Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 9593103, 32 pages
https://doi.org/10.1155/2022/9593103

Research Article

Multistrategy Integrated Marine Predator Algorithm Applied to 3D Surface WSN Coverage Optimization

Zhendong Wang, Hang Xiao, Shuxin Yang, Junling Wang, and Soroosh Mahmoodi

School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China

Correspondence should be addressed to Hang Xiao; xiaohang@mail.jxust.edu.cn

Received 27 May 2022; Revised 13 October 2022; Accepted 26 October 2022; Published 17 November 2022

Academic Editor: Tien-Wen Sung

Achieving maximum network coverage with a limited number of sensor nodes is key to node deployment of wireless sensor network (WSN). This paper proposes an improved marine predator algorithm (IMPA) for 3D surface wireless sensor network deployment. A population evolution strategy based on random opposition-based learning and differential evolution operator is proposed to enrich the population diversity and improve the global search capability of the algorithm. The grouping idea of the Shuffled Frog Leaping Algorithm (SFLA) is then introduced. A local search strategy based on the SFLA is proposed to replace the FADs effect of MPA and enhance the ability of the algorithm to escape from the local optimum. A quasireflected opposition-based learning strategy is also presented to improve the optimization accuracy, accelerate the convergence speed of the algorithm, and improve the quality of the solution. Fifteen benchmark functions are selected for testing. The results are compared with seven different algorithms. The results show that the improved algorithm has excellent optimization performances. Finally, the IMPA is applied to optimize WSN coverage on 3D surfaces. The experimental results show that the proposed IMPA has good terrain adaptation and optimal deployment capabilities. It can improve the coverage of the network, reduce the deployment cost, and extend the network life cycle.

1. Introduction

Wireless sensor network (WSN) is a type of network composed of sensor nodes deployed in the monitoring area through self-organization and multihop, which can sense, process, and transmit the information of the perceived region in real-time. WSN has been widely applied to military action, space exploration, environmental science, smart home, fine agriculture, emergency rescue, and other fields [1–5]. The coverage problem is the first fundamental issue faced by WSN configuration. The sensor nodes may be arbitrarily distributed in the configuration area, which reflects the status of a certain area of the WSN being monitored and tracked. And it involves communication quality, network life, and other issues. So, it directly affects the service quality of the entire network [6]. The sensor has a very limited battery due to the size of the sensor itself. Especially in the harsh deployment area, it is not feasible to replenish the energy of the sensors by manual means. Under certain coverage conditions, it is necessary to reasonably utilize the energy of the nodes to extend the network life cycle. As a result, achieving maximum network coverage with a limited number of sensor nodes becomes a key within the sensor node deployment.

At present, the coverage optimization algorithms for traditional 2D planes are very well developed. However, they cannot meet realistic and complex environments [7–12]. Compared with the 2D deployment environment, the 3D sensor node deployment is more complex and the network maintenance is more difficult. In addition, the existing 2D coverage methods cannot be directly applied in the 3D environment, otherwise a large number of coverage holes will be generated. Thus, the problem of WSN node deployment based on the 3D environments has become a vital research [13–15]. WSN node deployment for 3D environments can be divided into two categories. One category is node deployment in 3D space, such as smart cities [16], smart factories [17], and underwater wireless networks [18]. The other category is node deployment of WSNs on 3D surfaces, such as node deployment on mountains [19]. The WSN coverage of 3D surfaces as a particular coverage scenario for the 3D environments has been little studied in this aspect. However, there are still many scholars who have proposed appropriate coverage methods.

Swarm intelligence algorithm is an effective method for solving many practical problems [20–22], and it has been used by many scholars to solve the 3D surface coverage optimization problem. The marine predator algorithm (MPA) as a swarm intelligence algorithm was proposed by Faramarzi et al. [23] in 2020. The MPA was proposed to simulate the predatory behavior of marine organisms. Their research was inspired by the different hunting methods of ocean life and the speed of movement between predator and prey. Although the MPA has significant advantages in optimization problems and engineering applications, it still has some disadvantages such as the low solution accuracy, the slow convergence, and the poor local exploitation capability. In order to maximize the 3D surface WSN coverage and prolong the network life cycle, aiming at the defects of the MPA, this paper proposes a multistrategy integrated improved marine predator algorithm called IMPA. In the case of fewer sensor nodes, it can effectively improve the network coverage and extend the network life cycle.

This algorithm introduces a random opposition-based learning strategy and differential evolution operator to enrich the population diversity and expand the population search range. A grouping idea of the Shuffled Frog Leaping Algorithm (SFLA) is proposed to improve the exploitation ability of the algorithm. The global information and local search within the group are employed to guide the worst individual to approach the optimal individual. It can accelerate the convergence of the algorithm and reduce the risk of falling into the local optimum issues. To further speed up the convergence of the algorithm, a quasireflected opposition-based learning (QOBL) mechanism is proposed. And the boundary strategy is improved to enhance the quality of the solution. The algorithm is compared with the MSIMPA [24], the MPA, the Enhanced Sparrow Search Algorithm (ESSA) [25], the Improved Grey Wolf Optimizer (IGWO) [26], the Whale Optimization Algorithm (WOA) [27], the Particle Swarm Optimization (PSO) [28], and the Differential Evolution Algorithm (DE) [29] on 15 benchmark functions. The comparisons verify a strong optimization performance of the proposed algorithm. Finally, the proposed algorithm is applied to the WSN coverage optimization of two 3D terrains. The simulation results show that the IMPA can effectively improve the coverage of nodes, save the deployment cost of the network, and improve the network life cycle, which further verifies the practicability and effectiveness of the algorithm. In this study, the main contributions can be summarized as follows:

(1) Most researchers calculate coverage using a 0-1 perception model, which is overly idealized in 3D environment. Therefore, this paper proposes a probabilistic coverage model suitable for 3D surfaces

(2) A population evolution strategy based on random opposition-based learning and differential evolution operator is proposed to enrich the population diversity while improving the global search ability of the algorithm

(3) A local search strategy based on SFLA is proposed to replace the local search mechanism of the MPA to enhance the algorithm's local exploitation capability and the ability to jump out of the local optimum

(4) A novel 3D surface WSN coverage optimization technique based on multistrategy integrated improved marine predator algorithm is presented. Simulation experiments are conducted in two 3D terrains and the method has higher coverage and longer life cycle compared to other four algorithms

The rest of the paper is organized as follows. Section 2 presents the current research status and related work on 3D surface coverage and MPA algorithms. Section 3 introduces the relevant models for WSN node deployment on 3D surfaces. Section 4 describes the proposed IMPA algorithm in detail. Section 5 compares and analyzes the algorithms on the benchmark test functions. Section 6 presents the application of IMPA for WSN coverage in two 3D terrains. Finally, Section 7 concludes and proposes future research.

2. Related Work

At present, the research on WSN coverage control in 3D environment is a hot topic. For the WSN coverage optimization problem on 3D surfaces, some scholars use traditional techniques to improve coverage and ensure network service quality. For example, Anand et al. proposed a node coverage method applicable to 3D surfaces based on the terrain segmentation method of the Voronoi diagram [30]. Their simulation results demonstrated improvement in the algorithm's efficacy. A 3D deployment method based on the virtual force algorithm was proposed [31]. The simulation results show that the algorithm can be adapted to all types of surfaces and achieve full coverage while maintaining network connectivity. Kim [32] introduced a mobile sensor network for the coverage problem in mountainous terrain and proposed a 3D coverage method by considering the physical characteristics such as slopes in mountainous terrain. Simulation experimental results show that the method is significantly improved in terms of network coverage and coverage time. Full coverage of the monitoring area can be achieved by deploying an appropriate number of sensors. However, the method is not applicable in large-scale node deployment scenarios. These traditional techniques have provided important solutions and valuable research results for solving the coverage optimization problems of WSN, but there are some shortcomings. For example, the structure of some algorithms is too complex, which makes it difficult to meet the service requirements of users in practical deployment.

Since the swarm intelligence optimization algorithm has shown excellent results in various optimization problems and engineering applications, it opens up a new path for the research of coverage optimization of WSN, many researchers have utilized the swarm intelligence algorithm to optimize the WSN coverage of 3D surfaces [33–35]. For example, Li et al. [34] proposed a Modified Parallel Tunicate Swarm Algorithm (MPTSA) based on the improved traditional parallel strategy and applied it to the 3D surface coverage optimization. Their simulations and experimental results show the MPTSA can effectively improve the coverage. However, this method does not consider the lifetime of the network. Literature [36] proposed an innovative method for determining the 3D surface perception blind area for the 3D spatial target coverage problem. A differential evolution algorithm was employed to solve the 3D surface nodes deployment problem to realize full coverage of the monitoring area with the least number of nodes. However, the proposed method for judging the 3D perception blind area ignored the limitations where both sensing nodes and monitoring points are located on the terrain surface. Wang and Xie [37] proposed a more comprehensive method for determining the 3D surface perception blind area based on the mentioned limitations of perceptual blind area determination. They also innovatively enhanced the grid method commonly used in 2D WSN coverage to make it applicable to the coverage calculation of 3D surfaces. Moreover, an Enhanced Grey Wolf Optimizer (EGWO) using a hybrid inner and outer layer location update strategy was proposed and employed to the deployment optimization of 3D surfaces. The experimental results show that the EGWO has excellent deployment ability. However, the coverage perception model of this work is a binary perception model, which is too ideal in complex 3D terrain.

Currently, the research on MPA focuses on improving and applying the algorithm itself. Houssein et al. [38] proposed a MPAOBL-GWO algorithm incorporating the opposition-based learning and the Grey Wolf Optimizer (GWO). The GWO was applied as the local search method of MPA. It improved the exploitation capability of the algorithm and avoided falling into the local optimum issues. However, the convergence speed of the algorithm still needs to be improved. Abdel-Basset et al. [39] proposed an improved marine predator algorithm (IMPA) by using the improved population strategy based on the population's average fitness. This strategy enhanced the performance and convergence speed of the algorithm. However, the risk of falling into the local optimum searching for individuals while reducing the population diversity was increased. Literature [40] introduced a differential evolution operator into the MPA. It proposed an enhanced marine predator algorithm (EMPA) for identifying static and dynamic photovoltaic model parameters. The EMPA maintained the diversity of the population in the search process and increased the probability of the algorithm away from the local optimum. However, the enhancement of the optimization accuracy of the improved algorithm was very limited. In literature [24], a multistrategy improved marine predator algorithm (MSIMPA) was proposed. It employed the chaotic mapping and opposition-based learning strategy to improve the population's generation quality within the initial stage. Moreover, a grouping learning strategy to crossover the population operations was proposed to improve the algorithm's optimization accuracy and convergence speed. The simulation results show that the algorithm could effectively improve the deployment coverage of the 2D WSN.

Based on the above related works, some achievements have been achieved in the study of WSN coverage optimization for 3D surfaces and the improvement of the MPA algorithm. However, there are some shortcomings in these studies. For example, the coverage perception model is a binary model, which is too idealized in practical applications. In addition, since the 3D surface is more complex compared to the 2D plane, the coverage of some algorithms still needs to be further improved. Therefore, a lot of research is still needed for coverage optimization of 3D surface WSN to achieve higher coverage while obtaining a longer network life cycle using fewer sensor nodes. To address the above issues, this paper proposes a coverage optimization method based on multistrategy integrated improved marine predator algorithm.

3. 3D Surface WSN Coverage Model Description

3.1. Perceptual Model of Sensor Nodes. In WSN, suppose that the nodes of a wireless sensor are a set $S = \{s_1, s_2, \cdots, s_n\}$ where the 3D coordinate of s_j is denoted by (x_j, y_j, z_j) that $j = 1, 2, \cdots, n$. The set of the monitored nodes within the monitoring area is $M = \{m_1, m_2, \cdots, m_n\}$, where the 3D coordinate of m_k is denoted by (x_k, y_k, z_k) that $k = 1, 2, \cdots, n$. The sensing and communication radiuses of each node are R^s and R^c, respectively. The Euclidean distance between node s_j and node m_k can be defined by

$$\operatorname{dis}(s_j, m_k) = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2 + (z_j - z_k)^2}. \quad (1)$$

The common 0-1 perception model is mainly used in the WSN coverage of the 2D environment. In the 3D surface environment, it is necessary to establish a coverage model that is closer to the real world. Therefore, this paper adopts the probabilistic perception model. The probability of the monitoring node m_k can be perceived by the node s_j in the following:

$$P_{\text{cov}}(s_j, m_k) = \begin{cases} \dfrac{1}{(1 + \alpha * \operatorname{dis}(s_j, m_k))^{\beta}}, & \operatorname{dis}(s_j, m_k) \le R^s, \\ 0, & \operatorname{dis}(s_j, m_k) > R^s, \end{cases} \quad (2)$$

where α and β are type parameters related to the physical characteristics of the sensor. Usually, the value of β is from 1 to 4 and α is an adjustable parameter.

Therefore, the joint perception probability of all nodes in the set S to monitor node m_k can be evaluated by

$$P_C(s_{\text{all}}, m_k) = 1 - \prod_{i=1}^{n}\left(1 - P_{\text{cov}}(s_j, m_k)\right), \tag{3}$$

where s_{all} denotes all the nodes of the sensor within the monitoring range.

3.2. Judgment of 3D Perception Blind Area. There are often many obstacles in the deployment of sensor nodes in the real world. These obstacles will block the sensing signals of the sensor nodes, causing a substantial reduction in the effective monitoring area of the nodes.

As shown in Figure 1, even though nodes A, B, and C are all located in the perception range of node S, the monitoring points A and B are not sensed by the node S because the node signals are blocked by the obstacle (the shaded area in Figure 1). The lack of sensing causes the formation of a particular area for the 3D surface node deployment, which is called the 3D perception blind area [41].

For instance, the general method of the 3D perception blind area which is monitored from point B can be determined by the following procedure. Assuming that the coordinate of node S is (x_s, y_s, z_s), the coordinate of point B is (x_b, y_b, z_b), the spatial surface equation is $z = f(x, y)$, and point B is in the sensing range of node S. The line segment SB is obtained by connecting both points S and B, and then the equation of the straight spatial line SB can be created. As shown in Equation (4), the simultaneous equations of the spatial line SB and the spatial surface z. It can obtain the number η of solutions to Equation (4) under conditions $x \in (x_s, x_b)$ and $y \in (y_s, y_b)$.

$$\begin{cases} \dfrac{x - x_s}{x_b - x_s} = \dfrac{y - y_s}{y_b - y_s} = \dfrac{z - z_s}{z_b - z_s}. \\ z = f(x, y) \end{cases} \tag{4}$$

The determination of the 3D perception blind area between node S and point B depends on the value of η, which can be divided into the following two situations.

If $\eta = 0$, there is no perception blind area between node S and point B. It defines that the surface does not intersect with the line segment SB, which means the point B is covered by the node S.

If $\eta \geq 1$, there is a perception blind area between node S and point B, which means that the point B is not covered by the node S. Figure 1 intuitively shows that the surface intersects with the line SB, so the node S cannot monitor the point B.

The procedure mentioned above is an effective method to determine the existence of blind areas. However, there are some limitations. For instance, Figure 1 shows that the line segment AS has no intersection with the surface, but there is an obstacle between the point A and the node S. Therefore, the node S cannot cover the point A. For this situation, a determination rule is added in the following.

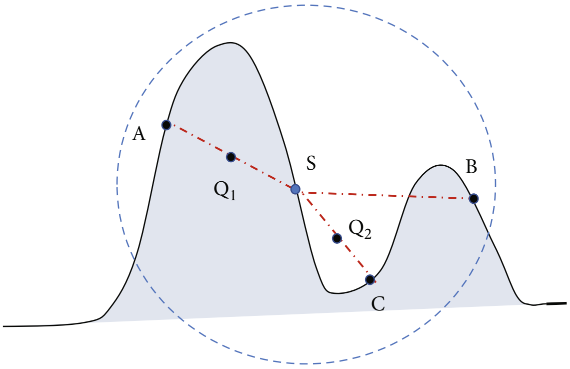


Figure 1: Schematic diagram of the 3D perception blind area.

Suppose the midpoint of the line segment AS is $Q1(x_q, y_q, z_q)$. If the value for the z-axis of the point Q1 is smaller than the value of the function corresponding to the surface equation $z = f(x, y)$, then there would be a perceptual blind area between both points, which means node S cannot monitor the point A. The condition of the method is presented in the following:

$$P_{\text{cov}}(S, A) = \begin{cases} 0 & \text{if } z_q \leq f\left(x_q, y_q\right), \\ 1 & \text{otherwise.} \end{cases} \tag{5}$$

It summarizes that a monitoring point should be perceived by the node S when the following three conditions are satisfied. First, the evaluated value of Equation (2) not equal to 0; second, Equation (4) has no solution, and finally, the evaluated value of Equation (5) becomes 1.

3.3. Coverage Description. It is not easy to divide the monitoring region into small regions of equal areas when solving the problem of 3D WSN coverage. Besides, it is not also necessary to do this division. A surface division based on the grid method was proposed [37]. The method gave a reasonable error condition for the number of grid divisions. The basic idea is to project the surface vertically into a two-dimensional plane. Then, the projection region is divided into several small grids with equal areas. The area of the small surface corresponding to each small grid can be calculated by the calculus idea and usage of the surface with the integral surface formula. The summation of the areas of the small surfaces corresponding to all small grids would be the total area of the whole surface. The calculation method of the surface area is shown in

$$\begin{cases} z = f(x, y), \\ S_F = \iint_{D_{xy}} \sqrt{1 + \left(\dfrac{dz}{dx}\right)^2 + \left(\dfrac{dz}{dy}\right)^2}\, dxdy, \end{cases} \tag{6}$$

where S_F denotes the area of the surface and D_{xy} is the integration region consisting of x and y.

