Hindawi
Scientific Programming
Volume 2022, Article ID 8312450, 20 pages
https://doi.org/10.1155/2022/8312450

Research Article

Multipopulation Particle Swarm Optimization Algorithm with Neighborhood Learning

XiaoMing Li,[1] ZiYi Wang,[2] Yi Ying,[1] and FangXiong Xiao[3]

[1] *School of Computer Science and Engineering, Sanjiang University, Nanjing 210012, Jiangsu, China*
[2] *College of Automation & College of Artificial Intelligence, Nanjing University of Post and Telecommunications, Nanjing 210023, Jiangsu, China*
[3] *School of Software Engineering, Jinling Institute of Technology, Nanjing 211169, Jiangsu, China*

Correspondence should be addressed to XiaoMing Li; lixiaoming9215@aliyun.com

Received 13 November 2021; Revised 14 December 2021; Accepted 28 December 2021; Published 25 January 2022

Academic Editor: Pengwei Wang

Particle swarm optimization (PSO) algorithm is widely used due to its fewer control parameters and fast convergence speed. However, as its learning strategy is only learning from the global optimal particle, the algorithm has the problem of low accuracy and easily falling into local optimization. In order to overcome this defect, a multipopulation particle swarm optimization algorithm with neighborhood learning (MPNLPSO) is proposed in this article. In MPNLPSO, a small-world network neighborhood learning strategy is proposed to make particles learn from the neighborhood optimal particles instead of only the global optimal particle. Furthermore, the concept of multipopulation cooperation is introduced to balance the ability of global exploration and local exploration. In addition, a dynamic opposition-based learning strategy is proposed to effectively activate the particles in the search stagnation state. Moreover, in order to improve the accuracy of the algorithm and, to some extent, avoid the population diversity decreases too fast, as the searching process continues, Lévy flight is introduced to randomly perturb the particles of historical optimal and neighborhood optimal. To verify the performance of the proposed algorithm experimentally, twenty benchmark functions are solved. Experimental results show that the proposed multipopulation particle swarm optimization algorithm with neighborhood learning presents high efficiency and performance with a certain robustness.

1. Introduction

Recent years have witnessed the emergence of many metaheuristics such as the monarch butterfly optimization (MBO) [1], slime mould algorithm (SMA) [2], moth search algorithm (MSA) [3], and colony predation algorithm (CPA) [4]. MBO [1] was firstly proposed by Wang in 2019, which simplifies and idealizes the migration behavior of eastern North American monarch population by updating the positions of the monarch butterflies in two ways. Li et al. [2] utilized the oscillation mode of slime mould in nature with adaptive weights to simulate the process of producing positive and negative feedback of the propagation wave of slime mould based on bio-oscillator to form the optimal path for connecting food with excellent exploratory ability and exploitation propensity and proposed SMA. Inspired by the phototaxis and Lévy flights of the moths, MSA was developed by Wang [3]. Natural moths are family insects, and in MS method, the best moth individual is viewed as the light source. Some moths that are close to the fittest one always display an inclination to fly around their own positions in the form of Lévy flights. On the contrary, due to phototaxis, the moths that are comparatively far from the fittest one will tend to fly toward the best one directly in a big step. These two features correspond to the processes of exploitation and exploration of any metaheuristic optimization method. And based on the corporate predation of animals in nature, CPA was proposed by Tu et al. [4], utilizing a mathematical mapping following the strategies and introducing new features of a unique mathematical model that uses a success rate to adjust the strategy and simulate hunting animals' selective abandonment behavior.

As a traditional optimization algorithm, particle swarm optimization (PSO), a type of evolutionary algorithm, based on swarm intelligence, which is originated from the simulation study of swarm intelligence behaviors of natural organisms in the evolution process, for example, the migration of birds [5] and the foraging of ant colonies [6], was firstly proposed by Kennedy and Eberhart [7] and Shi and Eberhart [8] in 1995 for the optimization of continuous problems. In this type of algorithm, the groups can effectively identify the characteristics of the problem by sharing evolutionary information and considering the environmental factors of individual's surroundings, and finally guide the group to search toward more promising areas. It established an evolutionary search model based on the intelligent behavior of swarms by studying the foraging behavior of the flock of birds and fish. In this model, each individual in the group can constantly move toward the position of the best individual in the group, which is affected by the interaction of its own and the best historical experience of any individual in the neighborhood, and thereby finds the global optimum of the problem. Since the algorithm was proposed, it has received a lot of research due to its advantages such as high efficiency and stability in solving complex problems and not being restricted by the characteristics of the problem. However, although most modified particle swarm optimization algorithm variants can satisfy the requirements for optimization problems, as the scale of optimization problems continues to increase, the existing PSO algorithm and its variants become more and more difficult to meet the performance requirements. Studies have shown that most of the improvements of PSO algorithm usually only pay attention to modify its global search ability and ignore its local search ability, which brings about problems such as insufficient population diversity and premature convergence when dealing with complex optimization problems [9]. In addition, limited by the relatively simple evolution strategy, which is that each particle only exchanges information with the best individual in its neighborhood and therefore ignores the important influence of other individuals on the evolution process of the entire group, the PSO algorithm usually has troubles in solving complex problems. Therefore, many researchers have conducted a lot of work on PSO algorithm from theoretical analysis to practical application, for example, the adaptive research of algorithm control parameters [10], the design of population topology [11], and the hybrid design with other evolutionary algorithms [12]. The PSO algorithm has the advantage of fewer control parameters that need to be adjusted, but these control parameters have a significant impact on the performance of the algorithm, for example, inertia weight and acceleration factor. Eltamaly [13] proposed a novel strategy that can determine the optimal values of control parameters of a PSO. The newly proposed strategy uses two nested PSO (NESTPSO) searching loops: the inner one contained the original objective function, and the outer one used the inner PSO as a fitness function. The control parameters and the swarm size acted as the optimization variables for the outer loop. These variables were optimized for the lowest premature convergence rate, the lowest number of iterations, and the lowest swarm size. Wei-Min et al. [14] proposed a more simple strategy of PSO algorithm called θ-PSO. In θ-PSO, an increment of phase angle vector replaces the increment of the velocity vector and the positions are decided by the mapping of phase angles. The results of benchmark testing of nonlinear functions show that the performance of θ-PSO is much more effective than that of the standard PSO. Tian et al. [15] proposed a chaotic PSO algorithm based on the sigmoid acceleration coefficient. A uniformly distributed initial particle population is generated through logical mapping rules, the sigmoid function is used to adaptively adjust the acceleration coefficient, and the particles are updated based on the chaotic reinitialization and Gaussian mutation strategy to maintain the diversity of the population, so that the algorithm can continue to explore the potential search area. Liu et al. [16] proposed an AWPSO algorithm based on adaptive weights of the sigmoid function. In this algorithm, considering the distance between each particle and the global optimal position, an adaptive acceleration coefficient update strategy based on the sigmoid function is designed to improve the convergence performance of the algorithm.

Generally, PSO algorithms only utilize the best historical experience of individuals and groups to guide the evolution process of the population. This strategy is simple and effective, but it is difficult to obtain high-quality results when dealing with complex problems. Therefore, the design of efficient evolution strategies has aroused extensive research. An adaptive PSO algorithm using scale-free network topology is proposed by Li et al. [17]. Based on the characteristics of scale-free network topology with power-law distribution, this novel algorithm can construct a corresponding neighborhood for each particle. Then, it selects the elite particles from the community to participate in the particle evolution process and considers full play to the guiding role of elite particles within the population search process. In addition, a new adaptive weight strategy and an introduction to the differential evolution operation for achieving a balance ability to the global and local exploration within the search process are proposed. As for the population topology, Chen et al. [18] proposed a hybrid ring topology in the early evolutionary process of evolution, a sparse topology is constructed to enhance the population diversity to help locate multiple optima, while in the later stage, the population communication topology is switched to a relatively dense topology for improving the convergence efficiency on the found optima, and the threshold that controls the switch of the topology and its effect is also analyzed in this study. In multimodal optimization problems, niching parameters are often used to inform the algorithm how far apart between two closest optima or the number of optima in the search space, and are typically difficult to set as they are problem dependent. Li [19] proposed a particle swarm optimization algorithm using a ring neighborhood topology without any niching parameters, and experimental results suggest that PSO algorithms using the ring topology are able to provide superior and more consistent performance over some existing PSO niching algorithms that require niching parameters. Qin et al. [20] presented an improved PSO algorithm with an interswarm interactive learning strategy (IILPSO) by overcoming the drawbacks of the canonical PSO algorithm's learning strategy. IILPSO is inspired by the phenomenon in human society that the interactive learning behavior takes place among different groups. Particles in IILPSO are divided into two swarms. The interswarm interactive learning (IIL) behavior is triggered when the best particle's fitness value of both the swarms does not improve for a certain number of iterations. According to the best particle's fitness value of each swarm, the softmax method and roulette method are used to determine the roles of the two swarms, such as the learning swarm and the learned swarm. In addition, the velocity mutation operator and global best vibration strategy are used to improve the algorithm's global search capability. The IIL strategy is applied to PSO with global star and local ring structures that are termed as IILPSO-G and IILPSO-L algorithm, respectively. Xu et al. [21] proposed a dimensional learning strategy (DLS) for discovering and integrating the promising information of the population best solution according to the personal best experience of each particle. Thereafter, a two-swarm learning PSO (TSLPSO) algorithm based on different learning strategies is proposed. One of the subpopulations constructs the learning exemplars by DLS to guide the local search of the particles, and the other subpopulation constructs the learning exemplars by the comprehensive learning strategy to guide the global search.

Moreover, a multi-objective approach to optimization has been attempted. Zhang et al. [22] proposed a modified particle swarm optimization (AMPSO) to solve the multimodal multi-objective problems. Firstly, a dynamic neighborhood-based learning strategy is introduced to replace the global learning strategy, which enhances the diversity of the population. Meanwhile, to enhance the performance of PSO, the offering competition mechanism is utilized. An improved version of the directed weighted complex network particle swarm optimization using the genetic algorithm (GDWCN-PSO) is presented by Bharti et al. [23]. This method uses the concept of the genetic algorithm after each update to enhance convergence and diversity.

Furthermore, the PSO algorithm and its variants have been applied to solve many realistic problems. Bharti et al. [23] have applied GDWCN-PSO to solve the optimal key-based medical image encryption. It is one of the most challenging problems in health IoTs for protecting sensitive and confidential patient data as well as addressing the major concern of integrity and security of data in today's advanced digital world. In addition, constrained optimization problems, compared to unconstrained problems, are more complex problems due to their multiple constraints with different requirements. In order to solve this kind of problem, Parsopoulos and Vrahatis [24] first proposed the idea of penalty function, turning the constraint problem into an unconstrained problem. However, in this way, an appropriate penalty function factor needs to be selected; otherwise, the efficiency of the algorithm will be significantly affected.

Although the above-mentioned details have introduced some modifications on the PSO algorithm, there is little research work analyzing the diversity of network topology. Thus, a multipopulation particle swarm optimization algorithm with neighborhood learning (MPNLPSO) is proposed. The main contributions of this study can be summarized as follows:

(1) The concept of population cooperation is introduced. In the iterative process, the particle swarm is divided into two populations according to the fitness value: elite population and shoddy population.

(2) Small-world network neighborhood learning strategy is proposed. A neighborhood for each particle is constructed to make them learn from the neighborhood optimal particles, thus getting rid of the situation of only learning from the global optimal particle.

(3) Dynamic opposition-based learning strategy is proposed. It mainly aims at activating the particles trapped in search stagnation in the shoddy population.

The rest of this article is arranged as follows: Section 2 briefly reviews the PSO algorithm and some details of the small-world network, opposition-based learning, Lévy flight, respectively, additionally, the concept of diversity analysis of particle swarm is introduced in particular. The algorithm we proposed is introduced in Section 3, which firstly gives details about two strategies: small-world network neighborhood learning strategy and dynamic opposition-based learning strategy, and then presents the framework of the proposed algorithm MPNLPSO. Section 4 presents the experiment results of MPNLPSO. Firstly, benchmark functions used in the experiment and the parameter settings of the comparison algorithms are introduced. Secondly, the diversity of the VN small-world network and NW small-world network is analyzed and the solution accuracy of the proposed algorithm is verified. Thirdly, the effectiveness of the strategy is analyzed. Lastly, the proposed algorithm is compared from the perspective of convergence.

