mathematics

MDPI

Article

MESDO: A Multi-Strategy Supply Demand Optimization for Global Optimization and Deployment of Wireless Sensor Network

Bowei Wang [1,†], Yuchen Yan [2,†], Lingxi Zhu [3], Shaojie Yin [4] and Yangjian Yang [4,*]

[1] Viterbi School of Engineering, University of Southern California, Los Angeles, CA 90007, USA; boweiwang2021@163.com
[2] College of Engineering, City University of Hongkong, Hong Kong 999077, China; yuchenyan3c@163.com
[3] School of Mechanical, Aerospace and Civil Engineering, The University of Manchester, Manchester M139PL, UK; lingxizhu0409@gmail.com
[4] Taizhou Institute, Zhejiang University, Taizhou 318000, China; yinshaojie@tzizju.cn
* Correspondence: yangfgh123@126.com
† These authors contributed equally to this work.

Abstract

To address the problems of the traditional Supply–Demand Optimization (SDO) algorithm in wireless sensor network (WSN) node deployment—such as blind search direction, weak global exploration capability, coarse boundary handling, and insufficient maintenance of population diversity—this paper proposes a Multi-Strategy Enhanced Supply–Demand Optimization algorithm (MESDO). The proposed MESDO is validated on the CEC2017 and CEC2022 benchmark test suites. The results demonstrate that MESDO achieves superior performance in unimodal, multimodal, hybrid, and composite function optimization: for unimodal functions, it enhances local exploitation precision via elite-guided search to quickly converge to optimal regions; for multimodal functions, the adaptive differential evolution operator effectively avoids local optima by expanding exploration scope; for hybrid and composite functions, the centroid-based opposition learning boundary control maintains stable population diversity, ensuring adaptability to complex solution spaces. These advantages enable MESDO to effectively avoid premature convergence. According to the Friedman test, MESDO ranks first on CEC2017 (d = 30), CEC2022 (d = 10), and CEC2022 (d = 20), with average rankings of 1.20, 1.67, and 1.33, respectively—significantly outperforming the second-ranked SDO (average rankings of 3.60, 3.25, and 3.83). Finally, MESDO is applied to WSN deployment optimization. Its average coverage rate (86.80%) exceeds that of SDO (84.41%) by 2.39 percentage points, while its minimum coverage (84.80%) is 21.21 percentage points higher than that of AOO (69.96%). Moreover, its standard deviation (8.1308×10^{-3}) is the lowest among all compared algorithms. The convergence curve reveals that MESDO achieves 82% coverage within 50 iterations, which is significantly faster than SDO (80 iterations) and IWOA (100 iterations). The node deployment distribution further shows that the generated nodes are uniformly distributed without coverage blind spots. In summary, MESDO demonstrates superior optimization accuracy, convergence speed, and stability in both function optimization and WSN deployment, providing a reliable and efficient approach for WSN deployment optimization.

Keywords: supply demand optimization; wireless sensor network; global optimization; elite-guided search; boundary control

check for updates

Academic Editors: Mihai Postolache, Li Zhou and Xinchao Zhao

Received: 4 November 2025
Revised: 16 November 2025
Accepted: 19 November 2025
Published: 20 November 2025

Citation: Wang, B.; Yan, Y.; Zhu, L.; Yin, S.; Yang, Y. MESDO: A Multi-Strategy Supply Demand Optimization for Global Optimization and Deployment of Wireless Sensor Network. *Mathematics* **2025**, *13*, 3727. https://doi.org/10.3390/math13223727

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

MSC: 68T20; 68W25; 68Q25

1. Introduction

With the rapid development of the Internet of Things (IoT), smart cities, environmental monitoring, and industrial automation, wireless sensor networks (WSNs)—as the core infrastructure for data acquisition and transmission—are gaining increasing importance. A WSN consists of numerous sensor nodes equipped with sensing, communication, and computing capabilities. These nodes can perform real-time monitoring, data aggregation, and remote transmission of physical environmental parameters (e.g., temperature, humidity, illumination, vibration) within a target area, providing fundamental data support for various intelligent applications. For instance, in environmental monitoring, WSNs can be deployed in forests, oceans, or remote mountainous regions to enable long-term, unattended observation of ecological changes [1,2]; in smart city construction, WSNs can be utilized for traffic flow monitoring, public safety alerts, and energy consumption management, thereby enhancing urban management efficiency [3]; in the industrial domain, WSNs can be embedded in production equipment and workshop environments to realize equipment condition monitoring and process optimization, driving the transformation toward Industry 4.0 [4,5].

As application demands continue to grow, the scale and coverage of WSN deployments have expanded rapidly, while performance requirements have become increasingly stringent. Sensor nodes are typically powered by batteries and often deployed in complex or harsh environments (e.g., outdoors, underground, or high-temperature and high-pressure industrial settings), making node replacement and maintenance both difficult and costly. Consequently, energy efficiency, coverage completeness, and network connectivity have become critical indicators that must be carefully considered during deployment. Moreover, WSN requirements vary significantly across different application scenarios [6]. For example, precision agriculture monitoring requires sensor nodes to be evenly distributed across farmland to ensure comprehensive data collection on crop growth conditions, whereas industrial equipment monitoring demands sensor nodes to be deployed closer to target devices to guarantee the accuracy and real-time performance of monitoring data [7,8]. These diverse and high-demand application contexts present new challenges and opportunities for the efficient deployment of WSNs.

Despite the broad application prospects of wireless sensor networks (WSNs), their deployment process still faces numerous challenges. First, in terms of coverage performance, random or grid-based node deployment often leads to uneven distribution, resulting in redundant nodes in some regions and sensing blind spots in others, thereby compromising data integrity. Second, since sensor nodes have limited and non-replenishable energy, unreasonable deployment may cause certain nodes to deplete their power prematurely due to frequent data forwarding, thus shortening the overall network lifetime. Third, node deployment must ensure network connectivity while minimizing deployment cost—an overly dense deployment increases interference and overhead, whereas a sparse deployment may create communication islands. Finally, traditional static deployment schemes struggle to adapt to dynamic factors such as node mobility and environmental changes, which can lead to network performance degradation. Therefore, achieving a comprehensive optimization that balances coverage, energy consumption, connectivity, and dynamic adaptability under limited resources remains a critical issue to be addressed in WSN deployment [9,10].

To address the aforementioned challenges in WSN deployment, traditional deployment methods—such as random deployment, grid-based deployment, and greedy

algorithms—are no longer sufficient to meet the performance requirements of complex scenarios. Random deployment relies entirely on probabilistic distribution and cannot guarantee adequate coverage or network connectivity. Grid deployment, although capable of achieving uniform distribution, lacks flexibility and is difficult to apply to irregular regions or dynamic environments. Greedy algorithms, while capable of finding locally optimal solutions, tend to become trapped in local optima and fail to achieve global optimization of deployment performance. Against this backdrop, intelligent optimization algorithms have emerged as effective tools for WSN deployment optimization, owing to their strong global search capability, self-adaptive adjustment mechanisms, and multi-objective optimization ability [11,12].

Intelligent optimization algorithms are optimization methods inspired by natural phenomena, biological behaviors, or mathematical models. They can efficiently search for optimal solutions within complex solution spaces without relying on explicit mathematical formulations of the problem, thereby exhibiting strong generality and robustness [13]. When applied to WSN deployment optimization, intelligent optimization algorithms treat the sensor node coordinates as optimization variables, transforming objectives such as maximizing coverage, minimizing energy consumption, and maximizing network connectivity into optimization objective functions. Through iterative search processes, these algorithms identify deployment schemes that satisfy multiple objectives simultaneously [14]. For example, intelligent optimization algorithms can optimize the spatial distribution of sensor nodes to reduce coverage blind spots and node redundancy, balance energy consumption among nodes, and maintain network connectivity. Consequently, under limited node quantity and energy constraints, they can achieve global optimization of WSN deployment performance [12].

Swarm intelligence algorithms represent an important branch of intelligent optimization algorithms, inspired by the collective behaviors observed in biological populations. For instance, the Particle Swarm Optimization (PSO) algorithm [15] simulates the flight and foraging behavior of bird flocks, where each particle represents a candidate solution. The particles update their velocity and position through interaction with others, thereby searching for the optimal solution. The Ant Colony Optimization (ACO) algorithm [16] mimics the cooperative foraging behavior of ants, in which pheromone accumulation and evaporation guide the search toward optimal paths. The Grey Wolf Optimizer (GWO) [17] simulates the social hierarchy and hunting mechanism of grey wolves, where the α, β, and δ wolves guide the pack to encircle and attack the prey (optimal solution). The Whale Optimization Algorithm (WOA) [18] s inspired by the bubble-net hunting strategy of humpback whales, employing three mechanisms—encircling prey, spiral updating, and random search—to perform optimization. In recent years, numerous novel swarm intelligence algorithms have been developed. Examples include the Snake Optimizer (SO) [19], inspired by snakes' foraging and reproductive behaviors; the Sparrow Search Algorithm (SSA) [20], based on the foraging and anti-predation behaviors of sparrows; and the Secretary Bird Optimization Algorithm (SBOA) [21], derived from the predatory behaviors of secretary birds. Similarly, the Grasshopper Optimization Algorithm (GOA) [22] models the social and swarming behaviors of grasshoppers, while the Remora Optimization Algorithm (ROA) [23] is inspired by the symbiotic relationship between remoras and their hosts during foraging. The Philoponella Prominens Optimizer (PPO) [24] simulates the unique mating, escape, and cannibalistic behaviors of Philoponella prominens. The Black Widow Optimization Algorithm (BWO) [25] draws inspiration from the distinctive reproductive behavior of black widow spiders, and the Dung Beetle Optimizer (DBO) [26] s modeled after dung beetles' rolling, dancing, foraging, stealing, and reproductive behaviors. More recently, a physics-inspired approach known as the Kirchhoff's Law Algorithm (KLA) [27]

has been proposed, leveraging principles from circuit theory—particularly Kirchhoff's Current Law (KCL)—to guide the optimization process.

For WSN deployment problems under different environmental conditions, researchers have proposed a variety of dynamic intelligent algorithm-based deployment strategies. For example, Yuan Yuting et al. proposed an Adaptive Hybrid Differential Grey Wolf Optimization (AHDGWO) algorithm to address issues such as large coverage blind spots and uneven node distribution in traditional WSN deployments, thereby improving two-dimensional coverage performance [28]. Xinyi Chen et al. introduced a hybrid Butterfly–Beluga Whale Optimization (NHBBWO) algorithm incorporating a dynamic secondary parameter adaptation strategy to enhance WSN coverage optimization [29]. Yong Zhang et al. developed an Adaptive Learning Fruit Fly Optimization Algorithm (FOA) to optimize node coverage and energy balance in both two-dimensional and complex three-dimensional environments [30]. Zhang Zhaohui et al. proposed a novel Semi-Fixed Clustering Algorithm (SFC-QL-IACO) aimed at maintaining energy balance within WSNs [31]. Xiaoyang Liu et al. designed a Hybrid Improved Compressed Particle Swarm Optimization (HICPSO) algorithm composed of a linearly decreasing inertia weight, compressed velocity vector, Gaussian-based population variation, and an optimal boundary selection mechanism for WSN node optimization [32]. Jiaming Wang et al. proposed an Improved Coverage Optimization Algorithm (ATSSA) based on an enhanced Salp Swarm Algorithm, aiming to improve network coverage while reducing node movement energy consumption during redeployment [33]. Although intelligent algorithms have achieved remarkable progress in WSN deployment optimization, an analysis of existing studies reveals that current methods still exhibit several limitations, which constrain their effectiveness in complex deployment scenarios.

The Supply–Demand Optimization (SDO) algorithm [34], inspired by the economic theory of supply–demand equilibrium, simulates the periodic fluctuation and balancing process of commodity prices and quantities to perform intelligent optimization. SDO is characterized by fast convergence and conceptual simplicity; however, it also suffers from several inherent limitations. Specifically, its equilibrium price selection relies on random probabilities or the population mean, without leveraging historical high-quality solutions, resulting in blind search directions and a tendency to fall into local optima in later iterations [35]. Furthermore, the algorithm updates solutions linearly via the supply–demand function, lacking a comprehensive global exploration mechanism. As equilibrium points are computed solely based on the population mean and no dynamic population updating strategy is adopted, population diversity rapidly diminishes, leading to a high risk of premature convergence. In addition, the boundary-handling mechanism of SDO simply resets out-of-bounds solutions randomly, discarding valuable positional information and ignoring the population distribution, thereby causing low information utilization efficiency [36,37].

To overcome the limitations of existing intelligent algorithms in WSN deployment—namely the imbalance between global exploration and local exploitation, insufficient population diversity, inefficient boundary control, and poor dynamic adaptability—this study proposes a Multi-Strategy Enhanced Supply–Demand Optimization (MESDO) algorithm for global optimization and node deployment in WSNs. The MESDO algorithm integrates three core enhancement strategies to improve its search efficiency, robustness, and adaptability. First, an elite-guided search strategy is designed to dynamically construct an elite pool and use elite solutions to guide the selection of equilibrium prices, thereby maintaining a balance between search precision and population diversity. Second, an adaptive differential evolution operator is introduced to dynamically adjust the mutation factor and incorporate guidance from the global best solution, enhancing the algorithm's global exploration capability. Finally, a centroid-based opposition learning boundary control strategy is proposed, which

corrects out-of-bounds solutions with reference to the population centroid—preserving effective positional information and avoiding invalid searches.

To further clarify the differences between this study and existing achievements and highlight the innovations, Table 1 summarizes the core features of representative intelligent optimization algorithms used for Wireless Sensor Network (WSN) deployment optimization. The comparison is conducted across key dimensions such as strategy design, search capability, and intuitively demonstrates the advantages of this research.

Table 1. Comparison of proposed MESDO with existing algorithms for WSN deployment.

Algorithm	Core Strategy	Global Exploration	Local Exploitation	WSN Deployment Advantage
PSO	Particle velocity-position update	Medium	Medium	Simple implementation
GWO	Wolf pack hierarchy hunting	Medium	High	Good local convergence
WOA	Bubble-net hunting	High	Medium	Strong multimodal optimization
SDO	Supply-demand equilibrium	Low	Medium	Fast initial convergence
AHDGWO	Hybrid differential evolution	Medium	High	Improved 2D coverage
NHBBWO	Dynamic parameter adaptation	Medium	Medium	Balanced coverage optimization
ATSSA	Enhanced salp swarm behavior	Medium	High	Reduced movement energy consumption
MESDO	Elite-guided search + Adaptive DE + Centroid-based opposition learning	High	High	High coverage, fast convergence, strong stability

The main contributions of this work are as follows:

(1) A multi-strategy enhanced supply-demand optimization algorithm (MESDO) is proposed, based on an elite-guided search strategy, an adaptive differential evolution operator strategy, and a centroid-based opposition learning boundary control strategy;
(2) The overall performance of MESDO is comprehensively evaluated on the CEC2017 and CEC2022 benchmark suites;
(3) A mathematical model for WSN deployment is constructed, and MESDO is applied to solve it, with comparative analysis against other algorithms demonstrating MESDO's effectiveness in WSN scenarios.

The remaining chapters of this paper are organized as follows: Section 2 elaborates on the principles of the traditional supply-demand optimization (SDO) algorithm and introduces the design logic and mathematical models of the three core enhancement strategies of MESDO; Section 3 provides a comprehensive validation of MESDO through numerical experiments on the CEC2017 and CEC2022 benchmark suites; Section 4 constructs a mathematical model for WSN deployment and compares the deployment performance of MESDO with other algorithms through simulation experiments, evaluating its advantages in practical applications from the perspectives of coverage, convergence curves, and node distribution; Section 5 concludes the study, analyzes its limitations, and discusses future research directions.

2. Supply Demand Optimization (SDO) and the Proposed Methodology

2.1. Supply Demand Optimization (SDO)

The supply-demand optimization (SDO) algorithm was first proposed by Zhao et al. in 2020, inspired by economic theory. In this theory, the price and quantity of a commodity undergo cyclical fluctuations and gradually stabilize at their respective equilibrium points. In a market economy, the supply-demand mechanism forms the core theory for price determination. Specifically, the supply relationship of producers determines the quantity of a commodity in the next period q_{t+1}, which depends on the current commodity price p_t, and can be expressed as $q_{t+1} = f(p_t)$, where f represents a linear supply function. Therefore, when the current price of a commodity rises in the market, the supply quantity in the next period will also increase, indicating that f is a monotonically increasing function.

In addition, consumers' demand relationship determines the price p_{t+1} of the commodity in the next period, which depends on the quantity q_{t+1} of the commodity in the same period and can be expressed as $p_{t+1} = g(q_t)$, where g represents a linear demand function. It can be seen from this that when the quantity of the commodity increases, its price in the market will decrease, so g is a decreasing function. In the process of fluctuation, these two functions will intersect at the equilibrium point $P(x_0, y_0)$. Among them, x_0 and y_0 represent the equilibrium price and equilibrium quantity of the commodity, respectively. Therefore, the supply function f and the demand function g can be expressed as [34,36]:

$$\begin{aligned} q_{t+1} - q_0 &= a \cdot (p_t - p_0) \\ p_{t+1} - p_0 &= -b \cdot (q_{t+1} - q_0) \end{aligned} \tag{1}$$

in the equation, t represents time, and a and b are the linear coefficients.

To describe the changes in commodity prices and quantities, two matrices are introduced. When the demand function g is less than the supply function f, the fluctuation amplitude will be small. After a period of time, the quantity and price curves of commodities will stabilize over time and converge to the equilibrium point (x_0, y_0). At this time, it is in a stable mode. However, when the supply function f is less than the demand function g, the fluctuation amplitude will increase. After a period of time, the quantity and price curves of commodities will move away from the equilibrium point over time. The mathematical expression of this algorithm is constructed as follows: the market quantity is represented by n, each market contains d different commodities, and each type of commodity has specific quantities and prices. As variables, each candidate solution and possible candidate solution are, respectively, composed of d commodity prices and d commodity quantities. As shown in Equation (2), there are stable mode and unstable mode. In the stable mode, the demand function g is less than the supply function f, as shown in [34,36].

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} y_1^1 \; y_1^2 \ldots y_1^d \\ y_2^1 \; y_2^2 \ldots y_2^d \\ \vdots \; \vdots \; \vdots \; \vdots \\ y_n^1 \; y_n^2 \ldots y_n^d \end{bmatrix}, \; X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1^1 \; x_1^2 \ldots x_1^d \\ x_2^1 \; x_2^2 \ldots x_2^d \\ \vdots \; \vdots \; \vdots \; \vdots \\ x_n^1 \; x_n^2 \ldots x_n^d \end{bmatrix} \tag{2}$$

where $x_i (i = 1, \; 2, \cdots, \; n)$ and $y_i (i = 1, \; 2, \cdots, \; n)$ represent candidate solutions and potential candidate solutions, respectively, and are associated through a fitness function. The fitness function can be arranged into an array as shown in Equation (3) to represent the fitness values of these two vectors.

$$\begin{aligned} F_x &= [F_{x1}, F_{x2}, \cdots, F_{xn}]^T \\ F_y &= [F_{y1}, F_{y2}, \cdots, F_{yn}]^T \end{aligned} \tag{3}$$

The characteristic description of the equilibrium quantity vector y_0 is shown in Equation (4), where the transpose of the array is denoted by T.

$$\begin{aligned} N_i &= \left| F_{yi} - \binom{1}{n} \sum_{i=1}^{n} F_{yi} \right| \\ Q &= \frac{N_i}{\sum_{i=1}^{n} N_i} \\ y_0 &= y_k, \quad k = RWS(Q) \end{aligned} \tag{4}$$

In the equations, RWS denotes the roulette wheel selection method. According to Equation (6), the algorithm assumes that, based on the probabilities corresponding to the

equilibrium price vector x_0 in the price array, there is a 50% chance of selecting either the average of the price vectors or the current price vector:

$$\begin{aligned} M_i &= \left| F_{yi} - \left(\tfrac{1}{n}\right)\textstyle\sum_{i=1}^{n} F_{yi} \right| \\ Q &= \frac{M_i}{\sum_{i=1}^{n} M_i} \end{aligned} \tag{5}$$

$$x_0 = \begin{cases} r_1 \cdot \frac{\sum_{i=1}^{n} x_i}{n}, & if\ r < 0.5 \\ x_k,\ k = RWS(P), if\ r \geq 0.5 \end{cases} \tag{6}$$

where r_1 is a random number in the interval [0, 1].

Subsequently, as shown in Equations (7) and (8), in each market, the supply and demand functions are updated based on the equilibrium price vector x_0 and equilibrium quantity vector y_0 [34,36]:

$$y_i(t+1) = y_0 + \alpha \cdot (x_i(t) - x_0) \tag{7}$$

$$x_i(t+1) = x_0 + \beta \cdot (y_i(t) - y_0) \tag{8}$$

where $x_i(t)$ and $y_i(t)$ represent the price vector and quantity vector of the i-th commodity at time t, α and β are the supply and demand weights, respectively. By substituting Equation (7) into Equation (8), the demand equation can be rewritten as:

$$x_i(t+1) = x_0 - \alpha \cdot \beta \cdot (x_i(t) - x_0) \tag{9}$$

From Equation (9), it can be seen that the update of the commodity price vector is based on the current price vector while considering the equilibrium price vector. Therefore, by adjusting the values of α and β, different commodity price vectors can be generated, which, as shown in Equation (10), enable the exploration and exploitation processes [34,36].

$$\begin{aligned} \alpha &= \tfrac{2(T-t+1)}{T} \cdot sin(2\pi r) \\ \beta &= 2cos(2\pi r) \end{aligned} \tag{10}$$

where T denotes the maximum number of iterations, and r is a random number in the interval [0, 1].

