$8.12\times10^{+03}$**
CEC23	**$7.58\times10^{+03}$**	**$9.60\times10^{+03}$**	$8.64\times10^{+03}$	$1.34\times10^{+04}$	$1.51\times10^{+04}$	$1.61\times10^{+04}$	$1.45\times10^{+04}$	$1.61\times10^{+04}$	$9.17\times10^{+03}$	$1.44\times10^{+04}$
CEC24	$1.29\times10^{+03}$	$1.33\times10^{+03}$	$1.42\times10^{+03}$	$1.47\times10^{+03}$	$1.38\times10^{+03}$	$1.39\times10^{+03}$	$1.42\times10^{+03}$	$1.43\times10^{+03}$	**$1.28\times10^{+03}$**	**$1.32\times10^{+03}$**
CEC25	$1.43\times10^{+03}$	$1.46\times10^{+03}$	$1.52\times10^{+03}$	$1.56\times10^{+03}$	$1.50\times10^{+03}$	$1.51\times10^{+03}$	$1.53\times10^{+03}$	$1.54\times10^{+03}$	**$1.41\times10^{+03}$**	**$1.45\times10^{+03}$**
CEC26	$1.59\times10^{+03}$	$1.60\times10^{+03}$	**$1.41\times10^{+03}$**	$1.67\times10^{+03}$	$1.46\times10^{+03}$	**$1.50\times10^{+03}$**	$1.67\times10^{+03}$	$1.68\times10^{+03}$	$1.59\times10^{+03}$	$1.62\times10^{+03}$
CEC27	**$2.49\times10^{+03}$**	**$2.71\times10^{+03}$**	$3.35\times10^{+03}$	$3.70\times10^{+03}$	$3.42\times10^{+03}$	$3.49\times10^{+03}$	$3.41\times10^{+03}$	$3.56\times10^{+03}$	$2.65\times10^{+03}$	$2.85\times10^{+03}$
CEC28	$2.15\times10^{+03}$	$3.44\times10^{+03}$	$2.87\times10^{+03}$	$5.95\times10^{+03}$	$1.91\times10^{+03}$	**$2.00\times10^{+03}$**	$6.09\times10^{+03}$	$7.13\times10^{+03}$	**$1.80\times10^{+03}$**	$3.85\times10^{+03}$
win	8	7	9	7	9	6	8	6	--	--
lose	16	18	15	17	15	18	16	19	--	--
draw	4	3	4	4	4	4	4	3	--	--

Table A3. The achieved optimal results of the ACMDE against the ALO, GWO, MFO, and PSO algorithms for the test function on 50D performance.