2. Related Work

2.1. Particle Swarm Optimization. In the standard particle swarm optimization algorithm, each particle $i\,(i = 1, 2, \ldots, N)$ represents a potential solution to the optimization problem and is defined by two vectors, velocity $V_i = [v_i^1, v_i^2, \ldots, v_i^D]$ and position $X_i = [x_i^1, x_i^2, \ldots, x_i^D]$. By initialization, particles are generated in a three-dimensional search space with random velocity and position values. During evolution, each particle updates its velocity and position according to the following learning strategies:

$$V_{ij}^{t+1} = V_{ij}^{t} + c_1 \times r_1\left(P\text{best}_{ij}^{t} - X_{ij}^{t}\right) + c_2 \times r_2\left(G\text{best} - X_{ij}^{t}\right), \tag{1}$$

$$X_{ij}^{t+1} = X_{ij}^{t} + V_{ij}^{t+1}, \tag{2}$$

where V_{ij}^{t+1} is the velocity of the j-dimensional component of particle i in generation $t+1$, and X_{ij}^{t+1} represents the position of the j-dimensional component of particle i in generation $t+1$. c_1 and c_2 are acceleration coefficients, and r_1 and r_2 are random numbers of [0, 1]. In order to control the influence of the particle's previous velocity on the current velocity, an inertia weight is usually added to the velocity in Equation (1), which is shown in Equation (3) as follows:

$$\begin{aligned} V_{ij}^{t+1} &= \omega \times V_{ij}^{t} + c_1 \times r_1\left(\text{Pbest}_{ij}^{t} - X_{ij}^{t}\right) \\ &\quad + c_2 \times r_2\left(\text{Gbest} - X_{ij}^{t}\right). \end{aligned} \tag{3}$$

2.2. Small-World Network.

2.2. Small-World Network. Watts and Strogtz [25] rewired the conventional network and introduced disordered lines. The resulting network has the characteristics of high agglomeration coefficient and low average path length. This network model is generally called the small-world network model. There are many forms of small-world network, among which WS small-world network and NW small-world network are widely used, and both are constructed on the basis of a ring network. WS small-world network is generated by the process of fixing a vertex in a ring network, then randomly selecting a node from the remaining nodes to reconnect with it, NW small-world network is constructed by adding edges between nodes randomly with connection probability p, and both processes must ensure that there are no duplicate edges in the network. The connection probability p is the probability of reconnection of each edge. $p = 0$ means that each edge is not reconnected, that is, it is still a ring network. $p = 1$ means that each edge is reconnected, that is, it becomes a composite network of random network and ring network. $p = (0,\ 1)$ indicates that some edges have been reconnected, that is, the NW small-world network.

The detail of the NW small-world network will be introduced due to its application in the proposed algorithm. The left side of Figure 1 shows a ring topology, where each node is connected with two points adjacent to itself on the left and right side respectively, and the connection probability is 0. The middle of Figure 1 shows the small-world network when the connection probability of the edge is 0.2. When the connection probability of the edge is 1, the small-world network will evolve into a combination of regular network and random network, which is shown on the right side of Figure 1.

In addition to building a small-world network based on the regular network above, Kleinberg [26] constructs small-world networks in another way. Rather than using a ring as the basic structure, we begin from a two-dimensional grid and make edges directed.

Figure 2 shows the Von Neumann network with swarms of size 5×5 and each particle has up, down, left, and right particles in its neighborhood, except the edge particles. Figure 3 shows that each node in the Von Neumann network is connected with its neighborhood, and in addition, randomly connected with two nodes, for instance, particle i is connected to two randomly selected particles j and k. Thus, a VN small-world network is generated.

2.3. Opposition-Based Learning. The opposition-based learning strategy [27] is a new concept that has emerged in the field of computational intelligence in recent years, and has been proven to be an effective concept to various optimization approaches [28–33]. When evaluating a solution x to a given problem, simultaneously computing its opposite solution will provide another chance for finding a candidate solution that is possibly closer to the global optimum.

Opposite number [29]: let $x \in [a, b]$ be a real number. The opposite of x is defined as follows:

$$x^{*} = a + b - x. \tag{4}$$

Similarly, the definition is generalized to higher dimensions as follows:

Opposite point [29]: let $X = (x_1, x_2, \ldots, x_D)$ be a point in a D-dimensional space, where $x_1, x_2, \ldots, x_D \in R$ and $x_j \in [a_j, b_j]$, $j = 1, 2, \ldots, D$. The opposite point $X^{*} = (x_1^{*}, x_2^{*}, \ldots, x_D^{*})$ is defined as follows:

$$x_j^{*} = a_j + b_j - x_j. \tag{5}$$

By applying the definition of opposite point, the opposition-based optimization can be defined as follows:

Opposition-based optimization [29]: let $X = (x_1, x_2, \ldots, x_{\mathrm{D}})$ be a point in a D-dimensional space (i.e., a candidate solution). Assume $f(X)$ is a fitness function, which is used to evaluate the candidate's fitness. According to the definition of the opposite point, $X^{*} = (x_1^{*}, x_2^{*}, \ldots, x_D^{*})$ is the opposite of $X = (x_1, x_2, \ldots, x_{\mathrm{D}})$. If $f(X^{*})$ is better than $f(X)$, then update X with X^{*}; otherwise keep the current point X. Hence, the current point and its opposite point are evaluated simultaneously in order to continue with the fitter one.

Particularly, in the particle swarm optimization algorithm, Equation (6) is used to calculate the opposition position of the particle.

$$X_i^{*} = X_{\max} + X_{\min} - X_i, \tag{6}$$

where $X_{\max}, X_{\min}$ represent the upper and lower bounds of particles, that is, the function definition domain.

2.4. Lévy Flight. Lévy flight [34] is a class of non-Gaussian random processes whose random walks are drawn from Lévy stable distribution. The generation of random numbers with Lévy flight consists of two steps: the choice of a random direction and the generation of steps that obey the chosen levy distribution. Random walks are drawn from Lévy stable distribution. Mantegna proposed a method to solve random numbers with normal distribution in 1994, that is, Mantegna method. The method of generating random walk obeying Lévy distribution by Mantegna method is as follows:

$$S = \frac{u}{|v|^{1/\beta}}, \tag{7}$$

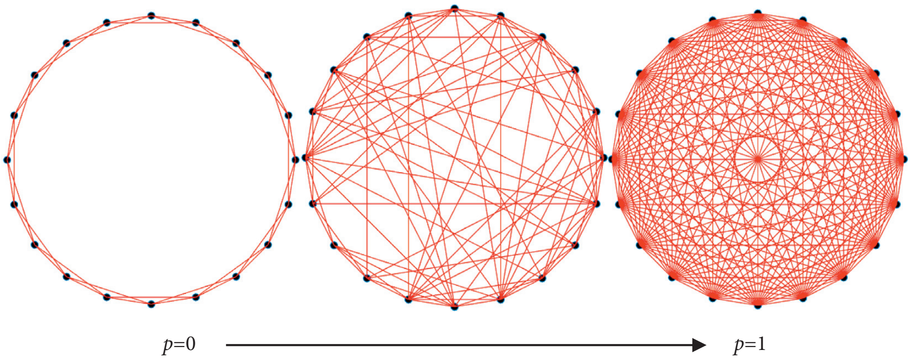

Figure 1: Evolution of NW small-world network.

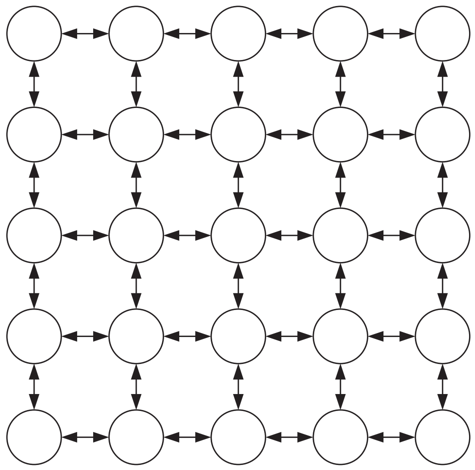

Figure 2: Swarm of particles 5 × 5.

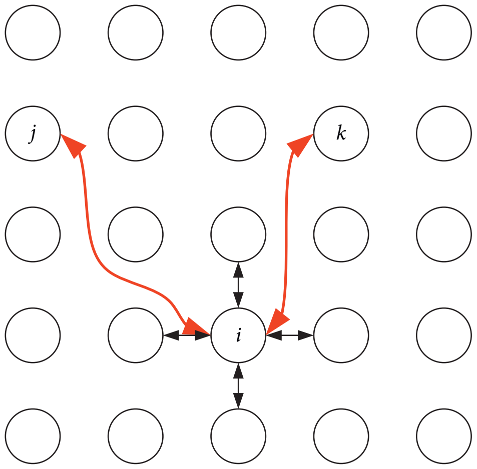

Figure 3: Small-world with 2 random particles.

where u and v obey normal distribution as follows:

$$u \sim N\left(0,\sigma_u^2\right).v \sim N\left(0,\sigma_v^2\right)$$
$$\sigma_u = \left\{\frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[1+\beta/2]\beta 2^{\beta-1/2}}\right\}^{1/\beta}. \tag{8}$$

The step size of random walk is calculated by the following formula:

$$\text{stepsize} = 0.01 \times S, \tag{9}$$

where 0.01 is the scale factor, which comes from $L/100$, and the stride L is the classical length scale. If step S is not modified, Lévy flight may jump out of the set area because the step is too large.

2.5. Diversity Analysis of Particle Swarm.

2.5. Diversity Analysis of Particle Swarm. By judging the difficulty of the problem from the fitness distance correlation, it can be concluded that when the problem presents simple characteristics, it shows that particle swarm optimization needs to enhance local exploitation ability, and when the problem presents difficult characteristics, it shows that particle swarm optimization needs to enhance global exploration ability. Therefore, it is necessary to judge the impact of two complex networks in the process of particle swarm optimization. Taking advantage of the characteristics of complex network topology can effectively enhance the robustness of particle swarm optimization algorithm.

There are many ways to evaluate population diversity, such as population diameter, population radius, average distance from population center to surrounding particles, standard average distance from population center to surrounding particles, and population consistency. In the process of particle swarm search, particle swarm will show different characteristics. In order to intuitively reflect the diversity of scale-free network and small-world network in the process of particle swarm search, this study selects the average distance from the population center to the surrounding particles to describe the diversity of particle swarm, that is, this study selects *Gbest* as the population center and calculates the average distance from other particles to *Gbest*. The calculation equation is shown as follows:

$$D = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\sum_{j=1}^{D}\left(x_i^j - \text{Gbest}\right)^2}, \tag{10}$$

where N is the particle swarm size and D is the dimension.

The above calculation method is chosen to describe the diversity of particle swarm so as to evaluate the diversity of particle swarm after every iteration, more specifically, it is more objective to take *Gbest* as the central position of particle swarm after different networks take effect on particle swarm, which can fully reflect the impact of different networks on particle swarm. If the result is small, it means that the network presents good convergence in the particle swarm optimization algorithm. Or, it means that the network has good diversity.

3. Proposed Algorithm

This section will introduce the proposed algorithm MPNLPSO in detail. Firstly, Section 3.1 introduces the neighborhood learning strategy of the small-world network. By building a small-world network to construct a neighborhood for each particle, it can make full use of the experience gained by the optimal particle within its neighborhood, thus getting rid of the situation of learning only from the global optimal particle. The dynamic opposition-based learning strategy is introduced in Section 3.2, which aims at reactivating particles in the search stagnation in the shoddy population.

3.1. Small-World Network Neighborhood Learning Strategy. As we all know, heuristic optimization algorithms based on multipopulation have always attracted researchers' interests. In order to improve the search efficiency of particle swarm optimization, the concept of population cooperation is introduced. The swarm is divided into elite population and shoddy population, and different learning strategies are adopted according to the characteristics of the two populations. The partition rule of particle swarm is as follows: in each iteration, the particle swarm is sorted according to the fitness value. Because all the problems in this study are minimum problems, the fitness values are sorted from small to large. The top 50% of the particles are defined as elite particle swarm, and the last 50% are shoddy population.

In this study, the dispersion analysis method introduced in Section 2.5 is used to calculate the dispersion of NW small-world network and VN small-world network. It is concluded that the dispersion of VN small-world network is greater than that of NW small-world network. The specific experimental results will be introduced in Section 4.

For the elite population, its ability of local search should be enhanced due to the location being more promising. Therefore, this study uses NW small-world network to build a neighborhood for each particle that belongs to the elite population. The specific steps are as follows:

(1) Build a ring-shaped regular network with particles of 10% of the population size before and after, respectively, for each particle.

(2) Set the probability p to 0.5 and randomly edge the regular network.

(3) After every particle in the population is traversed, the neighborhood topological space of small-world network is formed.

For the shoddy population, its global exploration ability should be enhanced because its location is far from the actual optimal value compared with the elite population. Therefore, this study uses VN small-world network to build a neighborhood for each particle. The specific steps are as follows:

(1) The particles in the inferior population are arranged into a 7×7 Von Neumann network according to the sequence number.

(2) After the von Neumann network is constructed, for each particle, in addition to its upper, lower, left, and right particles, two other particles are randomly selected to connect.

In order to ensure that the final network can present the characteristics of small-world network, the edges connected between particles are set as two-way edges, and the particles do not consider whether they have been previously connected when randomly selecting the target particles.

Figure 4 shows a simple example. Particle *i* is connected with *j* and *k*. The randomly selected target particle *j* has been connected with *i*, but it is still one of the two particles randomly selected by *i*. Only in this way can the final network show the characteristics of small-world network.