In this paper, the grid method is employed to calculate the coverage of the 3D surface. First, the projection area of the surface F mapped to the 2D plane is divided into $l * w$ grids, where l represents the number of grids in the x-axis direction, and w represents the number of grids in the y-axis direction. That is to say, the target area is divided into several cube grids of equal size. Consider the set $S^F = \{s_1^F, s_2^F, \cdots, s_n^F\}$ represents the area of the small surface, where $n = l * w$. The projection region is divided into several small grids of equal area and locates the monitoring point at the center of the small surface corresponding to the small grid. If the monitoring point can be perceived by the node S, it means that the node S could cover the small surface where the monitoring point is located. The final coverage area can be obtained by accumulating the areas of all the small surfaces which can be monitored. Therefore, the coverage C_F of the 3D surface can be described as follows:

$$C_F = \frac{1}{S_F} \sum_{x=1}^{l} \sum_{y=1}^{w} P_C\left(s_{all}, m_{w\cdot(x-1)+y}\right) \times s^F_{w\cdot(x-1)+y}, \quad (7)$$

where s_{all} denotes all the sensors deployed in the target area, the subscript $w(x-1)+y$ is used to calculate the index of the grid, that is, the index of the monitoring point. Equation (7) is the proposed objective function which is optimized by the algorithm presented in this research.

4. Improved Marine Predator Algorithm

4.1. Marine Predator Algorithm. The marine predator algorithm [23] is a novel metaheuristic algorithm that simulates the foraging strategy of marine organisms. It updates the position of predator and prey by switching between the Lévy flight and the Brownian motion. Both predator and prey are solution candidates, since each prey in the ecosystem is a potential predator. Like most metaheuristic algorithms, MPA generates the initial population randomly in the search space. The mathematical model is described as follows:

$$X_0 = \text{rand}\,(X_{\max} - X_{\min}) + X_{\min}, \quad (8)$$

where rand is a random number that obeys a uniform distribution of $[0, 1]$ and $X_{\min}$ and $X_{\max}$ denote the lower and upper limits of the search space, respectively.

In MPA, the location of the prey can be represented by a matrix. The corresponding mathematical expression is described as follows:

$$\text{Prey} = \begin{bmatrix} X_{1,1} & \cdots & X_{1,d} \\ \vdots & \ddots & \vdots \\ X_{n,1} & \cdots & X_{n,d} \end{bmatrix}_{n\times d}, \quad (9)$$

where n denotes the number of populations and d is the dimension of the search space. $X_{i,j}$ denotes the current position of the i-th prey in the j-th dimension. The individual with the best fitness value is called the top predator. It is used to construct the Elite matrix, whose size is the same as the Prey matrix. The mathematical expression of the Elite matrix is presented in the following:

$$\text{Elite} = \begin{bmatrix} X^I_{1,1} & \cdots & X^I_{1,d} \\ \vdots & \ddots & \vdots \\ X^I_{n,1} & \cdots & X^I_{n,d} \end{bmatrix}_{n\times d}, \quad (10)$$

where X^I represents the top predator vector. The Elite matrix is obtained by replicating this vector n times. The searching procedure of the MPA algorithm can be separated into three stages in accordance with the speed variation of marine biological predation, which is described as follows.

4.1.1. High-Speed Ratio Phase. This phase occurs in the first third of the number of iterations, when the population is in the exploration phase. At this time, the speed of the predator is greater than that of the prey, and the mathematical model of prey position update is described as follows:

$$\text{While}\, t \le \frac{1}{3} t_{\max}, i = 1, 2, \cdots, n, \quad (11)$$

$$\text{stepsize}_i = R_B \otimes (\text{Elite}_i - R_B \otimes \text{Prey}_i), \quad (12)$$

$$\text{Prey}_i = \text{Prey}_i + P \times R \otimes \text{stepsize}_i, \quad (13)$$

where t and $t_{\max}$ denote the current and the maximum number of iterations, respectively. R_B is a vector of random numbers obeying the normal distribution used to represent the Brownian motion. The symbol $\bigotimes$ denotes the entry-wise multiplication, R is a random number between 0 and 1, and $P = 0.5$ is a constant.

4.1.2. Unit-Speed Ratio Phase. This phase occurs between one-third and two-thirds of the number of iterations and gradually indicates the population's transition from exploration to exploitation. At this phase, the predator and the prey movement speed are equal. In this stage, the first half of the population is used for the exploration, and the movement strategy of the prey is the Lévy motion. The remaining half of the population would then be used for the exploitation where the movement strategy of the predator would be the Brownian motion. The mathematical model of this phase is described as follows:

$$\text{While} \frac{1}{3} t_{\max} < t < \frac{2}{3} t_{\max}, i = 1, 2, \cdots, n/2, \quad (14)$$

$$\text{stepsize}_i = R_L \otimes (\text{Elite}_i - R_L \otimes \text{Prey}_i), \quad (15)$$

$$\text{Prey}_i = \text{Prey}_i + P \times R \otimes \text{stepsize}_i, \quad (16)$$

$$\text{While} \frac{1}{3} t_{\max} < t < \frac{2}{3} t_{\max}, i = n/2, \cdots, n, \quad (17)$$

$$\text{stepsize}_i = R_B \otimes (R_B \otimes \text{Elite}_i - \text{Prey}_i), \quad (18)$$

$$\text{CF} = \left(1 - \frac{t}{t_{\max}}\right)^{(2t/t_{\max})}, \quad (19)$$

$$\text{Prey}_i = \text{Elite}_i + P \times CF \otimes \text{stepsize}_i, \tag{20}$$

where R_L is a vector of random numbers based on the Lévy distribution and CF is an adaptive parameter that controls the movement step of the predator.

4.1.3. Low-Speed Ratio Phase. A late iteration phase occurs in the last remaining third of the whole iteration. At this phase, the predator moves slower than the prey within a Lévy's motion, which adopts the population into a local exploitation strategy. The mathematical model of this phase is described as follows:

$$\text{While}\, t \geq \frac{2}{3} t_{\max}, i = 1, 2, \cdots, n, \tag{21}$$

$$\text{stepsize}_i = R_L \otimes (R_L \otimes \text{Elite}_i - \text{Prey}_i), \tag{22}$$

$$\text{Prey}_i = \text{Elite}_i + P \times \text{CF} \otimes \text{stepsize}_i. \tag{23}$$

4.1.4. Eddy Formation and FADs Effect. The formation of eddies or Fish Aggregation Devices (FADs) affects the foraging behavior of the marine predators. This strategy mainly improves searching for individuals to escape the local optimal solutions. Its mathematical model is described as follows:

$$\text{Prey}_i = \begin{cases} \text{Prey}_i + \text{CF}[X_{\min} + R_L \otimes (X_{\max} - X_{\min})] \otimes U & r \leq Pf, \\ \text{Prey}_i + [Pf(1-r) + r](\text{Prey}_{r1} - \text{Prey}_{r2}) & r > Pf, \end{cases} \tag{24}$$

where U is a binary vector containing 0 and 1 and $Pf = 0.2$ represents the probability of influence on the optimization process. r is a random number in the range $[0, 1]$, and the subscripts $r1$ and $r2$ represent the random index values of the prey matrix. In addition, the MPA algorithm has memory storage to remember the history of the best-reached position, which is similar to an essential greedy strategy. During the algorithm iteration, the fitness value of each individual in the current iteration is compared with the optimal solution of the previous iteration. The previous solution will then be replaced if the current's fitness is better.

4.2. Population Evolution Strategy Based on the Random Opposition-Based Learning and Differential Evolution Operator. The optimization process in each phase of MPA is highly depends on the prey matrix (described in section 4.1). The MPA algorithm randomly initializes the population and then relies only on the Brownian or Lévy motion strategies to update the individual positions. The late iteration of the algorithm will reduce population diversity, narrow search range, reduce exploration ability, and tends to stagnate in the local optimum. Usually, in the optimization process of swarm intelligence algorithms, when the searched individual is close to the optimal solution, the population will quickly approach the direction of the optimal solution and accelerate the convergence of the algorithm. On the contrary, the convergence rate becomes slower.

To solve this problem, opposition-based learning [42] (OBL) provides a search strategy that considers both the current solution and its corresponding inverse solution, expanding the search range and avoiding stagnation of the current solution. The research shows that the probability of the opposite solution of the population close to the global optimal solution is greater than that of the random solution. Therefore, many researchers have attempted to use OBL to improve both the quality of the population and the performance of the algorithm [43–48].

To enhance the diversity of the population further, Long et al. [49] proposed a random opposition-based learning (ROL) strategy based on OBL, which could effectively improve the ability of the algorithm to jump out of the local optimum. Suppose a feasible solution of a d-dimensional space is $X = (x_1, x_2, \cdots, x_d)$, where $x_1, x_2, \cdots, x_d \in R$ and $x_j \in [lb_j, ub_j]$. The random opposite solution of X would be $\widehat{X}_r = (\widehat{x}_{r,1}, \widehat{x}_{r,2}, \cdots, \widehat{x}_{r,d})$ and its calculation formula can be presented as follows:

$$\widehat{x}_{r,j} = lb_j + ub_j - R_1 \times x_j\, j = 1, 2, \cdots, d, \tag{25}$$

where $R_1 \in (0, 1)$ is a random number.

Given the effectiveness of the ROL strategy for the improvement of algorithm performance and inspired by the mutation, crossover, and selection operators in the differential evolution algorithm (DE), this paper proposes a population evolution strategy based on ROL and differential evolution operator (RDES) to generate a higher quality population.

First, the ROL population of the initial population is evaluated by Equation (25). Because the positions of all individuals in the ROL population may not be better than those of the initial individuals, both the initial and ROL populations are merged to expand the population searching range. Then, the differential evolution operator is considered based on the merged population to enrich the population's diversity further. The specific operations are as follows:

(1) *Mutation Operation*. Since the mutation operator in the DE has a significant influence on the algorithm, it is essential to choose an effective individual mutation method. This paper used the DE/current-to-best/1 mutation operator [29] to perform mutation operation on the merged population, as shown in the following formula:

$$V_i(t) = X_i(t) + F \cdot \left(X^{\text{best}}(t) - X_k(t) + X_m(t) - X_n(t)\right), \tag{26}$$

where F is scaling factor, $X_i(t)$ denotes the i-th individual of the current generation, $X^{best}(t)$ denotes the optimal individual of the current population. $X_k(t)$, $X_m(t)$, and $X_n(t)$ denote three different individuals randomly selected except the current individual, where the $k, m, n \in 1, 2, \cdots, 2 * N$ are not equal to each other, and N represents the population size. Equation (26) reveals that the mutation operator uses the individuals' local information and the population's

```
Initialize the population with $N$ individuals
For $i \leq N$ do
    Using Eq. (25) to generate the ROL population
End for
Merge initialization population and ROL population
For $i \leq 2N$ do
    Execute the mutation operation by Eq. (26)
    If rand < $P_c$ then
        Execute the crossover operation by Eq. (27)
    End if
    Using Eq. (28) to execute the selection operation
End for
Select the top $N$ fitness individuals from the merged
Population as the initial Prey matrix
```

ALGORITHM 1: Pseudocode of the RDES.

```
Initialization parameters: The number of groups $m$; the number of iterations within the group $t_m$;
For $i = 1 : m$ do
    For $j = 1 : t_m$ do
        Record $P_w, P_b, P_g$
        Update the worst Prey using Eq. (29)
        If $fitness(P_{\text{new}}) < \text{fitness}(P_w)$ then
            $P_w = P_{\text{new}}$
        Else
            Update the worst Prey using Eq. (30)
            $fitness(P_{new}) < \text{fitness}(P_w)$ then
                $P_w = P_{\text{new}}$
            Else
                Update the worst Prey using Eq. (32)
            End if
        End if
    End for
End for
```

ALGORITHM 2: Pseudocode of the local search strategy.

global optimal information to lead the population to find an optimal solution. It expands the diversity of the population and accelerates the searching speed of the algorithm simultaneously

(2) *Crossover Operation*. The original individual $X_i(t)$ and the new generated individual $V_i(t)$ are cross-operated after the differential mutation according to

$$U_{i,j}(t) = \begin{cases} V_{i,j}(t), & \text{rand} \leq P_c, \\ X_{i,j}(t), & \text{otherwise}, \end{cases} \tag{27}$$

where $U_{i,j}(t)$ represents the value of the j-th gene of the current i-th individual, rand is a random number uniformly distributed in $[0, 1]$, and P_c represents the crossover probability. The algorithm would keep the gene value of $V_{i,j}(t)$ if $\text{rand} < P_c$, otherwise, it replaces the corresponding gene value with $X_{i,j}(t)$

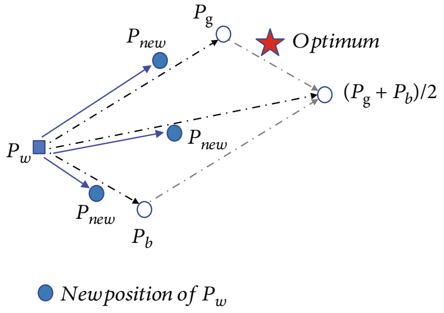


FIGURE 2: Schematic diagram of the position update of the worst individual (the black dashed line represents the guidance direction of the population information, and the solid blue line represents the trajectory of the worst individual position update).

(3) *Selection of Operation*. If the crossover operation results in a new individual with a better fitness value than the original individual, the original individual

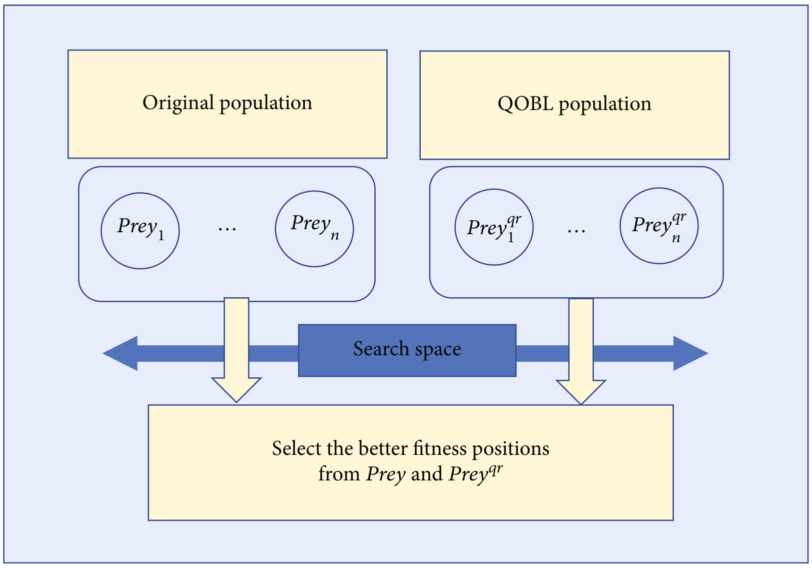


Figure 3: Schematic diagram of population selection based on QOBL.

```
Input: Number of population $N$, the maximum number of iterations $t_{\max}$, the crossover probability $P_c$, the number of groups $m$, the number of iterations within the group $t_m$, the dimensions of each individual $d$, and search range $[lb, ub]$
Output: Optimal fitness
Initialize the population of Preys with $N$ individuals
While $t \le t_{\max}$ do
    Execute RDES to generate a high-quality population (Algorithm 1)
    Calculate the fitness and build the Elite matrix
    If $t \le t_{\max}/3$ then
        Update Prey using Eq. (13)
    Else if $t_{\max}/3 < t < 2 * t_{\max}/3$
        For the first half of the population ($i = 1, 2, \cdots, N/2$)
        Update Prey using Eq. (16)
        For the remained half of the population ($i = N/2, \cdots, N$)
        Update Prey using Eq. (20)
    Else if $t \ge 2 * t_{\max}/3$
        Update Prey using Eq. (23)
    End if
    Complete memory saving and Elite update
    Execute the local search strategy (Algorithm 2)
    For $i = 1 : N$ do
        Calculate the QOBL position according to Eq. (33)
        Select the better individual according to Eq. (35)
    End for
    Complete memory saving and Elite updating
    Next iteration $t = t + 1$
End while
```

Algorithm 3: The proposed IMPA.

would be substituted. This process can be described as follows:

$$X_i(t+1) = \begin{cases} U_i(t), & f(U_i(t)) \le f(X_i(t)), \\ X_i(t), & \text{otherwise}. \end{cases} \tag{28}$$

Finally, the fitness values of all individuals in the merged population are calculated and sorted. The top N individuals with the best fitness value are selected to construct the prey matrix using the greedy strategy.

Based on the above descriptions, the mainframe of the RDES is given in Algorithm 1.

4.3. Local Search Strategy Based on Improved SFLA. SFLA is a swarm intelligence algorithm proposed by Eusuff et al. [50]. They simulated the population's mechanism of information sharing and communication during the frog foraging. The algorithm combines the advantages of the Memetic Algorithm

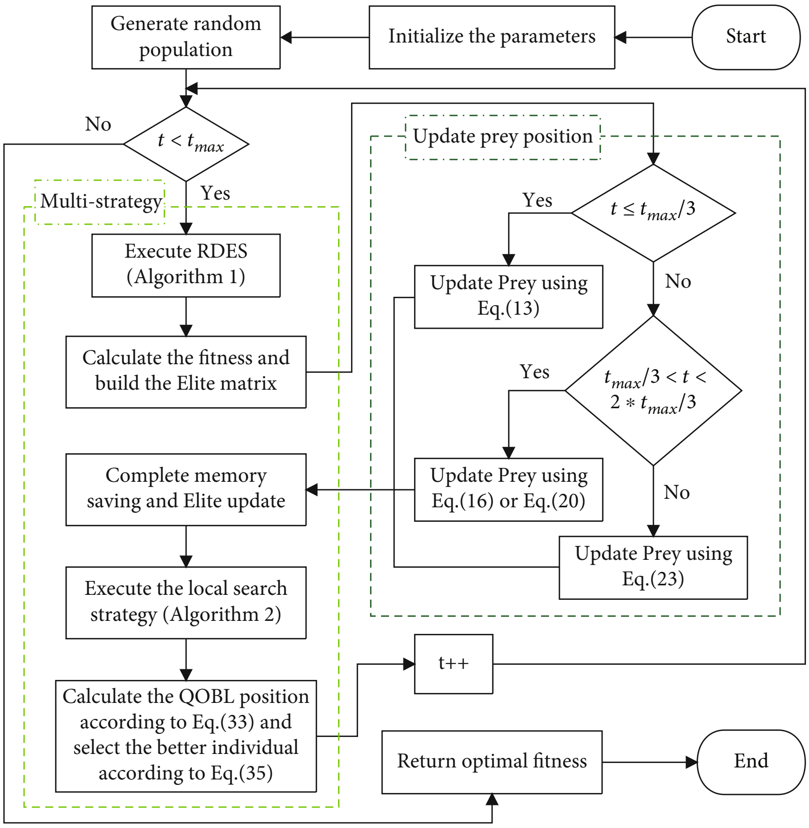

Start
Initialize the parameters
Generate random population
$t < t_{max}$
No
Yes
Multi-strategy
Execute RDES (Algorithm 1)
Calculate the fitness and build the Elite matrix
Complete memory saving and Elite update
Execute the local search strategy (Algorithm 2)
Calculate the QOBL position according to Eq.(33) and select the better individual according to Eq.(35)
Update prey position
$t \le t_{max}/3$
Yes
No
Update Prey using Eq.(13)
$t_{max}/3 < t < 2 * t_{max}/3$
Yes
No
Update Prey using Eq.(16) or Eq.(20)
Update Prey using Eq.(23)
t++
Return optimal fitness
End

Figure 4: IMPA flow chart.