50D	ALO	GWO	MFO	PSO	**ACMDE**

	BEST	MEAN	BEST	MEAN	BEST	BEST	MEAN	BEST	MEAN	BEST
CEC1	$9.37\times10^{+03}$	$1.95\times10^{+04}$	$9.60\times10^{+03}$	$1.63\times10^{+04}$	$2.56\times10^{+04}$	$3.11\times10^{+04}$	$1.14\times10^{+05}$	$1.30\times10^{+05}$	**$-1.40\times10^{+03}$**	**$-1.12\times10^{+03}$**
CEC2	$1.11\times10^{+01}$	$1.87\times10^{+01}$	$1.52\times10^{+01}$	$7.27\times10^{+01}$	$2.01\times10^{+01}$	$2.84\times10^{+01}$	$1.41\times10^{+01}$	$1.99\times10^{+02}$	**$4.80\times10^{+0}$**	**$8.41\times10^{+0}$**
CEC3	$1.41\times10^{+2}$	$8.21\times10^{+2}$	$3.24\times10^{+2}$	$1.25\times10^{+3}$	$2.06\times10^{+2}$	$7.76\times10^{+2}$	$5.48\times10^{+5}$	$1.49\times10^{+2}$	**$5.10\times10^{+1}$**	**$1.14\times10^{+1}$**
CEC4	**$1.11\times10^{+05}$**	**$1.35\times10^{+05}$**	$1.54\times10^{+05}$	$2.22\times10^{+05}$	$2.64\times10^{+05}$	$3.80\times10^{+05}$	$2.59\times10^{+05}$	$3.54\times10^{+05}$	$2.79\times10^{+05}$	$4.24\times10^{+05}$
CEC5	$2.51\times10^{+03}$	$4.35\times10^{+03}$	$1.69\times10^{+03}$	$4.63\times10^{+03}$	$1.51\times10^{+03}$	$2.35\times10^{+03}$	$1.57\times10^{+04}$	$2.12\times10^{+04}$	**$-8.65\times10^{+02}$**	**$2.24\times10^{+03}$**
CEC6	$6.93\times10^{+02}$	$1.55\times10^{+03}$	$7.64\times10^{+02}$	$1.80\times10^{+03}$	$2.01\times10^{+03}$	$2.60\times10^{+03}$	$1.86\times10^{+04}$	$2.31\times10^{+04}$	**$-6.27\times10^{+02}$**	**$-4.50\times10^{+02}$**
CEC7	**$-6.56\times10^{+02}$**	$-4.17\times10^{+02}$	$9.34\times10^{+03}$	$1.89\times10^{+05}$	$-3.35\times10^{+02}$	$4.81\times10^{+02}$	$2.95\times10^{+04}$	$1.72\times10^{+05}$	$-6.47\times10^{+02}$	**$-5.45\times10^{+02}$**
CEC8	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**
CEC9	**$-5.12\times10^{+02}$**	**$-5.00\times10^{+02}$**	$-4.62\times10^{+02}$	$-4.51\times10^{+02}$	$-4.42\times10^{+02}$	$-4.37\times10^{+02}$	$-4.67\times10^{+02}$	$-4.48\times10^{+02}$	**$-5.12\times10^{+02}$**	$-4.80\times10^{+02}$
CEC10	$1.40\times10^{+03}$	$2.46\times10^{+03}$	$2.62\times10^{+03}$	$4.21\times10^{+03}$	$1.11\times10^{+04}$	$1.32\times10^{+04}$	$1.35\times10^{+04}$	$1.60\times10^{+04}$	**$-2.53\times10^{+02}$**	**$2.05\times10^{+02}$**
CEC11	$1.82\times10^{+02}$	$4.27\times10^{+02}$	$1.24\times10^{+03}$	$1.77\times10^{+03}$	$8.90\times10^{+02}$	$1.00\times10^{+03}$	$1.59\times10^{+03}$	$1.82\times10^{+03}$	**$6.85\times10^{+01}$**	**$3.10\times10^{+02}$**
CEC12	**$3.63\times10^{+02}$**	**$5.60\times10^{+02}$**	$1.45\times10^{+03}$	$1.88\times10^{+03}$	$1.18\times10^{+03}$	$1.31\times10^{+03}$	$1.78\times10^{+03}$	$2.07\times10^{+03}$	$3.81\times10^{+02}$	$5.73\times10^{+02}$
CEC13	**$6.54\times10^{+02}$**	**$9.23\times10^{+02}$**	$1.24\times10^{+03}$	$2.17\times10^{+03}$	$1.31\times10^{+03}$	$1.44\times10^{+03}$	$1.99\times10^{+03}$	$2.21\times10^{+03}$	$8.41\times10^{+02}$	$1.11\times10^{+03}$
CEC14	$1.39\times10^{+04}$	**$1.74\times10^{+04}$**	$2.10\times10^{+04}$	$2.46\times10^{+04}$	$3.06\times10^{+04}$	$3.15\times10^{+04}$	$2.94\times10^{+04}$	$3.10\times10^{+04}$	**$1.17\times10^{+04}$**	$1.75\times10^{+04}$