After constructing the small-world network, each particle learns from its neighborhood optimal particle *Nbest*, so as to get rid of the situation that particle only learns from the global optimal particle. The speed update formula of small-world network neighborhood learning strategy is shown as follows:

$$\begin{aligned} V_{ij}^{t+1} &= \omega \times V_{ij}^{t} + c_1 \times r_1\left(p\text{best}_{ij}^{t} - X_{ij}^{t}\right) \\ &\quad + c_2 \times r_2\left(N\text{best} - X_{ij}^{t}\right). \end{aligned} \tag{11}$$

Even if a certain learning strategy is adopted, particle swarm optimization may fall into local optimization due to the uncertainty in the search process. In particular, the particles of historical optimal position and neighborhood optimal position are more important, as other particles will learn from them. This study takes advantages of the characteristics of Lévy flight to randomly perturb *Pbest* and *Nbest* to reduce the risk of falling into the local optimum. The improved velocity update equation is as follows:

$$\begin{aligned} V_{ij}^{t+1} &= \omega \times V_{ij}^{t} + c_1 \times r_1\left(\text{stepsize} \times p\text{best}_{ij}^{t} - X_{ij}^{t}\right) \\ &\quad + c_2 \times r_2\left(\text{stepsize} \times N\text{best} - X_{ij}^{t}\right). \end{aligned} \tag{12}$$

3.2. Dynamic Opposition-Based Learning Strategy. Based on the characteristics of shoddy population, we know that shoddy population has a higher probability of falling into search stagnation than elite population. Therefore, a dynamic opposition-based learning strategy is proposed. The basic idea of this strategy is setting a parameter *stag* with an initial value of 0. When the current solution is not significantly improved compared with the previous one, *stag* will increase by 1. When the solution searched by the population is better than the previous solution, *stag* will be reset to 0. As the searching process continues, the value of *stag* will keep or increase. In this study, the upper bound of *stag* is set to 5. When the upper bound is reached, the opposition-based

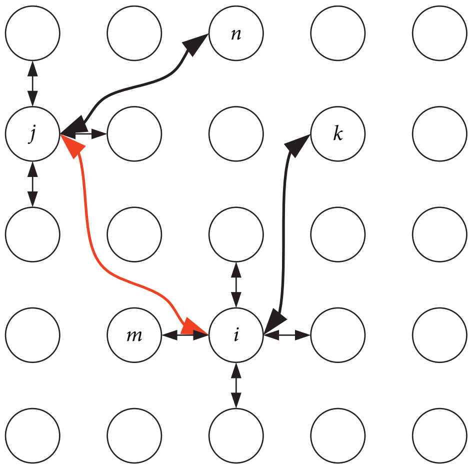

Figure 4: VN small-world network.

learning strategy will be adopted for the particle. After the solution based on opposition-based learning strategy is generated, it needs not to compare with the original solution, due to the obvious reason that the original solution has already made the particle trapped into search stagnation state, that is, the purpose of dynamic opposition-based learning is to increase the diversity of the particle and activate its search state, not just to find a better solution.

The pseudocode of dynamic opposition-based learning strategy is expressed as follows:

```
Input: shoddy population $N_2$, temporary variables $temp$ of last global optimal particle, current global optimal particle $Gbest$;
Output: $x_2$;
stag = 0;
For $i = 1 : N_2$
    If $stag > 5$
        $x_2 = x_2 + v_2$;
        $x_2 = U + L - x_2$;
    Else
        $x_2 = x_2 + v_2$;
    End if
End for
Evaluate particles and update $Gbest$;
If fitness($Gbest$)<fitness($temp$)
    fitness($temp$) = fitness($Gbest$);
    $stag = 0$;
Else
    $stag = stag + 1$;
End if
```

3.3. The Framework of the MPNLPSO Algorithm. According to the above learning strategy, a multipopulation particle swarm optimization algorithm with neighborhood learning strategy is proposed. After initializing and evaluating, the algorithm enters the iterative process. The particle swarm is sorted according to the fitness value. The top 50% of the particles is considered as the elite population, and the last 50% is the shoddy population. For elite population and shoddy population, NW small-world network and VN small-world network are constructed respectively, so that each particle can learn from the optimal particle in its neighborhood. For the shoddy population, the dynamic opposition-based learning strategy is additionally used to activate the particles trapped in the stagnant search. Finally, the elite population and shoddy population are combined to reevaluate the particles and update the historical optimal particles *Pbest* and global optimal particle *Gbest*.

Inspired by Li et al. [17], this study constructs a neighborhood for each particle, on the other hand, divides the particle swarm into elite population and shoddy population. It is considered that as the search process progresses, it is more reasonable to give each particle different inertia weights than to reduce the inertia weights linearly. As when the updated position of the particle is better than the average position of the particle swarm, the influence of the previous particle velocity should be reduced, which will enhance its local exploitation ability. And when the updated position of the particle is worse than the average position of the particle swarm, the influence of the previous particle velocity should be increased to enhance its ability of global exploration. Therefore, the inertia weight equation proposed in Li et al. [17] is used for calculation in this study.

$$\omega_i(t) = \begin{cases} \omega_{\min} + \dfrac{(\omega_{\max} - \omega_{\min})\left(f_{pbest}(t) - f_{\min}(t)\right)}{f_{avg}(t) - f_{\min}(t)}, & \text{if } f_{pbest}(t) \leq f_{avg}(t), \\ \omega_{\max}, & \text{if } f_{pbest}(t) \leq f_{avg}(t), \end{cases} \tag{13}$$

where $\omega_{\max} = 1.2$, $\omega_{\min} = 0.2$, and $f_{pbest}(t)$ is the historical optimal fitness of the ith particle in generation t. $f_{\min}(t)$ is the minimum fitness value in generation t. $f_{avg}(t)$ is the average fitness value in generation t.

The pseudocode of the MPNLPSO algorithm in this study can be expressed as follows:

```
Initialization: $N = 100$, $D = 30, 50$;
```

```
(2) Evaluate the particle swarm and get $Pbest$ and $Gbest$;
(3) While $Fes < maxFEs$ do
(4)     Calculate the inertia weight of each particle using equation (13);
(5)     Generate a set of random numbers obeying Lévy distribution;
(6)     Sort particle swarm by fitness in ascending order;
(7)     Constructing elite population $N_1$ using the top 50% of particles;
(8)     Constructing NW small-world network;
(9)     For $i = 1$: $N_1$
(10)        Select $Nbest$ for every particle;
(11)        Update velocity using equation (12);
(12)        Update position using equation (2);
(13)    End for
(14)    Constructing shoddy population $N_2$ using the top 50% of particles;
(15)    Constructing VN small-world network;
(16)    For $i = 1$: $N_2$
(17)        Select $Nbest$ for every particle;
(18)        Update velocity using equation (12);
(19)        Update position using strategy 2;
(20)    End for
(21) End while
```

4. Experiment

This study is performed on a computer with a Win_64 bit, Intel (R) core (TM) i5-6300hq CPU@ 2.30 GHz and 8 GB RAM. MATLAB r2019a was used to carry out the relevant experimental work, and the results were discussed and analyzed. Section 4.1 introduces the benchmark functions used in the experiment and the parameter settings of the comparison algorithms. The diversity of VN small-world network and NW small-world network is analyzed in detail in Section 4.2. Section 4.3 analyzes the effectiveness of the strategy. In Section 4.4, the solution accuracy of the proposed algorithm is verified in 30 and 50 dimensions respectively. In Section 4.5, the proposed algorithm is compared from the perspective of convergence.

4.1. Benchmark Functions and Parameter Settings. In order to verify the robustness of the MPNLPSO algorithm in solving complex problems, twenty widely used benchmark functions (ten unimodal functions and ten multimodal functions) are solved in this study. In addition, this study compares the MPNLPSO algorithm with seven other well-known particle swarm optimization algorithm variants, including BLPSO [35], CLPSO [36], SLPSO [37], ACPSO [37], SFAPSO [17], SWPSO-I [38], and RTPSO [39]. Where BLPSO, CLPSO, SLPSO, and ACPSO are four PSO variants on learning strategies, and SFAPSO, SWPSO-I, and RTPSO are three PSO variants on network topology. Table 1 clearly describes the names and upper and lower bounds of the benchmark functions. Table 2 shows the parameter settings of all algorithms. In order to ensure the fairness, the population size of all algorithms is set to 100, the dimensions are 30 and 50 respectively, and the maximum evaluation time is $D \times 1e4$. Other parameters are set according to the original values.

4.2. Diversity Analysis of Small-World Networks. This section uses the method described in Section 2.5 to calculate the diversity of VN small-world network and NW small-world network. Both of the two small-world networks are used to calculate twenty benchmark functions, with the population size of 100, the dimension of 30, and the iteration number of 2000. By solving each benchmark function 30 times, calculating the average distance from the center of the population to the surrounding particles, the average results of 30 runs are obtained and the data are listed in Table 3.

As we can see from Table 3, in terms of unimodal test functions, NW small-world network has greater diversity than VN small-world network in $f4$ and $f5$ functions, and shows better convergence ability in other eight test functions. As for multimodal functions, NW small-world network shows good diversity in $f11$, $f12$, $f18$, and $f19$ functions, and VN small-world network shows good diversity in the other six multimodal functions. On the whole, the results of fourteen functions of VN small-world network are better than that of NW small-world network, that is, VN small-world network shows better diversity in particle swarm optimization algorithm, and the convergence ability of NW small-world network is slightly better than that of VN small-world network.

4.3. Effectiveness of the Strategy. This part experiments on the effectiveness of the strategy, which is divided into five combinations. Comb. 1 is the standard particle swarm optimization algorithm, Comb. 2 is a multipopulation strategy, Comb. 3 uses the NW small-world network to build the neighborhood topology, Comb. 4 uses the VN small-world network to build the neighborhood topology, and Comb. 5 is the algorithm proposed in this study. All combinations were tested on 30 dimensions. Each combination ran 30 times independently to get the mean and standard deviation. It can be clearly seen from Table 4 that the experimental effect has not been improved when Comb. 2 does not use the network topology to build the neighborhood. Comb. 3 and Comb. 4 can significantly improve the result of $f5$ function without using the multipopulation strategy, and can reach the actual optimal value, but not all functions have improved. The convergence accuracy of Comb. 4 on $f13$ function is better than that of other combinations. Comb. 5, that is, the algorithm proposed in this study shows excellent optimization ability in most functions, which shows that there can be an effective cooperation between learning strategies and significant improvement in the convergence accuracy of the algorithm.

Table 1: Benchmark functions.

	Test function	Equation	Search range
$f1$	Brown function	$\sum_{i=1}^{n-1} (x_i{}^2)^{(x_{i+1}^2+1)} + (x_{i+1}{}^2)^{(x_i^2+1)}$	[−1, 4]
$f2$	Exponential function	$-\exp(-0.5 \sum_{i=1}^{n} x_i^2)$	[−1, 2]
$f3$	Griewank function	$1 + \sum_{i=1}^{n} x_i^2/4000 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i})$	[−600, 600]
$f4$	Powellsum function	$\sum_{i=1}^{n} \lvert x_i \rvert^{i+1}$	[−1, 1]
$f5$	Ridge function	$x_i + d * (\sum_{i=1}^{n} x_i^2)^{\alpha}, d = 1, \alpha = 0.5$	[−5, 5]
$f6$	Schwefel2.20 function	$\sum_{i=1}^{n} \lvert x_i \rvert$	[−100, 100]
$f7$	Schwefel2.21 function	$\max_{i=1,\ldots n} \lvert x_i \rvert$	[−100, 100]
$f8$	Schwefel2.22 function	$\sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[−100, 100]
$f9$	Schwefel2.23 function	$\sum_{i=1}^{n} x_i^{10}$	[−10, 10]
$f10$	Sphere	$\sum_{i=1}^{n} x_i^2$	[−5.12, 5.12]
$f11$	Ackley 1	$20 + \exp(1) - 20 * \exp(0.2\sqrt{\sum_{i=1}^{n} x_i^2/n}) - \exp[1/n \sum_{i=1}^{n} \cos(2\pi x_i)]$	[−35, 35]
$f12$	Alphine 1	$\sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1x_i \rvert$	[0, 10]
$f13$	Happy cat	$(\sum_{i=1}^{n} x_i^2 - n)^{\alpha} + \frac{1}{n}(0.2 \sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} x_i) + 0.5, \alpha = 0.5$	[−2, 2]
$f14$	Periodic	$1 + \sum_{i=1}^{n} \sin^2(x_i) - 0.1\exp(-\sum_{i=1}^{n} x_i^2)$	[−10, 10]
$f15$	Quatic function with noise	$\sum_{i=1}^{n} ix_1^4 + \text{rand}(0, 1)$	[−1.28, 1.28]
$f16$	Rastrign	$10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)]$	[−5.12, 5.12]
$f17$	Salomon	$1 - \cos(2\pi * \sqrt{\sum_{i=1}^{n} x_i^2}) + 0.1 * \sqrt{\sum_{i=1}^{n} x_i^2}$	[−100, 100]
$f18$	Xin-She Yang 1	$\sum_{i=1}^{n} a\lvert x_i \rvert^i, a = \text{rand}(0, 1)$	[−5, 5]
$f19$	Xin-She Yang 2	$(\sum_{i=1}^{n} \lvert x_i \rvert^i) * \exp[-\sum_{i=1}^{n} \sin(x_i^2)]$	[−2 π, 2 π]
$f20$	Xin-She Yang 4	$[\sum_{i=1}^{n} \sin^2(x_i) - \exp(-\sum_{i=1}^{n} x_i^2)] * \exp(-\sum_{i=1}^{n} \sin^2 \sqrt{\lvert x_i \rvert})$	[−10, 10]

Table 2: Parameter settings.

