

Article

Enhancing Firefly Algorithm with Dual-Population Topology Coevolution

Wei Li ¹, Wangdong Li ¹ and Ying Huang ^{2,*}

¹ School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China; liwei@jxust.edu.cn (W.L.); liwangdong@mail.jxust.edu.cn (W.L.)

² Institute of Mathematical and Computer Sciences, Gannan Normal University, Ganzhou 341000, China

* Correspondence: nhwshy@whu.edu.cn

Abstract: The firefly algorithm (FA) is a meta-heuristic swarm intelligence optimization algorithm. It simulates the social behavior of fireflies with their flash and attraction characteristics. Numerous researches showed that FA can successfully deal with some problems. However, too many attractions between the fireflies may result in high computational complexity, slow convergence, low solution accuracy and poor algorithm stability. To overcome these issues, this paper proposes an enhanced firefly algorithm with dual-population topology coevolution (DPTCFA). In DPTCFA, to maintain population diversity, a dual-population topology coevolution mechanism consisting of the scale-free and ring network topology is proposed. The scale-free network topology structure conforms to the distribution law between the optimal and potential individuals, and the ring network topology effectively reduces the attractions, and thereby has a low computational complexity. The Gauss map strategy is introduced in the scale-free network topology population to lower parameter sensitivity, and in the ring network topology population, a new distance strategy based on dimension difference is adopted to speed up the convergence. This paper improves a diversity neighborhood enhanced search strategy for firefly position update to increase the solution quality. In order to balance the exploration and exploitation, a staged balance mechanism is designed to enhance the algorithm stability. Finally, the performance of the proposed algorithm is verified via several well-known benchmark functions. Experiment results show that DPTCFA can efficiently improve the existing problems of FA to obtain better solutions.

Keywords: firefly algorithm; dual-population; coevolution; neighborhood enhanced search; staged balance

MSC: 68W50



Citation: Li, W.; Li, W.; Huang, Y. Enhancing Firefly Algorithm with Dual-Population Topology Coevolution. *Mathematics* **2022**, *10*, 1564. <https://doi.org/10.3390/math10091564>

Academic Editor: Alessandro Niccolai

Received: 1 April 2022

Accepted: 23 April 2022

Published: 6 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Many scholars were inspired by the behavioral characteristics of certain things in the natural world (e.g., biological systems) to simulate and use computation thinking to solve practical problems. However, as practical problems become more and more complex, traditional optimization algorithms (e.g., genetic algorithms) have difficulty in solving these problems [1]. In recent decades, some new metaheuristic optimization algorithms that simulate the intelligent characteristics of biological populations have been proposed [2], namely swarm intelligence optimization algorithms (SIOAs) [3]. Various studies show that SIOAs can solve the problems of the traveling salesman [4], path planning [5], workshop scheduling [6], and dynamic storage [7]. There are several popular SIOAs, e.g., particle swarm optimization (PSO) [8], ant colony optimization (ACO) [9], the firefly algorithm (FA) [10], cuckoo search (CS) [11], and artificial bee colony (ABC) [12].

Compared with other algorithm in SIOAs, FA has a short development history. It was first proposed by Yang in 2008. The idea of FA comes from simulating the flash courtship

behavior of the fireflies [13]. Due to the simple concept, easy implementation, and good optimization performance, it is widely used to solve various optimization problems. Compared with other algorithms among SIOAs, FA has better performance advantages in continuous numerical optimization problems such as benchmark functions and CEC functions, the same as discrete practical optimization problems such as vehicle path planning and the flexible job-shop scheduling problem (FJSP). The search process of FA mainly relies on the attraction between the fireflies to generate movement. A firefly with a better fitness value (brighter) has greater attractiveness, and it attracts fireflies with worse fitness values (darker) to move for finding a better solution. In the standard FA, a full attraction model is proposed, and the current selected firefly in a population can be attracted by all of the remaining fireflies. This model can effectively enhance the algorithm exploitation ability, but it will result in a higher computational complexity, and too many attractions among the fireflies cause search oscillation, and thereby lead to problems such as slow convergence, low solution accuracy and poor algorithm stability. In addition, the performance of FA is affected by parameter settings.

To tackle these issues, many scholars attempted to improve FA, and contributed various variants of FA. Fister et al. adaptively adjusted the step factor α in FA and used it to solve the image three-coloring problem. Experiments showed that the adaptive parameter strategy can greatly improve the solution accuracy in [14]. Similarly, 12 different chaos map functions were adopted to update the step factor α and the attractiveness β . It indicated that adding a chaos map strategy on the basis of FA can reduce the parameter sensitivity [15]. Furthermore, Wang et al. proposed a random attraction (RA) model to reduce the attractions in fireflies and lower the algorithm computational complexity [16]. The current selected firefly was compared with another randomly chosen one. Thence, each firefly only needed to move at most once. However, the RA model sometimes caused the individual not to move, thus falling into a local optimal value. Subsequently, Wang et al. designed a neighborhood attraction (NA) model based on the characteristics of the firefly population. This paper proved that compared with RA, NA obtained better results in the algorithm performance and stability [17]. Recently, Yu et al. improved the neighborhood search on the basis of the NA model. Unlike the NA model, when there was a better neighborhood individual than the current selected firefly, the individual moved towards a better neighborhood individual, and ended the iteration [18].

In the real world, there are a large number of multi-objective optimization problems (MOPs), and FA is mainly used to solve single-objective problems in continuous optimization problems [19]. It was a great challenge in finding a set of the pareto front for FA [20]. Therefore, Yang et al. extended the FA model from multiple strategies to solve MOPs. The results on the multi-objective function sets revealed that FA is capable of solving MOPs [21]. Lv et al. overcame the population constraint by introducing an iterative method of the compensation factor, so that FA obtained the pareto optimal solution in a short time [22]. Marichelvam et al. extended a new discrete FA to solve a two-objective hybrid flow shop scheduling problem, and verified that the algorithm was superior to other metaheuristic algorithms [23]. Then, for the FJSP, Karthikeyan et al. constructed a continuous function conversion mechanism, and combined with local search strategies to improve the information sharing between the fireflies. It was validated that the algorithm was an effective method to solve FJSP [24].

Although scholars have made an effort in the improvement of FA and applied it to solve practical optimization problems, there were still problems such as low solution accuracy and poor algorithm stability. In order to make full use of the learning experience of the optimal individual in the firefly population and balance the algorithm exploration and exploitation in the search process, this paper proposes an enhanced firefly algorithm with dual-population topology coevolution (DPTCFA). In DPTCFA, to maintain the population diversity, a scale-free network (SN) [25] and a ring network (RN) [26] neighborhood topology population coevolution mechanism is proposed, the SN topology conforms to the distribution law between the optimal and potential individuals of the firefly, and the RN

topology effectively reduces the attractions, and the algorithm thereby has a low computational complexity. The Gauss map strategy is introduced in the SN topology population to lower parameter sensitivity, and in RN topology population, a new distance strategy based on dimension difference is adopted to speed up the convergence. This paper improves a diversity neighborhood enhanced search strategy for firefly position update to strengthen the global optimization ability, and then to increase the solution quality. We designed a staged balance for the algorithm exploration and exploitation mechanism to ensure the global and local optimization abilities in the search process, thereby enhancing the algorithm stability [27]. Simulation results show that the proposed algorithm can fully maintain the population diversity, reduce attractions among the fireflies, lower the algorithm computational complexity, and effectively improve the algorithm performance and search ability. The main contributions of this paper are summarized as follows:

(1) Propose a dual-population topology coevolution mechanism. It can increase population diversity via constructing two different network topology populations. Simultaneously, the neighborhood topology structure can reduce attractions in fireflies, and then lower the algorithm computational complexity. Among them, the SN topology is a complex network structure with a power-law distribution, and this network characteristic just conforms to the distribution law between the optimal individual and the potential individual in the firefly population. In addition, the RN topology is a ring-shaped closed link, and each node is connected to its left and right neighboring nodes. In RN, all fireflies are arranged in a ring topology, and the improved NA search strategy is used to search and optimize. When the firefly population is initialized, the SN and the RN topology are used to construct sub-populations, i.e., the SN topology population and the RN topology population. Then the two sub-populations adopt the corresponding optimization strategy to synchronously iteratively evolve according to each network topology characteristic. After the optimization stage, the two sub-populations are merged to share information, and the roulette method is used to select some firefly individuals to complete the coevolution process.

(2) Improve a diversity neighborhood enhanced search strategy for firefly position update. In the SN and RN topology populations' coevolution process, when there are no other better individuals in the neighborhood of the current selected individual, the selected individual executes a location update operation based on the improved diversity neighborhood enhanced search strategy. This strategy can reduce the probability of the population falling into the local optimum and improve the global optimization ability.

(3) Design a staged balance mechanism. In DPTCFA, it is divided into the global and local optimization stages. The global optimization stage adopts a dual-population topology coevolution mechanism, and the local optimization stage introduces the Nelder–Mead simplex method (NMSM) to perform the fireflies' local fine-tuning to enhance the algorithm exploitation ability. The staged balance of the algorithm exploration and exploitation mechanism can sufficiently solve low solution accuracy problems for FA.