CEC15	**$1.36\times10^{+04}$**	**$1.87\times10^{+04}$**	$2.17\times10^{+04}$	$2.44\times10^{+04}$	$3.08\times10^{+04}$	$3.18\times10^{+04}$	$2.58\times10^{+04}$	$2.97\times10^{+04}$	$1.62\times10^{+04}$	$2.62\times10^{+04}$
CEC16	$2.04\times10^{+02}$	$2.05\times10^{+02}$	$2.03\times10^{+02}$	**$2.04\times10^{+02}$**	$2.04\times10^{+02}$	$2.05\times10^{+02}$	$2.04\times10^{+02}$	**$2.04\times10^{+02}$**	**$2.02\times10^{+02}$**	$2.05\times10^{+02}$
CEC17	$1.34\times10^{+03}$	$1.55\times10^{+03}$	$2.44\times10^{+03}$	$3.06\times10^{+03}$	$3.03\times10^{+03}$	$3.27\times10^{+03}$	$3.50\times10^{+03}$	$3.82\times10^{+03}$	**$9.59\times10^{+02}$**	**$1.25\times10^{+03}$**
CEC18	$1.73\times10^{+03}$	**$2.00\times10^{+03}$**	$2.84\times10^{+03}$	$3.22\times10^{+03}$	$3.06\times10^{+03}$	$3.37\times10^{+03}$	$3.63\times10^{+03}$	$3.98\times10^{+03}$	**$1.68\times10^{+03}$**	$2.18\times10^{+03}$
CEC19	$7.50\times10^{+03}$	$2.57\times10^{+04}$	$1.35\times10^{+04}$	$3.36\times10^{+04}$	$8.59\times10^{+05}$	$1.52\times10^{+06}$	$1.42\times10^{+05}$	$2.90\times10^{+05}$	**$6.01\times10^{+02}$**	**$7.24\times10^{+03}$**
CEC20	**$6.50\times10^{+02}$**	**$6.50\times10^{+02}$**	**$6.50\times10^{+02}$**	**$6.50\times10^{+02}$**	**$6.50\times10^{+02}$**	**$6.50\times10^{+02}$**	**$6.50\times10^{+02}$**	**$6.50\times10^{+02}$**	**$6.50\times10^{+02}$**	**$6.50\times10^{+02}$**
CEC21	$3.53\times10^{+03}$	$5.57\times10^{+03}$	$2.80\times10^{+03}$	$4.91\times10^{+03}$	$7.53\times10^{+03}$	$8.23\times10^{+03}$	$8.89\times10^{+03}$	$9.48\times10^{+03}$	**$1.13\times10^{+03}$**	**$1.36\times10^{+03}$**
CEC22	$1.60\times10^{+04}$	$2.15\times10^{+04}$	$2.70\times10^{+04}$	$2.96\times10^{+04}$	$3.23\times10^{+04}$	$3.31\times10^{+04}$	$3.24\times10^{+04}$	$3.37\times10^{+04}$	**$1.28\times10^{+04}$**	**$1.71\times10^{+04}$**
CEC23	**$1.71\times10^{+04}$**	**$2.35\times10^{+04}$**	$2.50\times10^{+04}$	$2.88\times10^{+04}$	$3.30\times10^{+04}$	$3.41\times10^{+04}$	$3.11\times10^{+04}$	$3.40\times10^{+04}$	$2.01\times10^{+04}$	$2.98\times10^{+04}$
CEC24	**$1.45\times10^{+03}$**	**$1.49\times10^{+03}$**	$1.67\times10^{+03}$	$2.03\times10^{+03}$	$1.61\times10^{+03}$	$1.62\times10^{+03}$	$1.71\times10^{+03}$	$1.74\times10^{+03}$	**$1.45\times10^{+03}$**	**$1.49\times10^{+03}$**
CEC25	$1.65\times10^{+03}$	$1.68\times10^{+03}$	$1.83\times10^{+03}$	$1.94\times10^{+03}$	$1.75\times10^{+03}$	$1.76\times10^{+03}$	$1.82\times10^{+03}$	$1.84\times10^{+03}$	**$1.61\times10^{+03}$**	**$1.65\times10^{+03}$**
CEC26	**$1.73\times10^{+03}$**	**$1.75\times10^{+03}$**	$1.87\times10^{+03}$	$1.93\times10^{+03}$	$1.91\times10^{+03}$	$1.91\times10^{+03}$	$1.87\times10^{+03}$	$1.90\times10^{+03}$	$1.74\times10^{+03}$	$1.79\times10^{+03}$
CEC27	$4.14\times10^{+03}$	**$4.38\times10^{+03}$**	$5.94\times10^{+03}$	$6.80\times10^{+03}$	$5.67\times10^{+03}$	$5.81\times10^{+03}$	$5.91\times10^{+03}$	$6.08\times10^{+03}$	**$4.12\times10^{+03}$**	$4.44\times10^{+03}$
CEC28	$8.51\times10^{+03}$	$1.07\times10^{+04}$	$1.75\times10^{+04}$	$2.09\times10^{+04}$	$1.18\times10^{+04}$	$1.31\times10^{+04}$	$1.84\times10^{+04}$	$1.96\times10^{+04}$	**$4.77\times10^{+03}$**	**$8.53\times10^{+03}$**
win	9	10	8	9	7	5	6	8	--	--
lose	18	16	18	16	19	21	19	18	--	--
draw	1	2	2	3	2	2	3	2	--	--