Algorithm	Parameter settings
CLPSO	$\omega = 0.9 \sim 0.2$, $c = 1.49445$, $gapm = 5$, $V_{\max} = 0.2 \times range$
BLPSO	$\omega = 0.9 \sim 0.2$, $c = 1.49445$, $I = E = 1$, $V_{\max} = 0.2 \times range$
SLPSO	$M = 100$, $m = M + \text{floor(d/10)}$, $c_3 = d/M \times 0.01$
ACPSO	$\omega = 0.9\text{~}0.4$, $c_1 = c_2 = 1.49445$, $swarmNum = 3$, $alpha = 0.1$, $beta = 0.1$, $\max = 0.2 \times range$
SFAPSO	$\omega_{\max} = 1.2$, $\omega_{\min} = 0.2$ $c1 = c2 = 2.0$, $\alpha = 0.5$, $g = 0.6$, $V_{\max} = 0.2 \times range$
SWPSO-I	$\omega = 0.9\text{~}0.4$, $c_1 = c_2 = 1.494$, $V_{\max} = 0.2 \times range$
RTPSO	$\chi = 0.7298$, $\varphi_1 + \varphi_2 = 4.1$, $V_{\max} = 0.2 \times range$
MPNLPSO	$\omega_{\max} = 1.2$, $\omega_{\min} = 0.2$, $c_1 = c_2 = 1.494$, $\beta = 1.5$, $V_{\max} = 0.2 \times range$

Table 3: Population diversity analysis.

	VN small-world network	NW small-world network
$f1$	**1.18 E + 00**	5.80 E − 01
$f2$	**2.97 E + 02**	4.05 E + 00
$f3$	**3.56 E + 01**	2.18 E + 01
$f4$	2.29E − 03	**2.95E − 02**
$f5$	**4.05 E + 02**	1.22 E + 02
$f6$	**6.37 E + 01**	5.70 E + 01
$f7$	6.81 E + 01	**7.51 E + 01**
$f8$	**1.11 E + 02**	6.00 E + 01
$f9$	**1.61 E + 01**	1.56 E + 01
$f10$	0.00 E + 00	0.00 E + 00
$f11$	4.15 E + 01	**5.17 E + 01**
$f12$	1.10 E + 01	**2.26 E + 01**
$f13$	**2.91 E + 02**	1.95 E + 02
$f14$	**1.21 E + 02**	1.96 E + 01
$f15$	**3.68 E + 01**	3.19 E + 01
$f16$	**1.25 E + 02**	9.41 E + 01
$f17$	**2.51 E + 02**	1.22 E + 02
$f18$	3.69 E + 01	**8.53 E + 01**
$f19$	2.01 E + 01	**2.23 E + 01**
$f20$	**2.14 E + 01**	1.72 E + 01

4.4. Comparison of PSO Algorithm Variants. This section compares MPNLPSO with six well-known PSO variants in 30 and 50 dimensions, respectively. The population size is 100, and the maximum number of evaluations is 10000 × D. In order to show the calculation error of each algorithm, each function is calculated 30 times, and then the minimum, average, and standard deviation are recorded. Tables 5 and 6 list the comparison results of the six well-known PSO variants on the benchmark test functions. It can be seen that each function contains three rows of values that are the minimum value, the average value, and the standard deviation.

It can be seen from the results of unimodal function in Table 5, that SFAPSO can converge to the actual optimal value on the $f1$ function. In the 30 results of the solution, MPNLPSO can converge to the actual optimal value, but it is not stable enough. On the $f2$ function, all algorithms fall into the local

Table 4: Effectiveness of the strategy.

		Comb. 1	Comb. 2	Comb. 3	Comb. 4	Comb. 5
*f*1	Mean	1.22E+01	4.02E+01	3.07E+00	5.40E-01	**2.29E-30**
	Std.	7.26E+00	3.06E+01	2.90E+00	5.11E-01	**1.16E-30**
*f*2	Mean	**1.00E+00**	1.00E+00	**1.00E+00**	**1.00E+00**	**1.00E+00**
	Std.	**0.00E+00**	6.78E-16	**0.00E+00**	**0.00E+00**	**0.00E+00**
*f*3	Mean	1.13E+00	2.89E+00	9.77E-01	6.97E-01	**0.00E+00**
	Std.	9.41E-02	2.18E+00	1.10E-01	1.53E-01	**0.00E+00**
*f*4	Mean	7.03E-07	5.28E-05	1.38E-08	8.83E-13	**3.08E-33**
	Std.	1.24E-06	1.28E-04	1.62E-08	3.07E-12	**0.00E+00**
*f*5	Mean	7.61E+01	2.91E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std.	3.36E+01	2.39E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
*f*6	Mean	1.21E+02	2.99E+02	7.36E+01	2.55E+01	**0.00E+00**
	Std.	3.94E+01	1.57E+02	1.51E+01	8.63E+00	**0.00E+00**
*f*7	Mean	1.55E+01	3.45E+01	1.06E+01	3.39E+00	**0.00E+00**
	Std.	3.11E+00	3.09E+00	2.60E+00	9.15E-01	**0.00E+00**
*f*8	Mean	3.69E+02	2.52E+32	2.46E+02	3.87E+01	**0.00E+00**
	Std.	9.74E+01	9.81E+32	6.61E+01	1.17E+01	**0.00E+00**
*f*9	Mean	1.16E+02	5.07E+05	1.29E+00	2.68E-04	**0.00E+00**
	Std.	2.43E+02	4.76E+05	4.14E+00	1.04E-03	**0.00E+00**
*f*10	Mean	1.86E+01	1.81E+02	3.64E-01	2.52E-01	**0.00E+00**
	Std.	1.85E+01	1.22E+02	4.47E-01	9.90E-01	**0.00E+00**
*f*11	Mean	8.42E+00	1.76E+01	5.97E+00	3.29E+00	**8.88E-16**
	Std.	1.24E+00	5.89E+00	1.23E+00	6.54E-01	**0.00E+00**
*f*12	Mean	6.31E+00	7.74E+00	3.18E+00	3.94E-01	**0.00E+00**
	Std.	2.45E+00	1.03E+01	1.49E+00	2.06E-01	**0.00E+00**
*f*13	Mean	4.85E-01	4.12E-01	2.94E-01	**3.05E-02**	9.42E-01
	Std.	1.13E-01	1.68E-01	9.14E-02	**1.11E-02**	1.22E-01
*f*14	Mean	8.12E-01	7.51E+00	4.98E-01	4.51E-01	**0.00E+00**
	Std.	2.80E-01	1.77E+00	2.89E-01	1.78E-01	**0.00E+00**
*f*15	Mean	1.89E-01	1.37E+01	5.56E-02	3.74E-03	**3.53E-04**
	Std.	9.82E-02	1.69E+01	4.67E-02	2.36E-03	**3.96E-04**
*f*16	Mean	6.83E+01	2.79E+02	5.17E+01	2.83E+01	**0.00E+00**
	Std.	1.53E+01	2.22E+01	1.22E+01	8.10E+00	**0.00E+00**
*f*17	Mean	3.38E+04	2.40E+07	6.67E+03	6.77E+02	**2.90E+01**
	Std.	7.75E-01	4.98E+00	5.64E-01	3.04E-01	**0.00E+00**
*f*18	Mean	3.10E+00	5.31E+08	1.01E-01	3.63E-06	**0.00E+00**
	Std.	7.61E+00	2.05E+09	1.70E-01	6.42E-06	**0.00E+00**
*f*19	Mean	2.48E-10	6.23E-09	9.46E-11	1.28E-11	**0.00E+00**
	Std.	4.10E-10	1.35E-08	1.22E-10	1.02E-11	**0.00E+00**
*f*20	Mean	1.00E+00	1.00E+00	1.00E+00	1.00E+00	**0.00E+00**
	Std.	4.46E-13	3.06E-09	1.75E-13	1.45E-13	**0.00E+00**

optimum. On the *f*3 function, BLPSO, SLPSO, and MPNLPSO can all converge to the actual optimal value, but compared to the other two algorithms, MPNLPSO is more stable. On the *f*4 function, BLPSO, SLPSO, and SFAPSO all present excellent optimization capability, especially, SFAPSO can converge to the actual optimal value and is relatively stable, and BLPSO and SLPSO show high accuracy, almost reaching the actual optimal value. On the contrary, as for *f*4 function, the performance of MPNLPSO is worse than BLPSO, SLPSO, SWPSO-I, and SFAPSO. As for the functions *f*5–*f*10, MPNLPSO has shown excellent optimization ability, converging to the actual optimum on both minimum and average value with good robustness. Among the *f*5–*f*10 functions, SLPSO can only converge to the actual optimal value on two functions. SWPSO-I and SFAPSO can converge to the actual optimal value on the *f*5 and *f*10 functions, but the stability of SWPSO-I on the *f*5 function is not as good as that of SFAPSO and MPNLPSO. Similar to SWPSO-I, ACPSO and RTPSO can converge to the actual optimal value on the *f*5 function, but the stability is not good enough.

It can be seen from the results of multimodal function in Table 5 that on the *f*11 function, the optimization effect of all algorithms is not good, and MPNLPSO is relatively better than all other algorithms. On the *f*13 function, SFAPSO can converge to the actual optimal value in all the 30 runs, but it can be seen from the average values that the stability is ordinary. The average values of ACPSO on the *f*13 function are slightly better than the other six algorithms, indicating that, to some extent, it has certain advantages in this function. It can be seen from the *f*14 function that, except for MPNLPSO, which exhibits excellent optimization capabilities, other algorithms all fall into local optima. Regarding *f*15 function, MPNLPSO is better than the other six algorithms in both the minimum and average values, although it fails to find the actual optimal value, it has a certain degree of competitiveness compared to other algorithms. Concluded from the results of functions *f*16–*f*20, MPNLPSO can converge to the actual optimal value, and it shows quite competitive stability. Especially on the *f*20 function, when

Table 5: Results of comparison algorithms on benchmark functions (30 *D*).