The rest of the paper is structured as follows: Section 2 summarizes the related work of this study. Section 3 introduces in detail an enhanced firefly algorithm with dual-population topology coevolution proposed in this paper. In Section 4, we analyze the experiment results and verify the optimized performance of the proposed algorithm. Relevant conclusions are discussed in Section 5.

2. Related Work

2.1. The Standard FA

Like other SIOAs, FA first starts with the population initialization. Each firefly individual in the population denotes a candidate solution, and it is in the decision variable range. Let N and D be the population size and the problem dimension, respectively. Then, each initial solution $X_i = (x_i^1, x_i^2, \dots, x_i^D)$ can be produced as below [28]:

$$x_i^d = Lb^d + rnd_1^d \times (Ub^d - Lb^d) \quad (1)$$

where $d = 1, 2, \dots, D, i = 1, 2, \dots, N$. $[Lb^d, Ub^d]$ is the lower and upper bounds on the problem dimension. $rnd_1^d \in [0, 1]$ is a uniform random number.

The attractiveness between the fireflies is determined by the light intensity, and the light intensity is generally measured by the objective function value. For a pair of fireflies X_i and X_j , the attractiveness is calculated as follows [28]:

$$\beta(r_{ij}) = \beta_0 e^{-\gamma r_{ij}^2} \tag{2}$$

$$r_{ij} = \|X_i - X_j\| = \sqrt{\sum_{d=1}^D x_i^d - x_j^d} \tag{3}$$

where r_{ij} is the Euclidean distance between two fireflies. The parameter β_0 is the attractiveness of the firefly individual X_i and X_j when the distance $r = 0$. γ represents the light absorption coefficient; it is set to $\frac{1}{\Gamma^2}$, where Γ is the length of the decision variable.

The current selected firefly individual X_i will be compared with all other individuals X_j , where $j = 1, 2, \dots, N \wedge i \neq j$. When the fitness value of the individual X_j is better than the individual X_i , the current selected individual X_i will move toward X_j due to attracting. According to the related literature, the movement equation of X_i is defined as follows [28]:

$$x_i^d(t + 1) = x_i^d(t) + \beta_0 e^{-\gamma r_{ij}^2} (x_j^d(t) - x_i^d(t)) + \alpha \varepsilon_i \tag{4}$$

Equation (4) is mainly composed of three parts: The first part represents the d-dimension position of the selected individual X_i under the current iteration. The second part is called the attraction term; it is related to the attractiveness between two firefly individuals. The third part denotes the random item, where α is the step factor and is a fixed value in the range $[0, 2]$. $\varepsilon_i \in [-0.5, 0.5]$ is a uniform random number.

In this paper, the following minimum problem is mainly solved:

$$\min_{x \in [Lb, Ub]} f(x)$$

where $f(x)$ is the objective value of the individual x , and represents the light intensity of each firefly.

FA implementation steps are as follows: (1) Randomly initialize all firefly individuals within the search space. (2) Evaluate the fitness value of each individual according to the objective function. (3) Each current selected individual X_i compares its fitness value with the remaining individuals X_j in the population. If the fitness value of X_j is better than X_i , then X_i updates its position according to Equation (4) and evaluates the fitness value of the new X_i . (4) Repeat step (3) until the termination condition is satisfied. Finally, sort according to the fitness value to find the global optimal solution of the firefly population.

To better explain the full attraction model of the standard FA, let us use a simple minimization problem to make an instruction. Suppose there are six individuals in the firefly population, $[X_1, X_2, X_3, X_4, X_5, X_6]$. The objective function values of these individuals are respectively $f(X_1) = 10.2, f(X_2) = 2.3, f(X_3) = 5.6, f(X_4) = 3.6, f(X_5) = 4.8$, and $f(X_6) = 12.8$. According to the objective function value, the fireflies can be sorted as: X_2, X_4, X_5, X_3, X_1 , and X_6 . Figure 1 shows the full attraction mechanism of the standard FA. As seen, the individual X_1 moves towards X_2, X_3, X_4 , and X_5 . X_3 moves towards X_2, X_4 , and X_5 . X_4 is only attracted by X_2 . X_5 is attracted by X_2 and X_4 . Among them, X_2 and X_6 are the best and worst individuals in the population, respectively. All firefly individuals are attracted by X_2 , and X_6 moves towards the remaining individuals in the population. The mechanism of attracting each other is called the full attraction model. Obviously, too many attractions make the algorithm higher computational complexity, and easily lead to search oscillations, thereby reducing the solution accuracy.

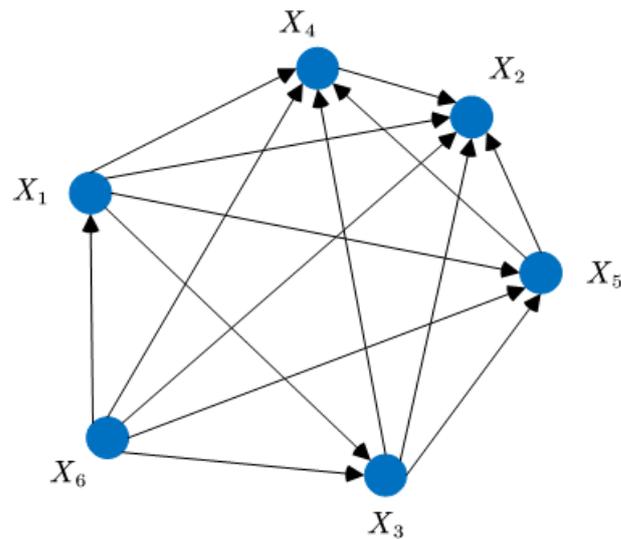


Figure 1. The full attraction mechanism.

In this paper, to tackle these issues, an SN and RN neighborhood topology populations coevolution mechanism is proposed, and it is used to solve the problems caused by the full attraction model. The diversity neighborhood enhanced search strategy is improved to increase the position disturbance for enhancing the global search ability. A staged balance of the algorithm exploration and exploitation mechanism is designed to maintain the consistency of the global and local search abilities.

2.2. Scale-Free Network

Scale-free networks (SNs) were first discovered by the physicist Barabási as network structural features [29]. The distribution of each node connection numbers in the network shows a power-law distribution, and its probability distribution function is as follows [30]:

$$P(k) = k^{-\lambda} \quad (5)$$

where k is the node degree, $P(k)$ represents the probability distribution of the node degree, and the power exponent λ is a parameter describing the network structure characteristic. Figure 2 is the function image of Equation (5). As seen, the node degree k increases, and $P(k)$ decreases, that is, a few nodes in the SN have a larger degree.

The biggest advantage of FA is to update the position by fireflies attracting each other. Figure 3 shows the SN topology. A few hub nodes (blue nodes) have a larger degree and are called optimal nodes, and most nodes (black nodes) have a smaller degree and are called potential nodes. Corresponding to FA, a small number of optimal individuals are in the hub position of the population, and have a greater impact on guiding the evolution direction. For the potential individuals, they have less impact on the algorithm performance.

Barabási et al. proposed a concise SN construction model (i.e., B-A model). The process mainly includes two parts: First, select some optimal individuals in the population to form a fully-connected network. Then, the remaining individuals are respectively connected to the network through loops, and the individuals with larger node degrees can be connected. Figure 4 shows the construction process of the SN topology population. The detailed steps are described as follows: (1) Evaluate the fitness value after population initialization, and sort the population according to the fitness value; (2) select some better individuals from the population as the optimal individual, and these better individuals will be connected to a fully-connected network; (3) in the iteration, the remaining potential individuals use a roulette selection algorithm to connect to the fully-connected network. The proportion of

the node degree to the total node degree is called the degree ratio. The degree ratio and cumulative probability are defined as follows:

$$P(k_i) = \frac{k_i}{\sum_{j=1}^n k_j} \tag{6}$$

$$C(k_i) = \sum_{j=1}^i P(k_j) \tag{7}$$

where k_i in Equation (6) indicates the individual i degree is k . n denotes the individual numbers in the completed network. $P(k_i)$ represents the proportion of the individual i degree to the total network degree, and Equation (7) is the cumulative probability.

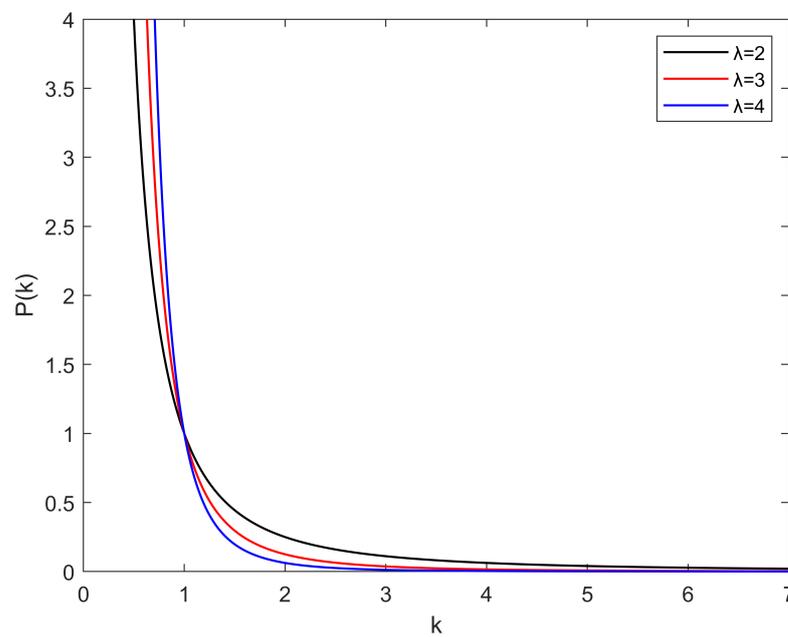


Figure 2. Power-law distribution function.