Table A4: Optimal results comparison of ACMDE with ALO, GWO, MFO, and PSO algorithms for 100D test function performance.

	ALO		GWO		MFO		PSO		**ACMDE**	
100D	BEST	MEAN	BEST	MEAN	BEST	BEST	MEAN	BEST	MEAN	BEST
CEC1	$8.37\times10^{+03}$	$1.95\times10^{+04}$	$8.60\times10^{+03}$	$1.63\times10^{+04}$	$2.56\times10^{+04}$	$3.11\times10^{+04}$	$1.14\times10^{+05}$	$1.30\times10^{+05}$	**$-1.40\times10^{+03}$**	**$-1.12\times10^{+03}$**
CEC2	$1.11\times10^{+01}$	$1.87\times10^{+01}$	$1.52\times10^{+01}$	$7.27\times10^{+01}$	$2.01\times10^{+01}$	$2.84\times10^{+01}$	$1.41\times10^{+01}$	$1.99\times10^{+02}$	**$4.50\times10^{+0}$**	**$8.41\times10^{+0}$**
CEC3	$1.81\times10^{+2}$	$2.21\times10^{+2}$	$3.24\times10^{+2}$	$1.25\times10^{+3}$	$2.06\times10^{+2}$	$7.76\times10^{+2}$	$5.48\times10^{+5}$	$1.49\times10^{+2}$	**$5.20\times10^{+1}$**	**$1.14\times10^{+1}$**
CEC4	**$1.11\times10^{+05}$**	**$1.35\times10^{+05}$**	$1.54\times10^{+05}$	$2.22\times10^{+05}$	$2.64\times10^{+05}$	$3.80\times10^{+05}$	$2.59\times10^{+05}$	$3.54\times10^{+05}$	$2.79\times10^{+05}$	$4.24\times10^{+05}$
CEC5	$2.51\times10^{+03}$	$4.35\times10^{+03}$	$1.69\times10^{+03}$	$4.63\times10^{+03}$	$1.51\times10^{+03}$	$2.35\times10^{+03}$	$1.57\times10^{+04}$	$2.12\times10^{+04}$	**$-8.65\times10^{+02}$**	**$2.24\times10^{+03}$**
CEC6	$6.93\times10^{+02}$	$1.55\times10^{+03}$	$7.64\times10^{+02}$	$1.80\times10^{+03}$	$2.01\times10^{+03}$	$2.60\times10^{+03}$	$1.86\times10^{+04}$	$2.31\times10^{+04}$	**$-6.27\times10^{+02}$**	**$-4.50\times10^{+02}$**
CEC7	$-5.56\times10^{+02}$	$-4.17\times10^{+02}$	$9.34\times10^{+03}$	$1.89\times10^{+05}$	$-3.35\times10^{+02}$	$4.81\times10^{+02}$	$2.95\times10^{+04}$	$1.72\times10^{+05}$	$-6.47\times10^{+02}$	**$-5.45\times10^{+02}$**
CEC8	$-5.79\times10^{+02}$	$-6.79\times10^{+02}$	$-6.79\times10^{+02}$	$-6.79\times10^{+02}$	$-6.79\times10^{+02}$	$-6.79\times10^{+02}$	$-6.79\times10^{+02}$	$-6.79\times10^{+02}$	**$-6.79\times10^{+02}$**	**$-6.79\times10^{+02}$**
CEC9	$-5.12\times10^{+02}$	**$-5.00\times10^{+02}$**	$-4.62\times10^{+02}$	$-4.51\times10^{+02}$	$-4.42\times10^{+02}$	$-4.37\times10^{+02}$	$-4.67\times10^{+02}$	$-4.48\times10^{+02}$	**$-5.12\times10^{+02}$**	$-4.80\times10^{+02}$
CEC10	$1.40\times10^{+03}$	$2.46\times10^{+03}$	$2.62\times10^{+03}$	$4.21\times10^{+03}$	$1.11\times10^{+04}$	$1.32\times10^{+04}$	$1.35\times10^{+04}$	$1.60\times10^{+04}$	**$-2.53\times10^{+02}$**	**$2.05\times10^{+02}$**
CEC11	$1.82\times10^{+02}$	$4.27\times10^{+02}$	$1.24\times10^{+03}$	**$1.77\times10^{+03}$**	$8.90\times10^{+02}$	$1.00\times10^{+03}$	$1.59\times10^{+03}$	$1.82\times10^{+03}$	**$6.85\times10^{+01}$**	**$3.10\times10^{+02}$**
CEC12	**$3.63\times10^{+02}$**	**$5.60\times10^{+02}$**	$1.45\times10^{+03}$	$1.88\times10^{+03}$	$1.18\times10^{+03}$	$1.31\times10^{+03}$	$1.78\times10^{+03}$	$2.07\times10^{+03}$	$3.81\times10^{+02}$	$5.73\times10^{+02}$