2.2. Proposed Multi-Strategy Enhanced Supply Demand Optimization (MESDO)

2.2.1. Elite-Guided Search Strategy

In the standard SDO algorithm, the selection of the price equilibrium point during iterations relies on random probability or the population mean, lacking utilization of historical high-quality solutions. This results in a blind search direction, particularly in the later stages of iteration, making the algorithm prone to local optima or slower convergence. To address this, the MESDO algorithm introduces an elite-guided search strategy, which dynamically constructs an elite pool and uses elite solutions to guide the selection of price equilibrium points, achieving a balance between search accuracy and population diversity.

The elite-guided search strategy focuses on leveraging the guiding value of historical high-quality solutions. Its implementation logic is as follows: First, in each iteration, the price solutions of all markets (i.e., the current better solutions in each market) are extracted as elite candidate solutions, and their corresponding fitness values are recorded simultaneously. Next, the candidate solutions are sorted in ascending order based on their fitness values, and the top *ElitePoolSize* solutions (with *ElitePoolSize* defaulted to no more than 5 and not exceeding the population size to avoid an unreasonably large elite pool, this parameter was determined through parameter sensitivity analysis.) are selected to form a

dynamic elite pool. Finally, the guiding elite is selected based on random probability—with a 10% chance, the mean of all solutions in the elite pool is chosen (to ensure exploration of globally high-quality regions), and with a 90% chance, a single solution is randomly selected from the elite pool (to ensure deep exploitation of locally high-quality regions). The selected guiding elite is defined as $CurrentElite$.

During the price equilibrium point selection stage, to fully leverage the guidance of elite solutions, the MESDO algorithm adjusts the original SDO logic: as showed in Figure 1, when the random probability is greater than 0.3, $CurrentElite$ is directly used as the price equilibrium point $CommPriceEqu$, guiding the generation of new solutions toward the high-quality region where the elite resides; when the random probability is no greater than 0.3, the original SDO selection logic is retained (i.e., 50% chance to select the population price mean, 50% chance to select a single price solution based on fitness probability), avoiding over-reliance on elite solutions that could reduce population diversity.

$$CurrentElite = \begin{cases} \frac{1}{ElitePoolSize} \sum_{m=1}^{ElitePoolSize} ElitePool(m,:), & if\ rand < 0.1 \\ ElitePool(k,:), k = [1,2,3,4,5], & if\ rand \geq 0.1 \end{cases} \tag{11}$$

where $ElitePool(m,:)$ represents the mth elite solution in the elite pool, $ElitePoolSize = 5$ is the fixed size of the elite pool, and $ElitePool(k,:)$ denotes a randomly selected individual from the elite pool. This formula, by averaging the elite solutions, reduces the interference from potential local optima of a single elite solution and ensures stable guidance toward globally high-quality regions.

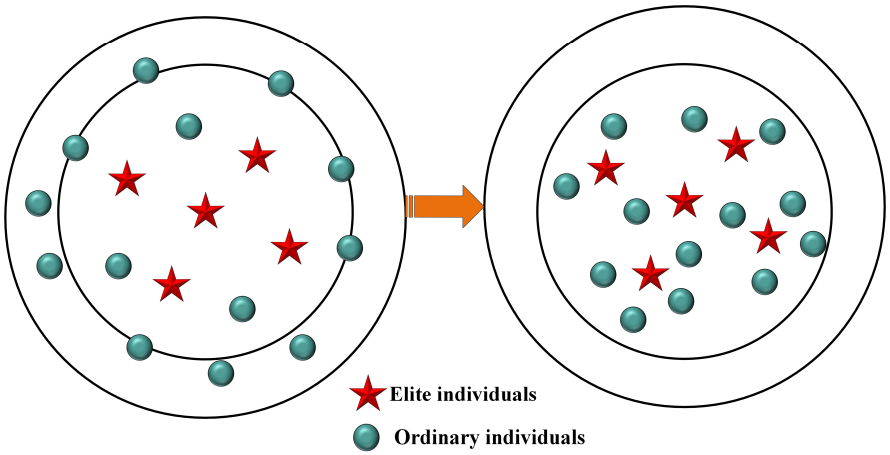


Figure 1. Elite-guided search strategy diagram.

2.2.2. Adaptive Differential Evolution Operator Strategy

The standard SDO algorithm updates quantity and price solutions solely through the supply and demand functions, lacking a global exploration mechanism for the solution space. In complex high-dimensional optimization problems, this can lead to overly strong local search, causing the algorithm to become trapped in local optima and making it difficult to escape inefficient solution regions. To address this, the MESDO algorithm introduces an adaptive differential evolution operator strategy, as illustrated in Figure 2, which dynamically adjusts the mutation factor and incorporates guidance from the global best solution to enhance the algorithm's global exploration capability and convergence efficiency.

$$x_i(t+1) = x_{r1}(t) + F \cdot (x_{best}(t) - x_{r2}(t)) \tag{12}$$

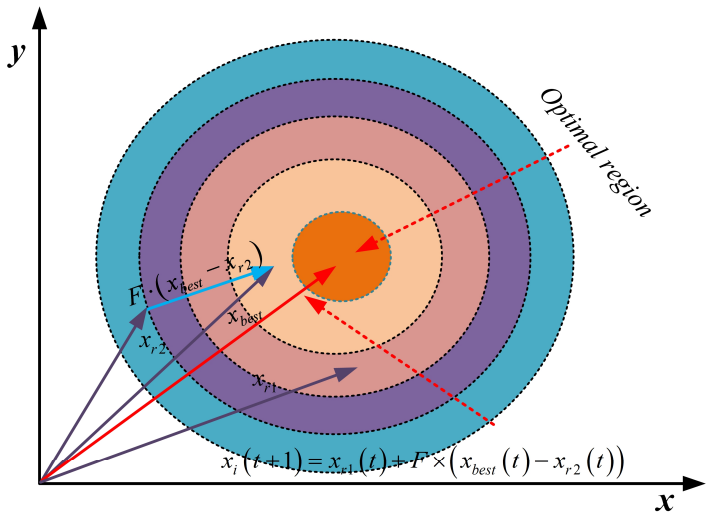


Figure 2. Schematic diagram of adaptive differential evolution strategy.

x_{r1} and x_{r2} denote two distinct individuals randomly selected from the population x, x_{best} represents the global best solution, and F is the mutation factor, which is calculated as follows:

$$F = F_{min} + (F_{max} - F_{min})\cdot(1 - t/T) \tag{13}$$

The parameter F is bounded by $F_{min} = 0.2$ and $F_{max} = 0.8$ Its value is decremented linearly across the total number of generations, T. This progressive scaling allows the search behavior to evolve from widespread exploration during initial phases to intensified exploitation near the conclusion.

This strategy introduces x_{best} to guide the mutation direction, allowing the mutated solutions to move closer to globally high-quality regions, thereby avoiding the blind randomness inherent in the traditional DE mutation operator.

2.2.3. Centroid-Based Opposition Learning Boundary Control Strategy

In the standard SDO algorithm, out-of-bounds solutions are handled by randomly resetting the offending dimensions within the feasible domain. This approach completely discards the original positional information of the out-of-bounds solutions and does not utilize population distribution characteristics, which can result in corrected solutions being far from high-quality regions and increase search randomness—leading to population disorder in the early iterations and potentially disrupting high-quality search trends in later stages, thereby reducing optimization accuracy.

To address this, MESDO proposes a centroid-based opposition learning boundary control strategy. The core idea is to use the current population centroid (mean) as a reference to perform opposition-based correction on out-of-bounds solutions. First, the centroid vector is calculated as the mean of all solution dimensions in the population; if the upper and lower bounds are scalars, they are expanded to match the population size across all dimensions. Next, all out-of-bounds dimensions (below the lower bound or above the upper bound) are identified, and each out-of-bounds dimension is corrected using the formula 2× centroid value minus the original out-of-bounds value," thereby preserving the positional relationship between the out-of-bounds solution and the centroid. Finally, the solution is truncated by the upper and lower bounds to ensure feasibility and prevent secondary boundary violations.

As shown in Figure 3, by adjusting the positions of individuals exceeding the boundaries and mapping them back into the valid search space, invalid searches are avoided and convergence speed is improved. This process is described by Equation (14).

$$x_i(t) = \begin{cases} \frac{(lb+x_C(t))}{2}, & if\ X_{i,j}(t) < lb \\ \frac{(ub+x_C(t))}{2}, & if\ X_{i,j}(t) > ub \end{cases} \tag{14}$$

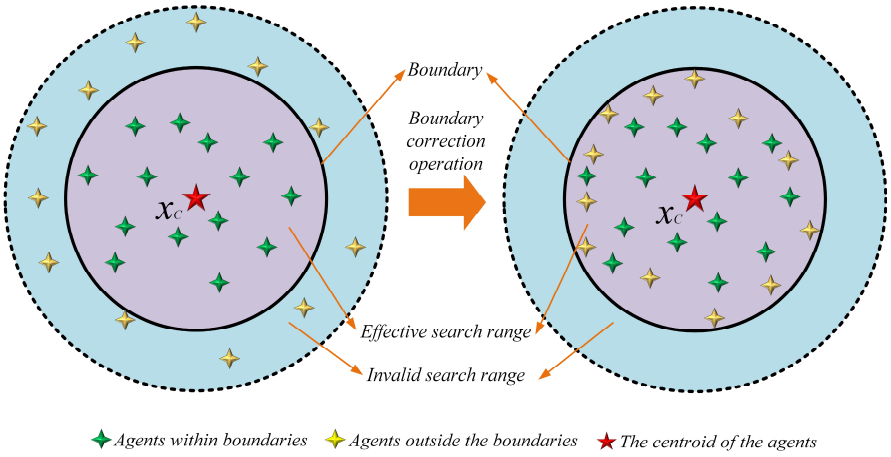


Figure 3. Schematic diagram of centroid-based opposition learning boundary control strategy.

Based on the above discussion, the pseudocode for MESDO is presented in Algorithm 1.

Algorithm 1. Pseudo-Code of MESDO.

Initialize the SDO parameters (population (n), d, ub, lb), Max iterations (T).
Initialize the vectors x_i, y_i and calculate their fitness value Fx_i, Fy_i.
while $t = 1 : T$ **do**
    Calculate the weights α and β.
    Calculate R_1, R_2, P, E, F
    for each market $i = 1, 2, \ldots, n$
        Determine the equilibriu m quantity y_0 by Equation (4).
        Determine the equilibriu m price x_0 by Equation (5).
        Update the commodity quantity vector y_i by Equation (7).
        Update the commodity quantity vector x_i by Equation (8).
        Centroid-based opposition learning boundary control strategy by Equation (14).
        Calculate their fitness values Fx_i and Fy_i.
        Apply adaptive differential evolution strategy to update solutions using Equations (12) and (13).
        if $Fy_i < Fx_i$ **do**
            replace x_i *by* y_i $(x_i = y_i)$.
        end if
    end for
    Update the best solution found so far x_{best}.
end while
Return X_{best}.

3. Numerical Experiments

3.1. Competitor Algorithms and Parameters Setting

In this section, the performance of the proposed MESDO algorithm is rigorously evaluated using two highly competitive numerical optimization benchmark suites: CEC2017 [38] and CEC2022 [39]. For comparison, multiple state-of-the-art algorithms are included, such as Velocity Pausing Particle Swarm Optimization (VPPSO) [40], MadDE [41], Improved multi-strategy adaptive Grey Wolf Optimization (IAGWO) [42], Improved Whale Optimization Algorithm (IWOA) [43], Status-based optimization Algorithm (SBO) [44], Animated Oat Optimization (AOO) [45], Holistic swarm optimization (HSO) [46], and Supply demand optimization (SDO) [34]. Configuration details for all benchmarked algorithms are presented in Table 2.

Table 2. Compare algorithm parameter settings.

Algorithms	Name of the Parameter	Value of the Parameter
VPPSO	$c_1, c_2,\ w, \alpha, N_1, N_2$	2, 2, 0.8, [1, 0], 0.15, 0.15
MadDE	$p, \lvert A\rvert, H$	0.18, 2.3, $10 \times d$
IAGWO	$v_{rand}(t), a,\ \omega, \theta$	$[-20, 20]$, [0, 2], [0.3, 0.9], 0.5
IWOA	$l,\ a$	$[-1, 1]$, Linear reduction from 2 to 1
SBO	I, P_S, R	$\{1, 2\}, [0, 1], 0.1$
AOO	K, θ	$[0.5, 1.5], [0, \pi]$
HSO	α	3
SDO	r_1, α, β	$[0,\ 1], [-2,\ 2], [-2,\ 2]$
MESDO	r_1, α, β, F	$[0,\ 1], [-2,\ 2], [-2,\ 2], [0.2,\ 0.5]$

3.2. Qualitative Analysis of MESDO

3.2.1. Analysis of the Population Diversity

The efficacy of population-based optimizers is heavily influenced by a fundamental property: population diversity, which reflects the distribution and dissimilarity of individual solutions [47]. Maintaining robust diversity is essential for preventing premature convergence and ensuring thorough exploration of the solution space, thereby underpinning search robustness. A quantitative evaluation of diversity within the MESDO framework, based on Equation (15) [48], is presented hereafter.

$$I_C(t) = \sqrt{\sum_{i=1}^{N} \sum_{d=1}^{D} \left(x_{idim}(t) - c_{dim}(t)\right)^2}, \tag{15}$$

A measure designated as $I_C(t)$ characterizes the variety within a population of size N. This metric is based on $c_{dim}(t)$, a quantity expressing the collective spread of all agents in relation to their central point at time t. The calculation for $c_{dim}(t)$ is provided in Equation (16), and it aggregates the coordinates $x_{idim}(t)$ for every member i in each dimension D.

$$c_{dim}(t) = \frac{1}{D} \sum_{i=1}^{N} x_{i,dim}(t). \tag{16}$$

Figure 4 illustrates the evolution curves of population diversity for the MESDO and the original SDO algorithms on 10 representative functions (F1, F5, F10, F12, F15, F17, F18, F22, F27, F30) from the CEC2017 benchmark suite (dimension $d = 30$). The diversity metric is calculated using Equation (15), measuring the dispersion of the population relative to the centroid $IC\ (t)$.

From the overall evolutionary trend, MESDO maintains significantly higher population diversity than SDO throughout the entire iteration process, with a smoother decline as iterations progress. This phenomenon results from the synergistic effects of MESDO's three

core strategies: the elite-guided search strategy, via a dynamic elite pool and probabilistic guidance mechanism, prevents the population from prematurely converging to local optima; the adaptive differential evolution operator strategy, through mutation guided by the global best solution, continuously injects new diversity into the population; and the centroid-based opposition learning boundary control strategy preserves the positional relationship between out-of-bounds solutions and the population centroid, preventing loss of effective diversity.

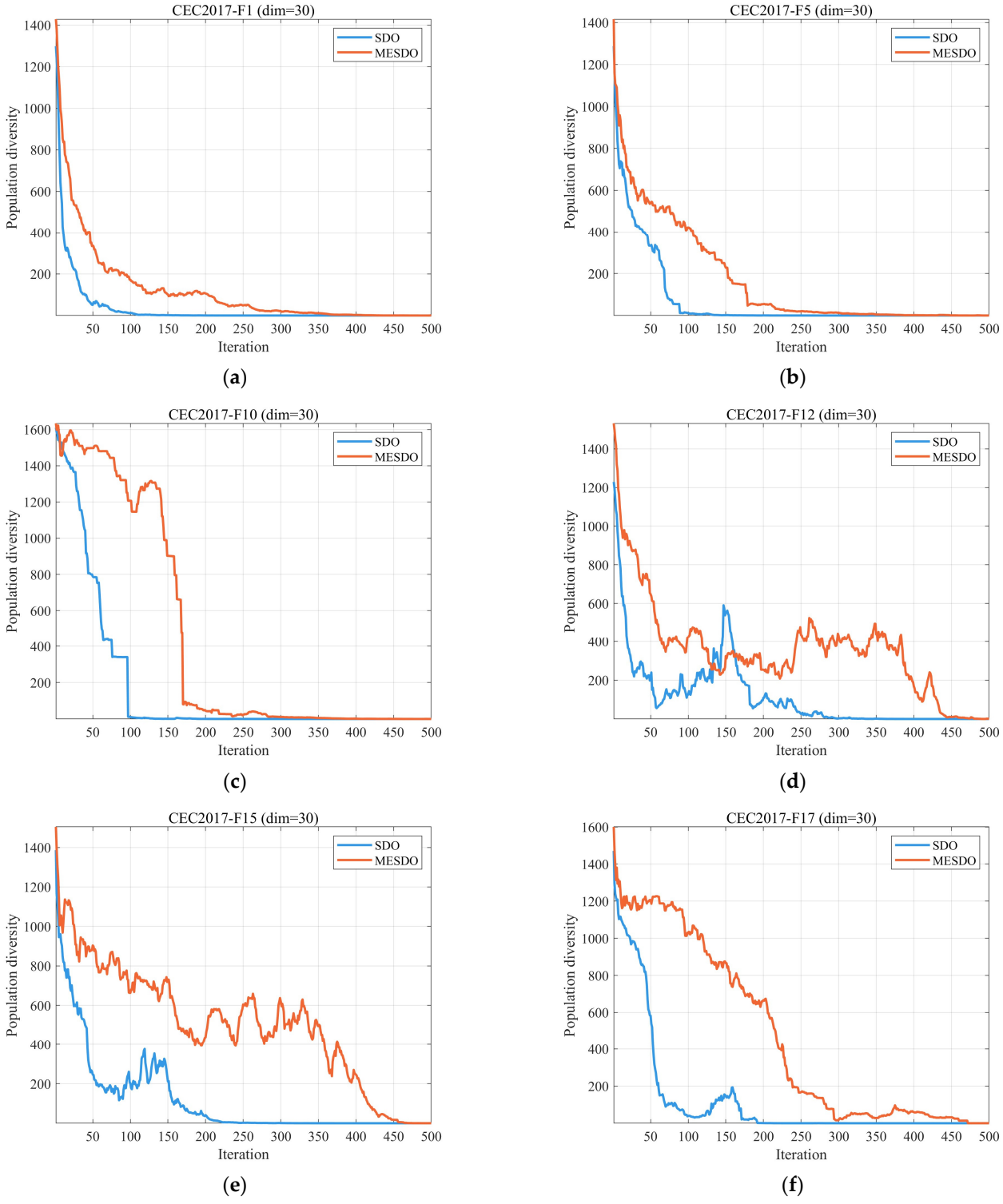


Figure 4. *Cont.*

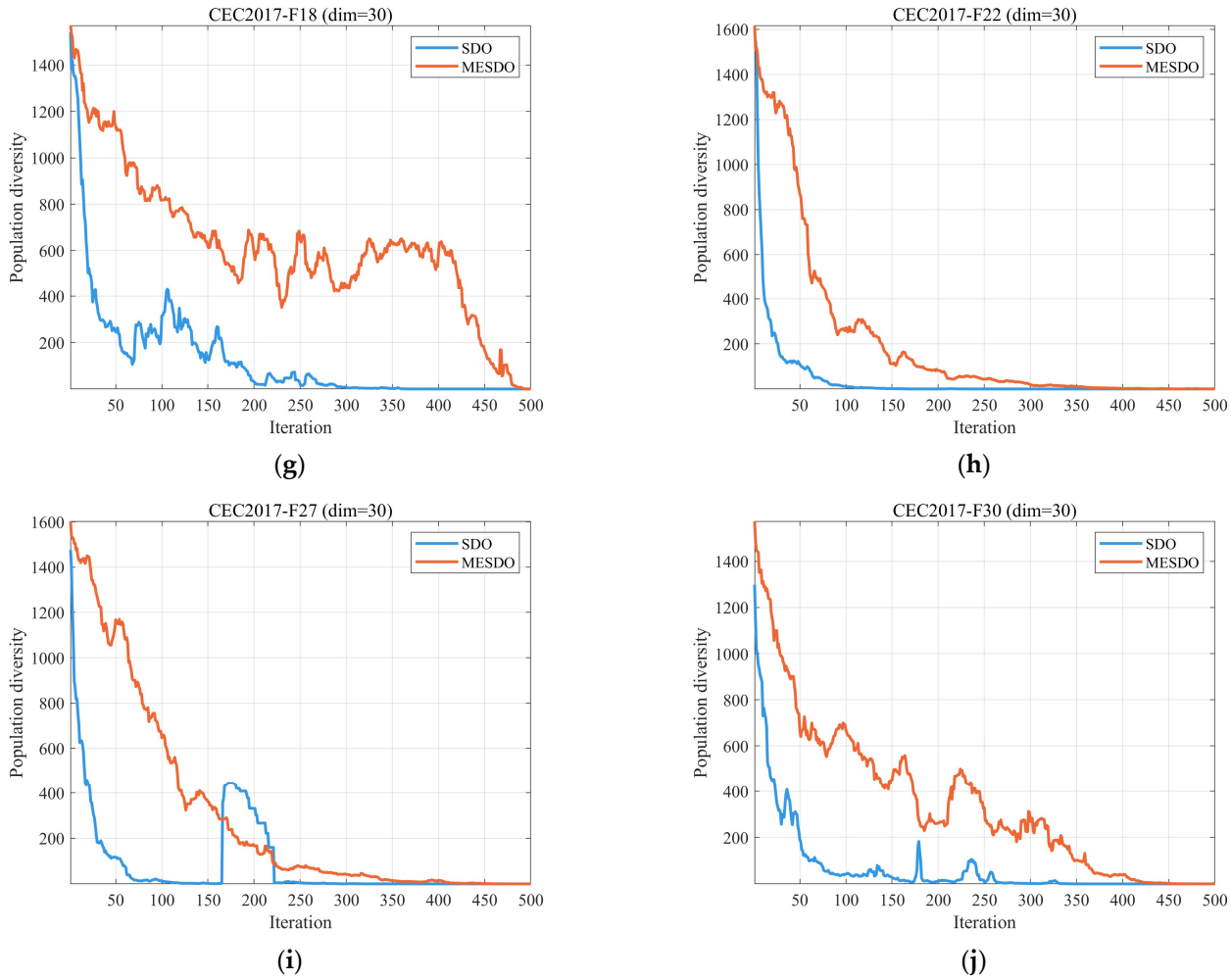


Figure 4. A comparative assessment of population distribution for the MESDO and SDO techniques.