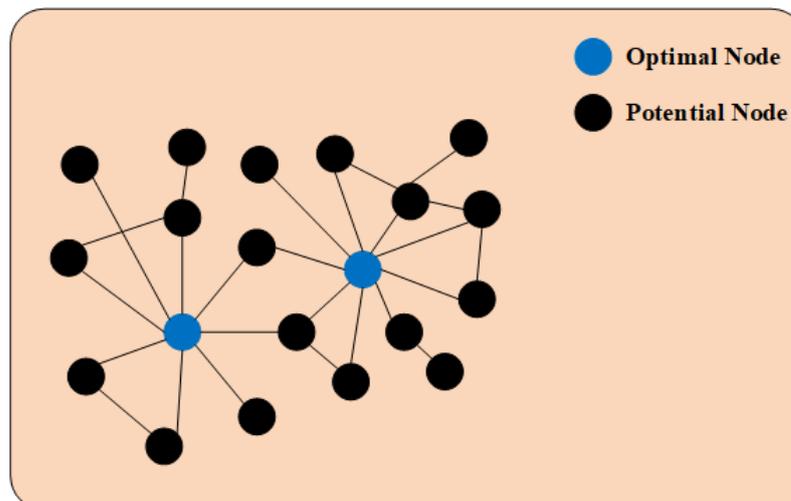


Figure 3. The SN topology.

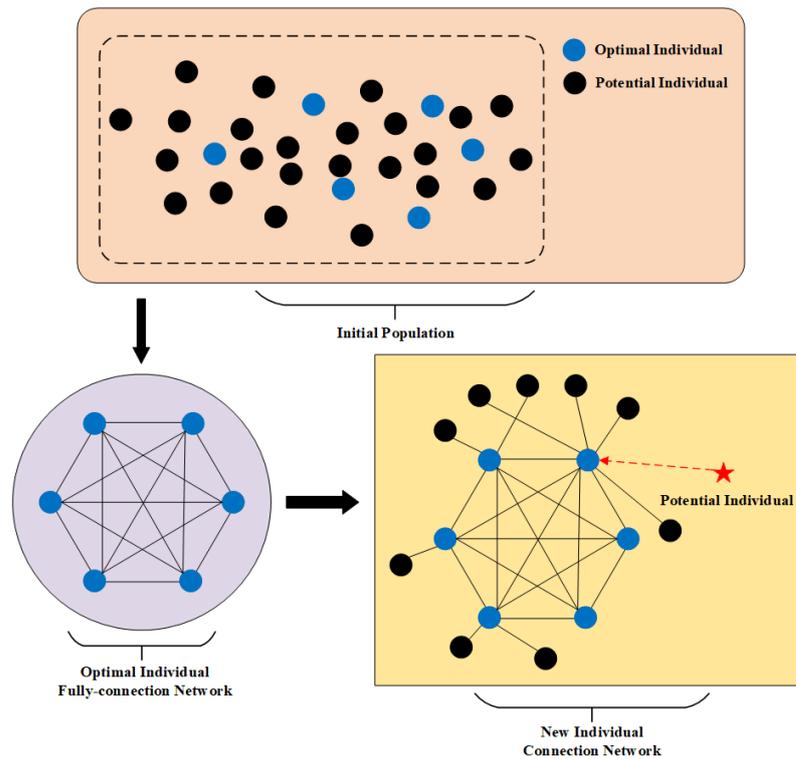


Figure 4. The SN topology population construction process.

After constructing the SN topology population, the population starts to search for the optimal. The current selected firefly individual uses the roulette selection algorithm to choose several firefly individuals from the SN neighborhood topology populations for attracting operation according to the fitness value. Similarly, the fitness value ratio and cumulative probability are described as follows:

$$P(x_i) = \frac{f(x_i)}{\sum_{j=1}^n f(x_j)} \tag{8}$$

$$C(x_i) = \sum_k^i P(x_k) \tag{9}$$

where $P(x_i)$ represents the fitness value ratio of individual x_i , and $C(x_i)$ is the cumulative probability.

Compared with the full attraction model in FA, the SN topology population optimization method effectively reduces attractions among the fireflies. In addition, the population structure with the power-law distribution is more in line with the natural biological evolution law, thereby maintaining adequately the population diversity.

2.3. Ring Network

Recently, Yu et al. proposed a modified neighborhood attraction (MNA) model based on the NA model [18]. Simulation experiments showed that the MNA model effectively reduces attractions between the fireflies. Based on this work, we construct a ring network (RN) topology.

In RN, all firefly individuals in the population are arranged in a ring topology. If the firefly individuals' indexes are adjacent to each other, they will be directly connected in the RN topology. However, this stipulates that the first individual is connected with the last individual for the special condition. The K-neighborhood of each individual X_i contains $2k + 1$ fireflies, i.e., $X_{(i-k)}, X_{(i-k+1)}, \dots, X_{(i-1)}, X_i, X_{(i+1)}, \dots, X_{(i+k-1)}, X_{(i+k)}$.

The NA model was first proposed by Wang, and the attraction among the fireflies only occurred in their K-neighborhood. The current selected firefly individual is compared with the remaining individuals in the K-neighborhood, and if the fitness value in the remaining individuals is better, the movement is carried out. Figure 5 clearly shows the mechanism of the NA model. However, in the RN topology population, other individuals' X_j indexes are set to $i - k, i - k + 1, \dots, i - 1, i$, that is, the individual X_i needs to compare with k individuals. However, the average attractions of each firefly are $((N - 1))/2$ in the standard FA, while the RN topology population only has $k/2$ at most. It can be seen that the RN topology population mechanism effectively lowers the computational complexity. Figure 6 is the attraction model in the RN topology population.

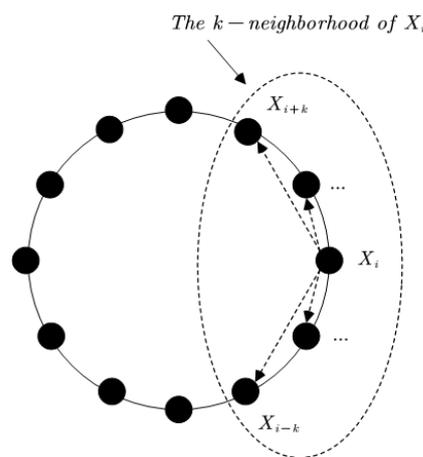


Figure 5. The NA model.

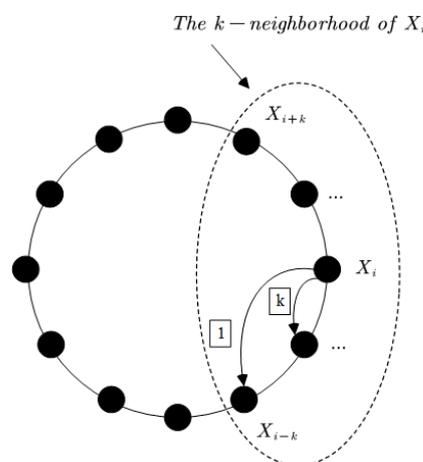


Figure 6. The attraction model in the RN topology population.

2.4. Adaptive Parameter

(1) Attractiveness with the Gauss map strategy

In FA, the attractiveness among the fireflies is calculated by Equation (2). The algorithm running time is too long due to the need to calculate the Euclidean distance r_{ij} between the individuals X_i and X_j . Therefore, a chaos map is used to generate the sequence instead of the attractiveness value calculated by Equation (2). The attractiveness with the Gauss map is defined as follows:

$$\beta(t + 1) = \begin{cases} 0, & \beta(t) = 0 \\ \text{mod}(\mu/\beta(t), 1), & \beta(t) \neq 0 \end{cases} \tag{10}$$

when $t = 0$, the initial attractiveness $\beta_0 \in [0, 1]$ is a uniform random number, and μ is usually set to 1. Correspondingly, the firefly movement can be calculated as follows:

$$x_i^d(t + 1) = x_i^d(t) + \beta(t + 1)(x_j^d(t) - x_i^d(t)) + \alpha \varepsilon_i \tag{11}$$

Experiments have proved that the chaos sequence generated by the Gauss map can reduce the time consumed by calculating the Euclidean distance. Compared with a pseudo-random number, the chaos sequence can achieve better performance [15]. Therefore, the attractiveness with the Gauss map strategy is used to update the attractiveness of the SN topology population dynamically.

(2) Step factor dynamic update

According to the related literature, the step factor α is dynamically updated as follows [31]:

$$\alpha(t + 1) = \alpha(t) \times (1/9000)^{\frac{1}{t}} \tag{12}$$

when $t = 0$, the initial step factor α_0 is set to 0.5.

The fixed step factor causes the disturbance term of Equation (4) to not change, thereby leading to a poor algorithm performance [32]. Numerous studies showed that an adaptive parameter strategy can improve the algorithm performance and solution accuracy [33]. Thus, in this work, the step factor α of the SN and RN topology populations adopts Equation (12) to update dynamically.