CEC13	**$6.54\times10^{+02}$**	**$8.23\times10^{+02}$**	$1.24\times10^{+03}$	$2.17\times10^{+03}$	$1.31\times10^{+03}$	$1.44\times10^{+03}$	$1.99\times10^{+03}$	$2.21\times10^{+03}$	$8.31\times10^{+02}$	$1.31\times10^{+03}$
CEC14	$1.39\times10^{+04}$	**$1.74\times10^{+04}$**	$2.10\times10^{+04}$	$2.46\times10^{+04}$	$3.06\times10^{+04}$	$3.15\times10^{+04}$	$2.94\times10^{+04}$	$3.10\times10^{+04}$	**$1.17\times10^{+04}$**	$1.75\times10^{+04}$
CEC15	**$1.35\times10^{+04}$**	**$1.87\times10^{+04}$**	$2.17\times10^{+04}$	$2.44\times10^{+04}$	$3.08\times10^{+04}$	$3.18\times10^{+04}$	$2.58\times10^{+04}$	$2.97\times10^{+04}$	$1.62\times10^{+04}$	$2.62\times10^{+04}$
CEC16	$2.07\times10^{+02}$	$2.05\times10^{+02}$	$2.03\times10^{+02}$	**$2.04\times10^{+02}$**	$2.04\times10^{+02}$	$2.05\times10^{+02}$	$2.04\times10^{+02}$	**$2.04\times10^{+02}$**	**$2.02\times10^{+02}$**	$2.05\times10^{+02}$
CEC17	$1.44\times10^{+03}$	$1.55\times10^{+03}$	$2.44\times10^{+03}$	$3.06\times10^{+03}$	$3.03\times10^{+03}$	$3.27\times10^{+03}$	$3.50\times10^{+03}$	$3.82\times10^{+03}$	**$9.59\times10^{+02}$**	**$1.25\times10^{+03}$**
CEC18	$1.83\times10^{+03}$	**$2.00\times10^{+03}$**	$2.84\times10^{+03}$	$3.22\times10^{+03}$	$3.06\times10^{+03}$	$3.37\times10^{+03}$	$3.63\times10^{+03}$	$3.98\times10^{+03}$	**$1.68\times10^{+03}$**	$2.18\times10^{+03}$
CEC19	$7.50\times10^{+03}$	$2.57\times10^{+04}$	$1.35\times10^{+04}$	$3.36\times10^{+04}$	$8.59\times10^{+05}$	$1.52\times10^{+06}$	$1.42\times10^{+05}$	$2.90\times10^{+05}$	**$6.01\times10^{+02}$**	**$7.24\times10^{+03}$**
CEC20	**$6.60\times10^{+02}$**	$6.80\times10^{+02}$	$6.50\times10^{+02}$	$6.40\times10^{+02}$	$6.30\times10^{+02}$	$6.70\times10^{+02}$	$6.80\times10^{+02}$	**$6.10\times10^{+02}$**	**$6.10\times10^{+02}$**	**$6.10\times10^{+02}$**
CEC21	$3.56\times10^{+03}$	$5.57\times10^{+03}$	$2.80\times10^{+03}$	$4.91\times10^{+03}$	$7.53\times10^{+03}$	$8.23\times10^{+03}$	$8.89\times10^{+03}$	$9.48\times10^{+03}$	**$1.13\times10^{+03}$**	**$1.36\times10^{+03}$**
CEC22	$1.60\times10^{+04}$	$2.15\times10^{+04}$	$2.70\times10^{+04}$	$2.96\times10^{+04}$	$3.23\times10^{+04}$	$3.31\times10^{+04}$	$3.24\times10^{+04}$	$3.37\times10^{+04}$	**$1.28\times10^{+04}$**	**$1.71\times10^{+04}$**
CEC23	**$1.71\times10^{+04}$**	**$2.35\times10^{+04}$**	$2.50\times10^{+04}$	$2.88\times10^{+04}$	$3.30\times10^{+04}$	$3.41\times10^{+04}$	$3.11\times10^{+04}$	$3.40\times10^{+04}$	$2.01\times10^{+04}$	$2.98\times10^{+04}$
CEC24	$1.46\times10^{+03}$	$1.49\times10^{+03}$	$1.67\times10^{+03}$	$2.03\times10^{+03}$	$1.61\times10^{+03}$	$1.62\times10^{+03}$	$1.71\times10^{+03}$	$1.74\times10^{+03}$	**$1.45\times10^{+03}$**	**$1.49\times10^{+03}$**
CEC25	$1.65\times10^{+03}$	$1.68\times10^{+03}$	$1.83\times10^{+03}$	$1.94\times10^{+03}$	$1.75\times10^{+03}$	$1.76\times10^{+03}$	$1.82\times10^{+03}$	$1.84\times10^{+03}$	**$1.61\times10^{+03}$**	**$1.65\times10^{+03}$**