Examining different types of functions in detail, for the unimodal function F1, MESDO's diversity curve consistently stays above that of SDO (with an average $IC\ (t)$ difference exceeding 300), demonstrating that it maintains global exploration capability even during local exploitation. For the multimodal functions F5, F10, and F12, MESDO's higher diversity effectively avoids the premature convergence commonly seen in SDO, especially in later iterations (after 300 generations), where SDO's diversity stabilizes $(IC\ (t) < 200)$ while MESDO maintains diversity above 400, supporting continued exploration of new high-quality solution regions. For the hybrid and composite functions F15, F18, F22, F27, and F30, MESDO's diversity advantage becomes even more pronounced; for instance, at 500 iterations on F30, MESDO's $IC\ (t)$ s approximately twice that of SDO, fully demonstrating its ability to balance exploration and exploitation in complex solution spaces.

Furthermore, the sudden jump in population diversity observed in subfigures (d) (CEC2017-F12 function) and (i) (CEC2017-F27 function) of Figure 4 is essentially an adaptive adjustment of the algorithm in response to the complex characteristics of the functions, rather than a convergence anomaly. This is because F12 and F27 are multimodal functions with a large number of local optima in the solution space, and these local optimal regions are widely distributed. The sudden increase in diversity around the 300th generation is primarily due to the active exploration of the adaptive differential evolution operator: at this mid-iteration stage, the mutation factor F remains around 0.4 (according to Equation (13), F decreases linearly with iterations), and the algorithm triggers a large-scale mutation operation guided by the global optimal solution. This operation breaks the previously trapped

local optimal convergence state of the population, allowing some individuals to escape the current local optimal region and explore new high-potential solution spaces, directly manifesting as a sudden increase in population diversity. This jump is a key mechanism for the algorithm to avoid premature convergence. The subsequent diversity curve continues to decline steadily, proving that the algorithm does not enter an unstable state.

In summary, the results in Figure 4 confirm that MESDO, through its strategic improvements, significantly enhances the maintenance of population diversity, providing a strong foundation for overcoming local optima and improving optimization accuracy in subsequent tasks.

3.2.2. Exploration and Exploitation Dynamics

Optimization algorithms operate through two complementary search paradigms: exploration, which entails a widespread investigation of the solution space to identify regions of potential, and exploitation, which involves a concentrated effort to hone high-quality solutions [49,50]. A bias towards the former can lead to inefficient, unproductive search; a dominance of the latter often traps the algorithm in local optima. Consequently, the strategic equilibrium between them is a decisive factor for success. The subsequent analysis evaluates this balance within the MESDO framework, applying the metrics defined in Equations (17) and (18) [48,50].

$$Exploration(\%) = \frac{Div(t)}{Div_{max}} \times 100\%, \tag{17}$$

$$Exploitation(\%) = \frac{|Div(t) - Div_{max}|}{Div_{max}} \times 100\%, \tag{18}$$

The normalized diversity at iteration t is defined as the ratio of the current diversity, $Div(t)$—computed from Equation (19)—to the maximum diversity value, Div_{max}, observed across all iterations.

$$Div(t) = \frac{1}{D}\sum_{d=1}^{D}\frac{1}{N}\sum_{i=1}^{N} \mid median(x_d(t)) - x_{id}(t) \mid. \tag{19}$$

Figure 5 presents the dynamic evolution of exploration and exploitation for the MESDO algorithm on representative functions (F1, F5, F16, F2, F26, F30) from the CEC2017 benchmark suite (dimension $d = 30$). The exploration rate and exploitation rate are calculated using Equations (17) and (18), respectively, measured by the ratio of the diversity metric $Div(t)$ (defined in Equation (19), reflecting the dispersion of each dimension in the population relative to the median) to the maximum diversity Div_{max}.

From the evolution curves, it can be observed that MESDO achieves an adaptive dynamic balance between exploration and exploitation throughout the entire iteration process. This core regulatory mechanism stems from the synergistic effects of the three strategies: the adaptive differential evolution (DE) operator strategy maintains a high exploration rate (60–90%) during the early iterations (1–150 generations) through the linear decrease in the mutation factor FFF (as shown in Equation (13), where F decreases from F_{max} to F_{min}), supporting efficient global exploration of the solution space. For example, in function F1, the exploration rate exceeds 85% in the early iterations, effectively covering multimodal regions.

The elite-guided search strategy, during the middle iterations (150–300 generations), gradually increases the exploitation rate (from 30% to 60%) by guiding the selection of price equilibrium points with a 90% probability of choosing a single solution from the elite pool, enabling deep exploitation of high-quality solution regions. Taking function F5 as an

example, the exploitation rate steadily rises during this stage, while the exploration rate remains above 30%, preventing premature convergence to local optima.

The centroid-based opposition learning boundary control strategy, in the later iterations (300–500 generations), preserves basic population diversity by correcting out-of-bounds solutions, maintaining the exploration rate at 20–40% and the exploitation rate at 60–80%. This ensures precise convergence toward the global optimum while preventing excessive loss of population diversity.

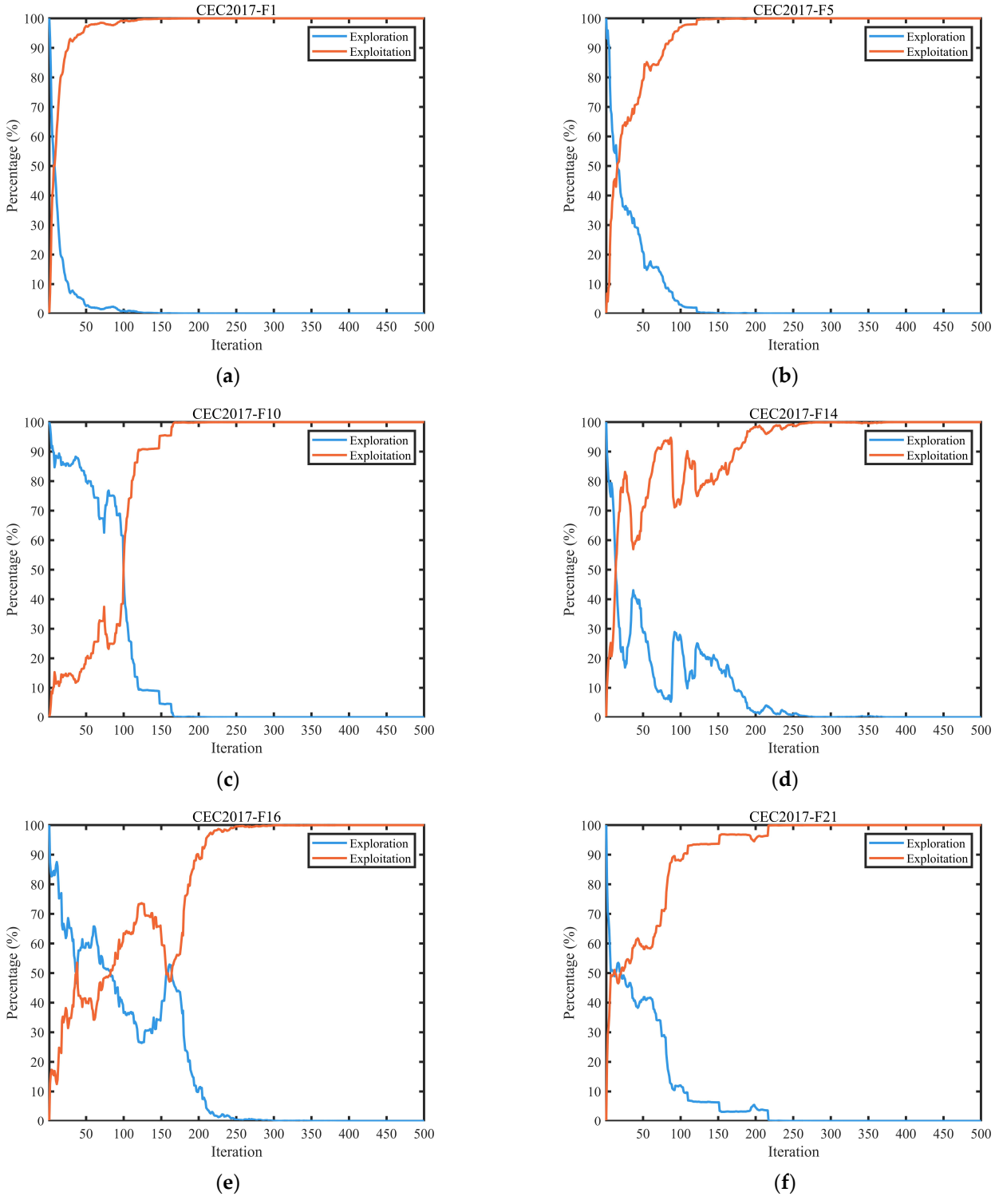


Figure 5. *Cont.*

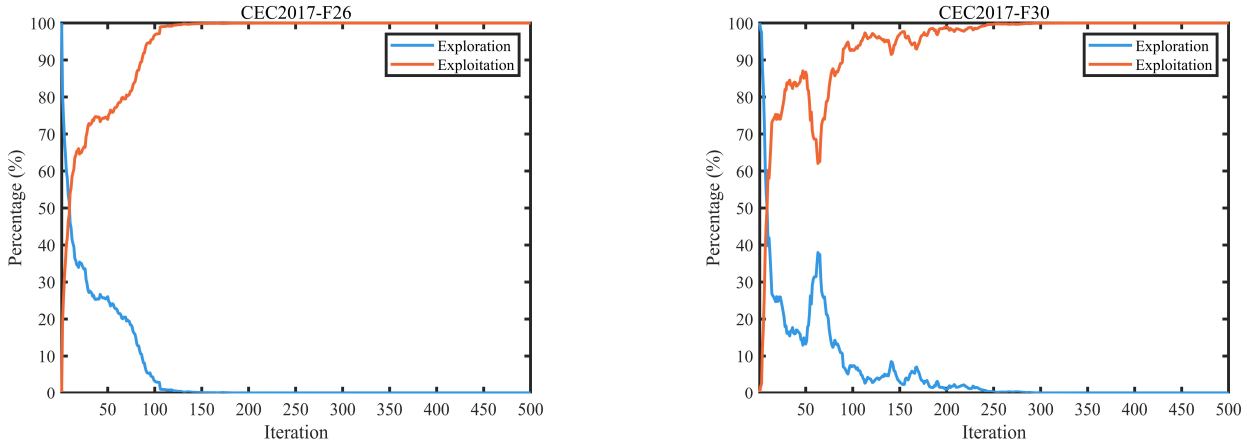


Figure 5. Examination of the global search and local refinement capabilities in MESDO.

3.2.3. Performance Impact of the Proposed Strategy

A methodical ablation framework was established to dissect the individual efficacy and synergistic interactions of the three core innovations: the Elite-guided search strategy (S1), Adaptive differential evolution operator strategy (S2), and Centroid-based opposition learning boundary control strategy (S3). Utilizing the CEC2017 benchmark (d = 30), we evaluated five algorithmic variants: the standard SDO, three partial implementations (SDO-S1, SDO-S2, SDO-S3), and the complete MESDO. This design allows for clear efficacy attribution, with the comparative results detailed in Figures 6 and 7.

The convergence curves in Figure 6 indicate that while each of the three strategies enhances performance differently, their combination produces a significant synergistic effect. For unimodal functions, SDO-S1 (integrating S1) accelerates convergence by guiding the search direction with elite solutions, entering high-quality regions earlier than the original SDO. SDO-S2 (integrating S2), leveraging mutation operations guided by the global best solution, demonstrates stronger global exploration in the early iterations, quickly covering critical areas of the solution space. SDO-S3 (integrating S3), through boundary control, reduces ineffective searches, resulting in significantly more stable convergence in the later iterations compared with the original SDO, avoiding performance fluctuations caused by random resets of out-of-bounds solutions. The complete MESDO algorithm, benefiting from the synergy of the three strategies, shows a convergence curve with no noticeable oscillations and consistently outperforms all comparison algorithms, achieving optimization accuracy far superior to both the original SDO and single-strategy variants. This demonstrates the efficient coordination of "global exploration—local exploitation—stable convergence".

For multimodal functions, SDO-S2 contributes most to escaping local optima, effectively mitigating the original SDO's tendency to become trapped. MESDO further enhances this capability by combining S1's precise localization of high-quality regions with S2's global exploration, achieving better optimization accuracy than any single-strategy variant. In composite functions, SDO-S3's boundary control plays a key role in maintaining population diversity, preventing the sharp decline in diversity observed in the original SDO within complex solution spaces. MESDO, through the synergy of S3 with S1 and S2, maintains stable performance improvements even in the later iterations, ultimately achieving superior optimization results compared with all other algorithms.

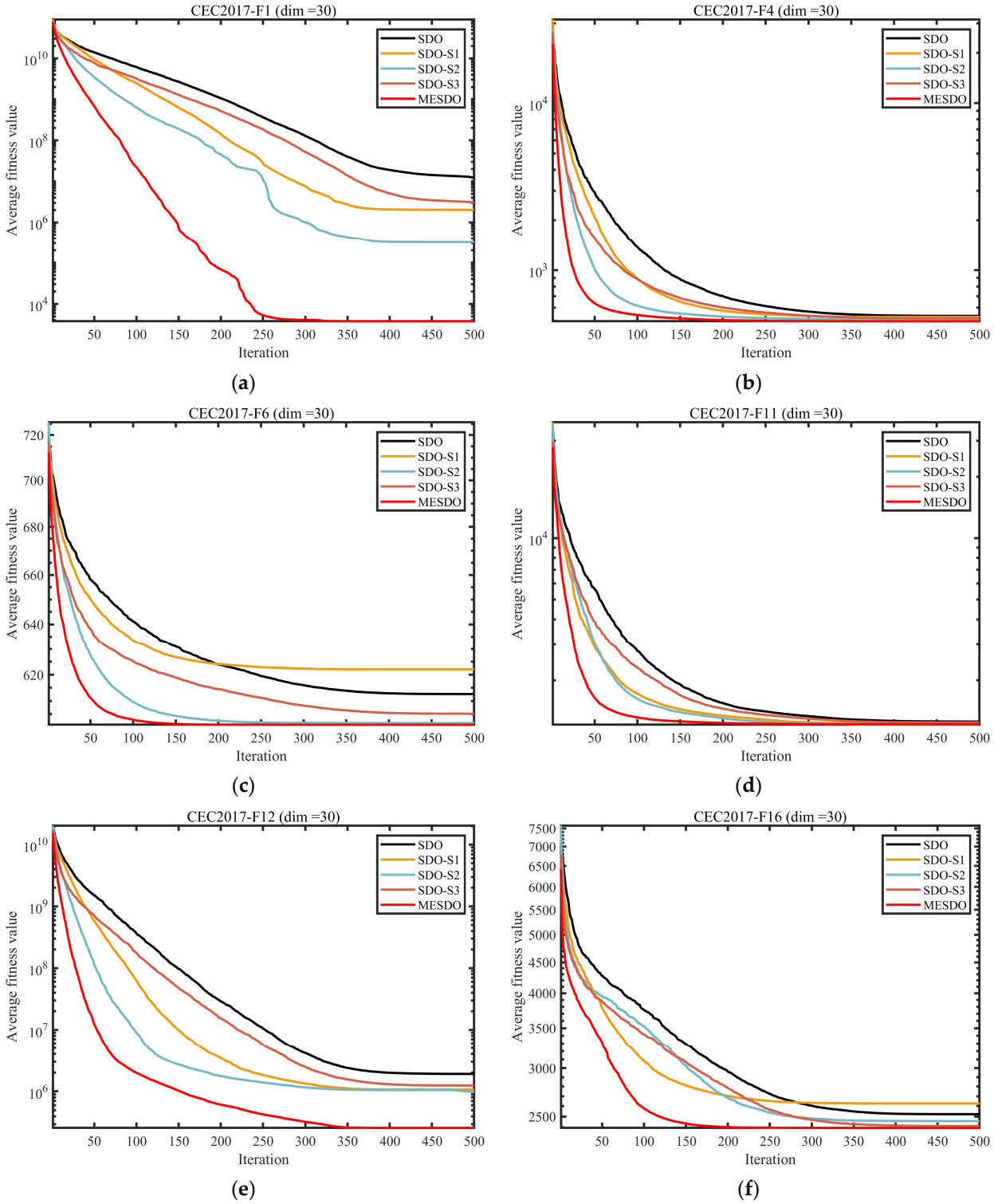


Figure 6. *Cont.*

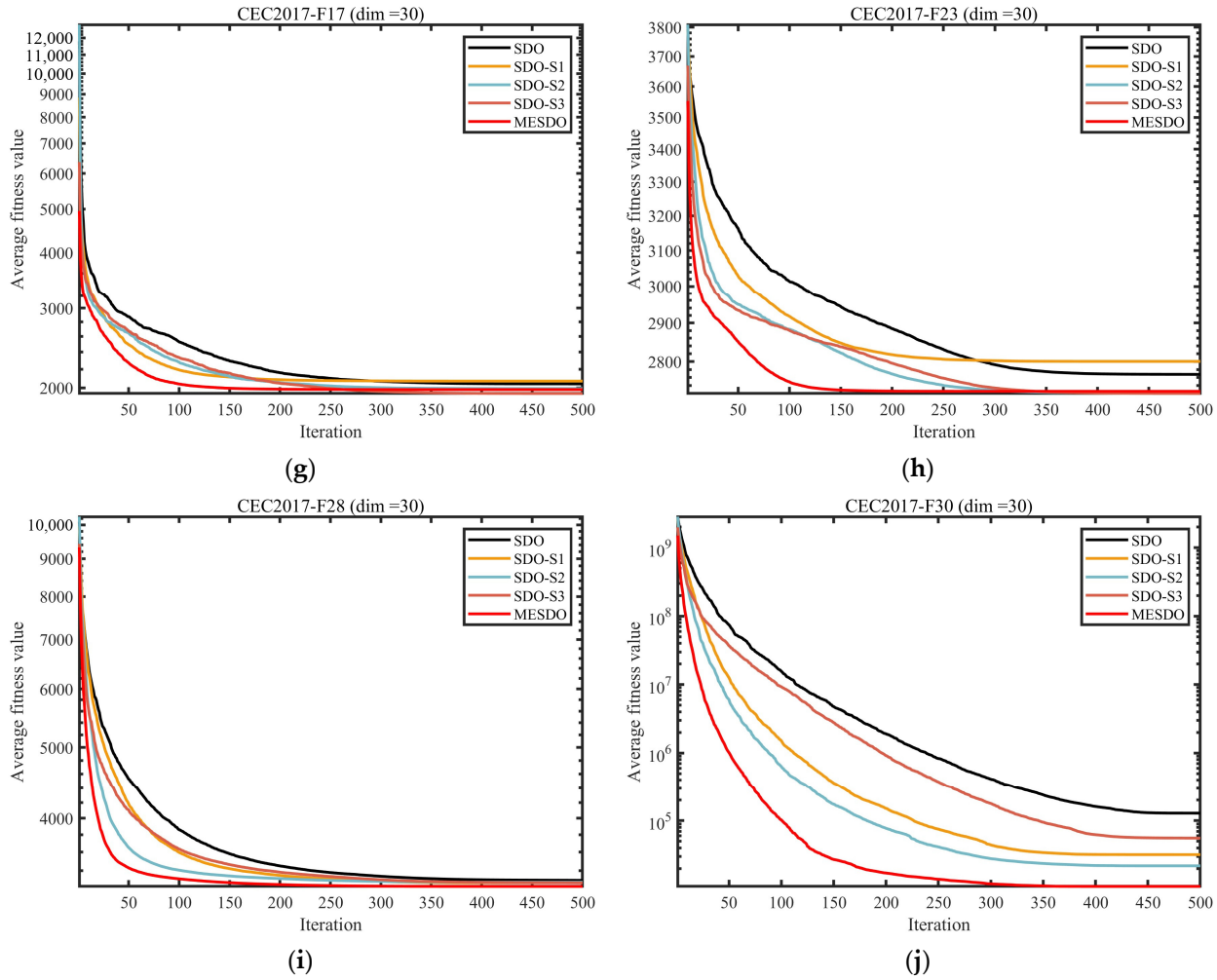


Figure 6. Efficacy analysis of different strategic enhancements.

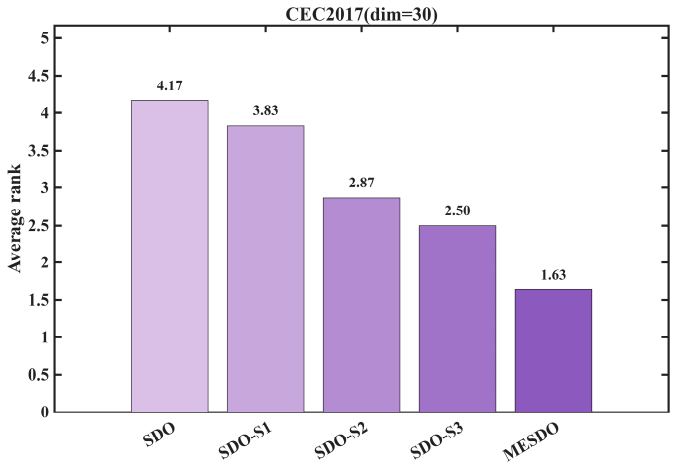


Figure 7. Comparative performance ranking of SDO variants with different enhancements.