		BLPSO	CLPSO	SLPSO	ACPSO	SWPSO-I	SFAPSO	RTPSO	SWNWPSO
*f*1	Min	1.53E − 152	5.44E + 01	3.13E − 145	3.37E − 06	4.19E − 05	**0.00E + 00**	7.70E − 01	**0.00E + 00**
	Mean	1.86E − 132	1.18E + 02	1.21E − 141	8.43E − 06	1.68E − 01	**0.00E + 00**	3.81E + 01	2.29E − 30
	Std.	1.02E − 131	4.70E + 01	6.42E − 141	4.46E − 06	5.30E − 01	**0.00E + 00**	5.15E + 01	1.16E − 30
*f*2	Min	1.00E + 00	1.00E + 00	1.00E + 00	**1.00E + 00**	1.00E + 00	1.00E + 00	1.00E + 00	**1.00E + 00**
	Mean	1.00E + 00	1.00E + 00	1.00E + 00	**1.00E + 00**	1.00E + 00	1.00E + 00	1.00E + 00	**1.00E + 00**
	Std.	7.93E − 11	1.64E − 05	6.78E − 16	**0.00E + 00**	6.78E − 16	6.78E − 16	2.03E − 06	**0.00E + 00**
*f*3	Min	0.00E + 00	3.05E + 00	0.00E + 00	3.59E − 06	5.77E − 06	8.51E − 07	6.86E − 02	**0.00E + 00**
	Mean	2.71E − 03	4.65E + 00	4.93E − 04	4.42E − 02	5.28E − 03	9.37E − 03	5.11E − 01	**0.00E + 00**
	Std.	4.81E − 03	5.79E − 01	1.88E − 03	3.46E − 02	6.67E − 03	1.23E − 02	3.85E − 01	**0.00E + 00**
*f*4	Min	5.32E − 251	6.06E − 04	0.00E + 00	7.52E − 33	7.96E − 39	**0.00E + 00**	6.66E − 16	3.08E − 33
	Mean	4.22E − 222	3.05E − 03	1.22E − 314	3.49E − 26	2.28E − 14	**0.00E + 00**	2.06E − 06	3.08E − 33
	Std.	0.00E + 00	2.06E − 03	0.00E + 00	1.17E − 25	1.00E − 13	**0.00E + 00**	7.94E − 06	0.00E + 00
*f*5	Min	1.78E − 07	6.41E − 01	9.37E − 09	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
	Mean	1.41E − 06	1.88E + 00	7.82E − 07	1.91E − 14	6.26E − 07	**0.00E + 00**	6.29E − 16	**0.00E + 00**
	Std.	1.20E − 06	6.50E − 01	6.14E − 07	3.04E − 14	1.63E − 06	**0.00E + 00**	3.45E − 15	**0.00E + 00**
*f*6	Min	1.03E − 93	4.21E + 02	5.32E − 73	6.37E − 02	3.36E − 02	7.05E − 02	1.09E + 00	**0.00E + 00**
	Mean	1.03E − 82	5.07E + 02	9.42E − 72	1.16E − 01	2.12E − 01	1.79E − 01	3.47E + 01	**0.00E + 00**
	Std.	4.12E − 82	4.03E + 01	9.72E − 72	2.93E − 02	1.32E − 01	6.69E − 02	4.28E + 01	**0.00E + 00**
*f*7	Min	5.18E − 03	3.89E + 01	3.53E − 39	1.61E − 01	1.63E − 02	3.55E − 02	2.71E − 01	**0.00E + 00**
	Mean	5.24E − 02	4.70E + 01	2.33E − 38	2.57E − 01	4.04E − 02	1.18E − 01	3.22E + 00	**0.00E + 00**
	Std.	3.93E − 02	3.39E + 00	2.60E − 38	5.93E − 02	1.67E − 02	6.25E − 02	3.73E + 00	**0.00E + 00**
*f*8	Min	1.28E − 88	1.47E + 21	1.30E − 71	6.84E − 02	3.86E − 02	3.44E − 02	1.38E + 00	**0.00E + 00**
	Mean	1.13E − 78	4.14E + 26	1.38E − 70	1.02E − 01	5.67E − 01	9.53E + 00	1.83E + 32	**0.00E + 00**
	Std.	3.20E − 78	1.47E + 27	1.57E − 70	2.75E − 02	1.55E + 00	5.06E + 01	8.19E + 32	**0.00E + 00**
*f*9	Min	6.32E − 232	2.51E + 06	**0.00E + 00**	1.53E − 23	1.37E − 21	5.19E − 21	1.71E − 07	**0.00E + 00**
	Mean	2.14E − 203	1.17E + 07	**0.00E + 00**	9.90E − 20	7.60E − 14	5.14E − 15	1.48E + 01	**0.00E + 00**
	Std.	0.00E + 00	7.36E + 06	**0.00E + 00**	2.38E − 19	4.14E − 13	1.93E − 14	5.25E + 01	**0.00E + 00**
*f*10	Min	1.06E − 57	3.03E + 02	**0.00E + 00**	2.34E − 06	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**	**0.00E + 00**
	Mean	7.33E − 34	3.63E + 02	**0.00E + 00**	1.75E − 05	**0.00E + 00**	**0.00E + 00**	9.24E + 01	**0.00E + 00**
	Std.	4.01E − 33	3.34E + 01	**0.00E + 00**	7.47E − 06	**0.00E + 00**	**0.00E + 00**	1.38E + 02	**0.00E + 00**
*f*11	Min	2.66E − 15	1.61E + 01	2.66E − 15	1.05E − 02	8.33E − 03	6.35E − 04	2.05E − 01	**8.88E − 16**
	Mean	6.69E − 15	1.74E + 01	5.74E − 15	1.70E − 02	8.09E − 02	1.11E + 00	4.76E + 00	**8.88E − 16**
	Std.	6.10E − 15	5.44E − 01	1.23E − 15	4.25E − 03	3.21E − 01	1.03E + 00	3.11E + 00	**0.00E + 00**
*f*12	Min	2.60E − 245	2.49E + 01	1.37E − 137	4.96E − 03	3.60E − 03	5.73E − 03	3.29E − 01	**0.00E + 00**
	Mean	8.12E − 14	2.97E + 01	7.40E − 18	7.27E − 03	8.99E − 03	8.46E − 02	1.03E + 01	**0.00E + 00**
	Std.	4.23E − 13	2.58E + 00	4.05E − 17	1.74E − 03	3.64E − 03	7.62E − 02	1.06E + 01	**0.00E + 00**
*f*13	Min	1.12E − 03	3.81E − 01	2.43E − 02	7.88E − 04	4.84E − 03	**0.00E + 00**	4.01E − 01	7.11E − 01
	Mean	3.76E − 03	5.60E − 01	4.05E − 02	**1.56E − 03**	2.17E − 02	2.12E − 02	6.18E − 01	9.42E − 01
	Std.	1.26E − 03	1.04E − 01	9.24E − 03	**3.94E − 04**	1.07E − 02	1.68E − 02	8.27E − 02	1.22E − 01
*f*14	Min	1.00E − 01	3.58E + 00	1.00E − 01	1.19E − 01	1.00E − 01	1.00E − 01	3.39E − 01	**0.00E + 00**
	Mean	1.02E − 01	5.43E + 00	1.00E − 01	1.39E − 01	1.01E − 01	1.00E − 01	6.18E + 00	**0.00E + 00**
	Std.	5.42E − 03	7.15E − 01	1.13E − 16	1.16E − 02	1.31E − 03	7.51E − 04	2.41E + 00	**0.00E + 00**
*f*15	Min	9.82E − 04	1.59E + 00	9.31E − 03	7.30E − 02	2.01E − 03	8.68E − 05	1.87E − 02	**6.22E − 06**
	Mean	1.99E − 03	6.66E + 00	1.36E − 02	4.83E − 01	6.16E − 03	1.86E − 03	3.07E − 01	**3.53E − 04**
	Std.	9.53E − 04	2.10E + 00	2.02E − 03	3.99E − 01	2.55E − 03	1.10E − 03	4.49E − 01	**3.96E − 04**
*f*16	Min	2.98E + 00	2.05E + 02	6.96E + 00	1.19E − 01	9.98E + 00	1.50E + 01	1.88E + 01	**0.00E + 00**
	Mean	8.89E + 00	2.36E + 02	1.56E + 01	5.21E − 01	1.60E + 01	2.49E + 01	1.63E + 02	**0.00E + 00**
	Std.	3.32E + 00	2.02E + 01	4.82E + 00	3.42E − 01	3.59E + 00	5.79E + 00	8.07E + 01	**0.00E + 00**
*f*17	Min	2.00E − 01	1.12E + 01	3.00E − 01	4.00E − 01	2.00E − 01	3.00E − 01	2.00E + 00	**0.00E + 00**
	Mean	2.03E − 01	1.29E + 01	3.07E − 01	5.00E − 01	2.03E − 01	4.40E − 01	4.65E + 00	**0.00E + 00**
	Std.	1.83E − 02	8.13E − 01	2.54E − 02	8.30E − 02	1.83E − 02	8.94E − 02	2.18E + 00	**0.00E + 00**
*f*18	Min	3.74E − 30	1.37E + 02	3.28E − 180	5.08E − 10	3.72E − 12	1.34E − 06	1.38E − 05	**0.00E + 00**
	Mean	1.95E − 13	4.55E + 04	1.62E − 169	1.88E − 06	2.25E − 08	5.04E − 03	2.33E + 00	**0.00E + 00**
	Std.	1.01E − 12	5.71E + 04	0.00E + 00	5.41E − 06	4.51E − 08	1.88E − 02	9.04E + 00	**0.00E + 00**

Table 5: Continued.

		BLPSO	CLPSO	SLPSO	ACPSO	SWPSO-I	SFAPSO	RTPSO	SWNWPSO
*f*19	Min	3.51 E − 12	3.93 E − 08	5.71 E − 12	3.52 E − 12	3.51 E − 12	3.66 E − 12	4.81 E − 11	**0.00 E + 00**
	Mean	3.91 E − 12	4.02 E − 07	8.15 E − 12	3.61 E − 12	3.56 E − 12	5.49 E − 12	8.70 E − 07	**0.00 E + 00**
	Std.	3.71 E − 13	4.42 E − 07	1.36 E − 12	8.27 E − 14	7.83 E − 14	3.25 E − 12	3.54 E − 06	**0.00 E + 00**
*f*20	Min	1.00 E + 00	1.00 E + 00	1.00 E + 00	1.00 E + 00	1.00 E + 00	1.00 E + 00	1.00 E + 00	**0.00 E + 00**
	Mean	1.00 E + 00	1.00 E + 00	1.00 E + 00	1.00 E + 00	1.00 E + 00	1.00 E + 00	1.00 E + 00	**0.00 E + 00**
	Std.	1.84 E − 16	4.40 E − 11	1.75 E − 16	2.06 E − 16	6.52 E − 16	6.00 E − 16	1.55 E − 11	**0.00 E + 00**

Table 6: Results of comparison algorithms on benchmark functions (50 dimensions).