CEC26	$1.93\times10^{+03}$	$1.75\times10^{+03}$	$1.87\times10^{+03}$	$1.93\times10^{+03}$	$1.91\times10^{+03}$	$1.91\times10^{+03}$	$1.87\times10^{+03}$	$1.90\times10^{+03}$	$1.74\times10^{+03}$	$1.79\times10^{+03}$
CEC27	$4.14\times10^{+03}$	$4.38\times10^{+03}$	$5.84\times10^{+03}$	$6.80\times10^{+03}$	$5.67\times10^{+03}$	$5.81\times10^{+03}$	$5.91\times10^{+03}$	$6.08\times10^{+03}$	$\mathbf{4.12\times10^{+03}}$	$4.44\times10^{+03}$
CEC28	$7.51\times10^{+03}$	$1.07\times10^{+04}$	$1.75\times10^{+04}$	$2.09\times10^{+04}$	$1.18\times10^{+04}$	$1.31\times10^{+04}$	$1.86\times10^{+04}$	$1.96\times10^{+04}$	$\mathbf{4.77\times10^{+02}}$	$\mathbf{8.53\times10^{+03}}$
win	8	9	8	9	8	9	5	6	--	--
lose	17	15	17	15	17	15	19	18	--	--
draw	3	4	3	4	3	4	4	4	--	--