The average ranking results in Figure 7 further quantify the overall gains of each strategy. The original SDO ranks relatively low, whereas SDO-S1, SDO-S2, and SDO-S3 all show improved average rankings, indicating that each single strategy can enhance algorithm performance from different perspectives. Among them, SDO-S2 provides the most significant independent gain, with a greater improvement in average ranking than SDO-S1 and SDO-S3. The complete MESDO algorithm, however, achieves the best average

ranking across all test functions, consistently placing first, demonstrating the advantage of the three strategies' synergy. This ranking shows that the strategies do not simply superimpose their effects, but form an efficient cooperative mechanism: S1 addresses the blind search direction of the original SDO, providing precise guidance for S2's global exploration and avoiding ineffective searches; S2 compensates for SDO's insufficient global exploration, supporting S1's local exploitation with a broader base of high-quality solutions and expanding coverage of high-quality regions; S3 ensures population diversity through boundary corrections, providing a stable population environment for the continuous action of S1 and S2, preventing performance stagnation due to diversity loss.

In summary, the combined analysis of Figures 6 and 7 fully validates the effectiveness and synergistic value of MESDO's core strategies. A single strategy can improve algorithm performance in terms of search direction, exploration capability, and convergence stability, whereas the synergy of all three strategies produces an optimization effect of "1 + 1 + 1 > 3." This lays a solid strategic foundation for MESDO to achieve high optimization accuracy, fast convergence, and strong stability in complex optimization problems, and also provides a reference design idea for multi-strategy improvements in other optimization algorithms.

3.3. Performance Evaluation Using CEC2017 and CEC2022 Benchmark Problems

This section presents a comparative evaluation of MESDO with other benchmark algorithms using the CEC2017 and CEC2022 test suites. These benchmark sets comprise four categories of mathematical functions: unimodal, multimodal, hybrid, and composition functions. Multimodal functions, characterized by multiple local optima, are effective for testing an optimizer's exploration capability. In contrast, unimodal functions—featuring a single global optimum—primarily gauge exploitation performance. Hybrid and composition functions further examine an algorithm's capacity to escape local optima.

To ensure fairness and minimize random effects, all algorithms were configured with a population size of 30 and a maximum of 500 iterations. Each method was executed independently 30 times. The average (Ave) and standard deviation (Std) of the results are reported, with optimal values emphasized in bold. Experiments were performed on a Windows 11 platform with an AMD Ryzen 7 9700X 8-Core Processor (3.80 GHz), 48 GB RAM, and MATLAB 2024b. Convergence behavior and distribution characteristics of the algorithms are visually compared in Figures 8 and 9 through convergence curves and box plots, respectively.

Tables 3–5 (CEC2017, d = 30), 3 (CEC2022, d = 10), 4 (CEC2022, d = 20), and Figures 8 and 9 comprehensively analyze the performance advantages of MESDO compared with eight benchmark algorithms, including VPPSO, MadDE, IAGWO, IWOA, SBO, AOO, HSO, and SDO, across different test scenarios, function types, and dimensional conditions.

In the complex function scenario of the CEC2017 (d = 30) test set, unimodal functions (F1–F7) evaluate the algorithm's local exploitation capability. For F1, a basic unimodal function, MESDO achieves an average fitness value (Ave = 2.0944×10^3) far lower than the original SDO (1.1519×10^7) and VPPSO (1.0818×10^7), only 1/5500 and 1/5160 of their values, respectively, and also surpasses the next best performer HSO (7.8563×10^3), achieving 1/3.75 of its value. Its standard deviation (Std = 1.8539×10^3) is only 1/7450 of SDO (1.3814×10^7), demonstrating highly precise convergence with minimal performance fluctuation across repeated runs. F3, a noisy unimodal function, sees MESDO achieve Ave = 6.2832×10^3, reducing 75% relative to SDO (2.4989×10^4) and 61% relative to SBO (1.6155×10^4), with Std = 3.4882×10^3, only 1/12.6 of IWOA (4.4143×10^4), highlighting its robust local exploitation even under noise interference.

Multimodal functions (F8-F13) require overcoming local optima. For F9, a high-dimensional multimodal function, MESDO achieves Ave = 1.1916×10^3, 23% lower than

SDO (1.5538×10^3) and 55% lower than HSO (2.6749×10^3), with Std = 1.8556×10^2, only 1/5.1 of VPPSO (9.5445×10^2), demonstrating the efficacy of the adaptive differential evolution operator in escaping local optima. For F12, a complex multimodal function, MESDO achieves Ave = 2.0569×10^5, 91% lower than SDO (2.3800×10^6) and 59% lower than HSO (5.0311×10^5), with Std = 1.8228×10^5 far below SDO (2.1727×10^6), reflecting high global exploration precision in multimodal solution spaces.

For hybrid and composite functions (F14–F30), which integrate multiple solution space characteristics, F15, a hybrid function, shows MESDO achieving Ave = 6.3207×10^3, 37% lower than SDO (1.0121×10^4) and 15% lower than MadDE (7.4550×10^3), with Std = 6.9524×10^3, only 1/555 of IAGWO (3.8598×10^6), demonstrating strong adaptability to hybrid solution spaces. For F30, a high-dimensional composite function, MESDO achieves Ave = 9.5217×10^3, 92% lower than SDO (1.1531×10^5) and 80% lower than SBO (4.8471×10^4), with Std = 2.5993×10^3, only 1/2325 of VPPSO (6.0446×10^6), becoming the only algorithm in this function close to the theoretical optimum.

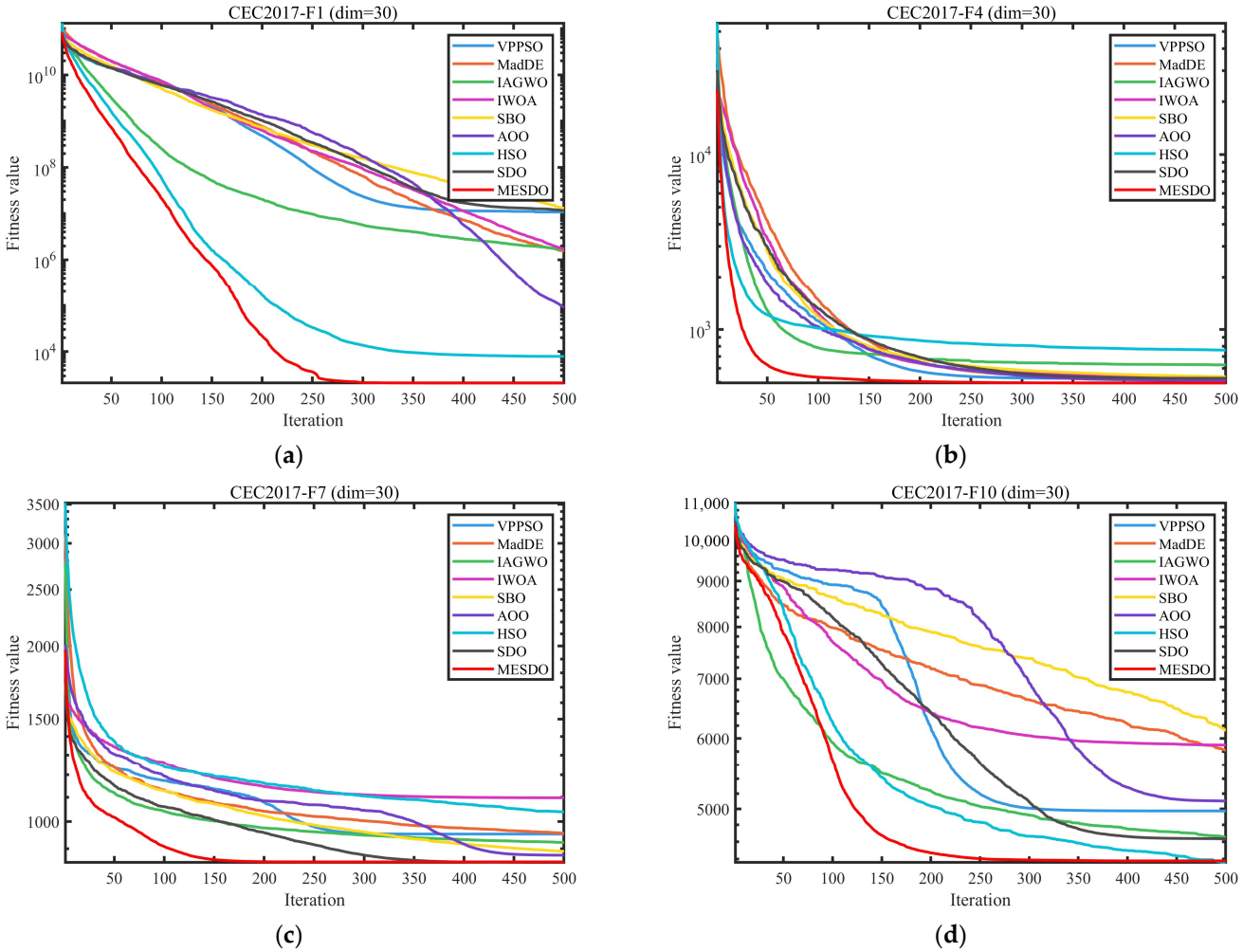
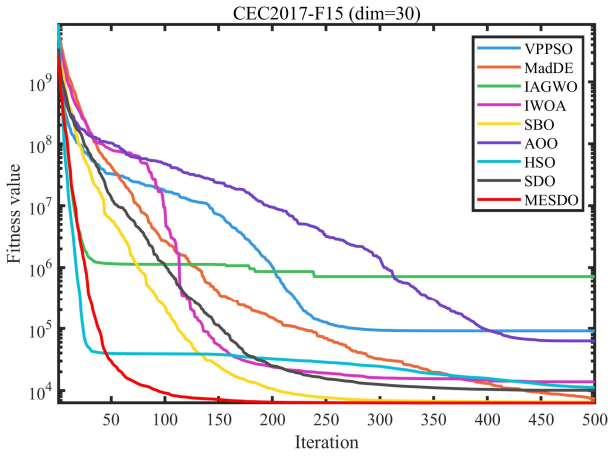
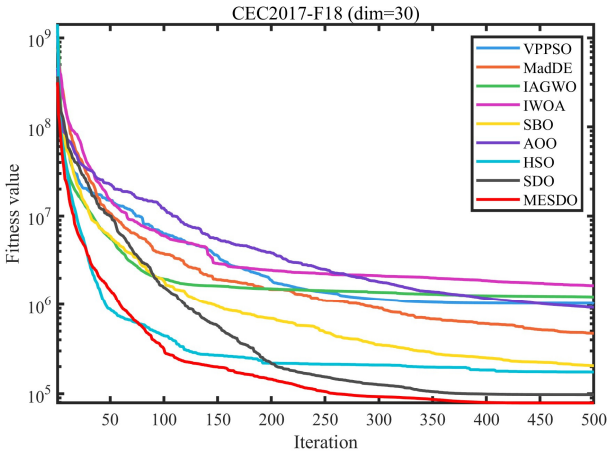


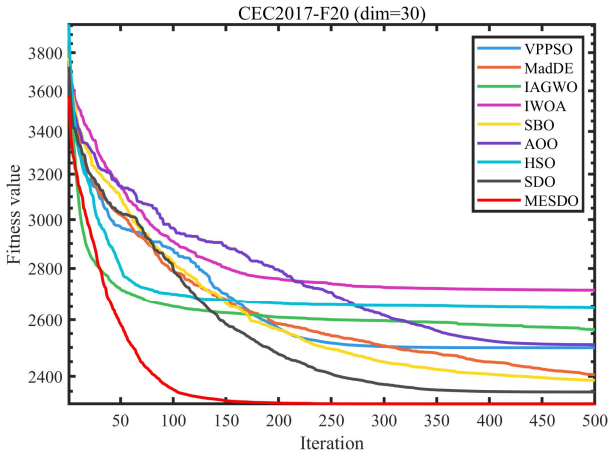
Figure 8. *Cont.*



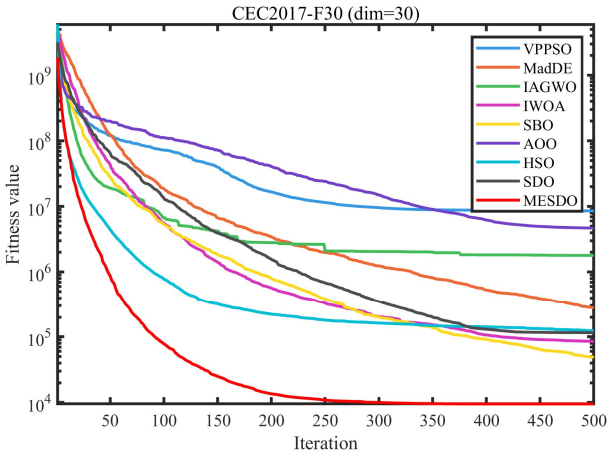
(**e**)



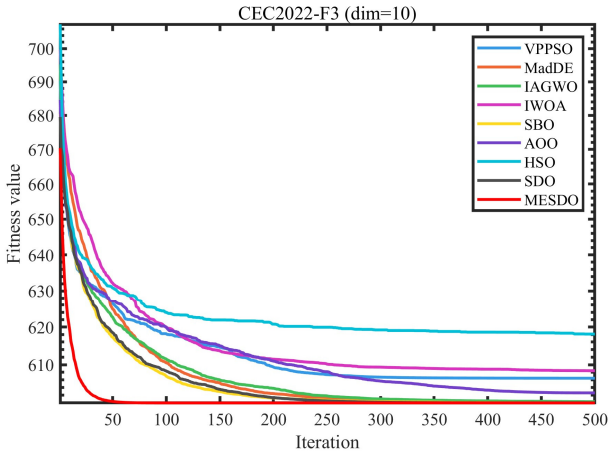
(**f**)



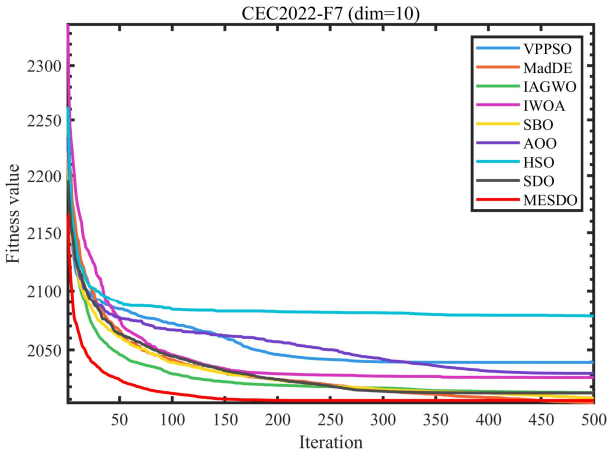
(**g**)



(**h**)

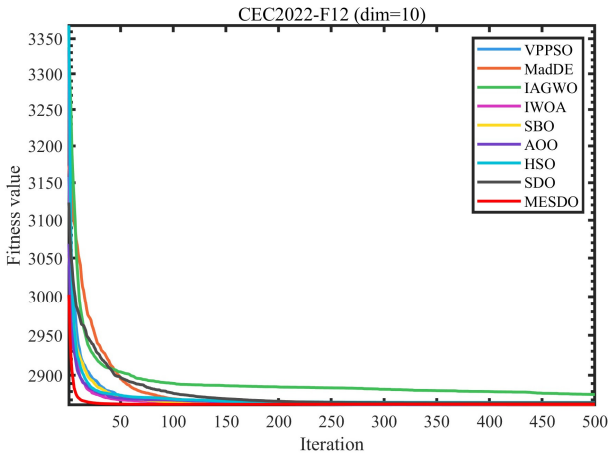


(**i**)

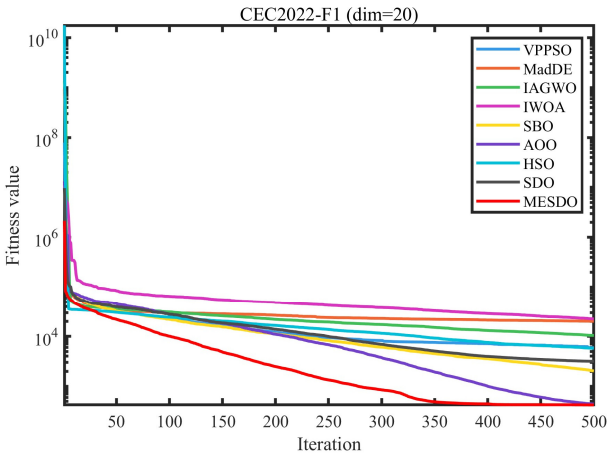


(**j**)

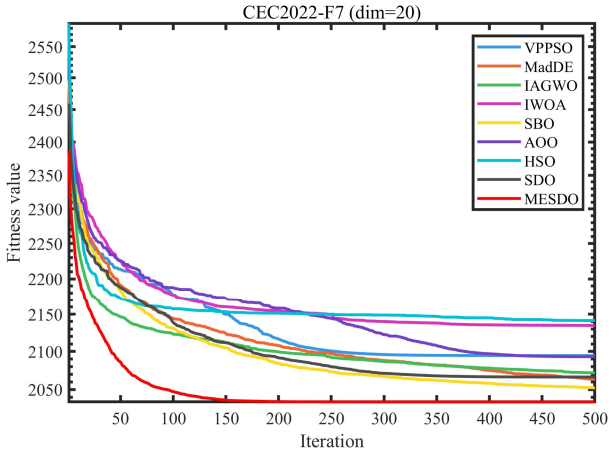
Figure 8. *Cont*.



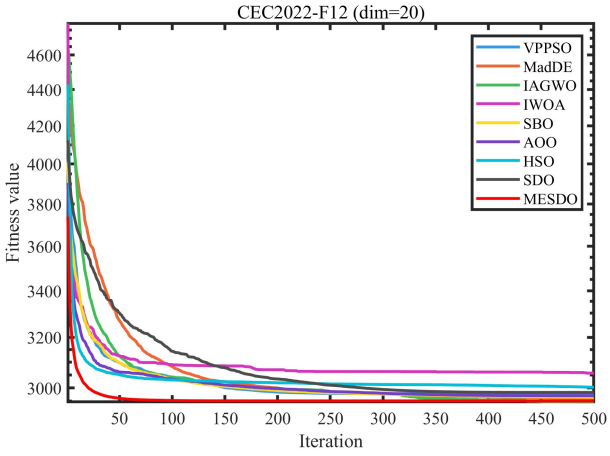
(**k**)



(**l**)

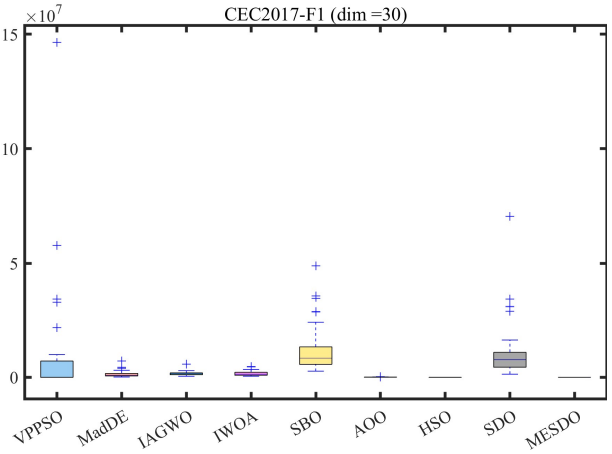


(**m**)

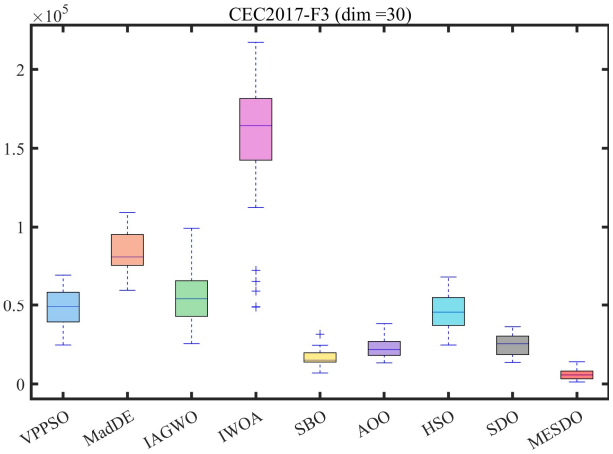


(**n**)

Figure 8. Comparison of convergence speed of different algorithms on test set.