2.5. New Distance Strategy

In standard FA, the attractiveness between the fireflies is related to the Euclidean distance. As seen from Equation (3), the Euclidean distance calculation will consume more running time due to the power calculation in the equation. Yu et al. designed a new equation for expressing distance based on the dimension difference [34].

$$r_{ij}^d = \frac{|x_i^d - x_j^d|}{Max^d - Min^d} \tag{13}$$

where $[Min^d, Max^d]$ represents each dimension range in the current population, and $d = 1, 2, \dots, D$. The attractiveness is redefined as follows:

$$\beta(r_{ij}) = rnd_2 e^{-r_{ij}^2} \tag{14}$$

where $rnd_2 \in [0, 1]$ is a uniform random number. Compared with the standard FA, the firefly movement equation is changed as follows:

$$x_i^d(t + 1) = x_i^d(t) + \beta(r_{ij})(x_j^d(t) - x_i^d(t)) + \alpha \varepsilon_i \Gamma \tag{15}$$

where Γ is the problem spatial range.

The distance strategy based on the dimension difference removes the power calculation of the Euclidean distance. It greatly reduces the algorithm running time. In this work, according to the structural characteristics of the RN topology population, Equation (13) is used to represent the distance between the fireflies.

2.6. Nelder–Mead Simplex Method

The Nelder–Mead Simplex Method (NMSM) was first proposed by Nelder and Mead in 1965 [35]. This method mainly uses operations such as reflection, expansion, inner contraction, and outer contraction to find a better solution and replace the worst individual in the population. This process is repeated until the termination condition is met, where X_{best} and X_{worst} represent the best individual and the worst individual in the population,

respectively. X_1 is the second-worst individual in the population. X_{Worst}^{cent} is the center point of all points except X_{worst} . X_{Worst}^{ref} , X_{Worst}^e , X_{Worst}^{ic} , and X_{Worst}^{oc} represent the reflection point, expansion point, and inner and outer contraction points, respectively. Figure 7 shows the location of different points in the NMSM.

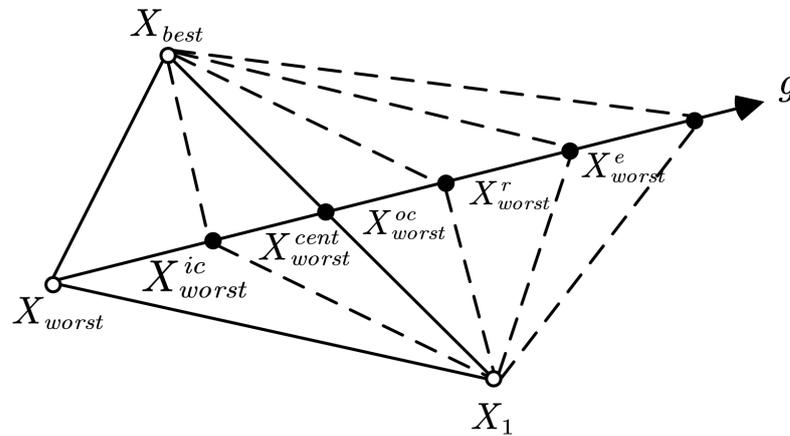


Figure 7. The location of different points in NMSM.

Except for the best individual X_{best} , the center of all other individuals X_{best}^{cent} is calculated as follows:

$$X_{best}^{cent} = \frac{1}{N-1} \left(\sum_{i=1}^N (X_i - X_{best}) \right) \tag{16}$$

The local fine-tuning of each individual in the population is defined as follows:

$$X_i(t+1) = \delta_0 X_i(t) + \delta_1 (X_{best}(t) - X_i(t)) + \delta_2 (X_{best}^{cent}(t) - X_i(t)) \tag{17}$$

where $X_i(t)$ represents the individual X_i of the $t - th$ iteration. δ is the reflection, expansion and contraction coefficient. δ_0 is usually set to 1, and δ_1 and δ_2 respectively are calculated by the following equations:

$$\begin{aligned} \delta_1 &= \frac{3\sigma}{N-1} \\ \delta_2 &= 3\sigma - 1 \end{aligned} \tag{18}$$

where $\sigma \in [0, 1]$ is a uniform random number.

The NMSM is mainly used for local fine-tuning of the firefly individuals to enhance the exploitation ability of the algorithm. In this work, the NMSM is used in the local optimization stage of the algorithm to balance the exploration and exploitation.

3. The Proposed Algorithm (DPTCFA)

In this section, an enhanced FA (DPTCFA) is proposed based on a dual-population topology coevolution mechanism, a staged balance mechanism, and an improved diversity neighborhood enhanced search strategy. In the proposed algorithm, in order to maintain the population diversity, an SN and RN neighborhood topology populations coevolution mechanism is proposed, that is, constructing two network topology populations, the SN and RN topology populations. The SN characteristics conform to the distribution law between the optimal and potential individuals, and RN effectively reduces attractions among the fireflies. To enhance the global search ability of FA, we improved the diversity neighborhood and enhanced the search strategy for the firefly position update. We designed a staged balance mechanism to keep the consistency of the global and local search capabilities. The whole algorithm process is divided into the global and local optimization stages. The global

optimization stage adopts the dual-population topology coevolution mechanism, and the local optimization stage introduces the NMSM to fine-tune the firefly individuals.

3.1. Diversity Neighborhood Enhanced Search Strategy

In addition to the adaptive parameter strategy and the distance strategy based on dimension difference described in related work, for the firefly position update operation, this paper improves the diversity neighborhood enhanced search strategy. In the full attraction model of the standard FA, if the fitness value of the remaining firefly individuals X_j in the population is not better than the current selected individual X_i , then X_i will not execute any operation [10]. Inspired by the related literature [36], in the SN and RN topology populations, when the individual X_i index is equal to another individual X_j index, X_i will move towards the global optimal individual X_{best} , and generate the trial individual LX_i , which is defined as follows:

$$LX_i = r_1 X_i + r_2 X_{best} + r_3 (X_c - X_d) \quad (19)$$

where $c, d \in [1, N] \wedge c \neq d \neq i$. $r_1, r_2, r_3 \in [0, 1]$ are uniform random numbers and $r_1 + r_2 + r_3 = 1$. If the fitness value of the test individual LX_i is better than X_i , replace X_i with LX_i .

3.2. Dual-Population Topology Coevolution Mechanism

The SN and RN neighborhood topology populations' coevolution mechanism mainly allows the SN and RN topology populations to iteratively optimize under corresponding strategies. In the SN topology population, the Gauss map strategy is used to calculate the attractiveness and reduce the parameter sensitivity. In the RN topology population, a new distance strategy based on dimension difference is designed to decrease the running time caused by calculating the Euclidean distance. In addition, the neighborhood search strategy based on the preset probability p_{ns} is carried out in the RN topology population. The strategy is mainly divided into two parts: Opposition-based learning (OBL) and Cauchy mutation (CM). The OBL and CM mainly perturb the position of firefly individuals to enhance the algorithm exploration ability [34]. Finally, both the SN and RN topology populations adopt the diversity neighborhood enhanced search strategy. When the firefly population is initialized, the SN and the RN topologies are used to construct sub-populations. Then, the two sub-populations adopt the corresponding optimization strategy to synchronously iteratively evolve. After the optimization stage, the two sub-populations are merged to share information, and the roulette method is used to select some firefly individuals to complete the coevolution process.

The detailed process of the SN and RN topology population optimizations is seen in the pseudo-code Algorithms 1 and 2.

3.3. Staged Balance Mechanism

In DPTCFA, in order to balance the algorithm exploration and exploitation, a staged balance mechanism is designed. The whole algorithm process is divided into global and local optimization stages. Among them, the global optimization stage mainly uses the dual-population topology coevolution mechanism for optimization. After this stage, merge the population and select N individuals. Subsequently, NMSM is mainly adopted for local fine-tuning of individuals. For the detailed implementation of NMSM, see the pseudo-code Algorithm 3.