(MA) [51] and the PSO [28]. Moreover, it benefits few control parameters, is an easy implementation, and has a strong global searching ability. Therefore, some researchers have hybridized SFLA with other algorithms to improve the optimization performance of the algorithm. For instance, in [52], an improved ant colony algorithm was combined with SFLA for solving the path planning problem, which accelerated the convergence of the algorithm. In [53], in order to improve the local search capability of the PSO, SFLA was integrated with PSO and used for the detection of sonar images.

The MPA uses the FADs effect (Equation (24)) to support individuals' searching for escape from the local optimum trap. This method lacks the guidance of population information to update the individuals' position and limited the local development capability. Finding a more optimal solution becomes complicated once the population falls into the local optimum. The SFLA divides the whole population into several groups based on its unique grouping operator. The evolution of the individual is guided by the local optimal solution in the group and the global optimal solution of the population. It realizes the information sharing and communication between individuals. Therefore, the algorithm would not easily fall into the local optimum and would be more conducive to find the global optimal solution [22]. In this paper, by considering the grouping idea of SFLA, both MPA and SFLA algorithms are combined, and a local search algorithm based on the improved SFLA is proposed. The proposed local search algorithm replaces the FADs effect to improve the local development ability of MPA.

The main idea of the SFLA is described in the following. First, all the population individuals are arranged in descending order according to their fitness values and divided into m groups. The first individual is assigned to the first group, the second to the second group, until the m-th to the m-th group. The $(m+1)$-th individual is then assigned to the first group, the $(m+2)$-th to the second group, and so on, until assigning the last one. Then, the positions of the worst individual P_w are updated according to Equations (29)–(31). The specific update method is as follows:

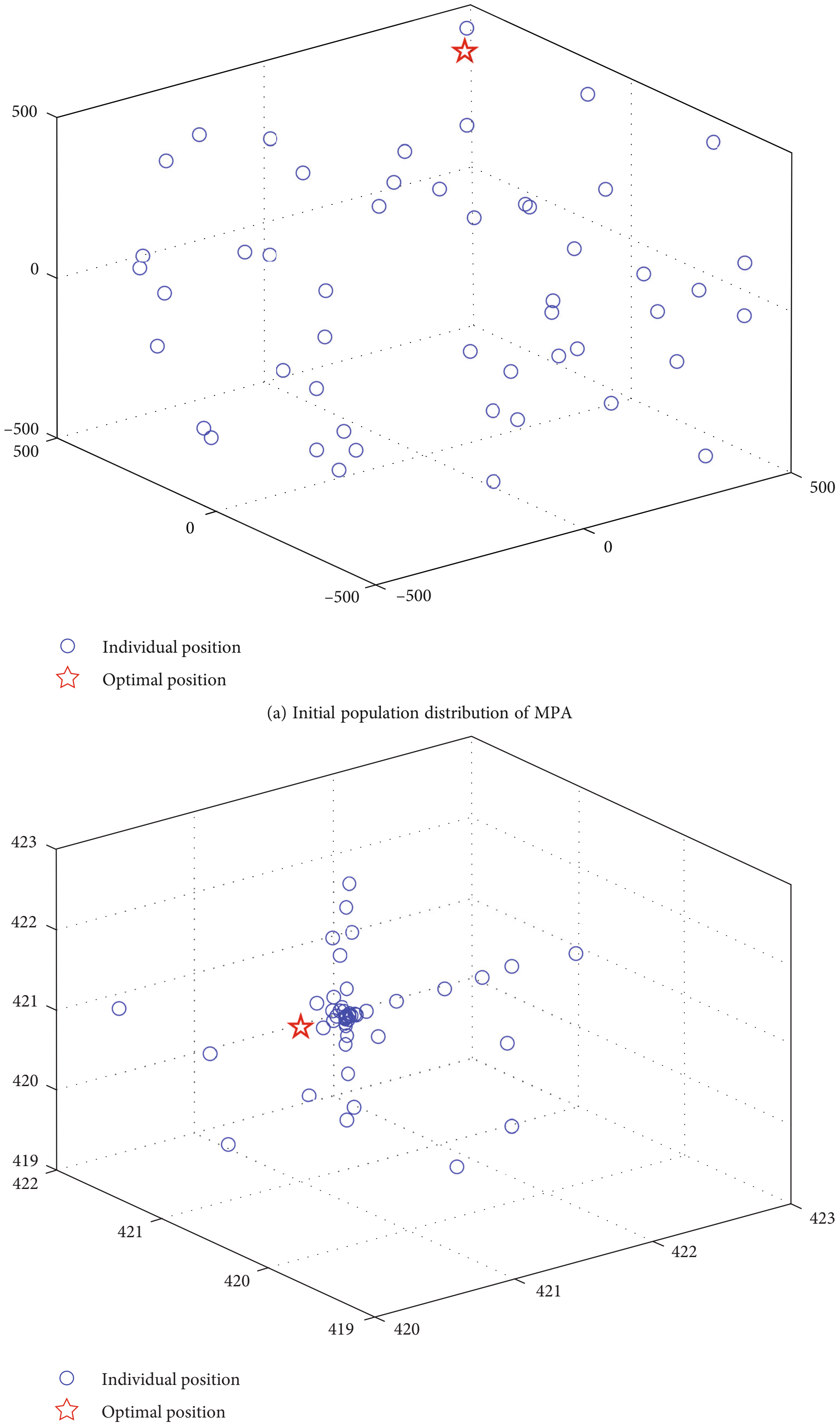
$$P_{\text{new}} = P_w + s \times \text{rand} \times (P_b - P_w), \tag{29}$$

$$P_{\text{new}} = P_w + s \times \text{rand} \times \left(P_g - P_w\right), \tag{30}$$

$$P_{\text{new}} = lb + \text{rand} \times (ub - lb), \tag{31}$$

where P_{new} denotes the updated position of the worst individual, P_b and P_w denotes the best and worst individuals in the group, rand is a random number between $[0, 1]$, P_{g} represents the position of the globally optimal individual for the entire population, s is a constant representing the step size factor and that its value can be adjusted according to the problems, and ub and lb are the upper and lower bounds of the search space, respectively.

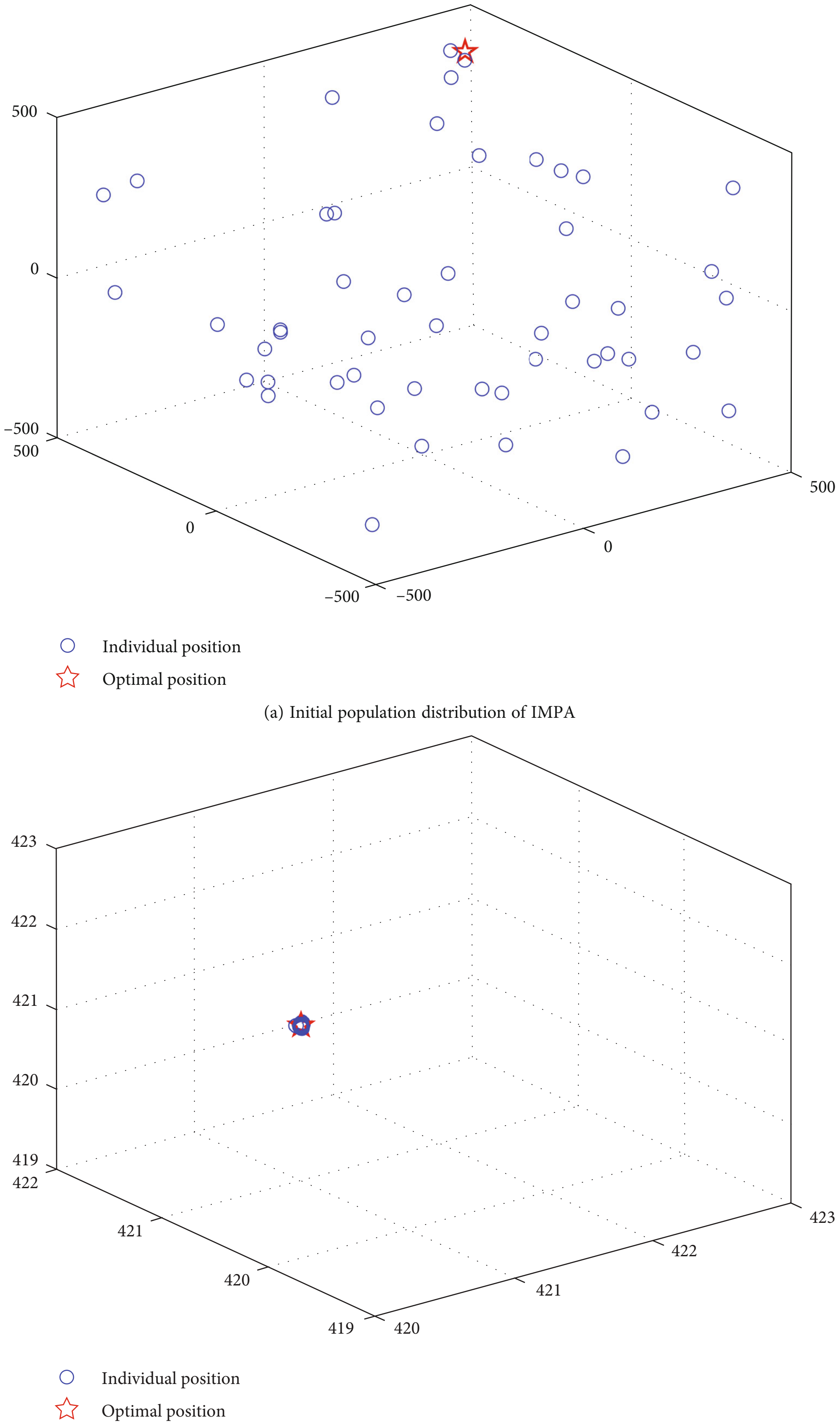
Since the third location update method (as shown in Equation (31)) of the basic SFLA algorithm is relatively random, it lacks the guidance of population information, and consequently, the searching speed is slow. Therefore, this



(a) Initial population distribution of MPA

(b) Final population distribution of MPA

Figure 5: Population distribution of MPA.



(a) Initial population distribution of IMPA

(b) Final population distribution of IMPA

Figure 6: Population distribution of IMPA.

Table 1: Benchmark test functions.

Function	Dim	Interval	Min
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30/100	[-100, 100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} x_i$	30/100	[-10, 10]	0
$F_3(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	30/100	[-30, 30]	0
$F_4(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30/100	[-100, 100]	0
$F_5(x) = \max_i\{\lvert x_i \rvert, 1 \leq i \leq n\}$	30/100	[-100, 100]	0
$F_6(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$	30/100	[-32, 32]	0
$F_7(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30/100	[-600, 600]	0
$F_8(x) = \sum_{i=1}^{n} i x_i^4 + \text{random}[0, 1)$	30/100	[-1.28, 1.28]	0
$F_9(x) = \sum_{i=1}^{n} - x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30/100	[-500, 500]	-418.98n
$F_{10}(x) = (\pi/n)\left\{10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + ((x_i + 1)/4)$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30/100	[-50, 50]	0
$F_{11}(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	2	[-10, 10]	0
$F_{12}(x) = \sum_{i=1}^{7} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0, 10]	-10.4029
$F_{13}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$	3	[0, 1]	-3.8628
$F_{14}(x) = (1.5 - x_1 + x_1x_2)^2 + \left(2.25 - x_1 + x_1x_2^2\right)^2 + \left(2.625 - x_1 + x_1x_2^3\right)^2$	2	[-4.5, 4.5]	0
$F_{15} = \sin^2(3\pi x_1) + (x_1 - 1)^2\left[1 + \sin^2(3\pi x_2)\right] + (x_2 - 1)^2\left[1 + \sin^2(2\pi x_2)\right]$	2	[-10, 10]	0

paper considers the usage of the local information of the group and the global information of the population to guide the algorithm for strengthening the ability of further search. A new location update method is proposed, as shown in

$$P_{new} = P_w + rand \times \left(\frac{P_g + P_b}{2} - P_w\right). \qquad (32)$$

The mainframe of the improved local search strategy is shown in Algorithm 2. First, the prey population is divided into m groups according to the grouping strategy which was introduced above. After the completion of grouping, the worst position of an individual would be updated using Equation (29). The original position is then replaced if the new position includes a better fitness value. Otherwise, the new position would be evaluated by Equation (30). If neither of the above two methods can generate a better position, the position of the worst individual in the group would be updated by Equation (32).

Figure 2 depicts the three updating methods for the position of the worst individual in each group. As it can be seen in Figure 2, Equation (29) means the best individual in the group guides the worst individual to move towards it. Equation (30) indicates that the worst individual approaches the global optimal individual, and Equation (32) represents that the population approaches the middle position between the global optimal and the intragroup optimal.

The improved local search method transmits the information according to the population grouping. It effectively combines the global information of the population and the local search. It also improves the exploitation ability of the population. In addition, the position of the worst individuals in the group is only updated each time after grouping, which reduces the computation time and increases the convergence rate of the algorithm. The fitness value of the population is inevitably improved accordingly when the individuals of all groups are remixed into the population. Therefore, it leads the population towards the optimal solution.

4.4. Dynamic Learning Mechanism. Ergezer et al. proposed a variant of OBL called quasireflected opposition-based learning (QOBL) [54]. They proved that the probability of the QOBL solution is closer to the optimal solution than the general OBL solution is 11/16. This paper proposes a QOBL strategy for the prey population to improve the optimization accuracy and accelerate the convergence speed of the algorithm further. The QOBL position (Prey_i^{qr}) of the prey in the generation can be described as follows:

$$\mathrm{Prey}_i^{qr}(t) = \mathrm{rand}\,(\mathrm{Prey}_i, c_i), \tag{33}$$

$$c_i = \frac{\min\,(\mathrm{Prey}_i) + \max\,(\mathrm{Prey}_i)}{2}, \tag{34}$$

where Prey_i represents the position of the i-th prey and c_i represents the midpoint of the dynamic boundary. Unlike the fixed boundary, the dynamic search boundary adjusts the maximum and minimum values in all dimensions of the current i-th individual as the upper and lower bounds of the search space. This method can further reduce the possibility of stagnating the population in the local optima, enhance the exploitation capability of the algorithm, and increase the convergence speed [55].

Finally, the elite strategy is employed to retain the individuals who carry better fitness value. If the fitness value of the QOBL position's fitness value is superior to the current individual, the new position would be assigned to the current individual. The strategy is modeled as follows:

$$\mathrm{Prey}_i(t+1) = \begin{cases} \mathrm{Prey}_i^{qr}(t) & \text{if } f\left(\mathrm{Prey}_i^{qr}(t)\right) < f(\mathrm{Prey}_i(t)), \\ \mathrm{Prey}_i(t) & \text{otherwise.} \end{cases} \tag{35}$$

Figure 3 shows the process of applying the dynamic learning mechanism based on QOBL to the MPA algorithm. First, the QOBL population of the prey population is evaluated. Then, the fitness values of both initial prey and the QOBL populations would be calculated separately. Finally, the individuals with better fitness values are retained for constructing new prey populations according to the Equation (35).

4.5. IMPA Algorithm. In this paper, a multistrategy integrated improved marine predator algorithm (IMPA) is proposed to increase both accuracy of solution and convergence speed and prevent falling into the local optimum of MPA. The pseudocode of IMPA is given in Algorithm 3, and the main framework of the algorithm is demonstrated in Figure 4.

Table 2: Parameter Settings for the comparing optimization algorithms.

Algorithm	Parameters
IMPA	$P = 0.5, P_c = 0.5$, $m = 8$, $t_m = 3$, $s = 2$
MSIMPA	FADs = 0.2, $P = 0.5$
MPA	FADs = 0.2, $P = 0.5$
ESSA	ST = 0.8, SD = 0.2
IGWO	$a = (2 \longrightarrow 0)$
WOA	$a = (2 \longrightarrow 0)$, $b = 1$
PSO	$\omega = 0.5$, $c_1 = 1.3$, $c_2 = 2.8$
DE	$P_c = 0.8$, $F = 0.5$

4.6. Algorithm Validity Test. In this section, the Schwefel2.26 Function is selected to test the validity of the three improved strategies. The population distribution maps of IMPA and MPA algorithms are drawn. The initial and final distributions of both algorithms are shown in Figures 5 and 6 while the population size and the maximum number of iterations are 50 and 20, respectively.

Figures 5(b) and 6(b) show that the population distribution of MPA is relatively scattered after 20 iterations and the convergence speed is low. Moreover, most individuals gather near the local optimum, which means that the MPA would not easily jump out of the local optimum when dealing with multidimensional complex functions. In contrast, all individuals in the IMPA converge to the global optimum quickly, proving the effectiveness of the proposed strategies.