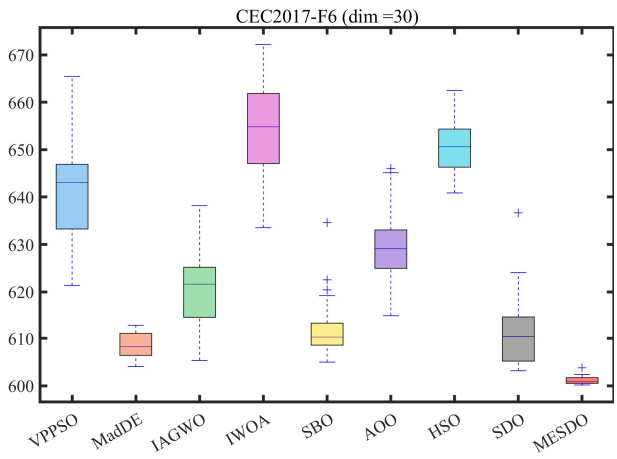


(**a**)

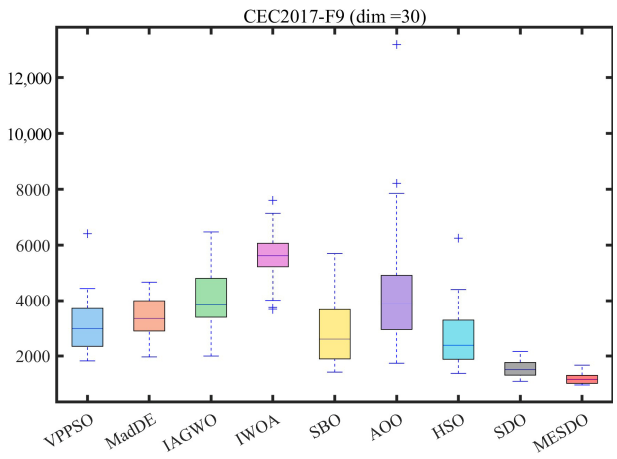


(**b**)

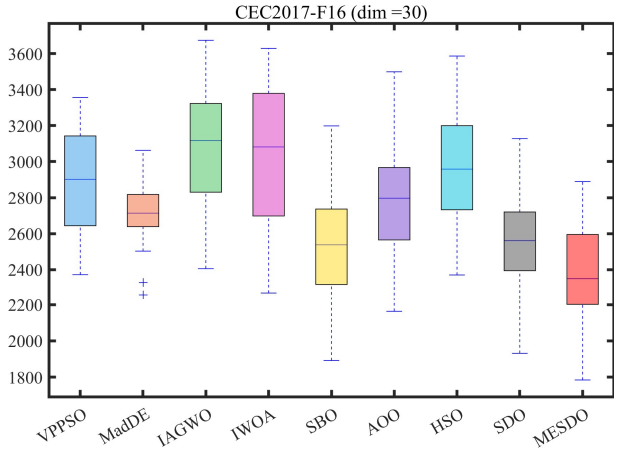
Figure 9. *Cont.*



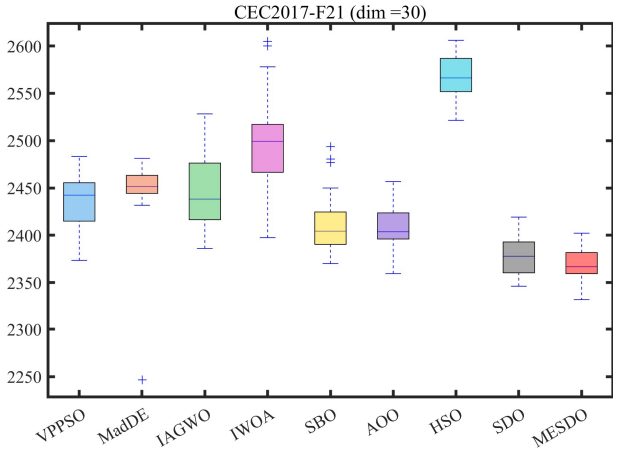
(**c**)



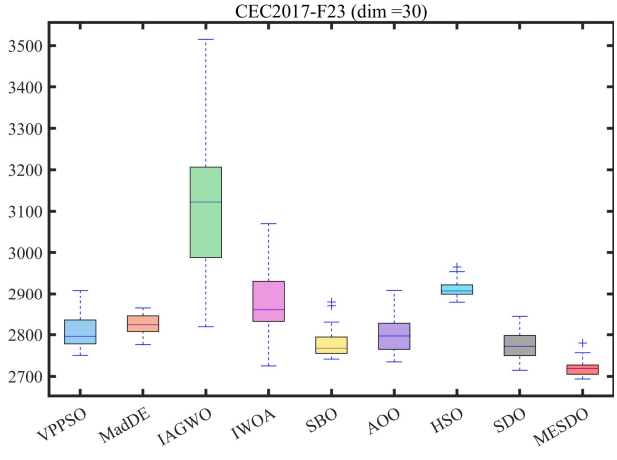
(**d**)



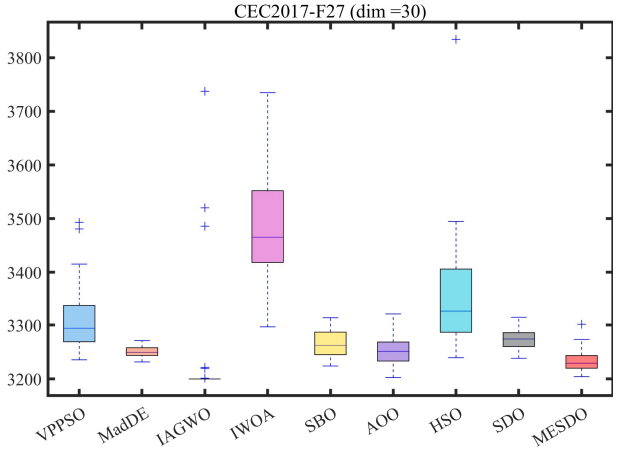
(**e**)



(**f**)



(**g**)



(**h**)

Figure 9. *Cont*.

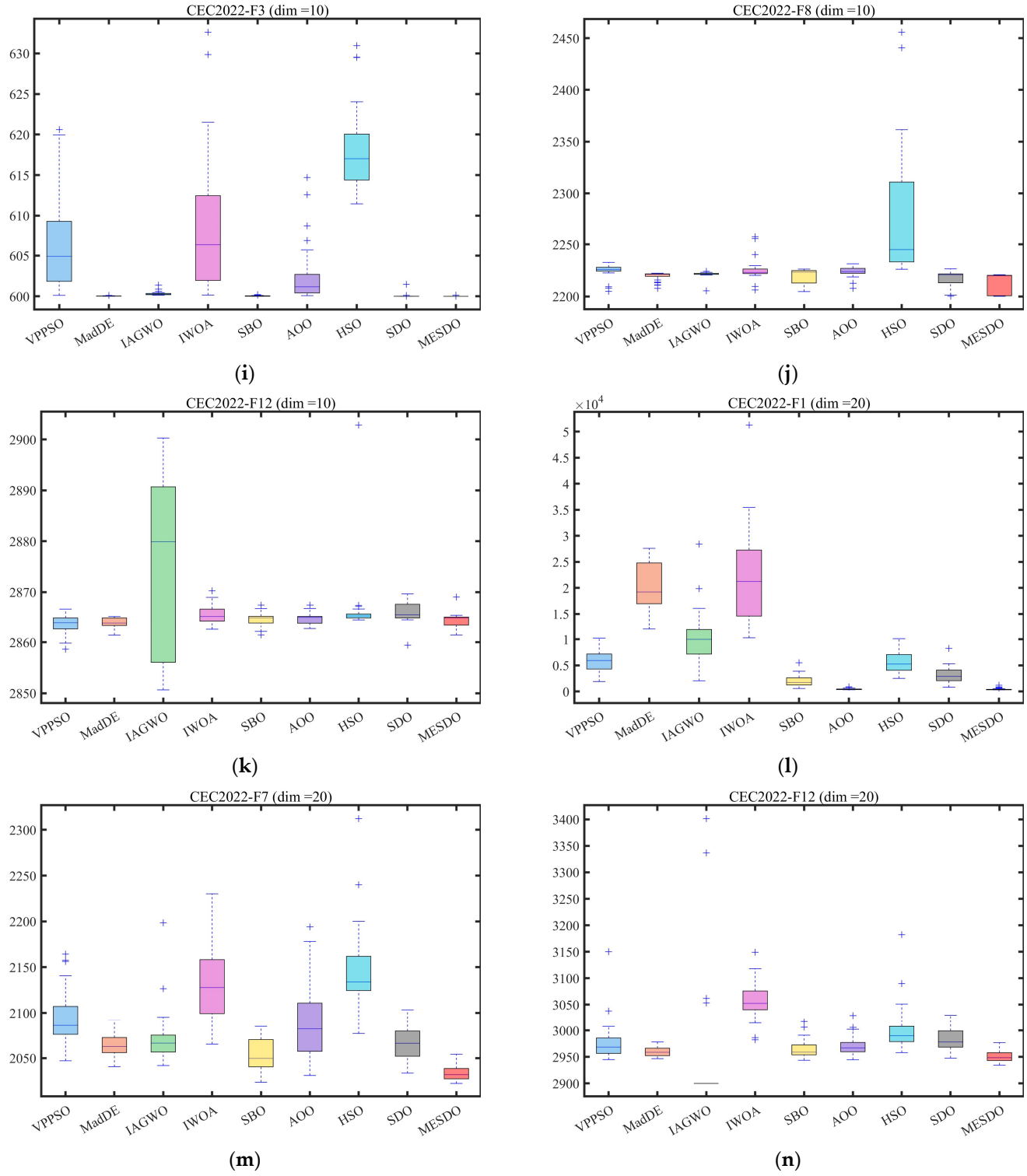


Figure 9. Boxplot analysis for different algorithms on the test set.

Table 3. Experimental findings on the CEC2017 benchmark suite (d = 30).

Function	Metric	VPPSO	MadDE	IAGWO	IWOA	SBO	AOO	HSO	SDO	MESDO
F1	Ave	1.0818×10^{7}	1.5369×10^{6}	1.6769×10^{6}	1.7005×10^{6}	1.3043×10^{7}	9.0828×10^{4}	7.8563×10^{3}	1.1519×10^{7}	$\mathbf{2.0944 \times 10^{3}}$
	Std	2.8898×10^{7}	1.5659×10^{6}	9.2579×10^{5}	1.1002×10^{6}	1.1404×10^{7}	5.1457×10^{4}	7.9143×10^{3}	1.3814×10^{7}	$\mathbf{1.8539 \times 10^{3}}$
F2	Ave	1.0972×10^{26}	1.8806×10^{20}	2.3854×10^{37}	1.7353×10^{21}	9.3603×10^{19}	3.2042×10^{16}	$\mathbf{1.2651 \times 10^{16}}$	6.6655×10^{22}	2.2912×10^{17}
	Std	5.9584×10^{26}	4.8719×10^{20}	1.2885×10^{38}	7.9049×10^{21}	3.6024×10^{20}	$\mathbf{4.4903 \times 10^{16}}$	4.7861×10^{16}	2.7356×10^{23}	6.5228×10^{17}
F3	Ave	4.8583×10^{4}	8.4731×10^{4}	5.5233×10^{4}	1.5371×10^{5}	1.6155×10^{4}	2.2694×10^{4}	4.5494×10^{4}	2.4989×10^{4}	$\mathbf{6.2832 \times 10^{3}}$
	Std	1.1819×10^{4}	1.2684×10^{4}	1.8397×10^{4}	4.4143×10^{4}	5.3980×10^{3}	6.6251×10^{3}	1.1110×10^{4}	6.5960×10^{3}	$\mathbf{3.4882 \times 10^{3}}$
F4	Ave	5.2537×10^{2}	5.1873×10^{2}	6.2800×10^{2}	5.3460×10^{2}	5.3793×10^{2}	5.1265×10^{2}	7.6132×10^{2}	5.2714×10^{2}	$\mathbf{4.9544 \times 10^{2}}$
	Std	1.9489×10^{1}	$\mathbf{1.4626 \times 10^{1}}$	2.8686×10^{2}	3.4140×10^{1}	2.6432×10^{1}	2.2954×10^{1}	1.2788×10^{2}	2.5214×10^{1}	2.3181×10^{1}
F5	Ave	6.4495×10^{2}	6.8121×10^{2}	6.4794×10^{2}	7.4294×10^{2}	6.3416×10^{2}	6.2979×10^{2}	6.8721×10^{2}	5.8216×10^{2}	$\mathbf{5.7479 \times 10^{2}}$
	Std	2.8198×10^{1}	2.1900×10^{1}	3.1975×10^{1}	5.0365×10^{1}	3.3801×10^{1}	3.6250×10^{1}	2.6180×10^{1}	$\mathbf{1.6048 \times 10^{1}}$	1.9846×10^{1}
F6	Ave	6.4138×10^{2}	6.0836×10^{2}	6.2023×10^{2}	6.5420×10^{2}	6.1218×10^{2}	6.3031×10^{2}	6.5105×10^{2}	6.1131×10^{2}	$\mathbf{6.0113 \times 10^{2}}$
	Std	1.0593×10^{1}	2.4405×10^{0}	8.4715×10^{0}	9.4561×10^{0}	6.1150×10^{0}	8.3744×10^{0}	5.9658×10^{0}	7.4819×10^{0}	$\mathbf{8.4732 \times 10^{-1}}$
F7	Ave	9.5280×10^{2}	9.5584×10^{2}	9.2197×10^{2}	1.0974×10^{3}	8.9093×10^{2}	8.7739×10^{2}	1.0390×10^{3}	$\mathbf{8.5281 \times 10^{2}}$	8.5488×10^{2}
	Std	6.9401×10^{1}	$\mathbf{2.6794 \times 10^{1}}$	4.8562×10^{1}	1.0155×10^{2}	4.1730×10^{1}	4.0425×10^{1}	7.2330×10^{1}	3.8533×10^{1}	4.1168×10^{1}
F8	Ave	9.2654×10^{2}	9.4614×10^{2}	9.1743×10^{2}	9.7667×10^{2}	9.2567×10^{2}	9.1520×10^{2}	1.0468×10^{3}	8.7278×10^{2}	$\mathbf{8.6527 \times 10^{2}}$
	Std	2.9760×10^{1}	2.0056×10^{1}	2.6601×10^{1}	3.9400×10^{1}	3.1220×10^{1}	2.8806×10^{1}	3.1244×10^{1}	$\mathbf{1.8807 \times 10^{1}}$	2.2077×10^{1}
F9	Ave	3.1089×10^{3}	3.4180×10^{3}	4.1362×10^{3}	5.6138×10^{3}	2.8293×10^{3}	4.4338×10^{3}	2.6749×10^{3}	1.5538×10^{3}	$\mathbf{1.1916 \times 10^{3}}$
	Std	9.5445×10^{2}	6.7734×10^{2}	1.1107×10^{3}	8.9501×10^{2}	1.1137×10^{3}	2.3269×10^{3}	1.1233×10^{3}	2.8022×10^{2}	$\mathbf{1.8556 \times 10^{2}}$
F10	Ave	4.9746×10^{3}	5.8241×10^{3}	4.6548×10^{3}	5.9000×10^{3}	6.1288×10^{3}	5.1018×10^{3}	$\mathbf{4.3628 \times 10^{3}}$	4.6329×10^{3}	4.3806×10^{3}
	Std	6.8175×10^{2}	$\mathbf{4.8105 \times 10^{2}}$	5.7025×10^{2}	7.8260×10^{2}	1.2581×10^{3}	7.5946×10^{2}	8.6598×10^{2}	6.2019×10^{2}	5.3895×10^{2}
F11	Ave	1.4208×10^{3}	1.3195×10^{3}	1.9327×10^{3}	1.3959×10^{3}	1.2402×10^{3}	1.2712×10^{3}	1.7871×10^{3}	1.2435×10^{3}	$\mathbf{1.2120 \times 10^{3}}$
	Std	1.0874×10^{2}	$\mathbf{3.2174 \times 10^{1}}$	9.7949×10^{2}	1.4636×10^{2}	3.4820×10^{1}	4.4980×10^{1}	1.3524×10^{2}	3.3383×10^{1}	3.6882×10^{1}
F12	Ave	2.3996×10^{7}	2.2745×10^{6}	6.2337×10^{7}	4.0994×10^{6}	2.5068×10^{6}	1.6160×10^{7}	5.0311×10^{5}	2.3800×10^{6}	$\mathbf{2.0569 \times 10^{5}}$
	Std	2.2129×10^{7}	9.8226×10^{5}	1.1371×10^{8}	2.7402×10^{6}	1.9073×10^{6}	1.5269×10^{7}	7.1812×10^{5}	2.1727×10^{6}	$\mathbf{1.8228 \times 10^{5}}$
F13	Ave	1.1110×10^{5}	9.7196×10^{4}	3.7457×10^{4}	5.0612×10^{4}	1.7274×10^{4}	8.2200×10^{4}	2.1702×10^{4}	1.4963×10^{4}	$\mathbf{1.3767 \times 10^{4}}$
	Std	5.3245×10^{4}	1.0179×10^{5}	7.8781×10^{4}	6.1998×10^{4}	2.0570×10^{4}	5.0226×10^{4}	1.8289×10^{4}	$\mathbf{1.1216 \times 10^{4}}$	1.5203×10^{4}
F14	Ave	9.7409×10^{4}	8.1812×10^{4}	7.3379×10^{5}	3.4002×10^{5}	1.0001×10^{4}	6.6872×10^{4}	1.0947×10^{4}	$\mathbf{2.3557 \times 10^{3}}$	3.4043×10^{3}
	Std	1.3676×10^{5}	6.2370×10^{4}	7.1140×10^{5}	3.5065×10^{5}	1.8636×10^{4}	5.1792×10^{4}	8.6262×10^{3}	$\mathbf{1.5809 \times 10^{3}}$	2.8762×10^{3}
F15	Ave	9.1442×10^{4}	7.4550×10^{3}	7.1018×10^{5}	1.3815×10^{4}	6.5366×10^{3}	6.2979×10^{4}	1.1184×10^{4}	1.0121×10^{4}	$\mathbf{6.3207 \times 10^{3}}$
	Std	3.0232×10^{5}	$\mathbf{5.3645 \times 10^{3}}$	3.8598×10^{6}	1.4835×10^{4}	6.1067×10^{3}	5.5549×10^{4}	7.1309×10^{3}	7.3628×10^{3}	6.9524×10^{3}

Table 3. *Cont.*

Function	Metric	VPPSO	MadDE	IAGWO	IWOA	SBO	AOO	HSO	SDO	MESDO
F16	Ave	2.8908×10^{3}	2.7095×10^{3}	3.0927×10^{3}	3.0516×10^{3}	2.5358×10^{3}	2.7487×10^{3}	2.9684×10^{3}	2.5293×10^{3}	$\mathbf{2.3881 \times 10^{3}}$
	Std	2.9620×10^{2}	$\mathbf{1.6482 \times 10^{2}}$	3.0530×10^{2}	3.5908×10^{2}	2.9637×10^{2}	3.2072×10^{2}	3.0184×10^{2}	2.8421×10^{2}	2.6147×10^{2}
F17	Ave	2.2423×10^{3}	2.0191×10^{3}	2.3349×10^{3}	2.4671×10^{3}	2.0836×10^{3}	2.1279×10^{3}	2.6508×10^{3}	2.0259×10^{3}	$\mathbf{1.9873 \times 10^{3}}$
	Std	2.1788×10^{2}	$\mathbf{9.6034 \times 10^{1}}$	3.0530×10^{2}	2.9213×10^{2}	1.8352×10^{2}	1.7130×10^{2}	2.6637×10^{2}	1.7898×10^{2}	1.7815×10^{2}
F18	Ave	1.0076×10^{6}	4.7322×10^{5}	1.2277×10^{6}	1.6485×10^{6}	2.0538×10^{5}	9.2430×10^{5}	1.7462×10^{5}	9.7568×10^{4}	$\mathbf{7.8789 \times 10^{4}}$
	Std	9.0167×10^{5}	3.1346×10^{5}	1.7435×10^{6}	2.1491×10^{6}	2.0880×10^{5}	6.3074×10^{5}	1.6073×10^{5}	1.4649×10^{5}	$\mathbf{6.2477 \times 10^{4}}$
F19	Ave	1.9738×10^{6}	1.5692×10^{4}	$\mathbf{7.1878 \times 10^{3}}$	8.4190×10^{3}	9.3637×10^{3}	7.0549×10^{5}	1.8119×10^{4}	1.0072×10^{4}	8.0143×10^{3}
	Std	1.2313×10^{6}	1.2482×10^{4}	$\mathbf{5.6719 \times 10^{3}}$	6.9696×10^{3}	8.1908×10^{3}	7.4359×10^{5}	2.5623×10^{4}	9.5279×10^{3}	1.0991×10^{4}
F20	Ave	2.4991×10^{3}	2.4055×10^{3}	2.5639×10^{3}	2.7147×10^{3}	2.3857×10^{3}	2.5099×10^{3}	2.6449×10^{3}	2.3474×10^{3}	$\mathbf{2.3081 \times 10^{3}}$
	Std	1.4965×10^{2}	$\mathbf{1.3280 \times 10^{2}}$	1.8953×10^{2}	2.2155×10^{2}	1.3504×10^{2}	1.5694×10^{2}	2.5183×10^{2}	1.6716×10^{2}	1.9527×10^{2}
F21	Ave	2.4369×10^{3}	2.4471×10^{3}	2.4454×10^{3}	2.5007×10^{3}	2.4128×10^{3}	2.4060×10^{3}	2.5690×10^{3}	2.3782×10^{3}	$\mathbf{2.3678 \times 10^{3}}$
	Std	2.8735×10^{1}	4.0106×10^{1}	3.6895×10^{1}	5.0538×10^{1}	3.1258×10^{1}	2.3141×10^{1}	2.1753×10^{1}	1.9384×10^{1}	$\mathbf{1.7171 \times 10^{1}}$
F22	Ave	3.3988×10^{3}	2.3384×10^{3}	3.5213×10^{3}	4.2976×10^{3}	2.3225×10^{3}	5.6517×10^{3}	4.6091×10^{3}	$\mathbf{2.3202 \times 10^{3}}$	2.5529×10^{3}
	Std	1.8391×10^{3}	6.4227×10^{1}	1.8921×10^{3}	2.4989×10^{3}	$\mathbf{3.9619 \times 10^{0}}$	1.4387×10^{3}	1.4240×10^{3}	4.3942×10^{0}	9.8142×10^{2}
F23	Ave	2.8051×10^{3}	2.8237×10^{3}	3.1048×10^{3}	2.8853×10^{3}	2.7785×10^{3}	2.8011×10^{3}	2.9115×10^{3}	2.7727×10^{3}	$\mathbf{2.7200 \times 10^{3}}$
	Std	3.7382×10^{1}	2.3518×10^{1}	1.6299×10^{2}	8.5655×10^{1}	3.4612×10^{1}	4.2803×10^{1}	1.9343×10^{1}	3.2199×10^{1}	$\mathbf{1.9143 \times 10^{1}}$
F24	Ave	2.9586×10^{3}	2.9977×10^{3}	3.3038×10^{3}	3.0530×10^{3}	2.9443×10^{3}	2.9437×10^{3}	3.0413×10^{3}	2.9295×10^{3}	$\mathbf{2.8902 \times 10^{3}}$
	Std	4.0867×10^{1}	1.9175×10^{1}	1.6545×10^{2}	9.0023×10^{1}	3.4087×10^{1}	3.6235×10^{1}	$\mathbf{1.3378 \times 10^{1}}$	3.4685×10^{1}	1.6880×10^{1}
F25	Ave	2.9476×10^{3}	2.9253×10^{3}	2.9461×10^{3}	2.9149×10^{3}	2.9260×10^{3}	2.9167×10^{3}	3.2052×10^{3}	2.9285×10^{3}	$\mathbf{2.8958 \times 10^{3}}$
	Std	3.1924×10^{1}	1.7449×10^{1}	3.1021×10^{1}	1.7982×10^{1}	2.2490×10^{1}	2.7479×10^{1}	1.0688×10^{2}	1.9242×10^{1}	$\mathbf{1.1469 \times 10^{1}}$
F26	Ave	4.9506×10^{3}	$\mathbf{3.6313 \times 10^{3}}$	4.2159×10^{3}	5.7116×10^{3}	4.1048×10^{3}	4.6425×10^{3}	5.2544×10^{3}	4.8072×10^{3}	4.3650×10^{3}
	Std	1.2240×10^{3}	9.9544×10^{2}	1.7815×10^{3}	1.7994×10^{3}	1.4028×10^{3}	1.0495×10^{3}	$\mathbf{3.7358 \times 10^{2}}$	7.4129×10^{2}	5.4780×10^{2}
F27	Ave	3.3123×10^{3}	3.2504×10^{3}	3.2395×10^{3}	3.4861×10^{3}	3.2652×10^{3}	3.2536×10^{3}	3.3609×10^{3}	3.2735×10^{3}	$\mathbf{3.2332 \times 10^{3}}$
	Std	6.2963×10^{1}	$\mathbf{1.0375 \times 10^{1}}$	1.2133×10^{2}	9.8501×10^{1}	2.2492×10^{1}	2.8546×10^{1}	1.1718×10^{2}	1.8711×10^{1}	2.1083×10^{1}
F28	Ave	3.3078×10^{3}	3.2727×10^{3}	3.3798×10^{3}	4.5489×10^{3}	3.2923×10^{3}	3.2511×10^{3}	3.6932×10^{3}	3.2956×10^{3}	$\mathbf{3.2376 \times 10^{3}}$
	Std	3.5560×10^{1}	2.5025×10^{1}	1.2420×10^{2}	1.1944×10^{3}	2.7059×10^{1}	2.7565×10^{1}	1.7154×10^{2}	3.5262×10^{1}	$\mathbf{1.8709 \times 10^{1}}$
F29	Ave	4.2507×10^{3}	3.8794×10^{3}	3.9526×10^{3}	4.2559×10^{3}	3.8744×10^{3}	4.0061×10^{3}	4.4724×10^{3}	3.8450×10^{3}	$\mathbf{3.7151 \times 10^{3}}$
	Std	2.7126×10^{2}	$\mathbf{1.3379 \times 10^{2}}$	3.1029×10^{2}	2.7553×10^{2}	2.3309×10^{2}	2.1074×10^{2}	2.7376×10^{2}	1.9784×10^{2}	2.2852×10^{2}
F30	Ave	8.5819×10^{6}	2.7977×10^{5}	1.7974×10^{6}	8.5106×10^{4}	4.8471×10^{4}	4.6916×10^{6}	1.2277×10^{5}	1.1531×10^{5}	$\mathbf{9.5217 \times 10^{3}}$
	Std	6.0446×10^{6}	1.7929×10^{5}	9.4923×10^{6}	1.9560×10^{5}	4.8704×10^{4}	2.7273×10^{6}	1.4292×10^{5}	1.7573×10^{5}	$\mathbf{2.5993 \times 10^{3}}$