Algorithm 1 Pseudo-code of the SN topology population optimization

```

1 Begin
2 Initialization: Population size  $N$ , the SN initial node number  $N_0$ , minimum number of nodes
   connected  $N_c$ , Initial value of chaotic attractiveness  $\beta_0$ ;
3 Select  $N_0$  optimal individuals as the initial nodes in the network;
4 Calculate the degree of each initial node in the current network:  $degree = N_0 - 1$ ;
5 for  $i = N_0 + 1 : N$  do
6   for  $j = 1 : N_c$  do
7     | The insertion node is connected to the network using the roulette selection algorithm;
8     | Update the degree of nodes to be inserted and connected;
9   end
10 end
11 for  $m = 1 : N$  do
12   | The roulette selection algorithm is used to select  $N_1$  individuals from the neighborhood of
   individual  $X_m$ ;
13   for  $n = 1 : N_1$  do
14     | if  $f(X_n) < f(X_m)$  then
15     |   | Update the chaotic attractiveness using Equation (10);
16     |   | Move  $X_m$  toward  $X_n$  according to Equation (11);
17     |   | Process the boundary of the new  $X_m$ ;
18     |   | Calculate the fitness value of the new  $X_m$ ;
19     |   end
20     | else
21     |   | Generate trial individuals  $LX_m$  according to Equation (19);
22     |   | Calculate the fitness values of  $X_m$  and  $LX_m$ ;
23     |   | Select the better of the two and replace  $X_m$ ;
24     |   end
25   end
26 end
27 End

```

Algorithm 2 Pseudo-code of the RN topology population optimization

```

1 Begin
2 Initialization: Population size  $N$ , k-neighborhood size  $K$ , neighborhood search probability  $P_{ns}$ ;
3 for  $i = 1 : N$  do
4   for  $j = i - K : i$  do
5     |  $j = (j + N) \% N$ ;
6     | if  $f(X_j) < f(X_i)$  then
7     |   | Update the attractiveness according to Equation (14);
8     |   | Move  $X_i$  toward  $X_j$  according to Equation (15);
9     |   | Process the boundary of the new  $X_i$ ;
10    |   | Calculate the fitness value of the new  $X_i$ ;
11    |   end
12    | else
13    |   | Generate trial individuals  $LX_i$  according to Equation (19);
14    |   | Calculate the fitness values of  $X_i$  and  $LX_i$ ;
15    |   | Select the better of the two and replace  $X_i$ ;
16    |   end
17  end
18 end
19 if  $random < P_{ns}$  then
20   | Execute neighborhood search: Opposition-based learning (OBL) and Cauchy mutation (CM) [34];
21 end
22 End

```

3.4. Implementation

From the above description of the proposed algorithm, in the DPTCFA, the algorithm is divided into two stages, i.e., the global search stage and the local search stage. To enhance the exploration, two topology populations are constructed in the global search stage—the SN and RN topology populations. According to dual-population topology characteristics, design and improve related strategies for optimization. After this stage, in order to balance

the algorithm exploration and exploitation, NMSM is used to enhance the local search ability in the second stage. The pseudo-code of DPTCFA is seen in Algorithm 4.

Algorithm 3 Pseudo-code of NMSM

```

1 Begin
2 Initialization: Population size  $N$ , maximum number of iterative search
    $MaxSearch$ , current iterations  $ms$ , reflection coefficient  $\delta_0$ ;
3 while  $ms < MaxSearch$  do
4   Calculate the fitness value of the population;
5   Find the best individual  $X_{best}$  in the population;
6   Calculate  $X_{best}^{cent}$  according to Equation (16);
7   for  $i = 1 : N$  do
8     Calculate  $\delta_1$  and  $\delta_2$  according to Equation (18);
9     Update firefly individual position according to Equation (17);
10    Process the boundary of the new individual;
11    Calculate the fitness value of the new individual;
12  end
13   $ms = ms + 1$ ;
14 end
15 End

```

Algorithm 4 Pseudo-code of DPTCFA

```

1 Begin
2 Initialization: The SN topology and RN topology population initialization, and
   parameter initialization, population size  $N$ , maximum number of iterations
    $MaxIt$ , current iterations  $t$ , step factor  $\alpha$ ;
3 Calculate the fitness values of the SN topology and RN topology population
   according to the objective function;
4 while  $t < MaxIt$  do
5   Execute Algorithm 1;
6   Record the optimal value of the SN topology population;
7   Execute Algorithm 2;
8   Record the optimal value of the RN topology population;
9   Update the step factor  $\alpha$  according to the Equation (12);
10   $t = t + 1$ ;
11 end
12 if the optimal value is not met then
13   Execute Algorithm 3;
14 end
15 End

```

4. Experiment Study

In this section, the simulation experiment is mainly divided into three parts: **(1) Parameter study:** When constructing the SN topology population, the proper initial node number N_0 and minimum node connection number N_c can improve the algorithm performance. **(2) Algorithm comparison:** To further highlight the performance of the proposed algorithm (DPTCFA), it is compared with several other FA variants. In the experiment, the maximum iteration number was uniformly set to 5000, each algorithm was run 30 times and the average optimal values were recorded. **(3) Strategy evaluation:** In order to verify the effectiveness of the proposed algorithm strategy, the strategy was evaluated through ablation experiments. The experimental platform was Intel(R) Core (TM) i7-8750H CPU @ 2.20 GHz, RAM 24.0 GB and MATLAB 2021a.

4.1. Benchmark Functions

To verify the performance of the proposed algorithm (DPTCFA), 26 benchmark functions were used in the experiments of this paper. Table 1 shows the basic information of the 26 benchmark functions. In addition, in order to facilitate the data observation, this paper uses the absolute error between the solved experiment results by the algorithm and the function optimal value as the criterion, that is, if the absolute error is 0, it means that the optimal value was obtained [37].

f_1 – f_7 are unimodal functions with only one global optimal value, f_8 – f_{24} are multimodal functions with multiple local minima, and f_{25} – f_{26} are composite functions. All benchmark functions are used to solve the problem of $\min f(x)$. In this paper, the problem dimension D is set to 30 and 50.

Table 1. The benchmark functions.

| Function No. | Function Name | Search Range |
|--------------|-----------------|---------------|
| f_1 | Brown | [−1, 4] |
| f_2 | Powell Sum | [−1, 1] |
| f_3 | Sum Squares | [−10, 10] |
| f_4 | Step 2 | [−100, 100] |
| f_5 | Schwefel 1.2 | [−100, 100] |
| f_6 | Schwefel 2.22 | [−100, 100] |
| f_7 | Schwefel 2.23 | [−10, 10] |
| f_8 | Schwefel 2.26 | [−500, 500] |
| f_9 | Griewank | [−100, 100] |
| f_{10} | Ridge | [−5, 5] |
| f_{11} | Sphere | [0, 10] |
| f_{12} | Zakharov | [−5, 10] |
| f_{13} | Alpine 1 | [−10, 10] |
| f_{14} | Ackley 4 | [−35, 35] |
| f_{15} | Periodic | [−10, 10] |
| f_{16} | Quartic | [−1.28, 1.28] |
| f_{17} | Rastrigin | [−5.12, 5.12] |
| f_{18} | Xin-She Yang 3 | [−20, 20] |
| f_{19} | Xin-She Yang 4 | [−10, 10] |
| f_{20} | Shubert | [−10, 10] |
| f_{21} | Shubert 3 | [−10, 10] |
| f_{22} | Shubert 4 | [−10, 10] |
| f_{23} | Styblinski-Tang | [−5, 5] |
| f_{24} | Rosenbrock | [−30, 30] |
| f_{25} | Penalized 1 | [−50, 50] |
| f_{26} | Penalized 2 | [−50, 50] |

4.2. Parameter Study

In DPTCFA, there are two important parameters, namely the initial node N_0 and the minimum number of connections N_c , which have a greater impact on the SN topology population optimization. Excessive N_0 and N_c will increase the attractions of firefly individuals, causing the algorithm to fail to achieve improved results. Too few N_0 and N_c can easily cause problems such as low accuracy and poor algorithm stability. Therefore, it is very necessary to determine the appropriate N_0 and N_c in order to improve the solution accuracy. In the SN topology population, the population size is $N = 30$. Generally, N_0 should not exceed $N/3$, and the range of N_c is normally between $[(2N_0)/3, N_0 - 1]$. Therefore, the parameter settings of N_0 and N_c can be divided into 12 groups, i.e., Pa1(10, 9), Pa2(10, 8), Pa3(10, 7), Pa4(9, 8), Pa5(9, 7), Pa6(9, 6), Pa7(8, 7), Pa8(8, 6), Pa9(8, 5), Pa10(7, 6), Pa12(7, 5), and Pa12(7, 4).

We set the parameters (N_0, N_c) to the above 12 groups of values, respectively, and ran them on DPTCFA. The results are shown in Tables 2 and 3. The “Mean” in the table is the average optimal fitness value obtained by running the algorithm 10 times, and the

optimal results are indicated in bold. It can be seen from the table that different parameter combinations have a greater impact on the algorithm performance. On the function f_4 , f_7 , f_8 , f_9 , f_{10} , f_{11} , f_{17} , f_{19} , and f_{21} , the algorithm converged to the optimal function value under 12 groups of parameters. However, on the functions f_{14} , f_{15} , f_{20} , f_{22} , and f_{23} , Pa1 could obtain the optimal value. In addition, compared to other parameter combinations, Pa4 had better results on the functions f_{16} , f_{18} , and f_{26} . Figure 8 lists the search curves of DPTCFA with different parameters (Pa1–Pa12).