		BLPSO	CLPSO	SLPSO	ACPSO	SWPSO-I	SFAPSO	RTPSO	SWNWPSO
*f*1	Min	4.96 E − 76	3.59 E + 02	5.00 E − 168	2.94 E − 08	7.66 E − 04	**0.00 E + 00**	9.60 E + 00	**0.00 E + 00**
	Mean	3.20 E − 75	1.51 E + 03	8.04 E − 166	2.03 E − 07	4.89 E − 01	**0.00 E + 00**	3.38 E + 02	2.90 E − 30
	Std.	3.18 E − 75	1.42 E + 03	0.00 E + 00	1.30 E − 07	1.25 E + 00	**0.00 E + 00**	6.39 E + 02	2.41 E − 30
*f*2	Min	**1.00 E + 00**	1.00 E + 00	1.00 E + 00	**1.00 E + 00**	1.00 E + 00	1.00 E + 00	1.00 E + 00	**1.00 E + 00**
	Mean	**1.00 E + 00**	1.00 E + 00	1.00 E + 00	**1.00 E + 00**	1.00 E + 00	1.00 E + 00	1.00 E + 00	**1.00 E + 00**
	Std.	**0.00 E + 00**	7.42 E − 08	4.34 E − 16	**0.00 E + 00**	2.26 E − 16	2.26 E − 16	8.21 E − 10	**0.00 E + 00**
*f*3	Min	**0.00 E + 00**	7.77 E + 00	**0.00 E + 00**	3.98 E − 08	1.21 E − 04	9.07 E − 06	9.38 E − 02	**0.00 E + 00**
	Mean	**0.00 E + 00**	9.07 E + 00	2.47 E − 04	9.93 E − 03	5.85 E − 03	7.46 E − 03	9.80 E − 01	**0.00 E + 00**
	Std.	**0.00 E + 00**	8.63 E − 01	1.35 E − 03	1.35 E − 02	7.66 E − 03	8.06 E − 03	5.48 E − 01	**0.00 E + 00**
*f*4	Min	1.13 E − 138	4.67 E − 04	**0.00 E + 00**	4.99 E − 45	**0.00 E + 00**	**0.00 E + 00**	8.35 E − 17	**3.08 E − 33**
	Mean	8.88 E − 131	2.22 E − 03	**0.00 E + 00**	2.53 E − 39	8.14 E − 14	**0.00 E + 00**	2.25 E − 08	**3.08 E − 33**
	Std.	4.83 E − 130	1.50 E − 03	**0.00 E + 00**	6.87 E − 39	3.43 E − 13	**0.00 E + 00**	8.96 E − 08	**0.00 E + 00**
*f*5	Min	3.99 E − 09	4.20 E + 00	1.27 E − 08	**0.00 E + 00**	**0.00 E + 00**	**0.00 E + 00**	**0.00 E + 00**	**0.00 E + 00**
	Mean	1.01 E − 06	5.05 E + 00	1.16 E − 06	**0.00 E + 00**	1.39 E − 06	**0.00 E + 00**	3.36 E − 02	**0.00 E + 00**
	Std.	1.24 E − 06	4.40 E − 01	1.08 E − 06	**0.00 E + 00**	3.26 E − 06	**0.00 E + 00**	1.08 E − 01	**0.00 E + 00**
*f*6	Min	5.85 E − 27	8.53 E + 02	1.62 E − 85	9.10 E − 04	3.51 E − 01	1.71 E − 01	3.17 E + 00	**0.00 E + 00**
	Mean	1.14 E − 26	9.97 E + 02	5.22 E − 84	3.76 E − 03	8.22 E − 01	9.52 E − 01	6.39 E + 01	**0.00 E + 00**
	Std.	3.98 E − 27	6.29 E + 01	5.06 E − 84	1.15 E − 03	7.90 E − 01	1.59 E + 00	6.43 E + 01	**0.00 E + 00**
*f*7	Min	6.86 E − 03	4.76 E + 01	1.04 E − 30	2.16 E − 01	9.00 E − 02	5.44 E − 01	6.88 E − 01	**0.00 E + 00**
	Mean	1.90 E − 02	5.41 E + 01	2.45 E − 06	4.15 E − 01	1.77 E − 01	2.12 E + 00	3.71 E + 00	**0.00 E + 00**
	Std.	1.08 E − 02	2.91 E + 00	1.34 E − 05	8.41 E − 02	5.35 E − 02	1.00 E + 00	1.98 E + 00	**0.00 E + 00**
*f*8	Min	6.23 E − 30	1.35 E + 45	7.43 E − 84	1.50 E − 03	3.07 E − 01	3.18 E − 01	2.43 E + 02	**0.00 E + 00**
	Mean	5.52 E − 29	2.63 E + 51	1.54 E − 82	3.45 E − 03	1.59 E + 00	9.28 E + 01	4.10 E + 63	**0.00 E + 00**
	Std.	4.43 E − 29	5.77 E + 51	1.59 E − 82	1.71 E − 03	2.64 E + 00	1.54 E + 02	2.25 E + 64	**0.00 E + 00**
*f*9	Min	2.04 E − 96	2.49 E + 07	**0.00 E + 00**	2.97 E − 25	1.03 E − 14	3.97 E − 16	3.54 E − 04	**0.00 E + 00**
	Mean	1.53 E − 90	9.24 E + 07	**0.00 E + 00**	2.15 E − 19	2.22 E − 10	1.68 E − 12	3.69 E + 01	**0.00 E + 00**
	Std.	3.79 E − 90	4.01 E + 07	**0.00 E + 00**	5.76 E − 19	5.66 E − 10	3.82 E − 12	1.24 E + 02	**0.00 E + 00**
*f*10	Min	2.41 E − 81	6.50 E + 02	**0.00 E + 00**	5.60 E − 08	**0.00 E + 00**	**0.00 E + 00**	**0.00 E + 00**	**0.00 E + 00**
	Mean	2.51 E − 10	7.66 E + 02	**0.00 E + 00**	3.57 E − 07	**0.00 E + 00**	**0.00 E + 00**	3.72 E + 02	**0.00 E + 00**
	Std.	3.89 E − 10	5.71 E + 01	**0.00 E + 00**	2.52 E − 07	**0.00 E + 00**	**0.00 E + 00**	3.51 E + 02	**0.00 E + 00**
*f*11	Min	2.66 E − 15	1.76 E + 01	2.66 E − 15	6.90 E − 04	3.64 E − 02	3.11 E − 02	2.11 E + 00	**8.88 E − 16**
	Mean	5.98 E − 15	1.83 E + 01	6.69 E − 15	1.72 E − 03	1.26 E + 00	2.19 E + 00	4.75 E + 00	**8.88 E − 16**
	Std.	9.01 E − 16	3.12 E − 01	2.03 E − 15	5.13 E − 04	8.96 E − 01	6.62 E − 01	1.65 E + 00	**0.00 E + 00**
*f*12	Min	3.50 E − 03	4.92 E + 01	2.11 E − 133	8.19 E − 04	1.51 E − 02	7.09 E − 02	1.51 E + 00	**0.00 E + 00**
	Mean	4.96 E − 03	5.84 E + 01	1.28 E − 16	4.05 E − 03	3.11 E − 02	2.46 E − 01	1.83 E + 01	**0.00 E + 00**
	Std.	5.94 E − 04	4.84 E + 00	2.01 E − 16	2.05 E − 03	1.46 E − 02	1.24 E − 01	1.60 E + 01	**0.00 E + 00**
*f*13	Min	9.46 E − 03	5.22 E − 01	9.17 E − 02	**3.91 E − 03**	4.14 E − 02	2.13 E − 02	5.46 E − 01	8.00 E − 01
	Mean	1.19 E − 02	6.44 E − 01	1.22 E − 01	**6.01 E − 03**	6.27 E − 02	7.28 E − 02	7.14 E − 01	9.97 E − 01
	Std.	1.47 E − 03	6.95 E − 02	2.09 E − 02	**1.22 E − 03**	1.14 E − 02	2.45 E − 02	7.31 E − 02	1.19 E − 01
*f*14	Min	3.76 E − 01	9.42 E + 00	1.00 E − 01	1.02 E − 01	1.01 E − 01	1.00 E − 01	9.64 E + 00	**0.00 E + 00**
	Mean	5.12 E − 01	1.18 E + 01	1.00 E − 01	1.77 E − 01	1.40 E − 01	1.03 E − 01	1.50 E + 01	**0.00 E + 00**
	Std.	6.70 E − 02	8.65 E − 01	7.06 E − 17	7.01 E − 02	1.22 E − 01	6.07 E − 03	2.19 E + 00	**0.00 E + 00**
*f*15	Min	4.64 E − 03	2.30 E + 01	2.08 E − 02	2.12 E − 01	7.04 E − 03	5.77 E − 05	1.54 E − 01	**1.26 E − 06**
	Mean	7.28 E − 03	3.31 E + 01	3.32 E − 02	6.27 E − 01	2.17 E − 02	9.77 E − 04	5.84 E − 01	**1.67 E − 04**
	Std.	1.50 E − 03	5.93 E + 00	4.35 E − 03	2.90 E − 01	1.11 E − 02	7.33 E − 04	5.14 E − 01	**1.92 E − 04**
*f*16	Min	1.57 E + 01	4.08 E + 02	1.79 E + 01	3.29 E − 01	1.31 E + 01	2.49 E + 01	9.27 E + 01	**0.00 E + 00**
	Mean	2.42 E + 01	4.53 E + 02	2.96 E + 01	4.26 E + 00	3.07 E + 01	4.93 E + 01	2.73 E + 02	**0.00 E + 00**
	Std.	3.06 E + 00	1.79 E + 01	6.11 E + 00	2.21 E + 00	7.33 E + 00	1.27 E + 01	8.12 E + 01	**0.00 E + 00**

Table 6: Continued.

		BLPSO	CLPSO	SLPSO	ACPSO	SWPSO-I	SFAPSO	RTPSO	SWNWPSO
*f*17	Min	2.00E − 01	1.59E + 01	5.01E − 01	6.00E − 01	2.00E − 01	5.00E − 01	2.30E + 00	**0.00E + 00**
	Mean	2.00E − 01	1.88E + 01	6.00E − 01	8.03E − 01	2.83E − 01	9.73E − 01	6.71E + 00	**0.00E + 00**
	Std.	3.07E − 08	1.18E + 00	5.22E − 02	1.10E − 01	5.92E − 02	2.35E − 01	2.45E + 00	**0.00E + 00**
*f*18	Min	4.40E − 52	7.52E + 08	2.94E − 219	3.16E − 09	4.10E − 11	1.50E − 05	4.62E − 04	**0.00E + 00**
	Mean	1.36E − 47	3.93E + 11	1.36E − 207	5.77E − 05	6.64E − 08	2.74E − 01	3.18E − 01	**0.00E + 00**
	Std.	3.55E − 47	5.41E + 11	**0.00E + 00**	3.03E − 04	1.54E − 07	9.65E − 01	1.47E + 00	**0.00E + 00**
*f*19	Min	1.65E − 20	4.80E − 12	2.72E − 20	1.23E − 20	1.21E − 20	1.24E − 20	8.42E − 15	**0.00E + 00**
	Mean	1.83E − 20	7.25E − 11	3.34E − 20	1.86E − 20	1.29E − 20	2.06E − 20	2.44E − 08	**0.00E + 00**
	Std.	1.38E − 21	7.97E − 11	3.46E − 21	4.09E − 21	8.62E − 22	3.07E − 20	7.49E − 08	**0.00E + 00**
*f*20	Min	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	**0.00E + 00**
	Mean	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	1.00E + 00	**0.00E + 00**
	Std.	0.00E + 00	7.14E − 17	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	**0.00E + 00**

other seven algorithms all fall into the local optimum, MPNLPSO can effectively jump out of the local optima and achieve the goal of finding the best.

From the results of unimodal function in Table 6, it can be obtained that MPNLPSO cannot converge to the actual optimal value on the *f*2 and *f*4 functions, but succeeds on the remaining eight functions. However, the average value of the *f*1 function indicates that MPNLPSO's optimizing ability on the *f*1 function is not stable enough. On the *f*2 function, all algorithms fall into local optima. On the *f*3 function, BLPSO and SLPSO present the same excellent optimization capability as MPNLPSO, but relatively, SLPSO is less stable than the other two comparison algorithms. In particular, on the *f*4 function, SLPSO, SWPSO-I, and SFAPSO can all converge to the actual optimal value, and the optimization ability is stronger than MPNLPSO. ACPSO, RTPSO, SWPSO-I, and SFAPSO all have excellent optimizing capabilities on the *f*5 function, but the stability of SWPSO-I is slightly worse. SLPSO can reach the actual optimal value on the *f*9 and *f*10 functions. SWPSO-I, RTPSO, and SFAPSO also show certain competitiveness on the *f*10 function.

From the results of multimodal function in Table 6, it can be seen that optimization capabilities of all algorithms are ordinary on the *f*11 function, while MPNLPSO can reach the actual optimal value on the *f*12 function with a certain degree of stability. On the *f*13 function, BLPSO's optimization ability is slightly stronger than other comparison algorithms, but it still cannot converge to the actual optimal value. What interests us is that MPNLPSO presents better robustness in *f*14 and *f*16–*f*20, and both the accuracy and stability of the solution are stronger than other algorithms. On the contrary, other comparison algorithms do not converge to the actual optimal value on the *f*16–*f*20 functions. Although SLPSO does not find the actual optimal value on the *f*18 function, its accuracy is high and it has a certain degree of stability. The other seven algorithms still fail to jump out of the local optimum on the f20 function, while MPNLPSO can easily jump out of the local optimum and reach the actual optimal value.

Combining Tables 5 and 6, we can conclude that MPNLPSO has presented certain competitiveness in solving both low-dimensional and high-dimensional complex optimization functions. Other comparison algorithms also have certain optimization capabilities, but regardless of overall optimization effect or stability, MPNLPSO is stronger than them.

In addition, this study uses two nonparametric tests with a significance level of 0.05, namely Wilcoxon rank sum test and Friedman test. For Wilcoxon rank sum test, "+" "–" "≈" in Tables 7 and 8 respectively indicate that the solution accuracy and stability of MPNLPSO are better than, worse than, and almost equal to the comparison algorithms. Tables 9 and 10 clearly show the ranking obtained by Friedman test.

As can be seen from Table 7, MPNLPSO has higher solution accuracy than BLPSO on fifteen functions and better performance than CLPSO on nineteen functions. SLPSO algorithm is excellent. The performance of four functions is equivalent to that of MPNLPSO, and three functions are better than MPNLPSO, but the solution accuracy of thirteen functions is inferior to MPNLPSO. SFAPSO has the same solving ability of three functions as MPNLPSO, and the performance of two functions is better than MPNLPSO. In Tables 8 and 10, the result of the Friedman test of MPNLPSO is 1.65. In Table 8, SLPSO ranks second, which is 2.75. BLPSO ranks third, which is 2.85. In Table 10, both BLPSO and SLPSO rank second. To sum up, the experimental analysis and comparison show that the proposed algorithm has strong competitiveness in both solution accuracy and convergence speed.

4.5. Convergence Analysis of the Comparison Algorithms. This section analyzes the convergence of MPNLPSO and seven other comparison algorithms. Figures 5 and 6 show the convergence curves of 30-dimensional unimodal and multimodal functions, and Figures 7 and 8 show the convergence curves of 50-dimensional unimodal and multimodal functions. It can be seen from Figure 5 that MPNLPSO can quickly jump out of the local optimum after falling into the local optimum in *f*1 function, but it falls into the local optimum in the later stage, and the final optimization effect is inferior to SLPSO. The *f*4 function can be solved quickly to a certain accuracy in the early

Table 7: Rank sum test (30 D).

MPNLPSO v.s.	+	−	≈
BLPSO	15	3	2
CLPSO	19	1	0
SLPSO	13	3	4
ACPSO	18	1	1
SFAPSO	15	2	3
SWPSO-I	17	1	2
RTPSO	18	1	1

Table 8: Rank sum test (50 D).

MPNLPSO v.s.	+	−	≈
BLPSO	15	3	2
CLPSO	19	1	0
SLPSO	13	3	4
ACPSO	18	1	1
SFAPSO	15	2	3
SWPSO-I	17	1	2
RTPSO	18	1	1

Table 9: Friedman test (30 D).

Algorithm	Rank
BLPSO	2.85
CLPSO	6.75
SLPSO	2.75
ACPSO	3.85
SFAPSO	3.45
SWPSO-I	3.80
RTPSO	6.00
MPNLPSO	1.65

Table 10: Friedman test (50 D).