4.7. Time Complexity Analysis. Time complexity is an essential indicator for predicting the performance of an algorithm. Suppose that N is the size of populations, D is the dimension of individuals, and T is the maximum number of iterations. The time complexity of the MPA is evaluated $O(N \bullet D \bullet T)$. The proposed IMPA algorithm would not change the structure of individual position updates in the three phases of the basic MPA algorithm. The analysis of the time complexity is explained in the following.

The increased complexity of the ROL population while performing the RDES (Algorithm 1) is $O(N)$, where this evaluation becomes $O(2 \bullet N \bullet D)$ while performing the DE evolution operation. While executing the local search strategy (Algorithm 2), the evaluated complexity is related to the number of the groups m and the number of iterations t_m, where $m \leq N/2$. Therefore, in the worst case, the increased complexity of this part would be $O(N)$. The increased complexity while computing the QOBL population would be $O(N)$. Finally, the evaluated time complexity of IMPA would be $O(T(N + 2 \bullet N \bullet D + N + N))$, which results in $O(N \bullet D \bullet T)$. It concluded that the improved strategies proposed in this study would not increase the time complexity of the proposed IMPA.

4.8. Global Convergence Analysis of IMPA. The convergence analysis of the three phases within the optimization process

Table 3: Comparison of benchmark function optimization results (30-dimension and fixed-dimension).

Function	Algorithm	Best	Ave	Std	Rank	Running time (s)
F1	IMPA	0	0	0	1	0.2197
	MSIMPA	0	0	0	1	0.4990
	MPA	3.249e-25	5.6984e-23	6.8011e-23	6	0.1168
	ESSA	3.626e-273	1.0578e-28	5.2877e-28	4	0.0929
	IGWO	3.6013e-30	2.0879e-28	3.8839e-28	5	0.3297
	WOA	7.8706e-85	1.2859e-72	4.9154e-72	3	0.0208
	PSO	8.3500e-08	3.9739e-02	1.4805e-01	7	0.1687
	DE	9.1636e-05	1.4571e-01	7.6655e-01	8	0.0705
F2	IMPA	0	0	0	1	0.2553
	MSIMPA	0	0	0	1	0.5151
	MPA	2.0622e-14	2.1332e-13	1.9902e-13	6	0.1194
	ESSA	0	5.0428e-17	2.3574e-16	5	0.0940
	IGWO	1.6695e-18	8.3521e-18	5.7087e-18	4	0.3340
	WOA	2.2386e-57	5.2527e-51	1.755e-50	3	0.0226
	PSO	2.9933e-03	7.9203e-02	1.6586e-01	8	0.1731
	DE	3.6882e-03	1.0604e-02	6.1972e-03	7	0.0716
F3	IMPA	2.2747e+01	2.3184e+01	2.0370e-01	3	0.2810
	MSIMPA	2.3646e+01	2.4331e+01	2.9350e-01	4	0.6913
	MPA	2.4583e+01	2.5351e+01	4.3584e-01	6	0.1373
	ESSA	2.2796e-19	2.4733e-05	4.5664e-05	1	0.1161
	IGWO	2.3720e+01	2.4335e+01	8.5325e-01	5	0.3542
	WOA	2.6976e+01	2.7968e+01	4.7406e-01	7	0.0312
	PSO	2.6555e-05	4.1230e-01	1.3376e+00	2	0.1801
	DE	2.6727e+01	7.7838e+01	5.1636e+01	8	0.0808
F4	IMPA	0	0	0	1	0.2494
	MSIMPA	1.2009e-09	7.5903e-09	5.2053e-09	3	0.4986
	MPA	1.4150e-08	1.2359e-04	6.4591e-04	4	0.1213
	ESSA	2.0396e-13	3.0063e-09	4.9969e-09	2	0.1020
	IGWO	3.3595e-05	3.7862e-02	8.6222e-02	6	0.3391
	WOA	8.0338e-02	3.8291e-01	2.2480e-01	8	0.0215
	PSO	9.3274e-06	3.0549e-01	1.2967e+00	7	0.1721
	DE	1.0879e-04	5.0759e-04	3.1912e-04	5	0.0715
F5	IMPA	0	0	0	1	0.2440
	MSIMPA	0	0	0	1	0.4935
	MPA	5.3530e-10	3.9982e-09	2.4312e-09	4	0.1151
	ESSA	0	2.7442e-14	1.4981e-13	3	0.0934
	IGWO	7.7868e-07	1.1347e-05	1.2372e-05	5	0.3325
	WOA	4.4745e+00	4.1640e+01	2.7544e+01	8	0.0209
	PSO	2.5094e-04	5.8164e-02	2.6117e-01	6	0.1698
	DE	5.7486e+00	1.5239e+01	6.1165e+00	7	0.0707
F6	IMPA	8.8818e-16	8.8818e-16	0	1	0.2431
	MSIMPA	8.8818e-16	8.8818e-16	0	1	0.5607
	MPA	1.7852e-13	1.3499e-12	8.3401e-13	6	0.1242
	ESSA	8.8818e-16	1.1428e-14	4.6812e-14	4	0.1093
	IGWO	4.3521e-14	6.4126e-14	1.2392e-14	5	0.3395
	WOA	8.8818e-16	4.6777e-15	2.4567e-15	3	0.0254

Table 3: Continued.

Function	Algorithm	Best	Ave	Std	Rank	Running time (s)
	PSO	4.1938e-04	4.3581e-02	1.2346e-01	8	0.1841
	DE	2.8525e-03	6.6528e-03	3.7812e-03	7	0.0804
F7	IMPA	0	0	0	1	0.2934
	MSIMPA	0	0	0	1	0.7127
	MPA	0	0	0	1	0.1413
	ESSA	0	0	0	1	0.1215
	IGWO	0	3.7034e-03	5.9281e-03	6	0.3616
	WOA	0	3.7007e-18	2.0270e-17	5	0.0331
	PSO	6.5533e-06	4.0149e+00	7.8970e+00	8	0.1906
	DE	3.7623e-04	7.4684e-03	2.1069e-02	7	0.0865
F8	IMPA	9.4248e-06	1.0125e-05	6.7652e-05	1	0.4376
	MSIMPA	3.4548e-06	7.6693e-05	7.7777e-05	2	1.6014
	MPA	3.6961e-04	1.2423e-03	5.8351e-04	4	0.2326
	ESSA	2.3136e-05	6.1201e-04	5.7285e-04	3	0.1838
	IGWO	9.4534e-04	2.5539e-03	1.3054e-03	5	0.4462
	WOA	2.4683e-05	2.5666e-03	2.9612e-03	6	0.0752
	PSO	5.9669e-03	4.8591e-02	3.7271e-02	8	0.2317
	DE	1.6948e-02	3.9809e-02	1.5271e-02	7	0.1256
F9	IMPA	-11977.2949	-10958.0172	374.2614	2	0.2557
	MSIMPA	-11412.6515	-10955.865	297.9579	3	0.7244
	MPA	-9904.4343	-8861.0457	465.2646	4	0.1415
	ESSA	-11029.4511	-8172.1176	1117.5627	6	0.1118
	IGWO	-11351.2906	-7855.7735	1753.1443	7	0.3589
	WOA	-12567.8533	-10576.3502	1797.3658	4	0.0332
	PSO	-12569.4866	-12066.0615	1024.5383	1	0.1825
	DE	-10862.8222	-8053.8889	1609.6519	8	0.0859
F10	IMPA	1.5705e-32	1.5705e-32	5.5674e-48	1	0.7041
	MSIMPA	3.4698e-10	1.3188e-09	9.3011e-10	3	1.3146
	MPA	1.5317e-09	4.1000e-05	2.2287e-04	5	0.4124
	ESSA	4.9661e-15	2.4918e-10	4.1901e-10	2	0.2997
	IGWO	2.7700e-06	2.9326e-04	1.2520e-03	6	0.6210
	WOA	6.1646e-03	2.3879e-02	2.0913e-02	7	0.1618
	PSO	6.2733e-08	1.8325e-05	2.6361e-05	4	0.3174
	DE	2.1047e-05	2.1863e-01	2.9378e-01	8	0.2135
F11	IMPA	0	0	0	1	0.1037
	MSIMPA	0	0	0	1	0.1860
	MPA	0	9.1953e-25	3.0168e-24	5	0.0553
	ESSA	0	2.6295e-32	1.4403e-31	4	0.0869
	IGWO	1.4018e-27	1.5287e-22	5.675e-22	6	0.2465
	WOA	5.4848e-06	9.8198e-04	1.0474e-03	7	0.0146
	PSO	3.0552e-04	4.1497e-01	8.0658e-01	8	0.1467
	DE	0	0	0	1	0.0560
F12	IMPA	-10.4029	-10.4029	1.3194e-16	1	0.2139
	MSIMPA	-10.4029	-10.4029	5.9774e-13	3	0.3531
	MPA	-10.4029	-10.4029	2.3540e-11	4	0.1048
	ESSA	-10.4029	-10.4029	7.3759e-16	2	0.1252

Table 3: Continued.

Function	Algorithm	Best	Ave	Std	Rank	Running time (s)
	IGWO	-10.4029	-10.4029	2.3253e-08	5	0.3106
	WOA	-10.4029	-7.4326	3.0665e+00	8	0.0338
	PSO	-10.4028	-10.3154	2.6727e-01	6	0.1729
	DE	-10.4029	-10.0045	1.5274e+00	7	0.0771
F13	IMPA	-3.8628	-3.8628	2.6662e-15	4	0.1102
	MSIMPA	-3.8628	-3.8628	2.6684e-15	5	0.2435
	MPA	-3.8628	-3.8628	2.3715e-15	1	0.0765
	ESSA	-3.8628	-3.8628	2.4184e-15	2	0.1033
	IGWO	-3.8628	-3.8628	2.4795e-15	3	0.2701
	WOA	-3.8628	-3.8582	0.0065032	7	0.0233
	PSO	-3.862	-3.8465	4.6015e-02	8	0.1550
	DE	-3.8628	-3.8628	2.7101e-15	6	0.0651
F14	IMPA	0	0	0	1	0.1241
	MSIMPA	0	3.5595e-30	1.3958e-29	3	0.1955
	MPA	3.4592e-28	3.0451e-22	1.5159e-21	5	0.0588
	ESSA	0	2.8995e-24	1.1012e-23	4	0.0891
	IGWO	7.4237e-21	7.1524e-17	1.4177e-16	6	0.2510
	WOA	1.4198e-13	1.0161e-01	2.6348e-01	8	0.0161
	PSO	1.2304e-04	1.3578e-03	9.7707e-04	7	0.1463
	DE	0	0	0	1	0.0575
F15	IMPA	1.3498e-31	1.3498e-31	6.6809e-47	1	0.1196
	MSIMPA	1.3498e-31	1.3498e-31	6.6809e-47	1	0.1903
	MPA	7.2938e-27	6.392e-23	1.3055e-22	6	0.0567
	ESSA	1.3498e-31	1.3498e-31	6.6809e-47	1	0.0890
	IGWO	7.8004e-30	6.2979e-26	2.7686e-25	5	0.2485
	WOA	2.2215e-09	8.1541e-06	1.6292e-05	7	0.0154
	PSO	1.2427e-07	1.6901e-03	2.9564e-03	8	0.1472
	DE	1.3498e-31	1.3498e-31	6.6809e-47	1	0.0561

for the basic MPA algorithm has been extensively discussed [23]. It approved that the MPA is a convergent algorithm. The conclusion of the mentioned study is reviewed in the following.

Assuming that $\{\text{Prey}(t)\}$ is a random sampling sequence of MPA, the MPA eventually converges to an Elite matrix (globally optimal position) consisting of the top predators, which is $\lim_{t\longrightarrow\infty} \text{Prey}(t) = \text{Elite}$.

The global convergence of the hybrid optimization algorithm was analyzed, and several sufficient conditions for the global convergence of the hybrid optimization algorithm were approved [56]. The approved relevant theorems of the mentioned research are represented below.

Lemma 1. *$\{f_n\}$ is a monotonically nonincreasing sequence with a lower limited bound in which $\lim_{t\longrightarrow\infty} f_n = \inf \{f_n\}$ [56].*

Lemma 2. *If an algorithm satisfies both following conditions (properties), the optimal sequence will converge globally according to a probability equal to 1 [56].*

Condition 3. The greedy strategy is used to retain the elite while comparing with the historical optimal solution during the iteration. The mentioned strategy is represented in the following:

$$f_{t+1}^{\text{best}} = \begin{cases} f(X), & f(X) < f_t^{\text{best}}, \\ f_t^{\text{best}}, & \text{otherwise}, \end{cases} \quad (36)$$

where f_t^{best} is the optimal fitness value found after t times of comparisons which retained to the moment t, and X is the solution used for $(t+1)$-th comparison with f_t^{best}.

Condition 4. The probability of transitioning from any non-global optimal point (might be set to X^*) to the corresponding level set $L(X^*) = \{X | f(X) < f(X^*), X \in S\}$ is not 0, where S represents the entire search space.

Theorem 5. *The proposed IMPA algorithm converges globally according to the probability equal to 1.*

Table 4: Comparison of benchmark function optimization results (100-dimension).

Function	Algorithm	Best	Ave	Std	Rank	Running time (s)
F1	IMPA	0	0	0	1	0.4799
	MSIMPA	0	0	0	1	1.7578
	MPA	2.0936e-20	2.0131e-19	1.8807e-19	5	0.2724
	ESSA	1.3162e-215	7.4533e-31	4.0189e-30	4	0.1457
	IGWO	3.4070e-13	2.7008e-12	2.0152e-12	6	0.5957
	WOA	4.056e-82	1.6867e-71	8.8984e-71	3	0.0481
	PSO	2.5648e-06	3.4789e-01	1.4948e+00	7	0.2561
	DE	6.8945e+02	1.2783e+03	5.3839e+02	8	0.1209
F2	IMPA	0	2.5048e-297	0	2	0.5490
	MSIMPA	0	0	0	1	1.9195
	MPA	1.2682e-13	1.418e-11	1.0044e-11	5	0.2828
	ESSA	4.6552e-62	2.6374e-14	1.4083e-13	4	0.1475
	IGWO	7.0279e-09	1.5616e-08	5.995e-09	6	0.6088
	WOA	5.5313e-61	2.3061e-49	9.5159e-49	3	0.0489
	PSO	4.8680e-03	3.9462e-01	4.8301e-01	7	0.2833
	DE	1.2305e+01	1.9098e+01	3.7135e+00	8	0.1235
F3	IMPA	9.2984e+01	9.3322e+01	1.7465e-01	3	0.6006
	MSIMPA	9.4100e+01	9.4491e+01	1.7690e-01	4	2.4819
	MPA	9.5854e+01	9.7101e+01	7.8353e-01	5	0.3002
	ESSA	8.5565e-09	3.6629e-04	5.9162e-04	1	0.1675
	IGWO	9.4975e+01	9.7472e+01	1.1160e+00	6	0.6292
	WOA	9.7476e+01	9.8149e+01	2.5179e-01	7	0.0606
	PSO	6.1575e-04	7.9545e+01	4.3079e+02	2	0.2824
	DE	3.7089e+05	1.5227e+06	1.0087e+06	8	0.1382
F4	IMPA	0	0	0	1	0.5204
	MSIMPA	7.3420e-02	5.7584e-01	3.4773e-01	3	1.6146
	MPA	3.0751e+00	3.9660e+00	6.4026e-01	4	0.2583
	ESSA	8.1256e-11	6.5052e-07	1.3268e-06	2	0.1420
	IGWO	5.4614e+00	7.8873e+00	1.0375e+00	6	0.5740
	WOA	1.6505e+00	4.1532e+00	9.5339e-01	5	0.0437
	PSO	3.0226e-07	3.4222e+01	1.3481e+02	7	0.2539
	DE	5.8683e+02	1.0714e+03	3.8677e+02	8	0.1172
F5	IMPA	0	0	0	1	0.4683
	MSIMPA	0	0	0	1	1.4869
	MPA	6.3693e-08	2.25e-07	1.1712e-07	4	0.2249
	ESSA	4.8795e-81	4.7458e-15	2.3905e-14	3	0.1178
	IGWO	1.1614e+00	4.6770e+00	2.8552e+00	6	0.5037
	WOA	2.3009e+01	8.6982e+01	1.6871e+01	7	0.0388
	PSO	3.4590e-04	9.5632e-03	1.0224e-02	5	0.2272
	DE	8.3771e+01	8.7385e+01	1.4535e+00	8	0.1050
F6	IMPA	8.8818e-16	8.8818e-16	0	1	0.5018
	MSIMPA	8.8818e-16	8.8818e-16	0	1	1.8198
	MPA	1.2297e-11	5.5801e-11	4.2214e-11	5	0.2448
	ESSA	8.8818e-16	1.4803e-15	1.3467e-15	3	0.1498
	IGWO	6.1236e-08	1.7033e-07	6.0547e-08	6	0.5292
	WOA	8.8818e-16	4.6777e-15	2.6279e-15	4	0.0486