Table 2. Results of the parameter analysis of DPTCFA ($N_0 = 10$ and $N_0 = 9$).

| Problems | Pa1 (10, 9) Mean | Pa2 (10, 8) Mean | Pa3 (10, 7) Mean | Pa4 (9, 8) Mean | Pa5 (9, 7) Mean | Pa6 (9, 6) Mean |
|----------|---|--|---|--|---|---|
| f_1 | 1.36×10^{-142} | 5.92×10^{-132} | 1.25×10^{-146} | 9.15×10^{-141} | 2.30×10^{-141} | 3.79×10^{-150} |
| f_2 | 2.09×10^{-156} | 4.97×10^{-164} | 1.12×10^{-148} | 3.94×10^{-155} | 2.08×10^{-166} | 7.56×10^{-165} |
| f_3 | 1.18×10^{-139} | 1.86×10^{-140} | 1.30×10^{-144} | 1.14×10^{-141} | 8.41×10^{-136} | 6.30×10^{-134} |
| f_4 | 0 | 0 | 0 | 0 | 0 | 0 |
| f_5 | 1.83×10^{-146} | 3.17×10^{-139} | 6.37×10^{-137} | 4.47×10^{-145} | 1.90×10^{-135} | 1.43×10^{-135} |
| f_6 | 6.73×10^{-72} | 7.27×10^{-73} | 3.37×10^{-70} | 7.76×10^{-71} | 1.09×10^{-65} | 7.43×10^{-68} |
| f_7 | 0 | 0 | 0 | 0 | 0 | 0 |
| f_8 | 0 | 0 | 0 | 0 | 0 | 0 |
| f_9 | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{10} | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{11} | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{12} | 4.15×10^{-143} | 2.49×10^{-131} | 1.81×10^{-136} | 2.48×10^{-145} | 8.25×10^{-138} | 7.13×10^{-130} |
| f_{13} | 6.97×10^{-69} | 2.69×10^{-72} | 1.28×10^{-68} | 8.20×10^{-73} | 1.04×10^{-73} | 1.83×10^{-77} |
| f_{14} | 0 | 0 | 3.87×10^{-03} | 7.65×10^{-05} | 4.16×10^{-03} | 0 |
| f_{15} | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{16} | 5.33×10^{-07} | 3.29×10^{-07} | 6.26×10^{-07} | 8.66×10^{-08} | 3.59×10^{-07} | 1.85×10^{-06} |
| f_{17} | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{18} | 1.36×10^{-04} | 2.28×10^{-03} | 3.31×10^{-06} | 3.04×10^{-08} | 1.36×10^{-04} | 9.01×10^{-07} |
| f_{19} | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{20} | 0 | 1.45×10^{-12} | 1.03×10^{-10} | 6.34×10^{-12} | 0 | 0 |
| f_{21} | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{22} | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{23} | 0 | 0 | 0 | 9.85×10^{-09} | 0 | 0 |
| f_{24} | $2.89 \times 10^{+01}$ | $2.90 \times 10^{+01}$ | $2.89 \times 10^{+01}$ | $2.90 \times 10^{+01}$ | $2.90 \times 10^{+01}$ | $2.89 \times 10^{+01}$ |
| f_{25} | 8.01×10^{-01} | 7.10×10^{-01} | 4.86×10^{-01} | 6.96×10^{-01} | 8.49×10^{-01} | 7.69×10^{-01} |
| f_{26} | $2.99 \times 10^{+00}$ | $2.99 \times 10^{+00}$ | $2.72 \times 10^{+00}$ | $2.66 \times 10^{+00}$ | $2.99 \times 10^{+00}$ | $2.99 \times 10^{+00}$ |

The optimal value is in bold.

It is clear from Tables 2 and 3 and Figure 8 that the setting of the parameter Pa1(10, 9) can obtain the optimal value in most cases, but to further verify the validity of the parameter settings, Table 4 gives the Friedman and Wilcoxon results of 12 groups of parameter settings, listing the mean rank of DPTCFA under different parameters. The Friedman test and Wilcoxon test can be used to represent the algorithm difference for problem solving the benchmark function to verify the algorithm stability. The larger the test result, the greater the difference in the algorithm’s ability to solve the problem, and the worse the algorithm stability [38]. As seen, the mean rank of Pa1(10, 9) is the smallest, so Pa1(10, 9) is the best choice for the inspection test. Based on the non-parametric test, Pa1(10, 9) was set to (N_0 , N_c) in the next experiment.

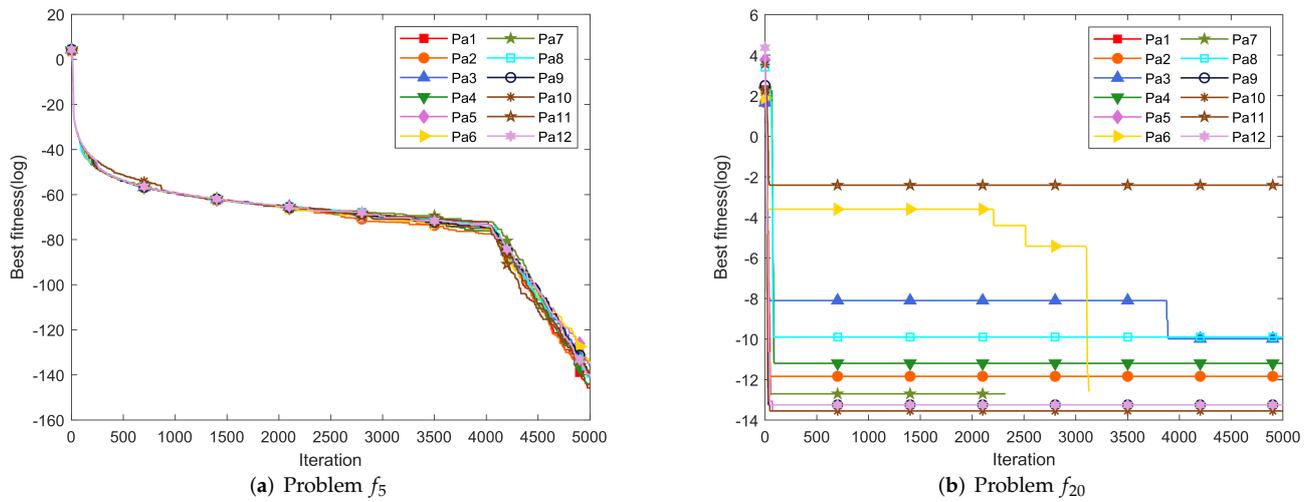


Figure 8. Search curves of DPTCFA with different parameters f_5 and f_{20} ($D = 30$).

Table 3. Results of the parameter analysis of DPTCFA ($N_0 = 8$ and $N_0 = 7$).

| Problems | Pa7 | Pa8 | Pa9 | Pa10 | Pa11 | Pa12 |
|----------|--|--|--|--|---|-------------------------|
| | (8, 7) Mean | (8, 6) Mean | (8, 5) Mean | (7, 6) Mean | (7, 5) Mean | (7, 4) Mean |
| f_1 | 3.52×10^{-136} | 1.13×10^{-144} | 3.33×10^{-138} | 2.71×10^{-144} | 6.32×10^{-140} | 3.07×10^{-137} |
| f_2 | 8.83×10^{-158} | 9.75×10^{-159} | 4.45×10^{-158} | 1.78×10^{-147} | 3.87×10^{-164} | 4.47×10^{-158} |
| f_3 | 8.88×10^{-142} | 9.84×10^{-132} | 8.22×10^{-138} | 3.08×10^{-136} | 8.07×10^{-138} | 1.09×10^{-141} |
| f_4 | 0 | 0 | 0 | 0 | 0 | 0 |
| f_5 | 5.69×10^{-142} | 1.43×10^{-141} | 6.39×10^{-141} | 3.21×10^{-141} | 1.23×10^{-140} | 1.88×10^{-141} |
| f_6 | 3.44×10^{-72} | 3.27×10^{-69} | 5.78×10^{-67} | 4.11×10^{-64} | 7.32×10^{-72} | 1.42×10^{-69} |
| f_7 | 0 | 0 | 0 | 0 | 0 | 0 |
| f_8 | 0 | 0 | 0 | 0 | 0 | 0 |
| f_9 | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{10} | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{11} | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{12} | 9.03×10^{-138} | 7.69×10^{-141} | 1.05×10^{-131} | 1.98×10^{-143} | 1.44×10^{-148} | 5.26×10^{-138} |
| f_{13} | 1.24×10^{-72} | 7.34×10^{-71} | 1.18×10^{-73} | 3.45×10^{-70} | 3.62×10^{-72} | 5.80×10^{-68} |
| f_{14} | 0 | 0 | 0 | 6.61×10^{-03} | 0 | 0 |
| f_{15} | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{16} | 4.12×10^{-07} | 2.23×10^{-06} | 1.20×10^{-06} | 7.58×10^{-07} | 2.94×10^{-06} | 5.68×10^{-07} |
| f_{17} | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{18} | 4.48×10^{-06} | 8.55×10^{-05} | 6.32×10^{-04} | 3.70×10^{-05} | 2.14×10^{-04} | 3.73×10^{-04} |
| f_{19} | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{20} | 0 | 1.26×10^{-10} | 5.68×10^{-14} | 2.84×10^{-14} | 3.90×10^{-03} | 5.68×10^{-14} |
| f_{21} | 0 | 0 | 1.19×10^{-07} | 0 | 0 | 0 |
| f_{22} | 0 | 0 | 0 | 0 | 0 | 5.59×10^{-12} |
| f_{23} | 0 | 0 | 0 | 0 | 0 | 0 |
| f_{24} | $2.89 \times 10^{+01}$ | $2.89 \times 10^{+01}$ | $2.89 \times 10^{+01}$ | $2.90 \times 10^{+01}$ | $2.90 \times 10^{+01}$ | $2.90 \times 10^{+01}$ |
| f_{25} | 5.88×10^{-01} | 8.53×10^{-01} | 4.15×10^{-01} | 3.36×10^{-01} | 4.68×10^{-01} | 7.44×10^{-01} |
| f_{26} | $2.99 \times 10^{+00}$ | $2.99 \times 10^{+00}$ | $2.99 \times 10^{+00}$ | $2.99 \times 10^{+00}$ | $3.00 \times 10^{+00}$ | $3.00 \times 10^{+00}$ |

The optimal value is in bold.