Algorithm	Rank
BLPSO	2.60
CLPSO	6.65
SLPSO	2.60
ACPSO	3.55
SFAPSO	3.80
SWPSO-I	3.75
RTPSO	6.10
MPNLPSO	1.65

stage, but SLPSO can achieve better accuracy in the later stage. In addition, MPNLPSO can quickly converge to the actual optimal value. It is precise because MPNLPSO can generally search the actual optimal value within the first 10 iterations, so the continuation of the convergence curve may not be clearly seen in the figure. Similarly, it can be seen from the multimodal function in Figure 6 that MPNLPSO has a very poor convergence effect on $f13$ function and shows good convergence ability on other multimodal functions. This shows that each complex optimization problem has different characteristics, and an algorithm cannot achieve very high solution accuracy and convergence speed for all complex functions. As an old saying goes, there is no free lunch in the world, and the algorithm proposed in this study cannot solve all complex optimization functions perfectly. As can be seen in Figures 7 and 8, MPNLPSO has no reduction in the convergence ability in 50 dimensions, and also has high solution accuracy and convergence speed.

The reasons why MPNLPSO has good robustness are summarized in the following aspects. Firstly, this algorithm divides the particles into two populations according to their fitness value for the purpose of cooperation, and a dynamic opposition-based learning strategy is proposed to activate the particles trapped in search stagnation in the shoddy population. Secondly, according to the different characteristics of the particles, corresponding learning strategies are adopted to enhance the abilities of different particles, which will contribute to the searching process. Thirdly, the small-world network topology is used to build a neighborhood for particles. By learning from the neighborhood optimal particles and getting rid of the situation of learning only from the global optimal particle, the risk of falling into local optimization can be reduced to a certain extent. Finally, in view of the fact that the historical optimal position of particles and the neighborhood optimal position of particles also have the possibility of falling into local optimization, this study takes advantages of Lévy flight to randomly disturb them to increase the diversity of these two particles, so that the

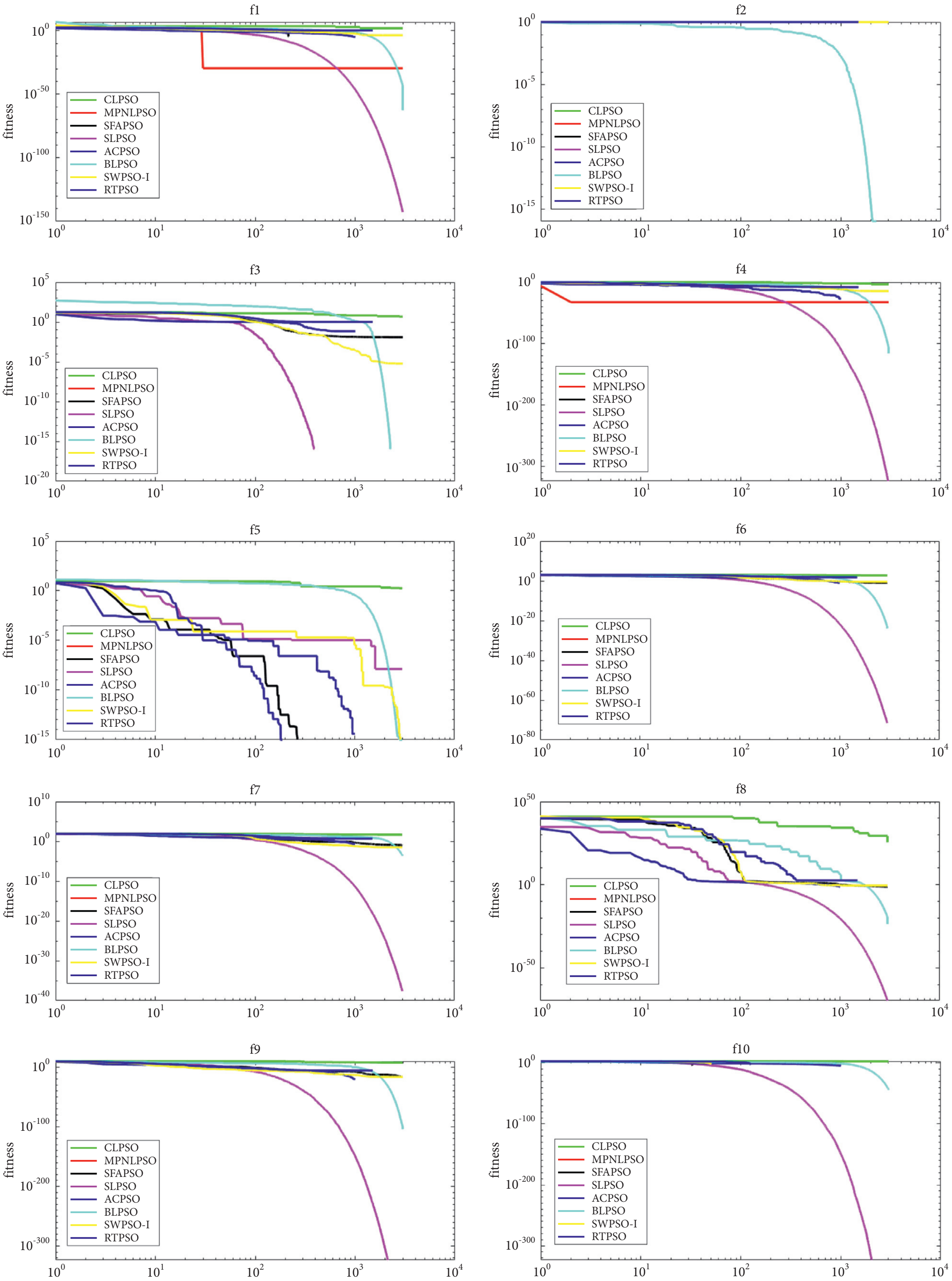

Figure 5: Convergence curve of unimodal function (30 D).

particle swarm can maintain good diversity from the beginning to the end of the search. To sum up, MPNLPSO has taken corresponding measures to strengthen both global exploration and local exploitation, effectively improving the solution accuracy of particle swarm optimization.

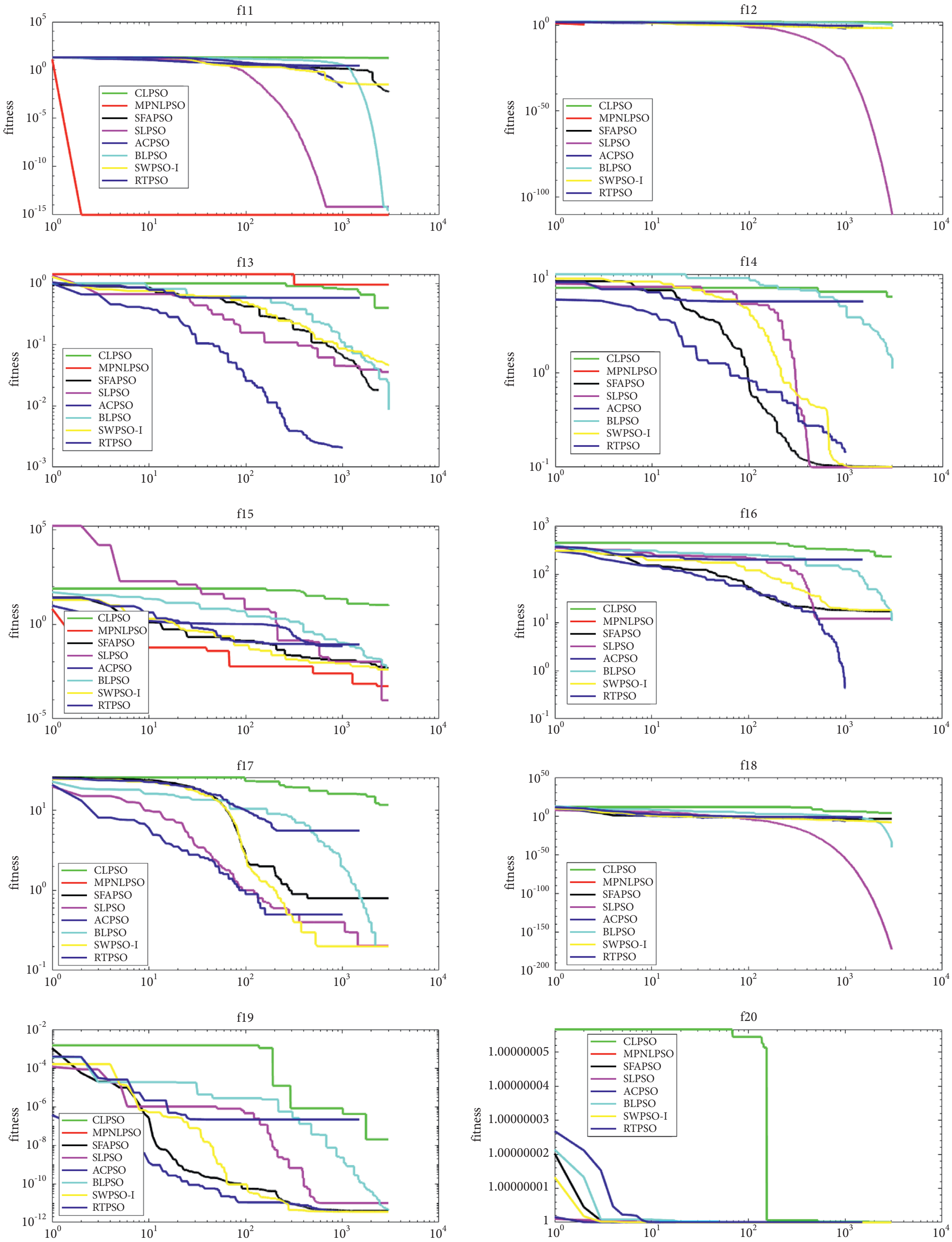

Figure 6: Convergence curve of multimodal function (30 D).

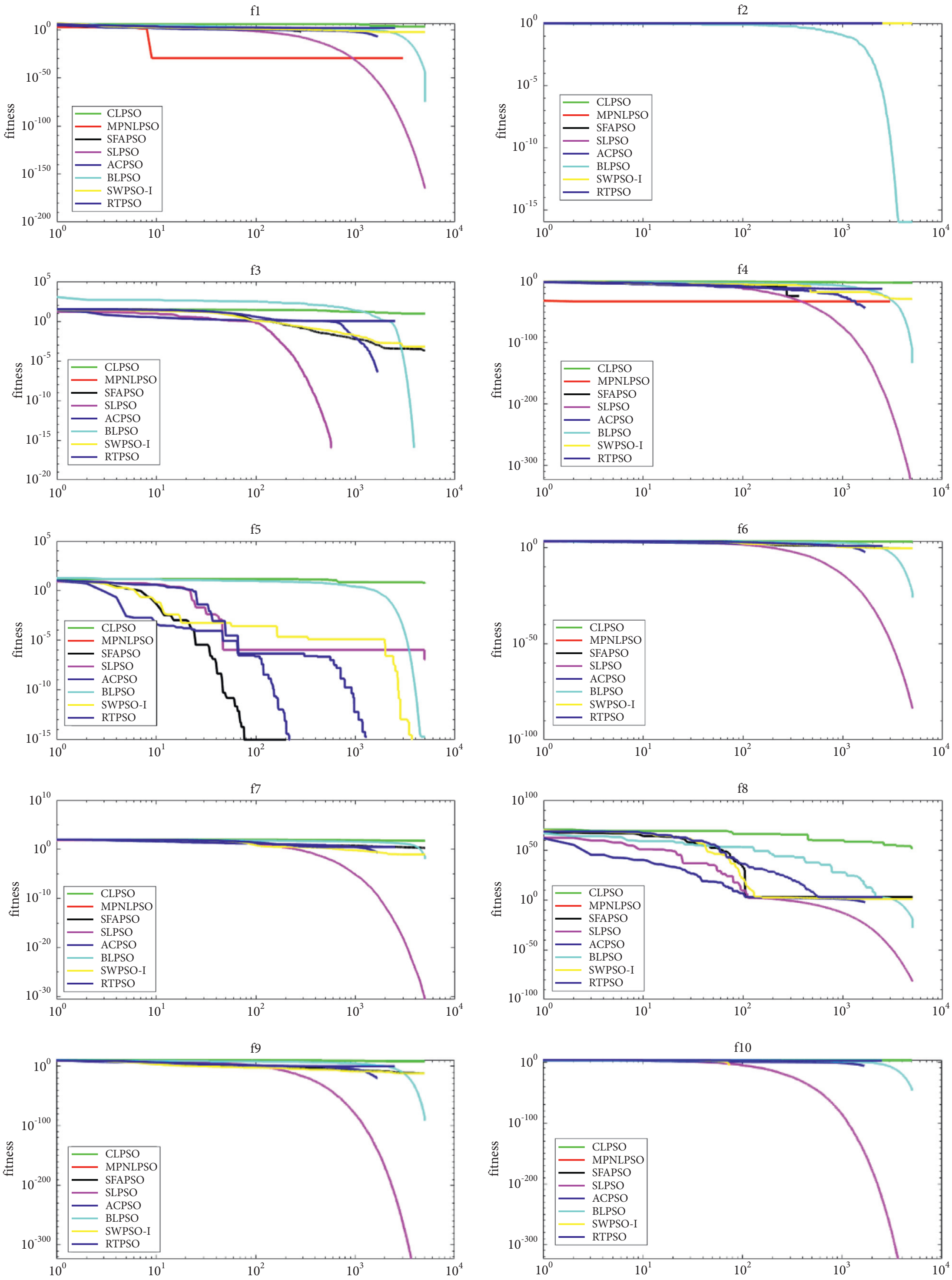

Figure 7: Convergence curve of unimodal function (50 *D*).