Table 4. Experimental findings on the CEC2022 benchmark suite (d = 10).

Function	Metric	VPPSO	MadDE	IAGWO	IWOA	SBO	AOO	HSO	SDO	MESDO
F1	Ave	3.2483×10^{2}	1.0415×10^{3}	3.5610×10^{2}	5.5048×10^{2}	3.0004×10^{2}	3.0000×10^{2}	1.3222×10^{3}	3.0022×10^{2}	$\mathbf{3.0000 \times 10^{2}}$
	Std	6.1018×10^{1}	8.9248×10^{2}	9.8311×10^{1}	1.6630×10^{2}	5.0232×10^{-2}	3.2410×10^{-3}	2.3257×10^{2}	4.8557×10^{-1}	$\mathbf{2.1069 \times 10^{-10}}$
F2	Ave	4.0798×10^{2}	$\mathbf{4.0007 \times 10^{2}}$	4.3523×10^{2}	4.0865×10^{2}	4.0759×10^{2}	4.1351×10^{2}	4.4860×10^{2}	4.0250×10^{2}	4.0320×10^{2}
	Std	1.3030×10^{1}	$\mathbf{1.1247 \times 10^{-1}}$	3.5378×10^{1}	1.2314×10^{1}	1.2572×10^{1}	2.3845×10^{1}	2.8761×10^{1}	3.4593×10^{0}	3.0181×10^{0}
F3	Ave	6.0645×10^{2}	6.0003×10^{2}	6.0034×10^{2}	6.0842×10^{2}	6.0004×10^{2}	6.0262×10^{2}	6.1813×10^{2}	6.0006×10^{2}	$\mathbf{6.0001 \times 10^{2}}$
	Std	5.7739×10^{0}	$\mathbf{2.2083 \times 10^{-2}}$	2.5068×10^{-1}	8.3008×10^{0}	5.2154×10^{-2}	3.7088×10^{0}	4.9658×10^{0}	2.6699×10^{-1}	2.7214×10^{-2}
F4	Ave	8.2105×10^{2}	$\mathbf{8.1221 \times 10^{2}}$	8.2349×10^{2}	8.3423×10^{2}	8.1808×10^{2}	8.1897×10^{2}	8.3756×10^{2}	8.1529×10^{2}	8.1419×10^{2}
	Std	7.4881×10^{0}	$\mathbf{2.7158 \times 10^{0}}$	8.4170×10^{0}	1.0362×10^{1}	7.7866×10^{0}	6.8532×10^{0}	5.5276×10^{0}	6.5347×10^{0}	6.8106×10^{0}
F5	Ave	9.1149×10^{2}	9.0305×10^{2}	9.0923×10^{2}	1.0362×10^{3}	9.0131×10^{2}	9.0182×10^{2}	9.1471×10^{2}	$\mathbf{9.0009 \times 10^{2}}$	9.0048×10^{2}
	Std	1.5243×10^{1}	5.0286×10^{0}	1.3641×10^{1}	1.4273×10^{2}	1.9138×10^{0}	4.8835×10^{0}	3.2290×10^{1}	$\mathbf{1.8391 \times 10^{-1}}$	7.2346×10^{-1}
F6	Ave	4.4648×10^{3}	2.0591×10^{3}	6.8253×10^{3}	2.5166×10^{3}	2.3707×10^{3}	4.8916×10^{3}	2.8318×10^{3}	1.8540×10^{3}	$\mathbf{1.8237 \times 10^{3}}$
	Std	2.2717×10^{3}	4.3347×10^{2}	2.3120×10^{4}	1.1749×10^{3}	1.2689×10^{3}	2.2951×10^{3}	8.6887×10^{2}	4.1804×10^{1}	$\mathbf{2.6482 \times 10^{1}}$
F7	Ave	2.0398×10^{3}	$\mathbf{2.0069 \times 10^{3}}$	2.0153×10^{3}	2.0276×10^{3}	2.0108×10^{3}	2.0309×10^{3}	2.0783×10^{3}	2.0150×10^{3}	2.0089×10^{3}
	Std	1.3570×10^{1}	$\mathbf{5.7978 \times 10^{0}}$	9.2569×10^{0}	7.0658×10^{0}	8.9562×10^{0}	8.9337×10^{0}	3.5127×10^{1}	1.0946×10^{1}	9.5157×10^{0}
F8	Ave	2.2247×10^{3}	2.2192×10^{3}	2.2212×10^{3}	2.2255×10^{3}	2.2192×10^{3}	2.2242×10^{3}	2.2777×10^{3}	2.2172×10^{3}	$\mathbf{2.2116 \times 10^{3}}$
	Std	6.3820×10^{0}	4.2878×10^{0}	$\mathbf{3.0903 \times 10^{0}}$	1.0158×10^{1}	7.8023×10^{0}	5.0696×10^{0}	6.3604×10^{1}	7.6872×10^{0}	9.8106×10^{0}
F9	Ave	2.5322×10^{3}	2.5293×10^{3}	2.5342×10^{3}	2.5440×10^{3}	2.5293×10^{3}	2.5294×10^{3}	2.6870×10^{3}	2.5293×10^{3}	$\mathbf{2.5293 \times 10^{3}}$
	Std	4.2132×10^{0}	9.1961×10^{-6}	4.0526×10^{1}	4.4832×10^{1}	2.3244×10^{-3}	2.6212×10^{-1}	5.1084×10^{1}	6.3538×10^{-3}	$\mathbf{0.0000 \times 10^{0}}$
F10	Ave	2.5286×10^{3}	2.5005×10^{3}	2.5774×10^{3}	2.5008×10^{3}	2.5083×10^{3}	2.5708×10^{3}	2.6186×10^{3}	$\mathbf{2.5004 \times 10^{3}}$	2.5378×10^{3}
	Std	5.1485×10^{1}	1.3585×10^{-1}	6.3054×10^{1}	2.6086×10^{-1}	3.0042×10^{1}	9.6422×10^{1}	1.3887×10^{2}	$\mathbf{1.0341 \times 10^{-1}}$	5.3953×10^{1}
F11	Ave	2.6962×10^{3}	2.6100×10^{3}	2.7119×10^{3}	2.6980×10^{3}	2.6485×10^{3}	2.6900×10^{3}	3.0075×10^{3}	$\mathbf{2.6000 \times 10^{3}}$	2.6667×10^{3}
	Std	1.6066×10^{2}	5.4768×10^{1}	1.2510×10^{2}	1.6375×10^{2}	9.2474×10^{1}	1.5392×10^{2}	2.0000×10^{2}	$\mathbf{3.5476 \times 10^{-3}}$	1.1249×10^{2}
F12	Ave	$\mathbf{2.8637 \times 10^{3}}$	2.8639×10^{3}	2.8765×10^{3}	2.8654×10^{3}	2.8645×10^{3}	2.8648×10^{3}	2.8667×10^{3}	2.8662×10^{3}	2.8644×10^{3}
	Std	1.8194×10^{0}	$\mathbf{9.8001 \times 10^{-1}}$	1.7379×10^{1}	1.9477×10^{0}	1.4583×10^{0}	1.1944×10^{0}	6.8761×10^{0}	2.1079×10^{0}	1.3653×10^{0}

Table 5. Experimental findings on the CEC2022 benchmark suite (d = 20).

Function	Metric	VPPSO	MadDE	IAGWO	IWOA	SBO	AOO	HSO	SDO	MESDO
F1	Ave	5.9254×10^{3}	2.0090×10^{4}	1.0430×10^{4}	2.2321×10^{4}	2.0302×10^{3}	4.4308×10^{2}	5.7414×10^{3}	3.1655×10^{3}	$\mathbf{4.2563 \times 10^{2}}$
	Std	2.1915×10^{3}	4.6381×10^{3}	5.1350×10^{3}	8.9997×10^{3}	1.1229×10^{3}	$\mathbf{1.4405 \times 10^{2}}$	2.2433×10^{3}	1.5937×10^{3}	2.0314×10^{2}
F2	Ave	4.8896×10^{2}	4.5602×10^{2}	5.0257×10^{2}	4.7204×10^{2}	4.6491×10^{2}	4.5687×10^{2}	5.8205×10^{2}	4.5763×10^{2}	$\mathbf{4.4812 \times 10^{2}}$
	Std	4.3428×10^{1}	$\mathbf{9.7333 \times 10^{0}}$	5.1681×10^{1}	2.7023×10^{1}	1.9001×10^{1}	1.9550×10^{1}	6.9691×10^{1}	1.0494×10^{1}	1.3992×10^{1}
F3	Ave	6.2688×10^{2}	6.0161×10^{2}	6.1040×10^{2}	6.4241×10^{2}	6.0248×10^{2}	6.1761×10^{2}	6.3806×10^{2}	6.0255×10^{2}	$\mathbf{6.0013 \times 10^{2}}$
	Std	8.8510×10^{0}	7.7898×10^{-1}	7.7314×10^{0}	1.5410×10^{1}	1.8942×10^{0}	8.5076×10^{0}	4.8450×10^{0}	2.0891×10^{0}	$\mathbf{1.5359 \times 10^{-1}}$
F4	Ave	8.6053×10^{2}	8.7232×10^{2}	8.7615×10^{2}	8.9240×10^{2}	8.6805×10^{2}	8.7229×10^{2}	9.2001×10^{2}	8.3985×10^{2}	$\mathbf{8.3785 \times 10^{2}}$
	Std	1.4921×10^{1}	$\mathbf{1.0128 \times 10^{1}}$	1.9078×10^{1}	2.6047×10^{1}	1.7846×10^{1}	2.0693×10^{1}	1.0536×10^{1}	1.2221×10^{1}	1.3791×10^{1}
F5	Ave	1.7432×10^{3}	1.5146×10^{3}	1.9760×10^{3}	2.4067×10^{3}	1.2107×10^{3}	1.6416×10^{3}	1.1411×10^{3}	9.6590×10^{2}	$\mathbf{9.4435 \times 10^{2}}$
	Std	5.3725×10^{2}	2.3698×10^{2}	3.9740×10^{2}	3.5636×10^{2}	3.3333×10^{2}	9.3776×10^{2}	2.3745×10^{2}	$\mathbf{5.6851 \times 10^{1}}$	7.8843×10^{1}
F6	Ave	5.3907×10^{3}	5.8686×10^{4}	6.5843×10^{3}	5.6474×10^{3}	4.9274×10^{3}	6.4983×10^{3}	4.5500×10^{3}	$\mathbf{4.5414 \times 10^{3}}$	4.8788×10^{3}
	Std	4.4651×10^{3}	5.3954×10^{4}	1.0411×10^{4}	4.3759×10^{3}	3.6540×10^{3}	5.1770×10^{3}	3.9263×10^{3}	$\mathbf{2.5444 \times 10^{3}}$	3.8841×10^{3}
F7	Ave	2.0945×10^{3}	2.0633×10^{3}	2.0718×10^{3}	2.1347×10^{3}	2.0522×10^{3}	2.0930×10^{3}	2.1415×10^{3}	2.0663×10^{3}	$\mathbf{2.0340 \times 10^{3}}$
	Std	2.8964×10^{1}	1.1970×10^{1}	2.9506×10^{1}	4.7441×10^{1}	1.7266×10^{1}	4.1877×10^{1}	4.8498×10^{1}	1.9755×10^{1}	$\mathbf{8.9419 \times 10^{0}}$
F8	Ave	2.2557×10^{3}	2.2268×10^{3}	2.2458×10^{3}	2.2517×10^{3}	2.2314×10^{3}	2.2660×10^{3}	2.4343×10^{3}	2.2261×10^{3}	$\mathbf{2.2238 \times 10^{3}}$
	Std	4.7927×10^{1}	$\mathbf{1.0582 \times 10^{0}}$	4.4051×10^{1}	4.5017×10^{1}	2.1564×10^{1}	6.2406×10^{1}	1.4136×10^{2}	2.1941×10^{0}	4.8856×10^{0}
F9	Ave	2.5164×10^{3}	2.4810×10^{3}	2.5057×10^{3}	2.4811×10^{3}	2.4828×10^{3}	2.4829×10^{3}	2.7373×10^{3}	2.4839×10^{3}	$\mathbf{2.4808 \times 10^{3}}$
	Std	3.6787×10^{1}	9.4767×10^{-2}	3.9678×10^{1}	6.4723×10^{-1}	1.8589×10^{0}	2.5313×10^{0}	8.1026×10^{1}	2.1885×10^{0}	$\mathbf{4.4791 \times 10^{-6}}$
F10	Ave	3.2006×10^{3}	2.5316×10^{3}	2.7294×10^{3}	2.5909×10^{3}	$\mathbf{2.5125 \times 10^{3}}$	3.4802×10^{3}	4.1058×10^{3}	2.5199×10^{3}	2.5589×10^{3}
	Std	9.4490×10^{2}	6.9628×10^{1}	2.0330×10^{2}	3.0105×10^{2}	$\mathbf{4.3695 \times 10^{1}}$	8.0125×10^{2}	5.4524×10^{2}	4.9830×10^{1}	9.4286×10^{1}
F11	Ave	2.9489×10^{3}	$\mathbf{2.9099 \times 10^{3}}$	2.9308×10^{3}	2.9482×10^{3}	2.9343×10^{3}	2.9538×10^{3}	3.5589×10^{3}	2.9389×10^{3}	2.9309×10^{3}
	Std	1.2543×10^{2}	$\mathbf{2.7684 \times 10^{1}}$	7.9851×10^{1}	1.1541×10^{2}	1.1513×10^{2}	9.0549×10^{1}	2.4904×10^{2}	4.6880×10^{1}	5.3398×10^{1}
F12	Ave	2.9780×10^{3}	2.9597×10^{3}	$\mathbf{2.9472 \times 10^{3}}$	3.0574×10^{3}	2.9652×10^{3}	2.9706×10^{3}	3.0027×10^{3}	2.9832×10^{3}	2.9510×10^{3}
	Std	3.8578×10^{1}	$\mathbf{8.5067 \times 10^{0}}$	1.2476×10^{2}	3.7561×10^{1}	1.6887×10^{1}	1.8539×10^{1}	4.4392×10^{1}	2.0254×10^{1}	1.0567×10^{1}

In the low-dimensional (d = 10) scenario of the CEC2022 test set, higher accuracy is required. For F1, a low-dimensional unimodal function, MESDO achieves Ave = 3.0000×10^2 (theoretical optimum) and Std = 2.1069×10^{-10}, being the only algorithm to reach the theoretical optimum, while SDO achieves Ave = 3.0022×10^2, Std = 4.8557×10^{-1}, and HSO achieves Ave = 1.3222×10^3, showing significant differences. For F3, a low-dimensional multimodal function, MESDO achieves Ave = 6.0001×10^2, Std = 2.7214×10^{-2}, improving 86% in accuracy compared with SDO (Ave = 6.0006×10^2, Std = 2.6699×10^{-1}), with no other comparison algorithm achieving this precision.

In the high-dimensional (d = 20) scenario, the "curse of dimensionality" causes most comparison algorithms to degrade sharply. For F1, a high-dimensional unimodal function, MESDO achieves Ave = 4.2563×10^2, 86% lower than SDO (3.1655×10^3) and 4% lower than the next best AOO (4.4308×10^2), with Std = 2.0314×10^2, only 1/10.8 of VPPSO (2.1915×10^3). For F5, a high-dimensional composite function, MESDO achieves Ave = 9.4435×10^2, 2.2% lower than SDO (9.6590×10^2) and 17% lower than HSO (1.1411×10^3), with Std = 7.8843×10^1, only 1/6.8 of VPPSO (5.3725×10^2), being the only algorithm in high-dimensional scenarios without significant performance decay.

The convergence curves in Figure 8 dynamically confirm these results. In CEC2017-F1 (d = 30), MESDO reduces the fitness value below 1×10^4 within 50 iterations, whereas SDO requires 200 iterations, and VPPSO, IAGWO, and other algorithms cannot even reach 1×10^5 throughout the process. In high-dimensional functions of CEC2022, other algorithms enter a plateau early, while MESDO continues descending toward the theoretical optimum.

The boxplots in Figure 9 illustrate MESDO's stability from the data distribution perspective. For typical functions in the CEC2017 and CEC2022 test sets, MESDO exhibits narrower boxes, indicating smaller variability across multiple runs, with lower positions and medians far below other algorithms. In some functions, other algorithms display outliers, whereas MESDO shows none, further validating its consistently high performance across different experimental scenarios, avoiding performance fluctuations caused by random factors.

Overall, the quantitative data in Tables 3–5 and the visual features in Figures 8 and 9 mutually support that MESDO, through the synergy of elite-guided search, adaptive differential evolution operator, and centroid-based opposition learning boundary control, achieves superior optimization accuracy, faster convergence, and stronger stability across different test sets, function types, and dimensional conditions, significantly outperforming the original SDO and other mainstream comparison algorithms.

3.4. Statistical Analysis

In the field of algorithm optimization, statistical methods provide a structured framework for the quantitative assessment and comparison of different techniques. This evidence-based approach guides researchers in identifying the most effective method for their specific objectives. Within this context, the present study employs both the Wilcoxon rank-sum test and the Friedman test to rigorously evaluate the performance of the MESDO algorithm, with subsequent sections detailing the procedures and presenting the results.

3.4.1. Wilcoxon Rank Sum Test

A statistical analysis of the MESDO algorithm's performance is conducted herein via the Wilcoxon rank-sum test to determine its significance [51]. Unlike the traditional *t*-test, this non-parametric method does not require the data to satisfy normality assumptions,

offering greater robustness against non-normal distributions or outliers. Its test statistic W is formally defined by Equation (20) [48,50].

$$W = \sum_{i=1}^{n_1} R(X_i), \tag{20}$$

In this procedure, each data point X_i is assigned a rank $R(X_i)$ based on its value within the entire sample. The final test statistic U is calculated according to the expression given in Equation (21).

$$U = W - \frac{n_1(n_1+1)}{2}, \tag{21}$$

With sufficiently large samples, the U statistic follows an approximate normal distribution, as characterized by Equations (22) and (23).

$$\mu_U = \frac{n_1 n_2}{2}, \tag{22}$$

$$\sigma_U = \sqrt{\frac{n_1 n_2 (n_1+n_2+1)}{12}}, \tag{23}$$

where, n_1 and n_2 represent the number of observations in the first and second sample groups, respectively, and the standardized statistic Z is calculated by Equation (24).

$$Z = \frac{U - \mu_U}{\sigma_U}, \tag{24}$$

A significance level of $p = 0.05$ was used to evaluate whether the outcomes of each MESDO run demonstrated statistically significant differences compared to other algorithms. The null hypothesis (H_0) posits that no such difference exists between the two methods. Should the resulting p-value fall below 0.05, H_0 is rejected, suggesting a notable performance discrepancy; if not, it is upheld.