Author Agreement

Title: "An Optimal Sensor Network Coverage Planning Based on Adaptive Crossover Mutation Differential Evolution"

This Author Agreement ("Agreement") is entered into on this [June 28, 2023] by and between:

[Thi-Kien Dao, 1]

[Trong-The Nguyen, 2]

[Zhenyu Meng, 3]

(hereinafter collectively referred to as the "Authors").

WHEREAS, the Authors have jointly authored a manuscript titled "An Optimal Sensor Network Coverage Planning Based on Adaptive Crossover Mutation Differential Evolution" (hereinafter referred to as the "Manuscript").

WHEREAS, the Authors have agreed to submit the Manuscript for publication.

WHEREAS, the Authors affirm that no conflict of interest exists in the submission of this Manuscript.

NOW, THEREFORE, in consideration of the mutual covenants and promises contained herein, the Authors agree as follows:

1. Submission and Approval: The Authors hereby submit the Manuscript for publication in the [Knowledge-Based Systems] or any other appropriate journal, as determined by the Authors.

2. Originality and Attribution: The Authors confirm that the Manuscript is an original work and does not infringe upon any copyright or proprietary right of any third party. The Authors further acknowledge that any sources utilized in the Manuscript have been appropriately cited and credited.

3. Indemnification: The Authors agree to indemnify and hold harmless the publisher, the journal, and their respective agents, employees, and representatives from any claims, suits, or actions arising out of any breach of this Agreement, including but not limited to claims of copyright infringement, plagiarism, or unauthorized use of third-party materials.

4. Amendments and Modifications: Any modifications, amendments, or changes to this Agreement must be agreed upon in writing by all the Authors.

5. Governing Law and Jurisdiction: This Agreement shall be governed by and construed in accordance with the laws of [Jurisdiction]. Any disputes arising under or in connection with this Agreement shall be subject to the exclusive jurisdiction of the courts of [Jurisdiction].

6. Entire Agreement: This Agreement constitutes the entire understanding between the Authors and supersedes any prior agreements or understandings, whether oral or written, relating to the subject matter hereof.

IN WITNESS WHEREOF, the Authors have executed this Author Agreement as of the date first above written.

[Thi-Kien Dao 1]: Kien ________ Date: _ June 28, 2023_

[Trong-The Nguyen 2]: Nguyen ___________ Date: __ June 28, 2023___

[Zhenyu Meng 3]: Meng _________ Date: ___ June 28, 2023____

Declaration of Interest Statement

Dear Editor:

We are Thi-Kien Dao, Trong-The Nguyen, and Zhenyu Meng are the authors of the manuscript titled: "An Optimal Sensor Network Coverage Planning Based on Adaptive Crossover Mutation Differential Evolution"

We are submitting a manuscript entitled " An Optimal Sensor Network Coverage Planning Based on Adaptive Crossover Mutation Differential Evolution, " for consideration for publication in the Knowledge-Based Systems journal.

Conflict of Interest: The Authors affirm that no conflict of interest exists in the submission of the Manuscript. The Authors declare that they have no financial or personal relationships that could influence the content or interpretation of the Manuscript. Should any potential conflicts of interest arise during the review or publication process, the Authors agree to disclose such conflicts promptly to the journal editor.

IN WITNESS WHEREOF, the Authors have executed this Author Agreement as of the date first above written.

[Thi-Kien Dao 1]: ________ Date: _ June 28, 2023_

[Trong-The Nguyen 2]: ___________ Date: __ June 28, 2023___

[Zhenyu Meng 3]: _________ Date: ___ June 28, 2023____

Kien

Nguyen

Meng