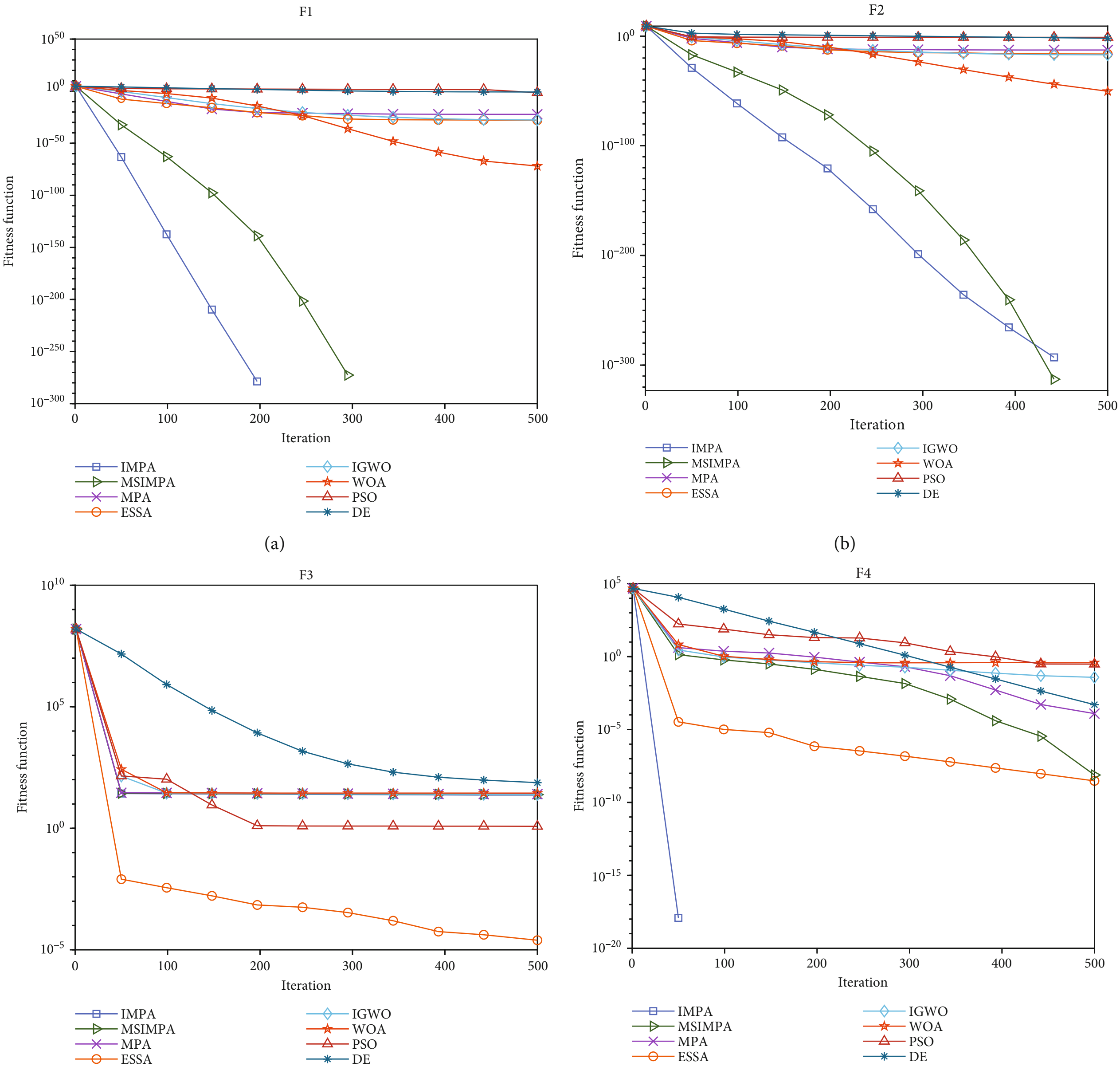
Table 4: Continued.

Function	Algorithm	Best	Ave	Std	Rank	Running time (s)
	PSO	3.1384e-04	5.8055e-02	1.8421e-01	7	0.2642
	DE	4.7890e+00	6.2446e+00	8.0832e-01	8	0.1407
F7	IMPA	0	0	0	1	0.5311
	MSIMPA	0	0	0	1	2.6065
	MPA	0	0	0	1	0.2776
	ESSA	0	0	0	1	0.1679
	IGWO	2.7289e-13	2.9180e-03	6.3530e-03	6	0.5641
	WOA	0	0	0	1	0.0631
	PSO	7.2014e-07	1.4439e+01	4.1840e+01	8	0.2778
	DE	5.0666e+00	1.3019e+01	6.3009e+00	7	0.1553
F8	IMPA	3.0053e-06	1.8797e-05	1.4073e-04	1	0.9748
	MSIMPA	2.2357e-06	7.5344e-05	6.8927e-05	2	2.0507
	MPA	3.8496e-04	1.8048e-03	5.9696e-04	4	0.5977
	ESSA	3.3627e-05	7.2115e-04	7.7153e-04	3	0.3942
	IGWO	0.0054203	1.5155e-02	4.5389e-03	6	0.8735
	WOA	2.6318e-05	5.3274e-03	5.2580e-03	5	0.2172
	PSO	4.9434e-03	6.7585e-02	4.1483e-02	7	0.4266
	DE	1.4769e+00	3.0316e+00	1.0210e+00	8	0.2811
F9	IMPA	-40708.7689	-33742.442	2302.8291	3	0.5864
	MSIMPA	-35220.1185	-33533.4499	895.5147	4	1.9733
	MPA	-26300.7559	-24710.3869	907.3275	5	0.2990
	ESSA	-28921.6901	-23487.2937	2711.1244	6	0.1623
	IGWO	-26070.9887	-16182.5565	6433.3800	7	0.5971
	WOA	-41895.7857	-34167.7492	5693.4744	2	0.0678
	PSO	-41898.2886	-40509.5626	3560.7589	1	0.2540
	DE	-10149.0836	-60561.6508	2047.6199	8	0.1443
F10	IMPA	4.7116e-33	4.7116e-33	1.3918e-48	1	1.8984
	MSIMPA	2.1453e-03	6.4522e-03	3.3446e-03	4	3.7580
	MPA	3.1036e-02	4.6095e-02	8.6771e-03	5	1.0844
	ESSA	1.0014e-15	4.9800e-09	1.0483e-08	2	0.7149
	IGWO	1.3000e-01	2.2248e-01	5.3100e-02	7	1.4428
	WOA	1.9735e-02	5.3398e-02	2.6769e-02	6	0.4637
	PSO	6.4788e-09	2.0935e-05	6.6560e-05	3	0.6519
	DE	1.4390e+04	4.3132e+05	5.7432e+05	8	0.5338

Assume that the objective function f of the IMPA algorithm optimization is set to find the minimum value, $\{f_n\}$ is the random sampling sequence, $\{f_t^{best}\}$ is the optimal sequence of sampling points, and f^{gbest} is the global optimal solution. As shown below, Theorem 5 can be proved by Lemma 1 and Lemma 2.

Proof. Because the three proposed improved strategies and the marine memory storage procedure in the IMPA algorithm all use the greedy strategy to retain the optimal solution of the population, the IMPA algorithm would satisfy Condition 3. □

According to Lemma 1, the limit of the optimal sequence $\{f_t^{\text{best}}\}$ exists. Let $\lim_{t\longrightarrow\infty} f_t^{\text{best}} = f_n^{\text{best}}$, and $f_n^{\text{best}} \neq f^{\text{gbest}}$. Because f^{gbest} is the global optimal solution, therefore $f_n^{\text{best}} > f^{\text{gbest}}$. Since $\lim_{t\longrightarrow\infty} f_t^{\text{best}} = \inf \{f_t^{\text{best}}\} = f_n^{\text{best}}$, it follows that for $\forall t \in Z^+$, $f_t^{\text{best}} \geq f_n^{\text{best}}$. Let X^* be the sampling point corresponding to f_n^{best}, so, $f_n^{\text{best}} = f(X^*)$. X^* is obviously a nonglobal optimal solution. Because the IMPA uses the same update operation as well as the MPA algorithm in the high-, unit-, and low-speed ratio phases, the convergence of the population sequence of IMPA would be consistent with that of the MPA and will eventually converge to the Elite matrix, $\lim_{n\longrightarrow\infty} f_n = \text{Elite} = f^{\text{gbest}}$. That is, the nonglobal optimal point X^* can converge to the Elite matrix (global optimal point). Therefore, when the X assigns value to a sufficiently small



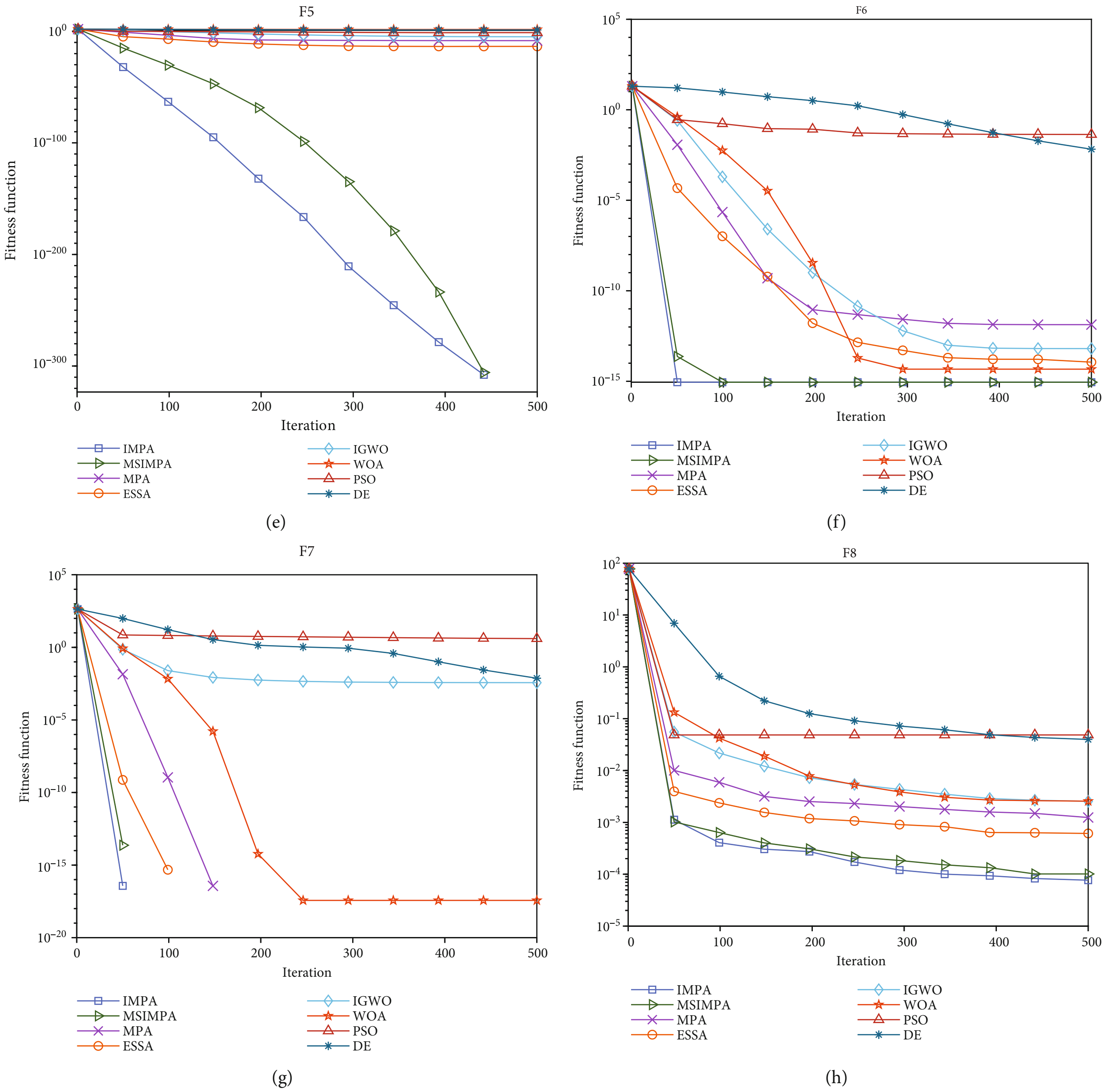
(a)

(b)

(c)

(d)

Figure 7: Continued.



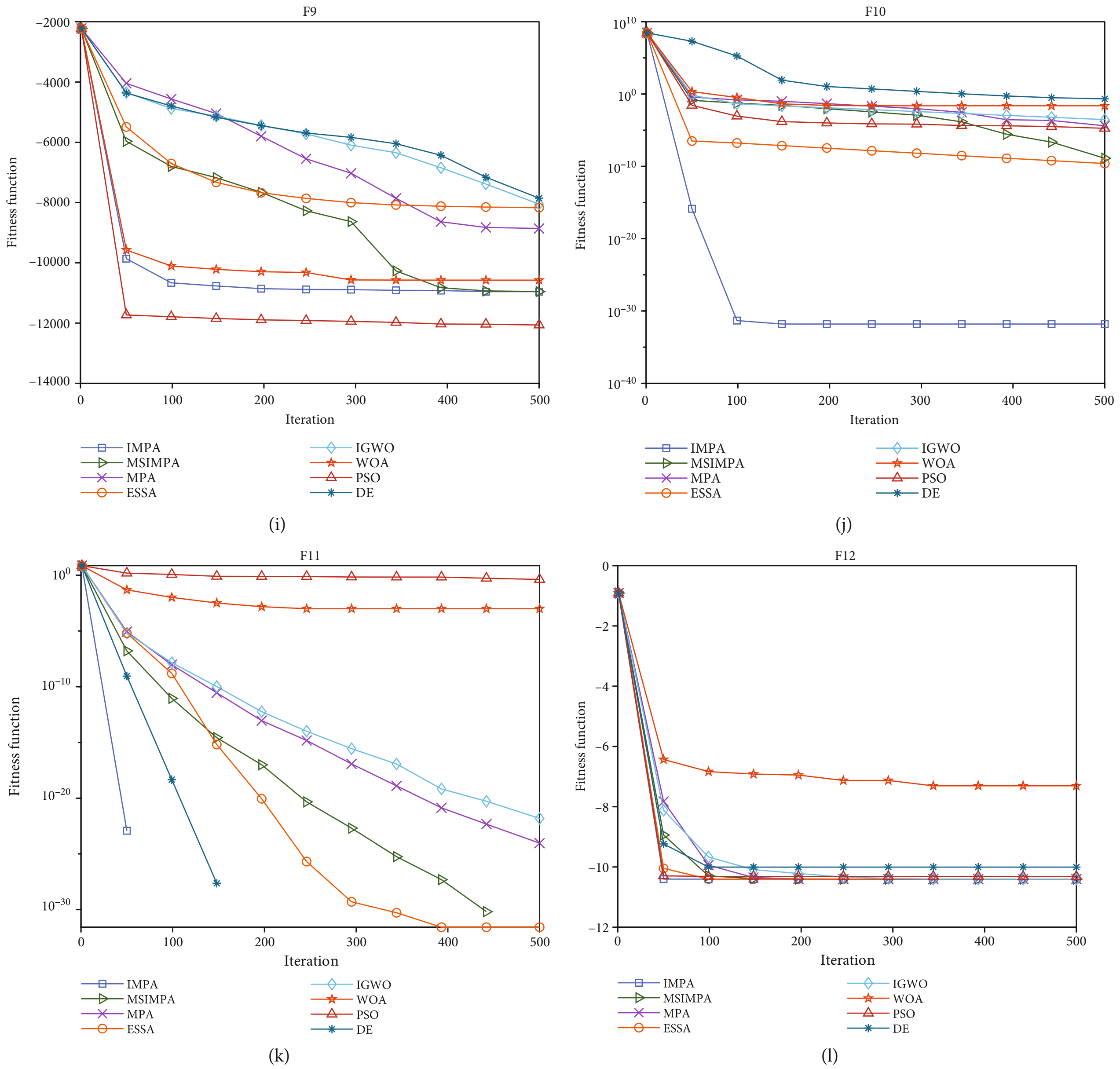
(e)

(f)

(g)

(h)

Figure 7: Continued.



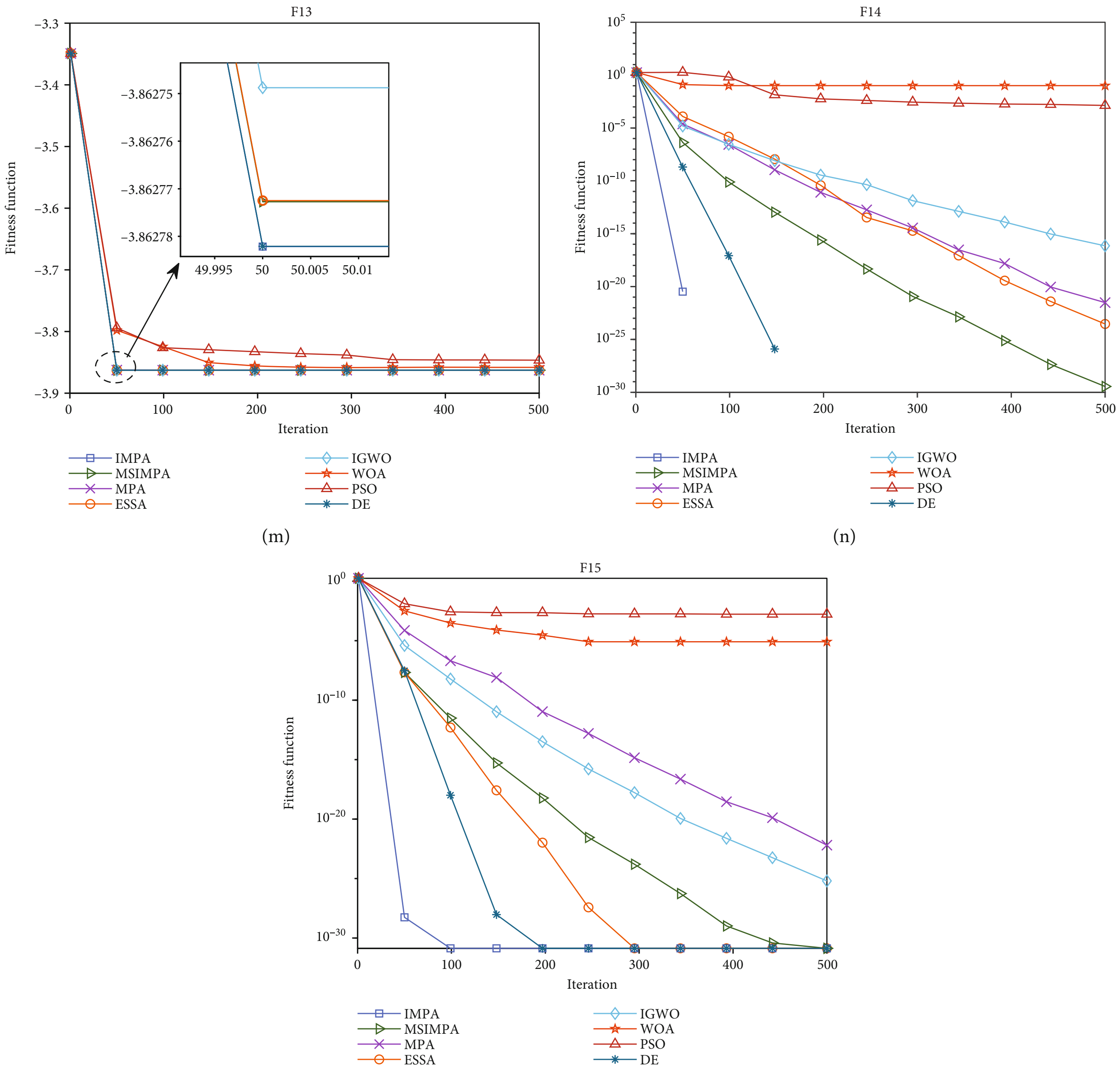
(i)

(j)

(k)

(l)

Figure 7: Continued.