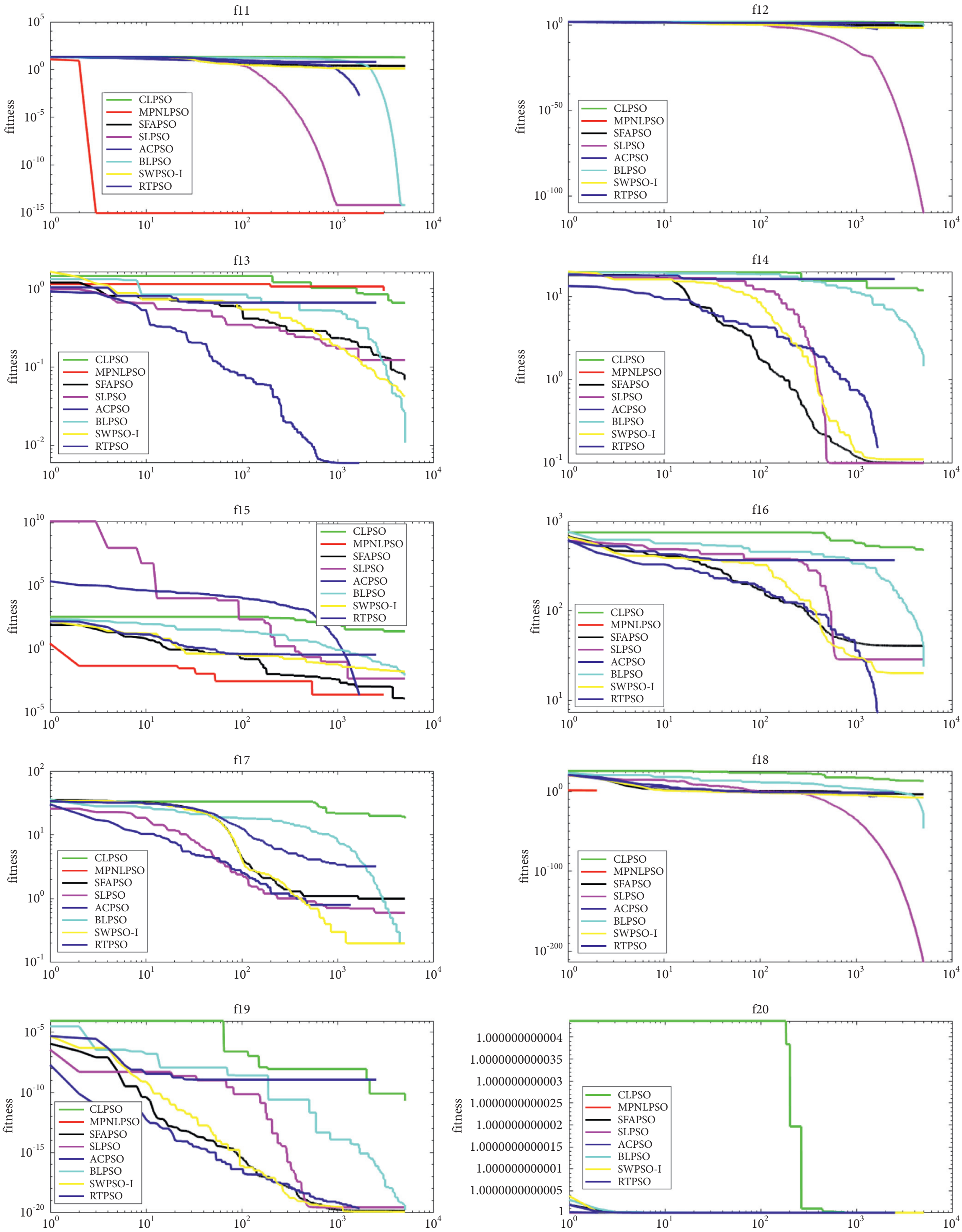

Figure 8: Convergence curve of multimodal function (50 D).

5. Conclusion

In this study, a multipopulation particle swarm optimization algorithm with neighborhood learning is proposed to improve the solution accuracy and convergence speed. As we all know, the reason why standard particle swarm optimization algorithm is easy to fall into local optimization is that all particles only learn from the global optimal particles. In order to get rid of this situation, this study proposes a small-world network neighborhood learning strategy, which makes the particles learn from the neighborhood optimal particles by constructing a neighborhood for each particle. At the same time, in view of the quality of the position of particles in the search process, this study adopts different small-world network neighborhood topologies for different populations, for the purpose of cooperation. For the shoddy population, a dynamic opposition-based learning strategy is proposed to activate the particles trapped in search stagnation. In addition, in order to maintain a certain diversity of the population from beginning to end, Lévy flight is used to randomly perturb the historically optimal and neighborhood optimal particles. In this study, twenty benchmark functions are tested, including ten unimodal test functions and ten multimodal test functions. The MPNLPSO algorithm can search the actual optimal value on most test functions with good stability. In addition, through rank sum test and Friedman test, MPNLPSO ranks first on the whole. At the same time, the convergence ability of MPNLPSO is proved by the convergence curve. Experimental results show that MPNLPSO effectively improves the solution accuracy and the convergence speed of the algorithm with certain robustness.

Although the algorithm proposed in this study has a certain improvement effect, it still has challenges in solving more complex problems. In the next step, we will focus on the analysis of other network topology characteristics and effectively utilize the unique characteristics of different network topologies, so that the algorithm can effectively solve more complex optimization problems. [40].

Data Availability

All data can be obtained by contacting the corresponding author's e-mail.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] G. G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Computing and Applications*, vol. 31, no. 7, pp. 1995–2014, 2019.

[2] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020.

[3] G. G. Wang, "Moth search algorithm: a bio-inspired metaheuristic algorithm for global optimization problems," *Memetic Computing*, vol. 10, no. 2, pp. 151–164, 2018.

[4] J. Tu, H. Chen, M. Wang, and A. H. Gandomi, "The colony predation algorithm," *Journal of Bionics Engineering*, vol. 18, no. 3, pp. 674–710, 2021.

[5] J. Kennedy, "Swarm intelligence," *Handbook of Nature-Inspired and Innovative Computing*, pp. 187–219, Springer, Berlin, Heidelberg, 2006.

[6] C. Blum and X. Li, "Swarm intelligence in optimization," *Swarm Intelligence. Natural Computing Series*, pp. 43–85, Springer, Berlin, Heidelberg, 2008.

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942–1948, 1995.

[8] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation Proceedings*, pp. 69–73, Anchorage, AK, USA, June 1998.

[9] W. Li, X. Meng, and Y. Huang, "Particle swarm optimization with A multidimensional mean learning strategy," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 527, no. 1, 2020.

[10] T. Blackwell and J. Kennedy, "Impact of communication topology in particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 689–702, 2018.

[11] M. Elbes, S. Alzubi, T. Kanan, A. Al-Fuqaha, and B. Hawashin, "A survey on particle swarm optimization with emphasis on engineering and network applications," *Evolutionary Intelligence*, vol. 12, no. 2, pp. 113–129, 2019.

[12] E. H. Houssein, A. G. Gad, K. Hussain, and P. N. Suganthan, "Major advances in particle swarm optimization: theory, analysis, and application," *Swarm and Evolutionary Computation*, vol. 63, Article ID 100868, 2021.

[13] A. M. Eltamaly, "A novel strategy for optimal PSO control parameters determination for PV energy systems," *Sustainability*, vol. 13, no. 2, 2021.

[14] Z. Wei-Min, L. Shao-Jun, and Q. Feng, "θ-PSO: a new strategy of particle swarm optimization," *Journal of Zhejiang University - Science*, vol. 9, no. 6, pp. 786–790, 2008.

[15] D. Tian, X. Zhao, and Z. Shi, "Chaotic particle swarm optimization with sigmoid-based acceleration coefficients for numerical function optimization," *Swarm and Evolutionary Computation*, vol. 51, Article ID 100573, 2019.

[16] W. Liu, Z. Wang, and Y. Yuan, "A novel sigmoid-function-based adaptive weighted particle swarm optimizer," *IEEE Transactions on Cybernetics*, vol. 51, pp. 1085–1093, 2019.

[17] W. Li, B. Sun, and Y. Huang, "Adaptive particle swarm optimization using scale-free network topology," *Journal of Network Intelligence*, vol. 6, no. 3, 2021.

[18] Z. G. Chen, Z. H. Zhan, D. Liu, S. Kwong, and J. Zhang, "Particle swarm optimization with hybrid ring topology for multimodal optimization problems," in *Proceedings of the 2020 IEEE International Conference on Systems, Man and Cybernetics*, pp. 2044–2049, Toronto, ON, Canada, October 2020.

[19] X. D. Li, "Niching without niching parameters: particle swarm optimization using a ring topology," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 150–169, 2009.

[20] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, "Particle swarm optimization with interswarm interactive learning strategy," *IEEE Transactions on Cybernetics*, vol. 46, no. 10, pp. 2238–2251, 2016.

[21] G. Xu, Q. Cui, X. Shi et al., "Particle swarm optimization based on dimensional learning strategy," *Swarm and Evolutionary Computation*, vol. 45, pp. 33–51, 2019.

[22] X. Zhang, H. Liu, and L. Tu, "A modified particle swarm optimization for multimodal multi-objective optimization," *Engineering Applications of Artificial Intelligence*, vol. 95, Article ID 103905, 2020.

[23] V. Bharti, B. Biswas, and K. K. Shukla, "A novel multiobjective gdwcn-pso algorithm and its application to medical data security," *ACM Transactions on Internet Technology*, vol. 21, no. 2, pp. 1–28, 2021.

[24] K. E. Parsopoulos and M. N. Vrahatis, "Particle swarm optimization method for constrained optimization problems," *Intelligent Technologies–Theory and Application: New Trends in Intelligent Technologies*, vol. 76, no. 1, pp. 214–220, 2002.

[25] D. J. Watts and S. H. Strogatz, "Collective dynamics of "small-world" networks," *Nature*, vol. 393, pp. 440–442, 1998.

[26] J. Kleinberg, *The Small-World Phenomenon: An Algorithmic Perspective*, Cornell University Ithaca, NY, USA, 1999.

[27] H. R. Tizhoosh, “Opposition-based learning: a new scheme for machine intelligence,” in *Proceedings of the International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC’06)*, pp. 695–701, Vienna, Austria, November 2005.

[28] A. R. Malisia and H. R. Tizhoosh, “Applying opposition-based ideas to the ant colony system,” in *Proceedings of the 2007 IEEE Swarm Intelligence Symposium*, pp. 182–189, Honolulu, HI, USA, April 2007.

[29] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, “Opposition-based differential evolution algorithms,” in *Proceedings of the 2006 IEEE International Conference on Evolutionary Computation*, pp. 2010–2017, Vancouver, BC, Canada, July 2006.

[30] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, “Opposition-based differential evolution for optimization of noisy problems,” in *Proceedings of the 2006 IEEE International Conference on Evolutionary Computation*, pp. 1865–1872, Vancouver, BC, Canada, July 2006.

[31] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, “Quasi-oppositional differential evolution,” in *Proceedings of the 2007 IEEE congress on evolutionary computation*, pp. 2229–2236, Singapore, September 2007.

[32] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, “Opposition-based differential evolution,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.

[33] M. Ventresca and H. R. Tizhoosh, “A diversity maintaining population-based incremental learning algorithm,” *Information Sciences*, vol. 178, no. 21, pp. 4038–4056, 2008.

[34] H.. Hüseyin and U. G. Harun, “A novel particle swarm optimization algorithm with Levy flight,” *Applied Soft Computing*, vol. 23, pp. 333–345, 2014.

[35] X. Chen, H. Tianfield, C. Mei, W. Du, and G. Liu, “Biogeography-based learning particle swarm optimization,” *Soft Computing*, vol. 21, no. 24, pp. 7519–7541, 2017.

[36] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, “Comprehensive learning particle swarm optimizer for global optimization of multimodal functions,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.

[37] R. Cheng and Y. Jin, “A social learning particle swarm optimization algorithm for scalable optimization,” *Information Sciences*, vol. 291, pp. 43–60, 2015.

[38] M. Vora and T. T. Mirnalinee, “Small world particle swarm optimizer for global optimization problems,” *Lecture Notes in Computer Science*, vol. 8251, pp. 575–580, 2013.

[39] X. Li, “Niching without niching parameters: particle swarm optimization using a ring topology,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 150–169, 2009.

[40] H. Mohammad, R. M. Mohammad, and M. E. Mohammad, “Adaptive cooperative particle swarm optimizer,” *Applied Intelligence*, vol. 39, pp. 397–420, 2013.