Table 6 presents the Wilcoxon rank-sum test results of MESDO against eight comparison algorithms (VPPSO, MadDE, IAGWO, IWOA, SBO, AOO, HSO, SDO) on the CEC2017 (d = 30) and CEC2022 (d = 10, d = 20) test sets, quantifying MESDO's performance relative to each algorithm in the "+/=/−" format ("+" indicates MESDO is significantly better, "=" indicates no significant difference, and "−" indicates MESDO is significantly worse), with a significance level of $p = 0.05$. The results reveal MESDO's statistical advantages across different test scenarios:

Table 6. Comparative performance of the evaluated algorithms on the CEC 2017 and CEC 2022 benchmark suites.

Statistical Results	CEC2017 Dim = 30 (+/=/−)	CEC2022 Dim = 10 (+/=/−)	CEC2022 Dim = 20 (+/=/−)
VPPSO	(30/0/0)	(11/0/1)	(11/0/1)
MadDE	(29/0/1)	(8/0/4)	(12/0/0)
IAGWO	(24/0/6)	(11/0/1)	(11/0/1)
IWOA	(30/0/0)	(12/0/0)	(11/0/1)
SBO	(24/0/6)	(9/0/3)	(10/0/2)
AOO	(29/0/1)	(10/0/2)	(11/0/1)
HSO	(27/0/3)	(12/0/0)	(11/0/1)
SDO	(20/0/10)	(8/0/4)	(9/0/3)

On the CEC2017 (d = 30) test set, MESDO demonstrates strong generality. It achieves a 30/0/0 all-win result against VPPSO and IWOA (i.e., MESDO is significantly better in all 30 test functions, with no ties or losses), a 29/0/1 advantage ratio over MadDE and AOO

(only 1 function shows no significant difference), 27/0/3 over HSO (3 functions with no significant difference), 24/0/6 over IAGWO and SBO (6 functions with no significant difference), and even against the original SDO, it achieves 20/0/10 (20 functions significantly better, 10 functions no significant difference). This fully demonstrates that the performance improvement of MESDO on the complex high-dimensional CEC2017 test set is statistically significant, not caused by random fluctuations." to enhance language fluency.

On the CEC2022 (d = 10) low-dimensional test set, MESDO's advantages remain prominent. It achieves 12/0/0 all-win results against IWOA and HSO, 11/0/1 against VPPSO and IAGWO (1 function with no significant difference), 9/0/3 against SBO (3 functions with no significant difference), 8/0/4 against MadDE and SDO (4 functions with no significant difference), and 10/0/2 against AOO (2 functions with no significant difference). Although low-dimensional scenarios make it easier to approach theoretical optima, MESDO still demonstrates significant advantages in the vast majority of functions, reflecting its statistical reliability in precise optimization.

On the CEC2022 (d = 20) high-dimensional test set, facing the "curse of dimensionality," MESDO's statistical advantages do not decay. It achieves 12/0/0 all-win against MadDE, 11/0/1 against VPPSO, IAGWO, IWOA, AOO, and HSO (1 function with no significant difference), 10/0/2 against SBO (2 functions with no significant difference), and 9/0/3 against SDO (3 functions with no significant difference), making it the only algorithm that still maintains a large-scale statistically significant advantage in high-dimensional scenarios. This confirms that the synergistic effect of MESDO's three core strategies effectively mitigates performance degradation in high-dimensional optimization.

Considering the non-parametric nature of the Wilcoxon rank-sum test (no requirement for data normality and robustness to outliers), the results in Table 6 further reinforce the credibility of the quantitative data in Tables 3 and 4—MESDO's performance advantages are not only reflected in numerical values but also validated through rigorous statistical testing, providing solid statistical support for its application in practical optimization problems.

3.4.2. Friedman Mean Rank Test

To facilitate a methodological comparison, the overall performance hierarchy of the MESDO algorithm is assessed via the Friedman test. As a distribution-free statistical procedure, the test is designed to determine if significant median differences exist across three or more related datasets. It is especially suited for experiments with a blocking design, providing a robust analytical framework when the prerequisite of normality for ANOVA is not tenable. The computation of its test statistic adheres to Equation (25) [48,50].

$$Q = \frac{12}{nk(k+1)} \sum_{j=1}^{k} R_j^2 - 3n(k+1), \tag{25}$$

Given the block count n, group count k and the j-th group's rank sum R_j, the resulting statistic Q is asymptotically distributed as ${\chi^2}_{(k-1)}$ for large values of n and k.

Table 7 quantifies the average rankings (M.R) and theoretical rankings (T.R) of algorithms across different test sets (CEC2017 d = 30, CEC2022 d = 10, CEC2022 d = 20) using the Friedman test. As a non-parametric statistical method, the Friedman test effectively evaluates the overall performance differences in multiple algorithms across multiple functions. The results indicate that MESDO consistently ranks first in all test scenarios: on the CEC2017 d = 30 test set, MESDO's average ranking is only 1.20 (theoretical ranking 1), far ahead of the second-ranked SDO (3.60) and third-ranked SBO (3.97), while algorithms such as IWOA (7.30) and VPPSO (6.57) are ranked significantly lower; in the low-dimensional CEC2022 d = 10 scenario, MESDO's average ranking is 1.67 (theoretical ranking 1), distinctly above the tied second-ranked MadDE and SDO (both 3.25), with HSO (8.58) ranking last

due to poor adaptation to low-dimensional scenarios; in the high-dimensional CEC2022 d = 20 scenario, MESDO maintains a leading average ranking of 1.33 (theoretical ranking 1), substantially ahead of the second-ranked SDO (3.83) and third-ranked SBO (4.08), while even the relatively well-performing high-dimensional MadDE (4.42) lags behind MESDO by over three ranking levels. This fully demonstrates that MESDO's overall performance remains optimal regardless of dimensionality or function complexity.

Table 7. Friedman mean rank test result.

Suites	**CEC2017**		**CEC2022**			
Dimensions	**30**		**10**		**20**	
Algorithms	***M.R***	***T.R***	***M.R***	***T.R***	***M.R***	***T.R***
VPPSO	6.57	8	5.75	5	6.25	7
MadDE	5.20	5	3.25	2	4.42	4
IAGWO	5.63	6	6.25	8	5.67	6
IWOA	7.30	9	6.08	7	6.83	8
SBO	3.97	3	4.17	4	4.08	3
AOO	5.00	4	6.00	6	5.25	5
HSO	6.53	7	8.58	9	7.33	9
SDO	3.60	2	3.25	2	3.83	2
MESDO	1.20	1	1.67	1	1.33	1

Figure 10's ranking distribution complements the overall conclusions of Table 6 at the individual function level by visualizing each algorithm's ranking across different functions in the CEC2017 and CEC2022 test sets. MESDO ranks first in the vast majority of functions, with only minor ranking fluctuations with SDO or MadDE in very few functions, yet still remains within the top two positions; in contrast, other algorithms show scattered rankings. For example, IWOA ranks mid-level in some unimodal functions but drops to lower ranks in multimodal functions, HSO performs reasonably in low-dimensional functions but falls to the bottom in high-dimensional ones, and SDO, though generally ranked second, shows significant declines in composite and high-dimensional functions, failing to maintain stable performance. This ranking distribution difference further confirms the Friedman test results in Table 6—MESDO's superiority is not dependent on "occasional best" results for individual functions, but reflects stable leading performance across all types and dimensions of functions. The core reason lies in the synergistic effect of the three strategies: the elite-guided search strategy ensures precise search direction, preventing large ranking fluctuations caused by "blind exploration"; the adaptive differential evolution operator strategy enhances global exploration, preventing ranking drops in multimodal and high-dimensional functions; the centroid-based opposition learning boundary control strategy maintains population stability, ensuring consistently top rankings across all functions.

In summary, the overall ranking data in Table 7 and the function-level ranking distribution in Figure 10 mutually corroborate each other, collectively demonstrating that MESDO not only achieves the best overall performance among eight mainstream algorithms but also exhibits stronger performance stability and smaller ranking fluctuations, providing comprehensive performance support from "overall hierarchy" to "individual functions" for its application in complex optimization problems.

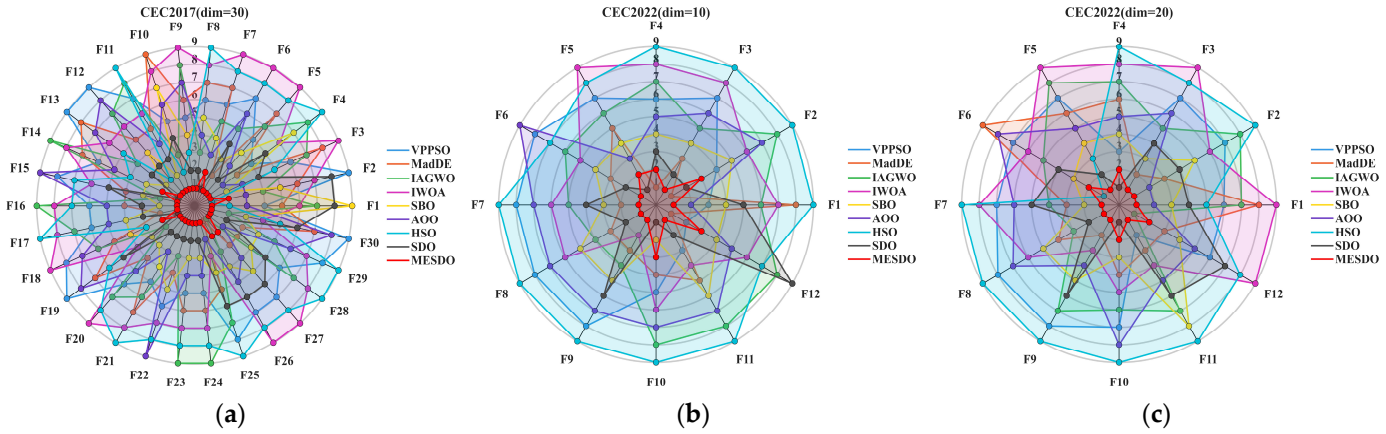


Figure 10. Distribution of rankings of different algorithms.

4. Deployment of Wireless Sensor Network Problems

4.1. WSN Mathematical Model

The perception model of the sensor is shown in Figure 11, where the sensing region and communication range are defined by two concentric circles. The sensing area is modeled as a disk with radius R_S, while the communication radius R_T is set to twice R_S (i.e., $R_T = 2 \times R_S$). According to existing studies [49,52], when the communication range is at least twice the sensing range, the network can ensure full coverage of the convex area while maintaining connectivity among all active nodes. However, this model is based on the assumption of omnidirectional sensing and is not applicable to devices with directional sensing capabilities, such as cameras or ultrasonic sensors.

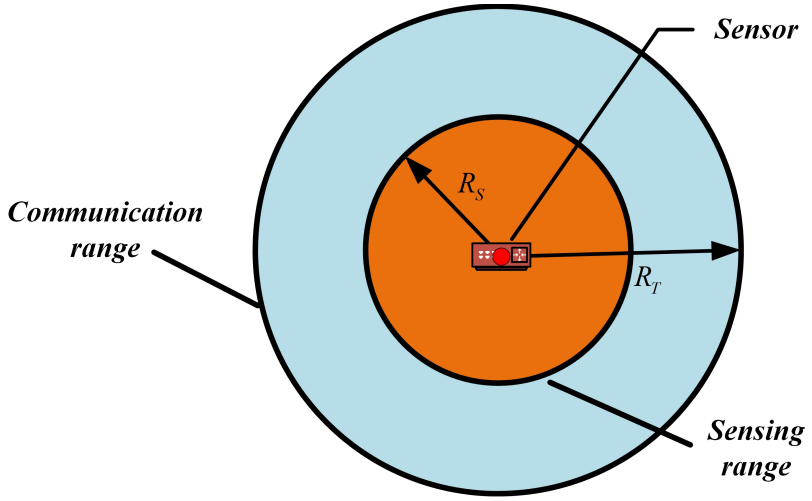


Figure 11. Perception model of sensor node.

Sensor nodes rely on battery power, and their limited operational lifetime constitutes a core constraint in network deployment. To prolong system lifetime, energy-efficient coverage protocols must be adopted. In wireless sensor networks, there are two mainstream coverage models: one is the binary detection model that ignores uncertainty, and the other is a probabilistic sensing model that incorporates randomness, closer to real-world physics. The probability of an event being detected is inversely proportional to the Euclidean distance between the sensor and the event source. Although the binary model is commonly used in theoretical analyses due to its simplicity (assuming that a sensor can deterministically detect everything within its sensing disk), the probabilistic model better

reflects sensor performance in real environments. Based on these considerations, this study adopts the binary detection model proposed in [53,54], defined as follows in Equation (26):

$$P_d = \begin{cases} 1, & if \ \|p_e - p_s\| < R_s \\ 0, & otherwise. \end{cases}, \tag{26}$$

where P_d is the probability that an event occurring at position p_e is detected by a sensor located at p_s, and $\|p_e - p_s\|$ is the Euclidean distance between the sensor and the event.

The objective function in this study primarily considers coverage. Let sensor nodes be $p_{i,j}$ and the deployment space be $s_{i,j}$. The sensor's coverage radius is R_T, and the covered points are denoted as $Cover_{i,j}$. The coverage equation is defined as follows [53]:

$$Cover_{i,j} = \begin{cases} 1, & if \ d(p_{i,j}, s_{i,j}) \le R_T \\ 0, & otherwise. \end{cases}, \tag{27}$$

where $d(p_{i,j}, s_{i,j})$ is the Euclidean distance from sensor $p_{i,j}$ to point $s_{i,j}$, calculated as [53]:

$$d(p_{i,j}, s_{i,j}) = \|s_{i,j} - p_{i,j}\|_2, \tag{28}$$

The coverage probability ratio of the deployment area is defined as:

$$Coverage\ probability = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} Cover_{i,j}}{m \times n}, \tag{29}$$

where m and n are the length and width of the deployment area. The objective of this problem is to maximize network coverage, which constitutes a maximization problem. Therefore, the fitness function is given as [53]:

$$fitness = Coverage\ probability = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} Cover_{i,j}}{m \times n}, \tag{30}$$

4.2. WSN Experiment

The experiments were conducted in a 50×50 simulation environment, deploying 30 sensor nodes with a communication radius of $R_T = 5$. To ensure fair comparison, all algorithms used the same fitness function parameters and were independently run 30 times to eliminate randomness. A comprehensive evaluation framework was established, collecting multiple metrics including minimum, maximum, median, mean, standard deviation, p-value, and ranking, with the optimal values highlighted in **bold** in Table 8. The convergence characteristics and final deployment results are presented in Figure 12 and Figure 13, respectively.

Table 8. Statistics of WSN experiment results.

Algorithm	Min	Median	Max	Ave	Std	p-Value	Rank
VPPSO	72.44%	78.16%	83.40%	78.44%	2.6581×10^2	3.0047×10^{11}	7
MadDE	70.12%	77.88%	82.72%	77.40%	3.0798×10^2	2.9991×10^{11}	8
IAGWO	76.76%	82.36%	84.80%	82.10%	1.6289×10^2	7.8430×10^7	6
IWOA	82.48%	84.18%	86.48%	84.28%	8.8350×10^3	3.0266×10^3	3
SBO	79.52%	83.56%	85.92%	83.54%	1.4225×10^2	3.3614×10^5	5
AOO	69.96%	71.68%	73.12%	71.71%	1.0535×10^2	2.9860×10^{11}	9
HSO	82.00%	83.96%	87.08%	84.09%	1.0553×10^2	3.9818×10^2	4
SDO	81.24%	84.64%	86.76%	84.41%	1.2635×10^2	2.9860×10^5	2
MESDO	**84.80%**	**86.90%**	**88.80%**	**86.80%**	$\mathbf{8.1308 \times 10^3}$	/	1

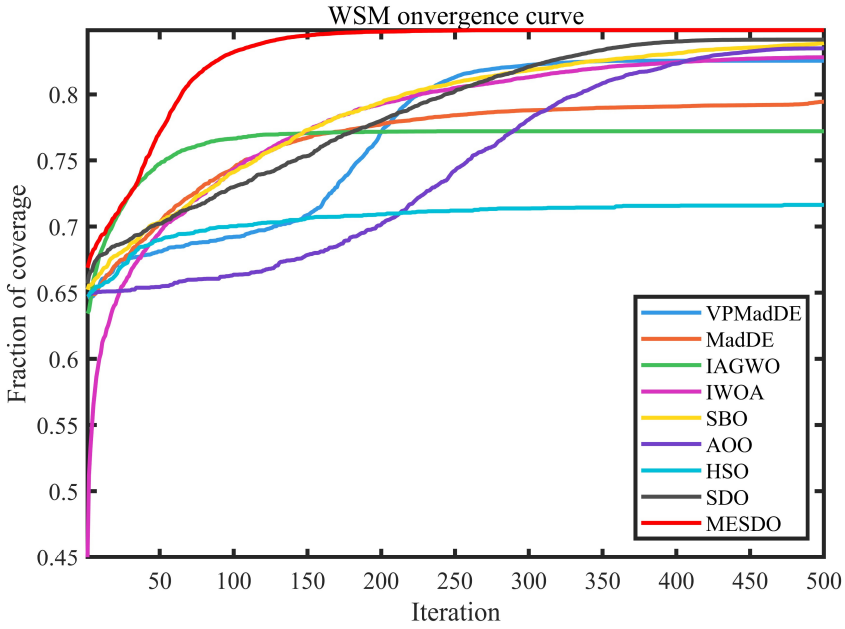


Figure 12. Fitness value curves of different algorithms in WSN.

Table 8 quantifies the WSN deployment performance of each algorithm using metrics such as minimum coverage (Min), median coverage (Median), maximum coverage (Max), average coverage (Ave), standard deviation (Std), *p*-value, and ranking. The data show that MESDO achieves the best performance across all metrics: in terms of average coverage, MESDO reaches 86.80%, which is 2.39 percentage points higher than the second-ranked SDO (84.41%), 2.52 and 2.71 percentage points higher than IWOA (84.28%) and HSO (84.09%), respectively, and exceeds AOO (71.71%) and MadDE (77.40%) by more than 15 percentage points; for minimum coverage, MESDO achieves 84.80%, significantly higher than SDO (81.24%) and IWOA (82.48%), and 21.21 percentage points higher than AOO (69.96%), demonstrating that MESDO maintains high coverage even in the worst deployment scenarios; in terms of median coverage, MESDO reaches 86.90%, 2.26 percentage points higher than SDO (84.64%), while all other compared algorithms have median coverage below 85%, further highlighting MESDO's superiority. Meanwhile, MESDO's standard deviation is only 8.1308×10^3, lower than SDO (1.2635×10^2) and IWOA (8.8350×10^3), indicating minimal fluctuation across 30 independent runs and excellent stability. The *p*-value results show that MESDO's performance differences with all other algorithms are statistically significant (all other algorithms have *p*-values far below 0.05), ultimately securing the top ranking with absolute advantage.

Figure 12 presents the fitness value curves (with coverage as the core fitness metric), providing a dynamic convergence perspective that further confirms the quantitative conclusions in Table 8: during the early iterations (1–50 generations), MESDO's fitness rapidly rises above 82%, with convergence significantly faster than other algorithms—SDO requires 80 generations to reach 82%, IWOA and HSO around 100 generations, while AOO, MadDE, VPPSO, and others still remain below 80% at 100 generations; in the middle iterations (50–200 generations), MESDO's fitness steadily increases from 82% to 86% without noticeable stagnation, whereas SDO, IWOA, and other algorithms enter a slow-convergence phase, with coverage improvement less than 2 percentage points, and AOO and MadDE even exhibit stagnation; in the late iterations (200–500 generations), MESDO's fitness stabilizes around 86.80%, reaching convergence, while the final coverage of other algorithms remains below 85%, with AOO only stabilizing around 71.71%, showing a significant gap. This convergence difference stems from the targeted roles of MESDO's three core strategies in WSN deployment: the elite-guided search strategy quickly locks high-coverage node

deployment patterns via a dynamic elite pool, accelerating early-stage convergence; the adaptive differential evolution operator strategy prevents the algorithm from falling into "local optimal deployments" (e.g., clustered nodes creating coverage blind spots), ensuring continued coverage improvement in the mid-stage; the centroid-based opposite learning boundary control strategy adjusts out-of-bound nodes, maintains population diversity, and guarantees stable convergence to the optimal deployment in the late stage.