(m)

(n)

(o)

Figure 7: Comparison of convergence curves.

neighborhood around the elite matrix, $P\{L(X^*)\} = P\{X|f(X) < f(X^*), X \in S\} \neq 0$, the Condition 4 is satisfied.

In summary, the IMPA algorithm satisfies both conditions of Lemma 2. Therefore, the IMPA algorithm converges globally according to the probability equal to 1.

5. Function Test and Result Analysis

In this study, 15 benchmark functions are selected for testing to verify the performance of the proposed IMPA algorithm. The results are compared with the seven algorithms of MSIMPA, MPA, ESSA, IGWO, WOA, PSO, and DE. The information of all the 15 test functions are shown in Table 1, where F1-F5 are complex unimodal functions, F6-F10 are multidimensional multimodal functions, and F11-F15 are fixed-dimensional multimodal functions. The population size of all algorithms and the maximum number of iterations are set to 30 and 500, respectively. The multidimensional functions (F1-F10) are tested in 30 and 100 dimensions, respectively. The main parameters of each algorithm are set as shown in Table 2. Each of the six algorithms is run 30 times independently. The best fitness value (Best), the average best fitness value (Ave), the standard deviation of the best fitness value (Std) results, and the average running time are recorded. To intuitively observe the performance difference of each algorithm, the algorithms are ranked according to the average best fitness value. If the average best fitness values are equal, the algorithms would then be ranked according to the best standard deviation value.

5.1. Numerical Analysis. Tables 3 and 4 revealed that the IMPA algorithm has a significant advantage over other

Table 5: The p value of the Wilcoxon rank sum test on 15 benchmark functions (30-dimension and fixed-dimension).

Function	MSIMPA p value S	MPA p value S	ESSA p value S	IGWO p value S	WOA p value S	PSO p value S	DE p value S
F1	N/A=	1.212e-12+	1.212e-12+	1.212e-12+	1.212e-12+	1.212e-12+	1.212e-12+
F2	N/A=	1.212e-12+	4.574e-12+	1.212e-12+	1.212e-12+	1.212e-12+	1.212e-12+
F3	3.020e-11+	3.020e-11+	3.020e-11+	3.020e-11+	3.020e-11+	5.573e-10+	3.020e-11+
F4	1.212e-12+	1.212e-12+	1.212e-12+	1.212e-12+	1.212e-12+	1.212e-12+	1.212e-12+
F5	N/A=	1.212e-12+	4.574e-12+	1.212e-12+	1.212e-12+	1.212e-12+	1.212e-12+
F6	N/A=	1.212e-12+	4.191e-02+	1.031e-12+	1.163e-09+	1.212e-12+	1.212e-12+
F7	N/A=	N/A=	N/A=	6.617e-04+	3.337e-01-	1.212e-12+	1.212e-12+
F8	4.515e-02+	3.020e-11+	3.081e-08+	3.020e-11+	7.739e-06+	3.020e-11+	3.020e-11+
F9	9.117e-01-	3.020e-11+	1.777e-10+	6.722e-10+	9.941e-01-	3.256e-07+	3.020e-11+
F10	1.212e-12+	1.212e-12+	1.212e-12+	1.212e-12+	1.212e-12+	1.212e-12+	1.212e-12+
F11	N/A=	4.574e-12+	3.337e-01-	1.212e-12+	1.212e-12+	1.212e-12+	N/A=
F12	2.758e-11+	1.449e-11+	8.435e-06+	1.449e-11+	1.449e-11+	1.136e-11+	7.338e-03+
F13	3.128e-01-	9.452e-09+	3.568e-07+	2.119e-05+	1.720e-12+	1.720e-12+	3.337e-03+
F14	2.19e-06+	1.212e-12+	1.657e-11+	1.212e-12+	1.212e-12+	1.212e-12+	N/A=
F15	N/A=	1.212e-12+	N/A	1.212e-12+	1.212e-12+	1.212e-12+	N/A=
+/=/-	6/7/2	14/1/0	13/1/1	15/0/0	13/0/2	15/0/0	12/3/0

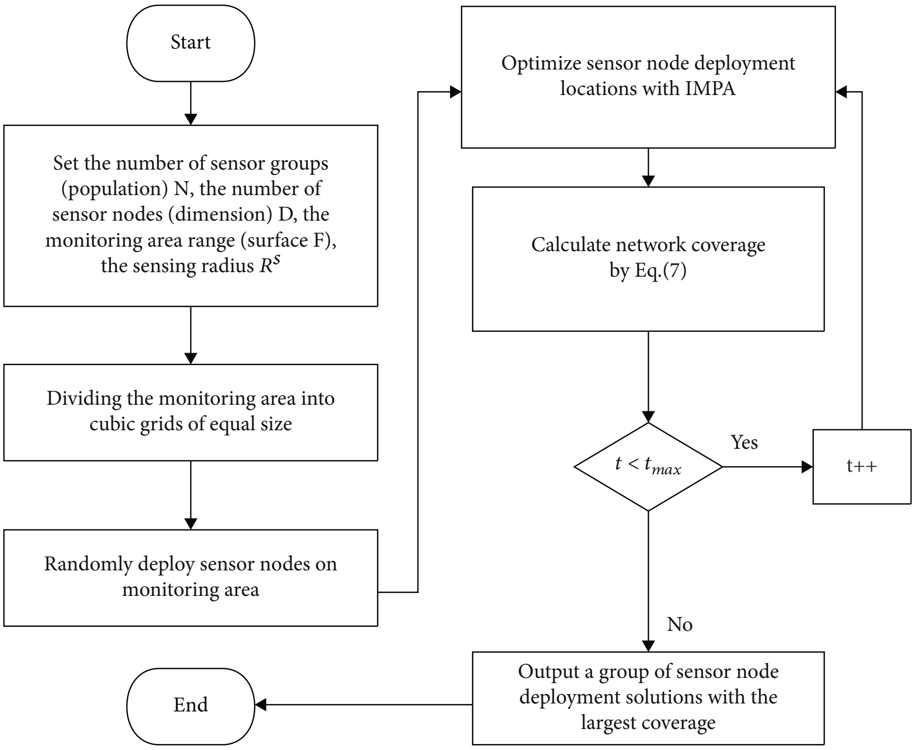
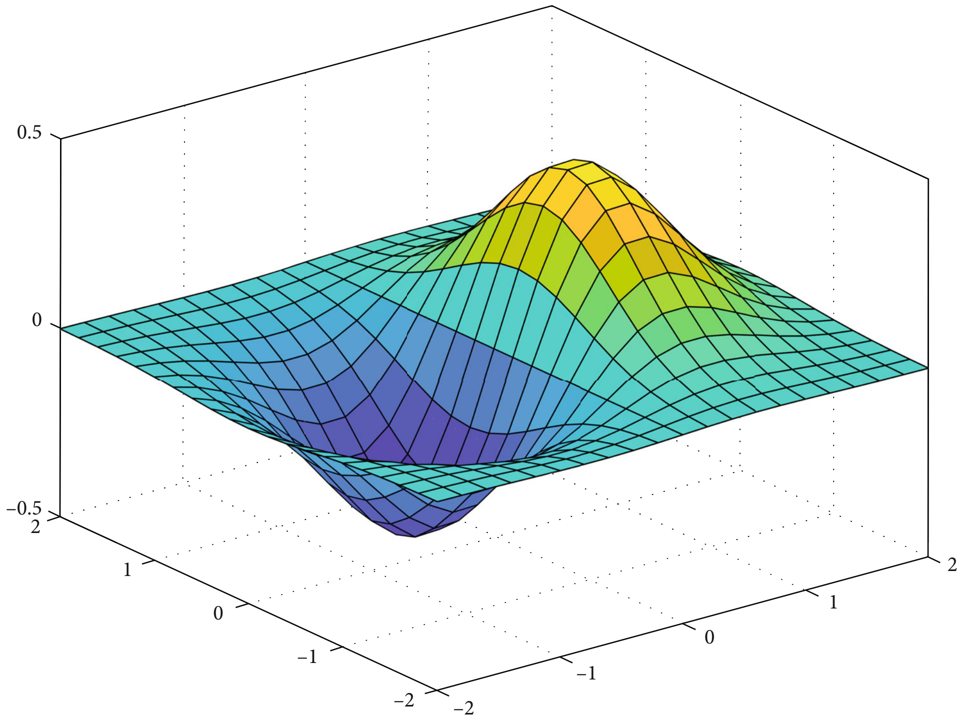
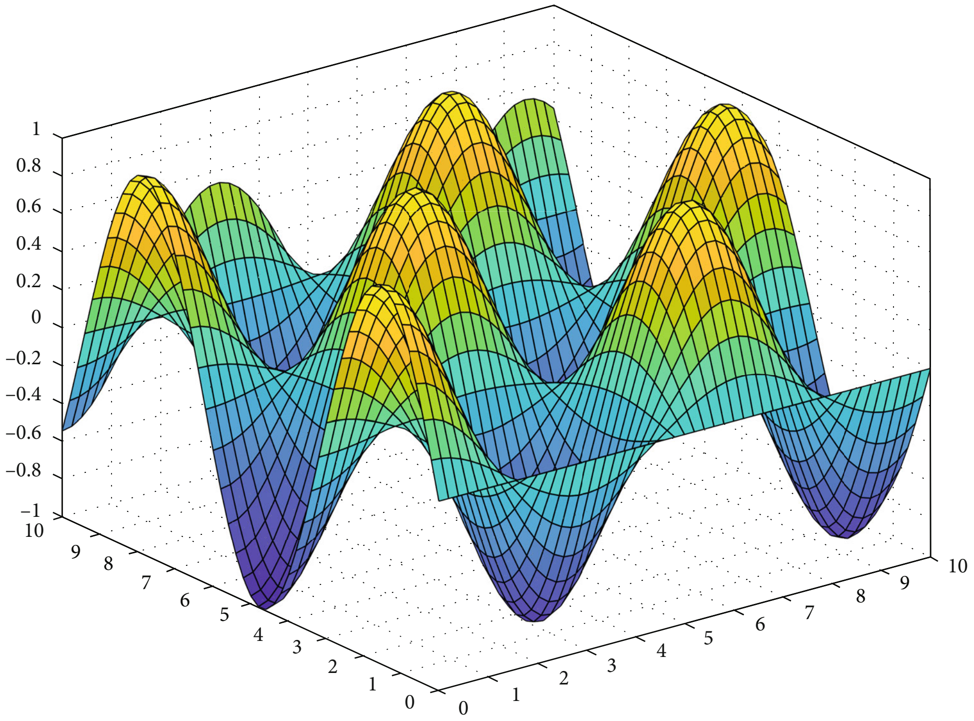

Start
Set the number of sensor groups (population) N, the number of sensor nodes (dimension) D, the monitoring area range (surface F), the sensing radius R^S
Dividing the monitoring area into cubic grids of equal size
Randomly deploy sensor nodes on monitoring area
Optimize sensor node deployment locations with IMPA
Calculate network coverage by Eq.(7)
$t < t_{max}$
Yes
t++
No
Output a group of sensor node deployment solutions with the largest coverage
End

Figure 8: The flow chart of the IMPA applied in WSN.

algorithms in the average best fitness and standard deviation values within the operation of both 30 and 100 dimensions. The IMPA algorithm ranks the best optimization performance over the other seven algorithms. Table 3 revealed that the IMPA algorithm could evaluate the theoretical optimum for F1-F2, F4-F5, F7, and F11-F14, the algorithm shows a strong global search capability. Although the IMPA could not assess the theoretical optimum for the F6, F8, F9-F10, and F15, it achieves the highest optimization accuracy over the other seven algorithms. None of the six algorithms in



(a) Undulating terrain



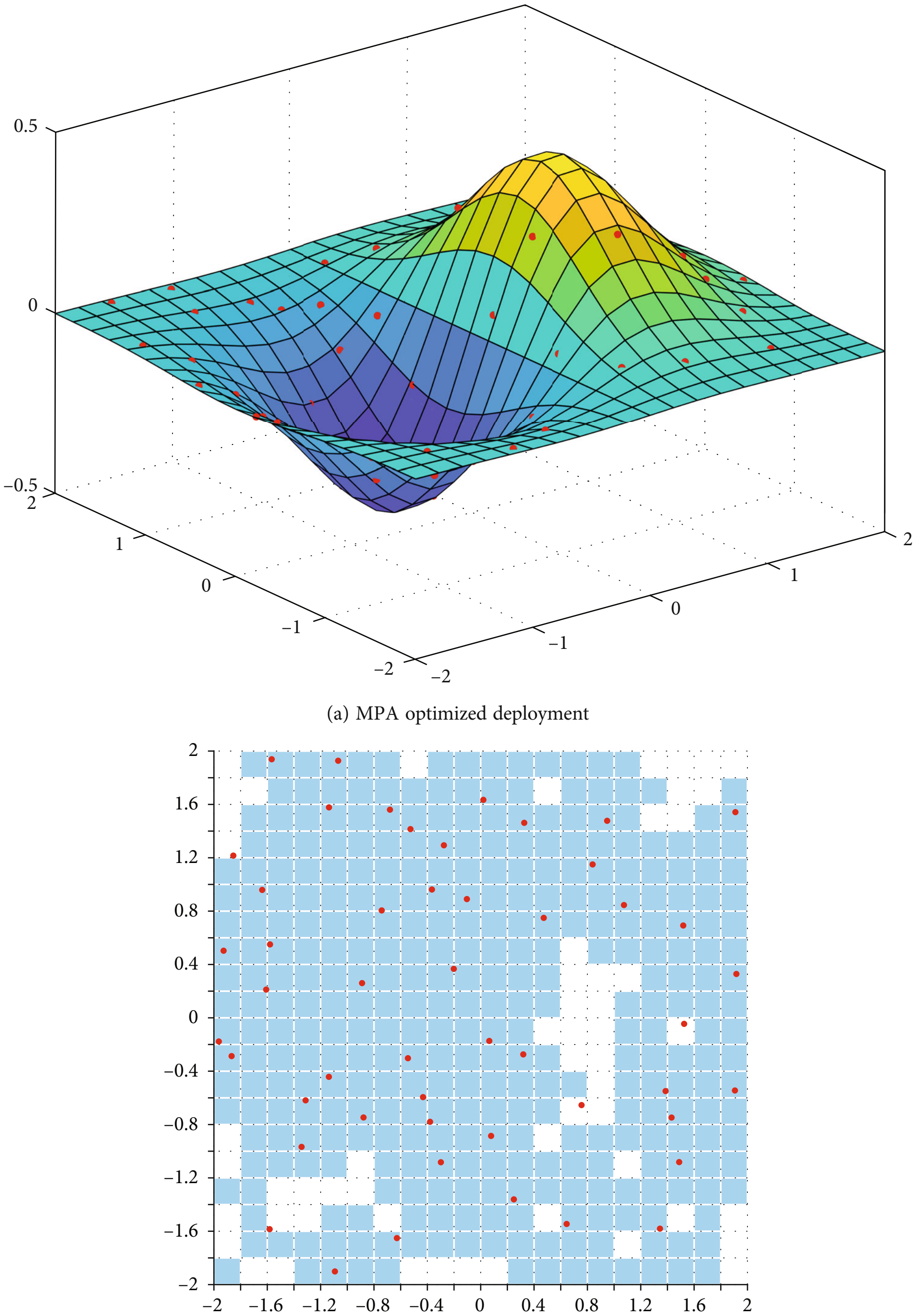
(b) Rugged terrain

Figure 9: 3D maps of two terrains.

F3 could evaluate the optimal value, but the ESSA achieves the highest optimization accuracy, exceeding five orders of magnitude. The optimization accuracy and stability of the MSIMPA algorithm, as well as the IMPA, show good optimization performance on the multiple functions. The optimization performance of the MPA in F13 ranks the best with good optimization ability and stability. However, the IMPA and MPA show a slight difference in the terms of standard deviation value. It indicates that the IMPA could not reduce the optimization capability of the algorithm.

Table 6: Comparison of average coverage of undulating terrain.