Table 4. The Friedman and Wilcoxon results in different parameters.

| Parameters | Friedman | Wilcoxon | Rank |
|------------|-------------|-------------|----------|
| Pa1(10, 9) | 6.08 | 3.42 | 1 |
| Pa2(10, 8) | 6.88 | 4.92 | 4 |
| Pa3(10, 7) | 6.38 | 4.81 | 3 |
| Pa4(9, 8) | 5.77 | 4.73 | 7 |
| Pa5(9, 7) | 6.83 | 6.12 | 2 |
| Pa6(9, 6) | 6.27 | 6.00 | 6 |
| Pa7(8, 7) | 5.46 | 5.73 | 5 |
| Pa8(8, 6) | 6.67 | 7.38 | 8 |
| Pa9(8, 5) | 6.77 | 7.85 | 9 |
| Pa10(7, 6) | 6.77 | 8.35 | 10 |
| Pa11(7, 5) | 6.79 | 8.92 | 11 |
| Pa12(7, 4) | 7.33 | 9.77 | 12 |

The optimal result is in bold.

4.3. Algorithm Comparison

In the second part of the experiment, DPTCFA was compared with some other FA variants. The considered algorithms are listed below:

- ▶ Standard FA (FA) [10].
- ▶ FA with chaos (CFA) [15].
- ▶ FA with adaptive parameter strategy (ApFA) [39].
- ▶ Random attraction for FA (RaFA) [16].
- ▶ Neighborhood attraction for FA (NaFA) [17].
- ▶ Modified neighborhood search for FA (MFANS) [18].
- ▶ The proposed algorithm (DPTCFA).

For the above seven algorithms, the population size N and the maximum iteration number $MaxIt$ were set to 30 and 5000, respectively. For the standard FA and ApFA, the parameters α , β_0 , and γ were set to 0.5, 1.0, and $\frac{1}{I^2}$, respectively. The parameter settings of CFA, RaFA, and NaFA can be found in the relevant literature [15–17]. In MFANS, the neighborhood size k and the preset probability p_{ns} were set to 2 and 0.05, respectively. N_0 , N_c , k , and p_{ns} were respectively set to 10, 9, 6, and 0.05 in DPTCFA. Finally, each algorithm was run 30 times in the same environment, and the average optimal fitness value was recorded. Table 5 shows the comparison results of DPTCFA and other FA variants on $D = 30$. The “Mean” in the table represents the average optimal fitness value, and the optimal results are indicated in bold. Obviously, the proposed algorithm (DPTCFA) obtained better results on most benchmark functions. Especially in 14 functions including $f_4, f_7, f_9-f_{11}, f_{14}, f_{15}$, and $f_{17}-f_{23}$, the function optimal value was obtained. On the function f_{24} , the result of DPTCFA was slightly insufficient compared to ApFA. In addition to the above functions, DPTCFA received a poor result on the composite functions f_{25} and f_{26} . However, the CFA using the chaos map achieved better results. Furthermore, CFA, ApFA, RaFA, NaFA, and MFANS achieved 6, 4, 4, 6, and 6 function optimal values, respectively. Due to space limitations, Figure 9 only shows the convergence images of DPTCFA and the other six comparison algorithms on some functions ($D = 30$).

Figure 9 gives the convergence images of the seven algorithms on the six test functions $f_2, f_5, f_6, f_{12}, f_{13}$, and f_{14} . The abscissa and ordinate respectively represent the iteration number and the logarithmic form of the optimal fitness value. In the image of the function f_2 , FA, CFA, ApFA, and RaFA fall into the local optimum, and finally obtain poor results. NaFA and MFANS only jump out of the local optimum after 4000 iterations, and the convergence speed is slow. DPTCFA shows better exploration ability in the global optimization stage (i.e., 1 to 4000 iterations), and the optimization ability level ranges from 10^0 to 10^{-100} . The local optimization stage enters after 4000 iterations, and the exploitation optimization ability level ranges from 10^{-100} to 10^{-150} . In the functions f_5, f_6, f_{12}, f_{13} , and f_{14} , CFA did not fall into the local optimal value, and it is sufficient to prove the effectiveness of

the chaotic strategy. On the function f_{14} , DPTCFA found the optimal value after about 500 iterations. It can be seen that the dual-population topology enhances the exploration ability. The convergence rate of NaFA on various functions was relatively slow; after 4000 iterations, this algorithm showed search oscillation, and the local optimal solution emerged. In comparison, the proposed approach considers the exploration and exploitation capabilities of FA. Therefore, DPTCFA achieved better results in 26 benchmark functions.

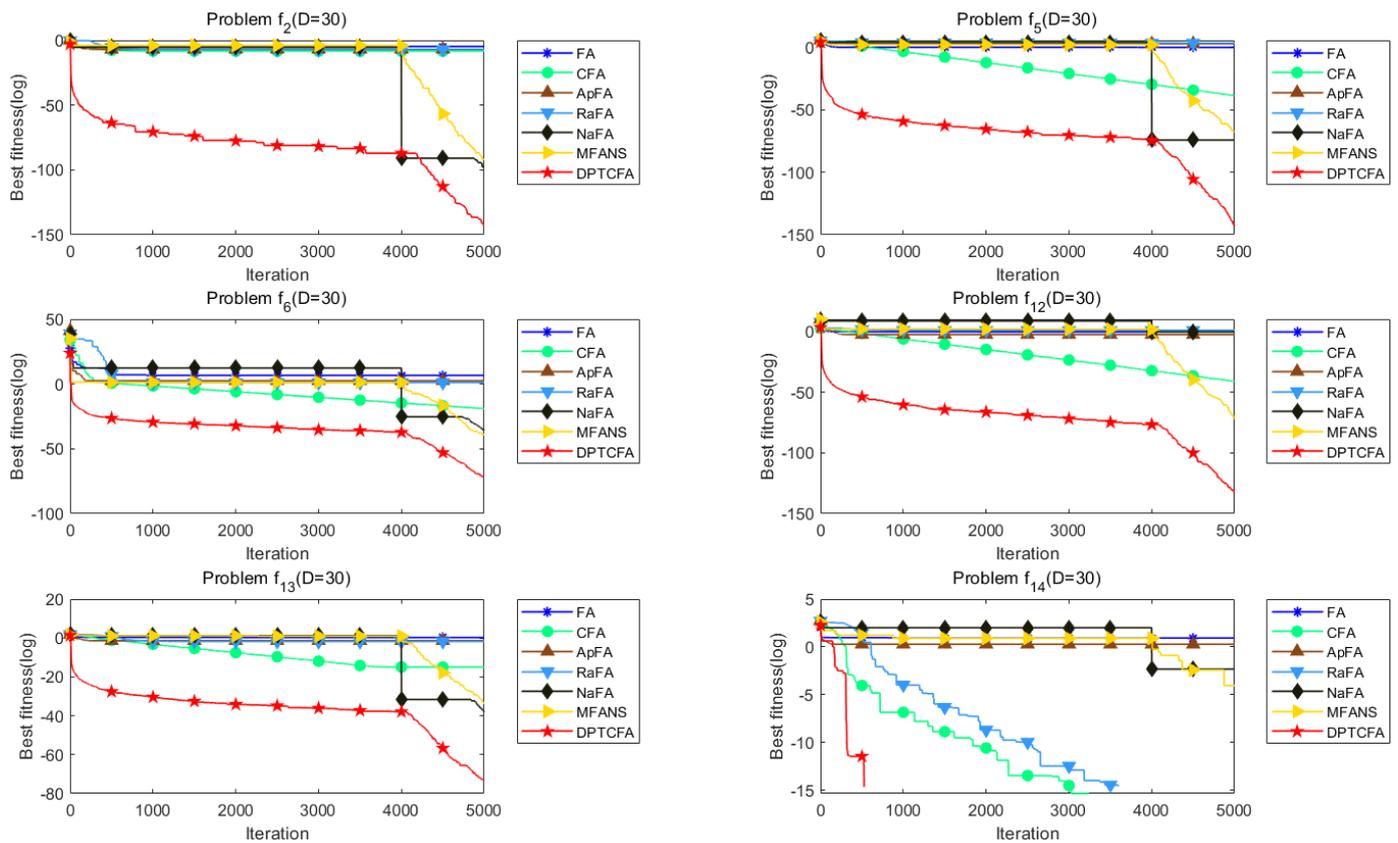


Figure 9. Search curves of DPTCFA and other FA variants ($D = 30$).