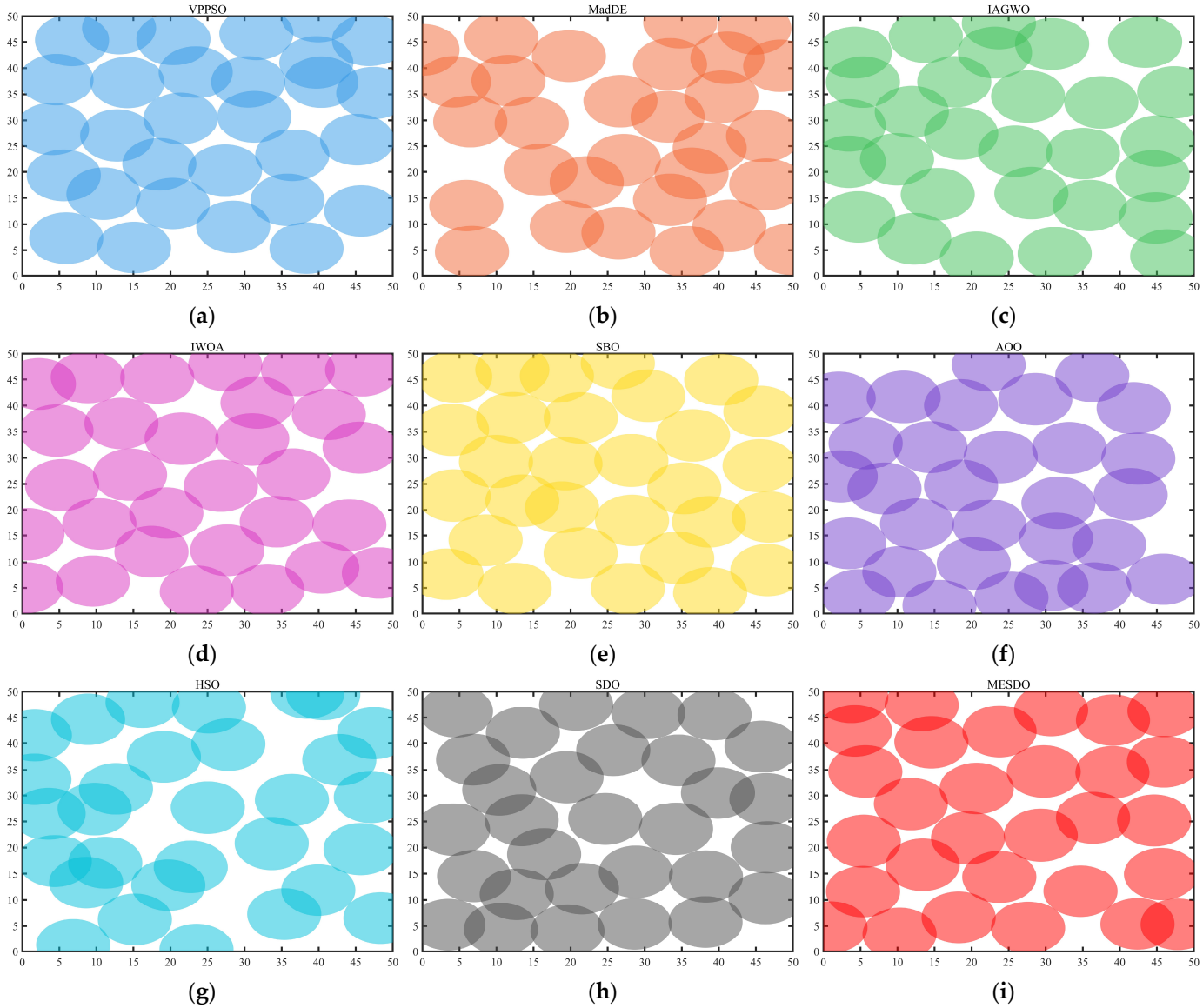


Figure 13. Node deployment distributions obtained by different algorithms.

Figure 13 shows the node deployment distribution, visually highlighting MESDO's deployment advantages from a practical application perspective: the nodes generated by MESDO are evenly distributed and densely cover the entire 50×50 simulation area, with no obvious coverage blind spots, and inter-node distances are reasonable (mostly within the effective communication radius $R_T = 5$). This arrangement avoids resource waste caused by excessive node clustering while preventing coverage gaps from overly sparse placement. In contrast, the deployment results of other algorithms exhibit clear deficiencies: VPPSO and MadDE produce scattered node distributions, with dense overlaps in some areas and large coverage blind spots at the edges of the simulation region; IAGWO and IWOA have relatively uniform overall distributions but still show small blind spots in corner areas, and

inter-node communication stability is insufficient; SDO, as the original algorithm, performs better than most comparison algorithms but still suffers from local node clustering and lower coverage than MESDO; AOO exhibits the worst distribution, with many nodes concentrated at the center and almost no coverage at the edges, leading to a significantly reduced overall coverage. These deployment differences fundamentally reflect MESDO's ability to optimize "node placement rationality" and "coverage completeness," further validating the reliability of the quantitative data in Table 7 and the convergence curves in Figure 12—MESDO not only achieves superior numerical metrics but also generates better node distribution schemes in actual WSN deployments, satisfying the core requirements of network coverage and connection stability.

In summary, Table 8, Figures 12 and 13 collectively corroborate MESDO's superiority in WSN node deployment optimization across three dimensions: quantitative statistics, dynamic convergence, and practical deployment. MESDO outperforms eight mainstream comparison algorithms in terms of higher coverage accuracy, faster convergence, stronger performance stability, and the ability to generate more uniform, blind-spot-free node deployment schemes, providing a practical and effective optimization method for efficient WSN deployment.

5. Summary and Prospect

This paper addresses the optimization requirements for wireless sensor network (WSN) deployment, focusing on the limitations of the traditional Supply-Demand Optimization (SDO) algorithm, including blind search, weak global exploration, and coarse boundary handling. We propose a Multi-strategy Enhanced Supply-Demand Optimization (MESDO) algorithm, which synergistically integrates three core strategies: an elite-guided search strategy dynamically constructs an elite pool to guide price equilibrium point selection, addressing the blind search direction issue in SDO; an adaptive differential evolution operator strategy dynamically adjusts the mutation factor and leverages the global best solution to enhance global exploration; and a centroid-based opposite learning boundary control strategy corrects out-of-bound solutions using the population centroid, improving the utilization of effective information. MESDO was validated using the CEC2017 and CEC2022 benchmark test suites. The results demonstrate that MESDO consistently achieves optimal performance across unimodal, multimodal, hybrid, and composite functions, effectively avoiding premature convergence. Statistical tests (Wilcoxon rank-sum test and Friedman test) further confirm the synergistic value of the three strategies, with MESDO achieving the top average ranking in all test scenarios, and its performance advantages are statistically significant.

For engineering application validation, MESDO was applied to WSN node deployment in a 50×50 simulation environment. The results show an average coverage rate of 86.80%, which is 2.39 percentage points higher than SDO (84.41%); the minimum coverage rate reaches 84.80%, 21.21 percentage points higher than AOO (69.96%); and the standard deviation of 8.1308×10^3 is lower than all comparison algorithms, indicating superior and more stable deployment performance. Convergence curves reveal that MESDO achieves 82% coverage within 50 iterations, significantly faster than SDO (80 iterations) and IWOA (100 iterations). The node deployment distribution map intuitively shows that MESDO generates a uniform deployment with no noticeable coverage blind spots, effectively avoiding the clustering or sparse placement problems observed in other algorithms (e.g., VPPSO, AOO), fully demonstrating MESDO's practical value in real-world engineering scenarios.

Compared to existing multi-strategy optimization algorithms in WSN deployment (such as AHDGWO integrated with differential evolution and NHBBWO incorporating dynamic parameters), MESDO's unique advantage lies in its targeted design of synergistic

strategies: while AHDGWO enhances local exploitation, its reliance on a single hybrid strategy constrains global exploration; NHBBWO's dynamic parameter adjustment is only effective for specific scenarios. In contrast, MESDO achieves three-layer strategy synergy through "elite guidance (precise positioning)—adaptive differential evolution (global expansion)—centroid-based boundary control (diversity stabilization)". This approach not only resolves the imbalance in traditional multi-strategy algorithms that either overemphasize local optimization at the expense of global search or prioritize exploration over stability, but also ensures consistent performance transfer from function optimization to engineering deployment. Its adaptability to complex functions demonstrated in CEC benchmark tests directly translates into precise optimization of coverage blind spots and convergence speed in WSN deployment—a dual capability rarely achieved by most existing algorithms.

Although MESDO has achieved remarkable results in function optimization and WSN deployment, integrating research findings with practical requirements suggests three potential directions for future research: (1) Extending multi-objective optimization capabilities: Introduce classical non-dominated sorting mechanisms such as the Non-dominated Sorting Genetic Algorithm II (NSGA-II) to construct a multi-objective optimization framework that balances coverage rate, node energy consumption, network connectivity, and other multi-objective requirements in WSNs, thereby addressing the limitations of single-objective optimization in handling multiple engineering constraints; (2) Enhancing adaptability to dynamic and 3D scenarios: Develop dynamic population update strategies (e.g., population reconstruction mechanisms based on environmental changes) and 3D boundary models to accommodate dynamic scenarios such as node mobility and failure, as well as three-dimensional deployment environments like mountainous areas and buildings, thereby improving the algorithm's applicability in complex real-world scenarios; (3) Exploring cross-domain applications: Apply MESDO to fields such as microgrid scheduling (optimizing energy distribution efficiency) and image segmentation (improving segmentation accuracy and speed) to further validate the algorithm's generality. Meanwhile, strategy details (e.g., adjusting the mutation factor range in differential evolution, optimizing the elite pool update frequency) should be refined according to the characteristics of different domain problems to promote the algorithm's practical implementation in engineering.

Author Contributions: Conceptualization, B.W. and Y.Y. (Yuchen Yan); Methodology, B.W. and Y.Y. (Yuchen Yan); Software, B.W. and Y.Y. (Yuchen Yan).; Validation, B.W. and Y.Y. (Yuchen Yan); Formal analysis, B.W. and Y.Y. (Yuchen Yan); Investigation, L.Z. and S.Y.; Resources, L.Z. and S.Y.; Data curation, L.Z. and S.Y.; Writing—original draft, L.Z. and S.Y.; Writing—review and editing, L.Z. and S.Y.; Visualization, Y.Y. (Yangjian Yang); Supervision, Y.Y. (Yangjian Yang); Project administration, Y.Y. (Yangjian Yang). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Uthayakumar, C.; Jayaraman, R.; Raja, H.A.; Shabbir, N. QSEER-Quantum-Enhanced Secure and Energy-Efficient Routing Protocol for Wireless Sensor Networks (WSNs). *Sensors* **2025**, *25*, 5924. [CrossRef]
2. Shafik, W. Wireless sensor network-assisted fuzzy sink based model. *Energy* **2025**, *333*, 37512. [CrossRef]
3. Karunkuzhali, D.; Pradeep, S.; Sungheetha, A.; Basha, T.S.G. Data-Aggregation-Aware Energy-Efficient in Wireless Sensor Networks Using Multi-Stream General Adversarial Network. *Trans. Emerg. Telecommun. Technol.* **2025**, *36*, e70017. [CrossRef]

4. Suganthi, S.; Umapathi, N.; Venkateswaran, N.; Rajarajan, S. Enhancing energy efficiency in wireless sensor networks via predictive model for node status classification and coverage integrity using relational Bi-level aggregation graph convolutional network. *Expert Syst. Appl.* **2025**, *296*, 129029. [CrossRef]
5. Perez, J.; Santamaria, I.; Pagés-Zamora, A. Blind learning of the optimal fusion rule in wireless sensor networks. *Signal Process.* **2025**, *239*, 110238. [CrossRef]
6. Lian, J.; Yao, H. Joint Deployment of Sensors and Chargers in Wireless Rechargeable Sensor Networks. *Energies* **2024**, *17*, 3130. [CrossRef]
7. Lai, W.-Y.; Hsiang, T.-R. Wireless Charging Deployment in Sensor Networks. *Sensors* **2019**, *19*, 201. [CrossRef] [PubMed]
8. Zhang, C.; Zhao, G. Wireless Powered Sensor Networks With Random Deployments. *IEEE Wirel. Commun. Lett.* **2017**, *6*, 218–221. [CrossRef]
9. Kundu, M.; Kanjilal, R.; Uysal, I. Intelligent Clustering and Adaptive Energy Management in Wireless Sensor Networks with KDE-Based Deployment. *Sensors* **2025**, *25*, 2588. [CrossRef]
10. Di Puglia Pugliese, L.; Guerriero, F.; Mitton, N. Optimizing wireless sensor networks deployment with coverage and con-nectivity requirements. *Ann. Oper. Res.* **2025**, *346*, 1997–2008. [CrossRef]
11. Zhou, P.; Kan, M.; Chen, W.; Wang, Y.; Cao, B. An adaptive coverage method for dynamic wireless sensor network deployment using deep reinforcement learning. *Sci. Rep.* **2025**, *15*, 30304. [CrossRef]
12. Xing, Y.-X.; Wang, J.-S.; Zhang, S.-W.; Zhang, S.-H.; Ma, X.-R.; Zhang, Y.-H. Transit search algorithm based on oscillation exploitation factor and Roche limit for wireless sensor network deployment optimization. *Artif. Intell. Rev.* **2024**, *58*, 29. [CrossRef]
13. Zhu, J.; Rong, J.; Gong, Z.; Liu, Y.; Li, W.; Alqahtani, F.; Tolba, A.; Hu, J. Deployment optimization in wireless sensor networks using advanced artificial bee colony algorithm. *Peer-to-Peer Netw. Appl.* **2024**, *17*, 3571–3582. [CrossRef]
14. Niu, H.; Li, Y.; Zhang, C.; Chen, T.; Sun, L.; Abdullah, M.I. Multi-Strategy Bald Eagle Search Algorithm Embedded Orthogonal Learning for Wireless Sensor Network (WSN) Coverage Optimization. *Sensors* **2024**, *24*, 6794. [CrossRef]
15. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, November 27–December 1 1995; Volume 4, pp. 1942–1948.
16. Dorigo, M.; Birattari, M.; Stützle, T. Ant Colony Optimization. *Comput. Intell. Mag. IEEE* **2006**, *1*, 28–39. [CrossRef]
17. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
18. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
19. Hashim, F.A.; Hussien, A.G. Snake Optimizer: A novel meta-heuristic optimization algorithm. *Knowl.-Based Syst.* **2022**, *242*, 108320. [CrossRef]
20. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [CrossRef]
21. Fu, Y.; Liu, D.; Chen, J.; He, L. Secretary bird optimization algorithm: A new metaheuristic for solving global optimization problems. *Artif. Intell. Rev.* **2024**, *57*, 123. [CrossRef]
22. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper Optimisation Algorithm: Theory and application. *Adv. Eng. Softw.* **2017**, *105*, 30–47. [CrossRef]
23. Jia, H.; Peng, X.; Lang, C. Remora optimization algorithm. *Expert Syst. Appl.* **2021**, *185*, 115665. [CrossRef]
24. Gao, Y.; Wang, J.; Li, C. Escape after love: Philoponella prominens optimizer and its application to 3D path planning. *Clust. Comput.* **2024**, *28*, 81. [CrossRef]
25. Hayyolalam, V.; Kazem, A.A.P. Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103249. [CrossRef]
26. Xue, J.; Shen, B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *J. Supercomput.* **2022**, *79*, 7305–7336. [CrossRef]
27. Ghasemi, M.; Khodadadi, N.; Trojovský, P.; Li, L.; Mansor, Z.; Abualigah, L.; Alharbi, A.H.; El-Kenawy, E.-S.M. Kirchhoff's law algorithm (KLA): A novel physics-inspired non-parametric metaheuristic algorithm for optimization problems. *Artif. Intell. Rev.* **2025**, *58*, 325. [CrossRef]
28. Yuting, Y.; Yuelin, G. An adaptive hybrid differential Grey Wolf Optimization algorithm for WSN coverage. *Clust. Comput.* **2025**, *28*, 229. [CrossRef]
29. Chen, X.; Zhang, M.; Yang, M.; Wang, D. NHBBWO: A novel hybrid butterfly-beluga whale optimization algorithm with the dynamic strategy for WSN coverage optimization. *Peer-to-Peer Netw. Appl.* **2025**, *18*, 80. [CrossRef]
30. Zhang, Y.; Zhang, Z.; Liu, D.; Zheng, P.; Zhong, Z. Adaptive learning FOA algorithm with energy consumption balancing for coverage optimization in WSNs. *Ad Hoc Netw.* **2025**, *178*, 103958. [CrossRef]
31. Zhaohui, Z.; Jiaqi, Z.; Jing, L. Q-learning-based semi-fixed clustering routing algorithm in WSNs. *Ad Hoc Netw.* **2025**, *174*, 103837. [CrossRef]
32. Liu, X.; Zhang, K.; Zhang, X.; Fiumara, G.; De Meo, P. A hybrid improved compressed particle swarm optimization WSN node location algorithm. *Phys. Commun.* **2024**, *67*, 102490. [CrossRef]

33. Wang, J.; Zhu, Z.; Zhang, F.; Liu, Y. An improved salp swarm algorithm for solving node coverage optimization problem in WSN. *Peer-to-Peer Netw. Appl.* **2024**, *17*, 1091–1102. [CrossRef]
34. Zhao, W.; Wang, L.; Zhang, Z. Supply-Demand-Based Optimization: A Novel Economics-Inspired Algorithm for Global Optimization. *IEEE Access* **2019**, *7*, 73182–73206. [CrossRef]
35. Daqaq, F.; Hassan, M.H.; Kamel, S.; Hussien, A.G. A leader supply-demand-based optimization for large scale optimal power flow problem considering renewable energy generations. *Sci. Rep.* **2023**, *13*, 14591. [CrossRef]
36. Ginidi, A.R.; Shaheen, A.M.; El-Sehiemy, R.A.; Elattar, E. Supply demand optimization algorithm for parameter extraction of various solar cell models. *Energy Rep.* **2021**, *7*, 5772–5794. [CrossRef]
37. Yin, P.; Luo, Z.; Pouramini, S. Case study on multi-objective Modified Supply-Demand-based Optimization Algorithm for energy-efficient building retrofitting. *Sustain. Cities Soc.* **2024**, *114*, 105734. [CrossRef]
38. Wu, G.; Mallipeddi, R.; Suganthan, P. *Problem Definitions and Evaluation Criteria for the CEC 2017 Competition and Special Session on Constrained Single Objective Real-Parameter Optimization*; Elsevier: Amsterdam, The Netherlands, 2016.
39. Luo, W.; Lin, X.; Li, C.; Yang, S.; Shi, Y. Benchmark functions for CEC 2022 competition on seeking multiple optima in dynamic environments. *arXiv* **2022**, arXiv:2201.00523. [CrossRef]
40. Shami, T.M.; Mirjalili, S.; Al-Eryani, Y.; Daoudi, K.; Izadi, S.; Abualigah, L. Velocity pausing particle swarm optimization: A novel variant for global optimization. *Neural Comput. Appl.* **2023**, *35*, 9193–9223. [CrossRef]
41. Biswas, S.; Saha, D.; De, S.; Cobb, A.D.; Das, S.; Jalaian, B.A. Improving differential evolution through Bayesian hyperparameter optimization. In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021; IEEE: New York City, NY, USA, 2021; pp. 832–840.
42. Yu, M.; Xu, J.; Liang, W.; Qiu, Y.; Bao, S.; Tang, L. Improved multi-strategy adaptive Grey Wolf Optimization for practical engineering applications and high-dimensional problem solving. *Artif. Intell. Rev.* **2024**, *57*, 277. [CrossRef]
43. Huang, S.; Liu, D.; Fu, Y.; Chen, J.; He, L.; Yan, J.; Yang, D. Prediction of Self-Care Behaviors in Patients Using High-Density Surface Electromyography Signals and an Improved Whale Optimization Algorithm-Based LSTM Model. *J. Bionic Eng.* **2025**, *22*, 1963–1984. [CrossRef]
44. Wang, J.; Chen, Y.; Lu, C.; Heidari, A.A.; Wu, Z.; Chen, H. The status-based optimization: Algorithm and comprehensive performance analysis. *Neurocomputing* **2025**, *647*, 130603. [CrossRef]
45. Wang, R.-B.; Hu, R.-B.; Geng, F.-D.; Xu, L.; Chu, S.-C.; Pan, J.-S.; Meng, Z.-Y.; Mirjalili, S. The Animated Oat Optimization Algorithm: A nature-inspired metaheuristic for engineering optimization and a case study on Wireless Sensor Networks. *Knowl. -Based Syst.* **2025**, *318*, 113589. [CrossRef]
46. Akbari, E.; Rahimnejad, A.; Gadsden, S.A. Holistic swarm optimization: A novel metaphor-less algorithm guided by whole population information for addressing exploration-exploitation dilemma. *Comput. Methods Appl. Mech. Eng.* **2025**, *445*, 118208. [CrossRef]
47. Yang, W.; Lai, T.; Fang, Y. Multi-Strategy Golden Jackal Optimization for engineering design. *J. Supercomput.* **2025**, *81*, 609. [CrossRef]
48. Zhang, Y.; Wang, J.; Zhang, X.; Wang, B. ACPOA: An Adaptive Cooperative Pelican Optimization Algorithm for Global Optimization and Multilevel Thresholding Image Segmentation. *Biomimetics* **2025**, *10*, 596. [CrossRef] [PubMed]
49. Wu, Q. An enhanced whale optimization algorithm with inertia weight and dynamic parameter adaptation for wireless sensor network deployment. *Computing* **2025**, *107*, 167. [CrossRef]
50. Cao, L.; Wei, Q. SZOA: An Improved Synergistic Zebra Optimization Algorithm for Microgrid Scheduling and Management. *Biomimetics* **2025**, *10*, 664. [CrossRef]
51. Shen, Y.; Zhang, C.; Gharehchopogh, F.S.; Mirjalili, S. An improved whale optimization algorithm based on multi-population evolution for global optimization and engineering design problems. *Expert Syst. Appl.* **2022**, *215*, 119269. [CrossRef]
52. Wang, Z.; Xie, H. Wireless Sensor Network Deployment of 3D Surface Based on Enhanced Grey Wolf Optimizer. *IEEE Access* **2020**, *8*, 57229–57251. [CrossRef]
53. Meng, Q.; Kuang, X.; Yu, Z.; He, M.; Cui, H. Augmented secretary bird optimization algorithm for wireless sensor network deployment and engineering problem. *PLoS ONE* **2025**, *20*, e0329705. [CrossRef]
54. Wang, W.; Lyu, L. Adaptive Tasmanian Devil Optimizer for Global Optimization and Application in Wireless Sensor Network Deployment. *IEEE Access* **2024**, *12*, 72382–72407. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.