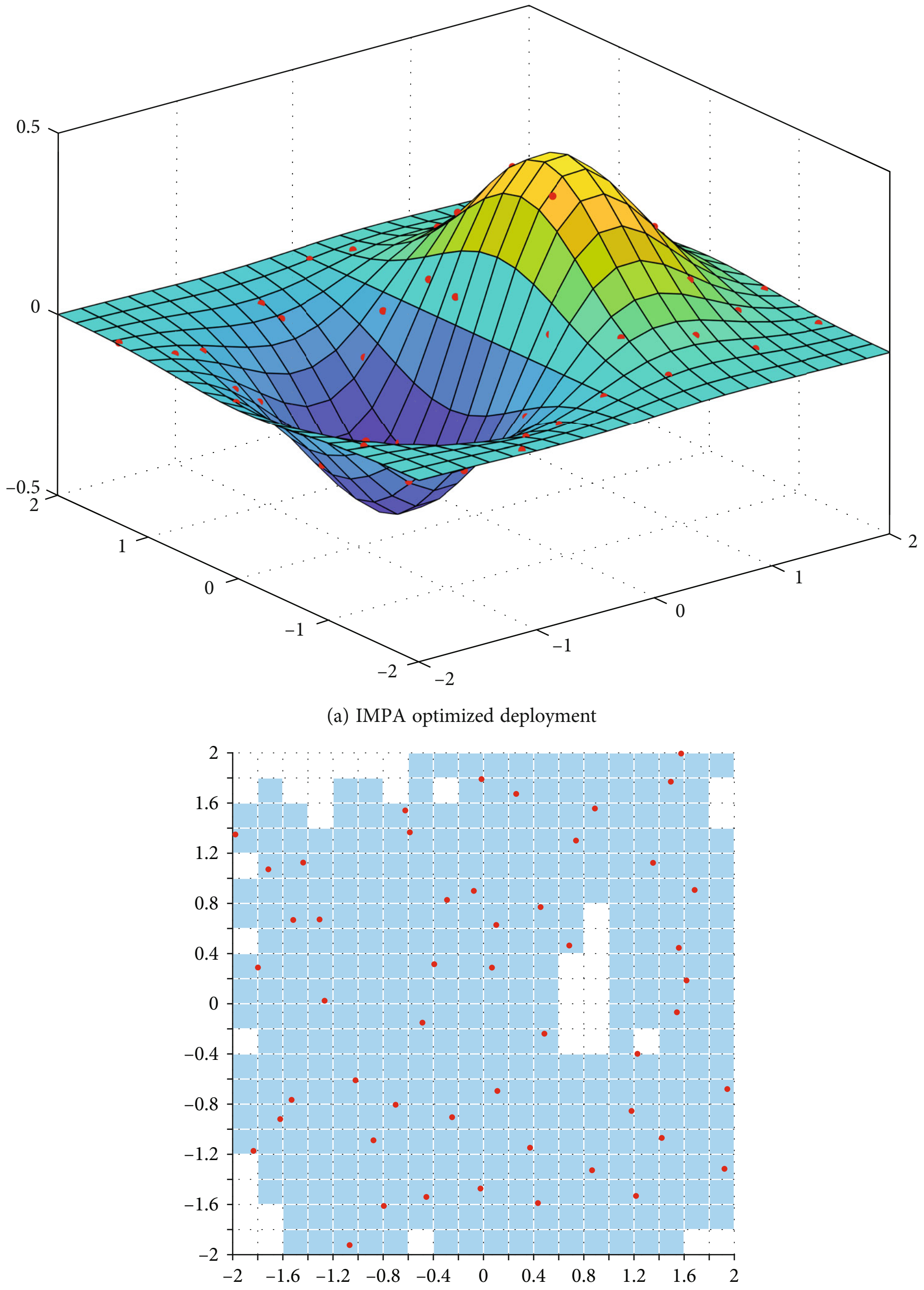
Algorithm	Average coverage (%)					
	30 nodes	40 nodes	50 nodes	60 nodes	70 nodes	80 nodes
WOA	61.94	69.46	77.58	79.01	84.1	88.98
ESSA	64.07	71.76	79.48	81.39	84.84	87.13
MPA	70.15	80.79	86.61	90.28	94.37	94.8
MSIMPA	72.21	81.37	87.67	91.63	94.77	95.33
IMPA	**74.19**	**83.2**	**90.05**	**93.6**	**96.59**	**96.81**



(a) MPA optimized deployment

(b) MPA optimized deployment of 2D view

Figure 10: MPA optimized deployment on undulating terrain (50 nodes).



(a) IMPA optimized deployment

(b) IMPA optimized deployment of 2D view

Figure 11: IMPA optimized deployment on undulating terrain (50 nodes).

The IMPA can still maintain a high solution accuracy in the 100-dimensional test. For instance, all the six algorithms fall into the local optimum in F10. However, the average optimization accuracy of the IMPA is at least 24 times greater than the magnitude of other algorithms. It shows a stronger capability to escape the local optimal. Moreover, it proves the effectiveness of the local search strategy based on the improved SFLA.

5.2. Convergence Analysis. In this paper, the average convergence curves of the algorithms are selected to evaluate the convergence ability of the optimization algorithms. Figure 7 shows each algorithm's 30-dimensional average convergence curves on the 15 benchmark test functions. The IMPA has excellent convergence speed and convergence accuracy on the remaining 14 test functions except for the F3. In F1, F2, F5, and F7, both IMPA and MSIMPA converge to the theoretical optimum quickly. In F12 and F13, the optimization precision and convergence speed of the IMPA improved a little over the other seven algorithms. It is worth mentioning that the DE can quickly converge to the global optimal value in F11, F14 and F15, which shows

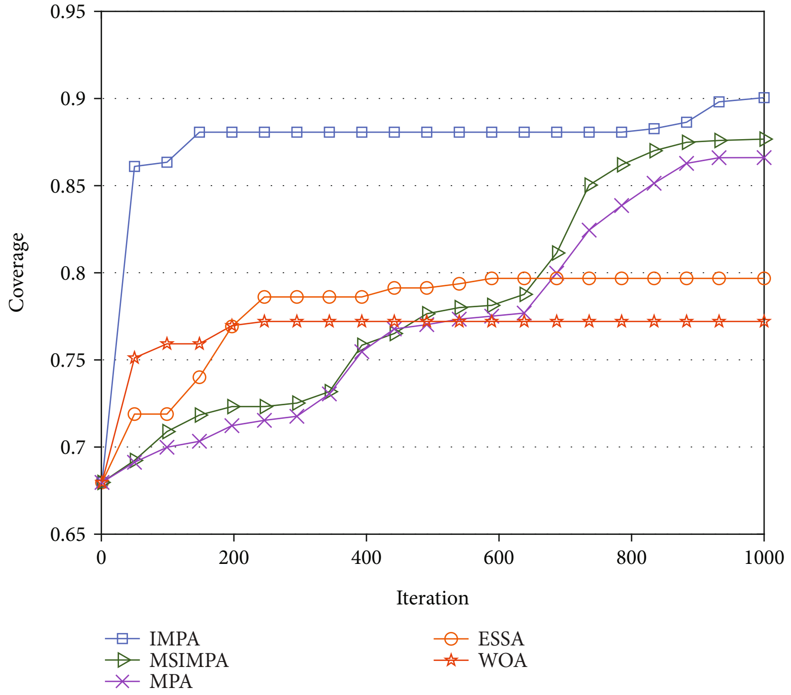


Figure 12: Convergence comparison in undulating terrain (50 nodes).

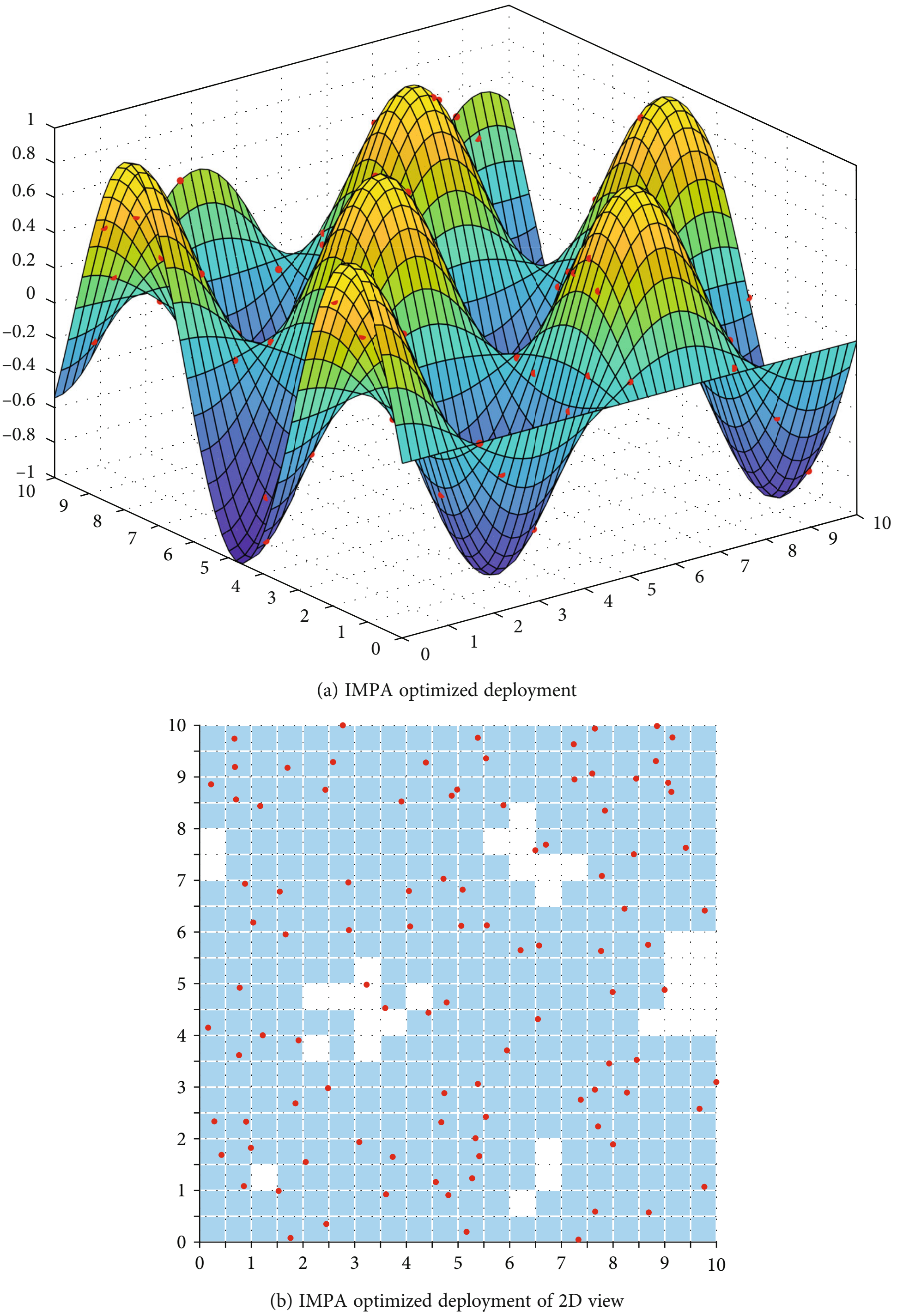
excellent optimization ability. In addition, it can be seen from Tables 3 and 4 that the running time of the proposed IMPA algorithm on most test functions is about twice that of MPA, but the optimization accuracy of IMPA has been greatly improved. Therefore, it is worth sacrificing a little time to obtain a higher quality of the solution. IMPA can solve the issues of low optimization accuracy and low convergence rate of MPA and shows significant improvements in the convergence speed and solution accuracy. These improvements are due to the implementation of the population evolution strategy and the dynamic learning mechanism, which effectively enhance the algorithm's ability for the global searching while enriching the diversity of the population.

5.3. The Wilcoxon Rank Sum Test. Only relying on the comparison of mean and standard deviation values, the superiority of the IMPA cannot be approved entirely [57]. Therefore, it needs to be tested at the statistical level. In this paper, to distinguish the significant differences in the optimization performance of the IMPA over the other seven algorithms, Wilcoxon statistical tests are performed. Test results of 30-dimension and fixed-dimension on the benchmark functions at the significant level of $p = 5\%$ are obtained. The test results are shown in Table 5. When $p < 5\%$, the zero hypothesis considers a rejection, indicating that there is a significant difference between both algorithms. When $p > 5\%$, it indicates that the difference between both algorithms is not substantial. Since the best algorithm on a benchmark function cannot be compared with itself, the best algorithm on each benchmark function is marked as N/A, indicating that the performance is comparable and optimal between the two. "S" represents the saliency judgment result, and the symbols "+", "-", and "=," respectively, indicate that the performance of IMPA is better than, worse than, and equivalent to the comparison algorithm. As shown in Table 5, most of the p values are less than 5%, indicating that the IMPA algorithm is significantly different in most of the test functions among the other seven algorithms. For F1, F2, F5-F7, F11, and F15, the performance of IMPA and MSIMPA are comparable, and all can obtain the best value. In general, the IMPA algorithm shows significant advantages over the other seven algorithms.

6. Application of 3D Surface WSN Coverage Based on IMPA

In this section, simulation of the performed experiments is conducted under two kinds of 3D terrains named undulating and rugged terrains. The flow chart of the proposed IMPA algorithm applied to 3D surface WSN coverage optimization is shown in Figure 8. The three-dimensional maps of both terrains are shown in Figure 9. In this research, to simplify the problem model, both terrains are simulated using the surface equation of the function. And the simulation results for IMPA, MSIMPA, MPA, ESSA, and WOA are compared.

6.1. Undulating Terrain Simulation Experiment. The surface equation of the simulated undulating terrain is $z = xe^{-x^2-y^2}$. The length and width of the mapped surface into the 2D plane are set to 4 m. The sensing radius of the sensor nodes is 0.5 m. The population size and iterations set to 30 and 1000, respectively. The experiments are independently run 20 times for each group to obtain the average result. The surface is divided into 400 sections using the grid division



(a) IMPA optimized deployment

(b) IMPA optimized deployment of 2D view

Figure 13: IMPA optimized deployment on rugged terrain.

method, and deployed with $\{30, 40, \cdots, 80\}$ nodes. The simulation results are as shown in Table 6.

Table 6 revealed that the average coverage increases with the increase of sensor nodes. Under the same conditions, the network coverage of sensor nodes optimally deployed by the IMPA algorithm proposed in this paper is the largest. The coverage optimization ability of WOA and ESSA is poor, and there is a big gap compared with the other three algorithms. When the number of nodes is 50, the average coverage of the IMPA algorithm is 3.44% greater than that of the MPA and 12.47% higher than the WOA. When the number of nodes is 80, the average coverage of the IMPA reaches to 96.81%. The average coverage of the IMPA does not increase significantly after 70 nodes. It is mainly due to the redundancy of nodes in the network, weakening the algorithm's performance.

Figures 10 and 11 show the optimized deployment results of the MPA and IMPA at 50 nodes, respectively. As

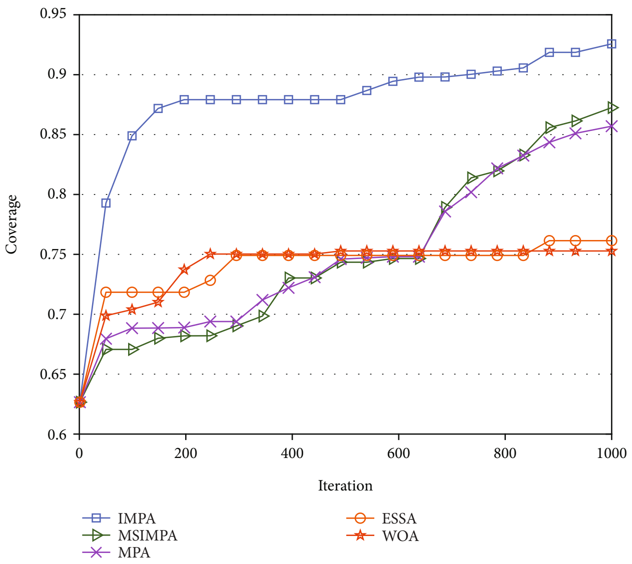


Figure 14: Convergence comparison in rugged terrain.

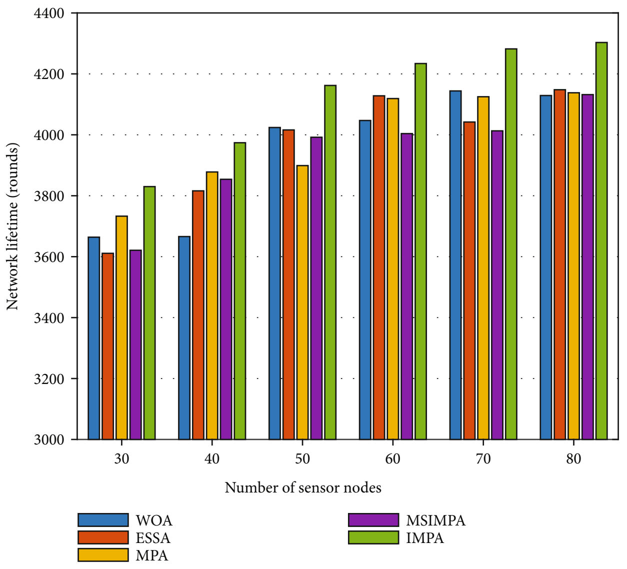


Figure 15: Comparison of the life cycle of the five algorithms in undulating terrain.

shown in Figure 10(b), in order to understand the distribution location and coverage areas of the nodes, we map the 3D coverage results into a 2D plane. The red dots indicate the deployment locations of the sensor nodes and the blue areas indicate the coverage areas. Figure 12 shows the coverage iteration curves of the five algorithms.

Figures 10(b) and 11(b)show that the sensor nodes deployed by the IMPA are more uniform, and the coverage area is larger. Figure 12 shows that the convergence rate of the IMPA is more rapid compared to the MPA. At about the 200-th generation, the coverage rate of the IMPA reaches to 88%, while the MPA algorithm is only about 72%, a difference of about 16% between the two. This shows that the proposed improved strategy can speed up the optimization speed of the algorithm.

6.2. Rugged Terrain Simulation Experiment. Simulations using more complex rugged terrain are conducted to further validate the deployment capabilities of the IMPA algorithm. The simulated surface equation of the rugged terrain is $z = \cos(x)\sin(y)$. The length and width of the mapped surface into the 2D plane are adjusted to 10 m. The sensing radius of the sensor nodes is 1 m. The population size and iterations are set to 50 and 300, respectively. The simulations are independently run 20 times for each group to obtain the average value. In this simulation, the number of sensor nodes is adjusted to 100 because the rugged terrain is complex, and the 3D perception blind area is larger. The surface is also divided into 400 sections. The results of the optimized deployment of the IMPA are shown in Figure 13.

Figure 13 shows that the optimized nodes by the IMPA are uniformly deployed on the surface. The coverage area significantly becomes greater while comparing to the initial deployment. However, deploying nodes in the convex region of the 3D surface due to the existence of the 3D perception blind area would be difficult. It results in a relatively concentrated coverage hole which is shown in the blank area of Figure 13(b). Figure 14 illustrates the coverage convergence curves of both algorithms in rugged terrain. It shows that the convergence rate of the IMPA is very high within the early stages. Around the 200-th generation, the coverage quickly reaches to 87.5% where its final coverage rate is optimized to 92.57%. The results indicate that the final coverage for the IMPA is 6.82% greater than the MPA (85.75%). It shows that the IMPA algorithm integrating the three proposed improved strategies of this research could maintain good deployment performance within the complex rugged terrain. Moreover, the IMPA algorithm shows good terrain adaptability.

In summary, by comparing both simulation results for the different terrains, the IMPA could effectively improve the coverage while solving the coverage problem for the complex 3D surface WSN. The nodes could be deployed more uniformly and have more robust adaptability to the different terrains. Finally, it verifies the improvement and effectiveness of the proposed strategies.

6.3. Network Life Cycle Comparison. The control of energy consumption is very important to the network life of WSN and is directly related to the life cycle of the entire network. Under the same conditions, the smaller the energy consumption of WSN, the longer the life cycle of the network. The LEACH protocol is a low-energy routing algorithm commonly used in WSN. The basic idea is to distribute the energy load of the entire network evenly to each sensor node by randomly and cyclically selecting the cluster head node, thereby reducing the energy consumption of the network and improving the overall survival time of the network. In this paper, the LEACH protocol is used for data transmission, and the specific energy consumption model is shown in the literature [58]. Reference [59] uses the death time of the last node to represent the life cycle of the network. Therefore, this paper also uses the death time of the last node to test the network life cycle.

The comparison of different algorithms in undulating terrain is shown in Figure 15. It can be seen from the figure

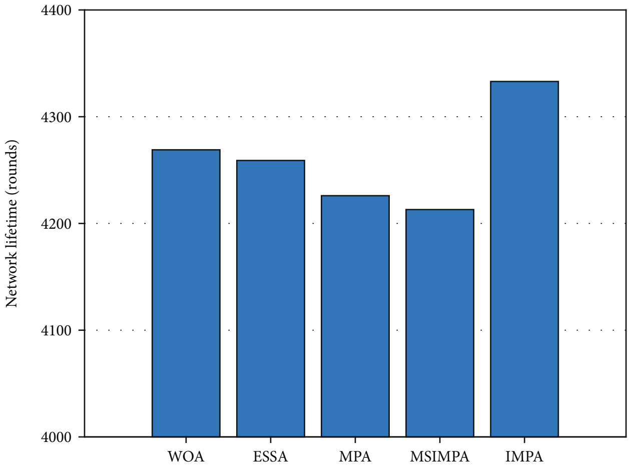


Figure 16: Comparison of the life cycle of the five algorithms in rugged terrain.

that the network lifetime increases as the number of deployed sensor nodes increases and the proposed IMPA algorithms all obtain the highest network life cycle. Interestingly, the MSIMPA is able to achieve a higher coverage than the MPA, but his network life cycle is shorter than MPA. The comparison of the network life cycle of rugged terrain is shown in Figure 16. It can be seen that the network life cycle optimized by IMPA is the highest, reaching 4333 rounds, followed by the WOA algorithm, but the network coverage rate optimized by WOA is not ideal. Overall, the WSN optimized by the proposed IMPA algorithm can achieve higher network coverage and longer life cycle.

7. Conclusion and Future Work

This study proposed a multistrategy integrated improved marine predator algorithm (IMPA) for coverage optimization of WSN on the 3D surfaces. The fusion of the random opposition-based learning strategy and the differential evolution operator was employed to expand the search range and improve the diversity of the population to enhance the global search ability. Furthermore, the improved SFLA strategy, as a local search algorithm, was employed to replace the original FADs effect of MPA, reducing the blind search of individuals and improving the algorithm's local exploitation ability. Moreover, a dynamic learning mechanism based on quasireflected opposition-based learning was proposed to improve the algorithm's convergence speed and optimization search accuracy. The IMPA algorithm is compared with the other seven algorithms in 15 benchmark test functions. The results show that IMPA has significant advantages in solution accuracy and convergence speed in both 30 and 100 dimensions. Finally, IMPA is applied to the node deployment of WSN on two 3D surfaces. The simulation results show that the nodes optimized by IMPA are more uniform, have higher coverage, longer network life cycle, and have faster convergence speed. It also shows that the IMPA has strong adaptability to the two 3D simulated terrains and has certain practicability and effectiveness.

Because of the complexity of the 3D surfaces and the existence of 3D perception blind area, it is difficult to ensure the connectivity of the deployed networks. In the future, we will study appropriate methods to ensure the connectivity of WSN deployed on 3D surfaces.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflict of interest with respect to the research, authorship and/or publication of this article.

Acknowledgments

This work is supported by the Natural Science Foundation of China (62062037, 61562037) and the Natural Science Foundation of Jiangxi Province (20212BAB202014, 20171BAB202026).

References

[1] J. Wang, Y. Gao, K. Wang, A. K. Sangaiah, and S.-J. Lim, "An affinity propagation-based self-adaptive clustering method for wireless sensor networks," *Sensors*, vol. 19, no. 11, 2019.

[2] S. Abdollahzadeh and N. J. Navimipour, "Deployment strategies in the wireless sensor network: a comprehensive review," *Computer Communications*, vol. 91-92, pp. 1–16, 2016.

[3] T. H. F. Khan and D. S. Kumar, "Ambient crop field monitoring for improving context based agricultural by mobile sink in WSN," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 4, pp. 1431–1439, 2020.

[4] W.-H. Nam, T. Kim, E.-M. Hong, J.-Y. Choi, and J.-T. Kim, "A wireless sensor network (WSN) application for irrigation facilities management based on information and communication technologies (ICTs)," *Computers and Electronics in Agriculture*, vol. 143, pp. 185–192, 2017.

[5] P. Chaturvedi and A. K. Daniel, "A comprehensive review on scheduling based approaches for target coverage in WSN," *Wireless Personal Communications*, vol. 123, no. 4, pp. 3147–3199, 2022.

[6] Z.-G. Du, J.-S. Pan, S.-C. Chu, H.-J. Luo, and P. Hu, "Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks," *IEEE Access*, vol. 8, pp. 8583–8594, 2020.

[7] Z. Wang, H. Xie, D. He, and S. Chan, "Wireless sensor network deployment optimization based on two flower pollination algorithms," *IEEE Access*, vol. 7, pp. 180590–180608, 2019.

[8] A. Prasanth and S. Jayachitra, "A novel multi-objective optimization strategy for enhancing quality of service in IoT-enabled WSN applications," *Peer-to-Peer Networking and Applications*, vol. 13, no. 6, pp. 1905–1920, 2020.

[9] L. Cao, Y. Yue, Y. Cai, and Y. Zhang, "A novel coverage optimization strategy for heterogeneous wireless sensor networks based on connectivity and reliability," *IEEE Access*, vol. 9, pp. 18424–18442, 2021.

[10] L. Wang, W. Wu, J. Qi, and Z. Jia, "Wireless sensor network coverage optimization based on whale group algorithm,"

Computer Science and Information Systems, vol. 15, no. 3, pp. 569–583, 2018.

[11] H. T. T. Binh, N. T. Hanh, L. Van Quan, and N. Dey, "Improved cuckoo search and chaotic flower pollination optimization algorithm for maximizing area coverage in wireless sensor networks," *Neural Computing and Applications*, vol. 30, no. 7, pp. 2305–2317, 2018.

[12] X. Lu, Y. Su, Q. Wu, Y. Wei, and J. Wang, "An improved coverage gap fixing method for heterogenous wireless sensor network based on Voronoi polygons," *Alexandria Engineering Journal*, vol. 60, no. 5, pp. 4307–4313, 2021.

[13] M.-C. Zhao, J. Lei, M.-Y. Wu, Y. Liu, and W. Shu, "Surface coverage in wireless sensor networks," in *IEEE INFOCOM 2009 - The 28th Conference on Computer Communications*, pp. 109–117, Rio de Janeiro, Brazil, Apr. 2009.

[14] S. Temel, N. Unaldi, and O. Kaynak, "On deployment of wireless sensors on 3-D terrains to maximize sensing coverage by utilizing cat swarm optimization with wavelet transform," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 1, pp. 111–120, 2014.

[15] M. Zafer, M. R. Senouci, and M. Aissani, "On coverage of 3D terrains by wireless sensor networks," in *Proceedings of the 2019 Federated Conference on Computer Science and Information Systems*, pp. 501–504, Leipzig, Germany, Sep. 2019.

[16] B. Cao, J. Zhao, P. Yang, P. Yang, X. Liu, and Y. Zhang, "3-D deployment optimization for heterogeneous wireless directional sensor networks on Smart City," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1798–1808, 2019.

[17] B. Cao, J. Zhao, Z. Lv, X. Liu, X. Kang, and S. Yang, "Deployment optimization for 3D industrial wireless sensor networks based on particle swarm optimizers with distributed parallelism," *Journal of Network and Computer Applications*, vol. 103, pp. 225–238, 2018.

[18] J. Zhou, G. Qi, and C. Liu, "A chaotic parallel artificial fish swarm algorithm for water quality monitoring sensor networks 3D coverage optimization," *Journal of Sensors*, vol. 2021, Article ID 5529527, 12 pages, 2021.

[19] W. Fu, Y. Yang, G. Hong, and J. Hou, "WSN deployment strategy for real 3D terrain coverage based on greedy algorithm with DEM probability coverage model," *Electronics*, vol. 10, no. 16, p. 2028, 2021.

[20] A. Yildiz, "A comparative study on the optimal non-linear seat and suspension design for an electric vehicle using different population-based optimisation algorithms," *International Journal of Vehicle Design*, vol. 80, no. 2/3/4, pp. 241–256, 2019.

[21] A. Yildiz, "Parametric synthesis of two different trunk lid mechanisms for sedan vehicles using population-based optimisation algorithms," *Mechanism and Machine Theory*, vol. 156, article 104130, 2021.

[22] Z. Li, J. Zeng, Y. Chen, G. Ma, and G. Liu, "Death mechanism-based moth–flame optimization with improved flame generation mechanism for global optimization tasks," *Expert Systems with Applications*, vol. 183, article 115436, 2021.

[23] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: a nature-inspired metaheuristic," *Expert Systems with Applications*, vol. 152, article 113377, 2020.

[24] M. A. Chi, Z. E. N. G. Guohui, H. U. A. N. G. Bo, and L. I. U. Jin, "Marine predator algorithm based on chaotic opposition learning and group learning," *Computer Engineering and Applications*, pp. 1–14, 2021, http://kns.cnki.net/kcms/detail/11.2127.TP.20210730.1554.011.html.

[25] W. A. N. G. Zhendong, W. A. N. G. Jiabao, and L. Dahai, "Study on WSN optimization coverage of an enhanced sparrow search algorithm," *Chinese Journal of Sensors and Actuators*, vol. 34, no. 6, pp. 818–828, 2021.

[26] M. H. Nadimi-Shahraki, S. Taghian, and S. Mirjalili, "An improved grey wolf optimizer for solving engineering problems," *Expert Systems with Applications*, vol. 166, article 113917, 2021.

[27] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[28] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, WA, Australia, 1995.

[29] H. Peng, Y. Han, C. Deng, J. Wang, and Z. Wu, "Multi-strategy co-evolutionary differential evolution for mixed-variable optimization," *Knowledge-Based Systems*, vol. 229, article 107366, 2021.

[30] N. Anand, R. Ranjan, B. S. Rai, and S. Varma, "A novel computational geometry-based node deployment scheme in 3D wireless sensor network," *International Journal of Sensor Networks*, vol. 25, no. 3, pp. 135–145, 2017.

[31] N. Boufares, P. Minet, I. Khoufi, and L. Saidane, "Covering a 3D flat surface with autonomous and mobile wireless sensor nodes," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1628–1633, Valencia, Spain, Jun. 2017.

[32] K. Kim, "Mountainous terrain coverage in mobile sensor networks," *IET Communications*, vol. 9, no. 5, pp. 613–620, 2015.

[33] L.-G. Zhang, F. Fan, S.-C. Chu, A. Garg, and J.-S. Pan, "Hybrid Strategy of Multiple Optimization Algorithms Applied to 3- D Terrain Node Coverage of Wireless Sensor Network," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6690824, 2021.

[34] J. Li, G.-C. Li, S.-C. Chu, M. Gao, and J.-S. Pan, "Modified parallel tunicate swarm algorithm and application in 3D WSNs coverage optimization," *Journal of Internet Technology*, vol. 23, no. 2, pp. 227–244, 2022.

[35] J.-S. Pan, Q.-W. Chai, S.-C. Chu, and N. Wu, "3-D terrain node coverage of wireless sensor network using enhanced black hole algorithm," *Sensors*, vol. 20, no. 8, p. 2411, 2020.

[36] S. Sun, L. Sun, and S. Chen, "Research on the target coverage algorithms for 3D curved surface," *Chaos, Solitons & Fractals*, vol. 89, pp. 397–404, 2016.

[37] Z. Wang and H. Xie, "Wireless sensor network deployment of 3D surface based on enhanced grey wolf optimizer," *IEEE Access*, vol. 8, pp. 57229–57251, 2020.

[38] E. H. Houssein, M. A. Mahdy, A. Fathy, and H. Rezk, "A modified marine predator algorithm based on opposition based learning for tracking the global MPP of shaded PV system," *Expert Systems with Applications*, vol. 183, article 115253, 2021.

[39] M. Abdel-Basset, R. Mohamed, M. Elhoseny, R. K. Chakrabortty, and M. Ryan, "A hybrid COVID-19 detection model using an improved marine predators algorithm and a ranking-based diversity reduction strategy," *IEEE Access*, vol. 8, pp. 79521–79540, 2020.

[40] M. Abd Elaziz, S. B. Thanikanti, I. A. Ibrahim et al., "Enhanced marine predators algorithm for identifying static and dynamic photovoltaic models parameters," *Energy Conversion and Management*, vol. 236, article 113971, 2021.

[41] Q. Li and N. Liu, "Coverage blind area repair based on perceived multimedia data driven in mobile wireless sensor networks," *Advances in Multimedia*, vol. 22, Article ID 2354024, 2022.

[42] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, vol. 1, pp. 695–701, Vienna, Austria, 2005.

[43] Z. Liang, J. Zhang, L. Feng, and Z. Zhu, "A hybrid of genetic transform and hyper-rectangle search strategies for evolutionary multi-tasking," *Expert Systems with Applications*, vol. 138, article 112798, 2019.

[44] M. Abd Elaziz, D. Oliva, and S. Xiong, "An improved opposition-based sine cosine algorithm for global optimization," *Expert Systems with Applications*, vol. 90, pp. 484–500, 2017.

[45] H. Wang, H. Li, Y. Liu, C. Li, and S. Zeng, "Opposition-based particle swarm algorithm with cauchy mutation," in *2007 IEEE Congress on Evolutionary Computation*, pp. 4750–4756, Singapore, Sep. 2007.

[46] W. Dong, L. Kang, and W. Zhang, "Opposition-based particle swarm optimization with adaptive mutation strategy," *Soft Computing*, vol. 21, no. 17, pp. 5081–5090, 2017.

[47] Y. Zhou, J.-K. Hao, and B. Duval, "Opposition-based memetic search for the maximum diversity problem," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 5, pp. 731–745, 2017.

[48] Q. Fan, H. Huang, Q. Chen, L. Yao, K. Yang, and D. Huang, "A modified self-adaptive marine predators algorithm: framework and engineering applications," *Engineering with Computers*, vol. 38, no. 4, pp. 3269–3294, 2022.

[49] W. Long, J. Jiao, X. Liang, S. Cai, and M. Xu, "A random opposition-based learning grey wolf optimizer," *IEEE Access*, vol. 7, pp. 113810–113825, 2019.

[50] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering Optimization*, vol. 38, no. 2, pp. 129–154, 2006.

[51] J. D. Knowles and D. W. Corne, "M-PAES: a memetic algorithm for multiobjective optimization," in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, vol. 1, pp. 325–332, La Jolla, CA, USA, Jul. 2000.

[52] X. Pu, C. Xiong, and L. Zhao, "Path planning for robot based on IACO-SFLA hybrid algorithm," in *2020 Chinese Control And Decision Conference (CCDC)*, pp. 4886–4893, Hefei, China, Aug 2020.

[53] X. Wang, S. Liu, X. Teng, J. Sun, and J. Jiao, "SFLA with PSO Local Search for detection sonar image," in *2016 35th Chinese Control Conference (CCC)*, pp. 3852–3857, Chengdu, China, Jul. 2016.

[54] M. Ergezer, D. Simon, and D. Du, "Oppositional biogeography-based optimization," in *2009 IEEE International Conference on Systems, Man and Cybernetics*, pp. 1009–1014, San Antonio, TX, USA, Oct 2009.

[55] M. K. Naik, R. Panda, and A. Abraham, "Adaptive opposition slime mould algorithm," *Soft Computing*, vol. 25, no. 22, pp. 14297–14313, 2021.

[56] Z. Liu, Z. Qin, P. Zhu, and H. Li, "An adaptive switchover hybrid particle swarm optimization algorithm with local search strategy for constrained optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 95, article 103771, 2020.

[57] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.

[58] Z. Cui, Y. Cao, X. Cai, J. Cai, and J. Chen, "Optimal LEACH protocol with modified bat algorithm for big data sensing systems in internet of things," *Journal of Parallel and Distributed Computing*, vol. 132, pp. 217–229, 2019.

[59] Y. Du, "Method for the optimal sensor deployment of WSNs in 3D terrain based on the DPSOVF algorithm," *IEEE Access*, vol. 8, pp. 140806–140821, 2020.