Table 5. Comparison of results obtained by DPTCFA and other FA variants ($D = 30$).

| Problems | FA Mean | CFA Mean | ApFA Mean | RaFA Mean | NaFA Mean | MFANS Mean | DPTCFA Mean |
|----------|------------------------|-------------------------|-------------------------|-------------------------|-------------------------|------------------------|-------------------------|
| f_1 | 4.79×10^{-01} | 7.27×10^{-43} | 1.72×10^{-88} | 4.83×10^{-40} | 1.91×10^{-60} | 1.78×10^{-63} | 7.14×10^{-134} |
| f_2 | 1.74×10^{-05} | 2.96×10^{-09} | 9.18×10^{-08} | 1.07×10^{-07} | 6.16×10^{-82} | 5.78×10^{-80} | 4.67×10^{-142} |
| f_3 | $3.87 \times 10^{+00}$ | 8.92×10^{-41} | 4.38×10^{-08} | 9.78×10^{-02} | 2.17×10^{-63} | 2.26×10^{-60} | 8.01×10^{-131} |
| f_4 | $2.23 \times 10^{+01}$ | 3.33×10^{-02} | $3.07 \times 10^{+03}$ | 6.67×10^{-02} | 0 | 0 | 0 |
| f_5 | 5.06×10^{-01} | 2.28×10^{-39} | $5.81 \times 10^{+02}$ | $1.85 \times 10^{+03}$ | 2.18×10^{-66} | 8.75×10^{-67} | 4.39×10^{-126} |
| f_6 | $6.77 \times 10^{+10}$ | $2.73 \times 10^{+00}$ | $3.36 \times 10^{+02}$ | $6.50 \times 10^{+01}$ | 8.93×10^{-32} | 1.37×10^{-31} | 1.26×10^{-65} |
| f_7 | 2.64×10^{-07} | 1.99×10^{-210} | 7.46×10^{-199} | 1.23×10^{-194} | 2.38×10^{-316} | 0 | 0 |
| f_8 | $8.51 \times 10^{+03}$ | $4.00 \times 10^{+03}$ | $9.92 \times 10^{+03}$ | $3.12 \times 10^{+03}$ | $8.00 \times 10^{+03}$ | 1.18×10^{-01} | 4.32×10^{-02} |
| f_9 | 2.12×10^{-02} | 3.78×10^{-03} | 5.00×10^{-01} | 7.39×10^{-04} | 0 | 0 | 0 |
| f_{10} | 1.77×10^{-07} | 0 | 0 | 0 | 1.35×10^{-06} | 5.32×10^{-06} | 0 |
| f_{11} | 3.24×10^{-02} | 1.05×10^{-42} | 4.94×10^{-85} | 4.50×10^{-40} | 0 | 0 | 0 |
| f_{12} | 4.42×10^{-01} | 1.10×10^{-41} | 7.16×10^{-01} | $3.01 \times 10^{+01}$ | $1.15 \times 10^{+00}$ | 1.37×10^{-02} | 7.13×10^{-128} |
| f_{13} | $1.46 \times 10^{+00}$ | 7.00×10^{-03} | 5.99×10^{-02} | 1.51×10^{-02} | 5.19×10^{-35} | 2.14×10^{-32} | 6.91×10^{-66} |
| f_{14} | $3.00 \times 10^{+01}$ | 0 | $3.07 \times 10^{+01}$ | 0 | 1.88×10^{-03} | 3.37×10^{-03} | 0 |
| f_{15} | 3.58×10^{-01} | 1.00×10^{-01} | 1.00×10^{-01} | 1.00×10^{-01} | 0 | 0 | 0 |
| f_{16} | 1.30×10^{-01} | 3.21×10^{-03} | 1.61×10^{-02} | 1.91×10^{-02} | 8.55×10^{-05} | 2.33×10^{-04} | 1.05×10^{-06} |
| f_{17} | $7.60 \times 10^{+01}$ | $4.08 \times 10^{+01}$ | $2.49 \times 10^{+01}$ | $2.32 \times 10^{+01}$ | 0 | 0 | 0 |

Table 5. Cont.

| Problems | FA Mean | CFA Mean | ApFA Mean | RaFA Mean | NaFA Mean | MFANS Mean | DPTCFA Mean |
|----------|------------------------|--|--|------------------------|------------------------|------------------------|------------------------|
| f_{18} | $1.00 \times 10^{+00}$ | $1.00 \times 10^{+00}$ | $1.00 \times 10^{+00}$ | $1.00 \times 10^{+00}$ | 3.20×10^{-05} | 6.33×10^{-04} | 0 |
| f_{19} | $1.00 \times 10^{+00}$ | $1.00 \times 10^{+00}$ | $1.00 \times 10^{+00}$ | $1.00 \times 10^{+00}$ | 0 | 2.81×10^{-16} | 0 |
| f_{20} | 1.40×10^{-02} | 0 | 2.84×10^{-15} | 9.47×10^{-15} | $1.23 \times 10^{+00}$ | $1.83 \times 10^{+01}$ | 0 |
| f_{21} | 1.15×10^{-05} | 0 | 0 | 1.18×10^{-16} | 1.22×10^{-03} | 3.31×10^{-03} | 0 |
| f_{22} | 5.24×10^{-06} | 0 | 0 | 0 | 4.63×10^{-04} | 1.45×10^{-03} | 0 |
| f_{23} | 8.63×10^{-06} | 0 | 0 | 0 | 3.76×10^{-03} | 8.27×10^{-03} | 0 |
| f_{24} | $6.63 \times 10^{+01}$ | $4.05 \times 10^{+01}$ | $2.83 \times 10^{+01}$ | $1.55 \times 10^{+02}$ | $2.90 \times 10^{+01}$ | $2.90 \times 10^{+01}$ | $2.89 \times 10^{+01}$ |
| f_{25} | $7.93 \times 10^{+00}$ | 1.57×10^{-32} | $8.74 \times 10^{+00}$ | 3.57×10^{-03} | 8.75×10^{-01} | 8.54×10^{-01} | 6.72×10^{-01} |
| f_{26} | 4.67×10^{-02} | 1.37×10^{-32} | $3.36 \times 10^{+01}$ | 7.32×10^{-04} | $2.92 \times 10^{+00}$ | $2.97 \times 10^{+00}$ | $2.86 \times 10^{+00}$ |

The optimal value is in bold.

To further verify the effectiveness of the algorithm, in this work, DPTCFA and other FA variants were tested from two aspects.

(1) Increasing the problem dimension: In addition to the 30-dimension experiment, the 7 algorithms were also tested with a 50-dimension condition. Table 6 lists the experiment results of the 7 algorithms on the 50-dimension. Figure 10 is the convergence image of DPTCFA and other FA variants in the case of $D = 50$. Similar to the convergence of $D = 30$, the increase in dimensionality did not make the DPTCFA effect worse, which verifies that the proposed algorithm has strong stability.

Table 6. Comparison results obtained by DPTCFA and other FA variants ($D = 50$).

| Problems | FA Mean | CFA Mean | ApFA Mean | RaFA Mean | NaFA Mean | MFANS Mean | DPTCFA Mean |
|----------|------------------------|--|------------------------|------------------------|-------------------------|-------------------------|---|
| f_1 | $1.37 \times 10^{+00}$ | 1.82×10^{-42} | $2.75 \times 10^{+00}$ | 5.31×10^{-05} | 2.88×10^{-65} | 8.36×10^{-65} | 5.38×10^{-133} |
| f_2 | 4.87×10^{-02} | 5.04×10^{-09} | 1.30×10^{-07} | 7.00×10^{-08} | 3.24×10^{-83} | 9.12×10^{-79} | 1.09×10^{-145} |
| f_3 | $1.98 \times 10^{+01}$ | 6.71×10^{-40} | 3.43×10^{-02} | $6.97 \times 10^{+00}$ | 5.71×10^{-62} | 2.61×10^{-57} | 3.22×10^{-132} |
| f_4 | $2.83 \times 10^{+00}$ | 6.67×10^{-02} | $7.81 \times 10^{+03}$ | $5.43 \times 10^{+00}$ | 0 | 0 | 0 |
| f_5 | $2.69 \times 10^{+00}$ | 6.82×10^{-38} | $7.28 \times 10^{+03}$ | $9.54 \times 10^{+03}$ | 3.46×10^{-59} | 8.17×10^{-63} | 3.29×10^{-131} |
| f_6 | $5.22 \times 10^{+25}$ | $7.16 \times 10^{+01}$ | $6.38 \times 10^{+02}$ | $2.96 \times 10^{+02}$ | 2.46×10^{-32} | 1.56×10^{-32} | 1.33×10^{-65} |
| f_7 | 1.51×10^{-05} | 6.02×10^{-209} | 8.07×10^{-79} | 1.03×10^{-04} | 1.90×10^{-307} | 5.72×10^{-313} | 0 |
| f_8 | $1.28 \times 10^{+04}$ | $7.75 \times 10^{+03}$ | $1.75 \times 10^{+04}$ | $6.29 \times 10^{+03}$ | $1.51 \times 10^{+04}$ | 1.63×10^{-01} | 5.05×10^{-02} |
| f_9 | 3.76×10^{-02} | 1.48×10^{-03} | $1.67 \times 10^{+00}$ | 6.33×10^{-03} | 0 | 0 | 0 |
| f_{10} | 1.85×10^{-07} | 0 | 0 | 0 | 4.92×10^{-09} | 1.63×10^{-06} | 0 |
| f_{11} | 2.33×10^{-01} | 4.07×10^{-42} | $9.06 \times 10^{+00}$ | $1.65 \times 10^{+01}$ | 0 | 0 | 0 |
| f_{12} | $1.40 \times 10^{+00}$ | 6.93×10^{-41} | $2.27 \times 10^{+02}$ | $1.52 \times 10^{+02}$ | $1.48 \times 10^{+01}$ | $1.16 \times 10^{+00}$ | 4.14×10^{-131} |
| f_{13} | $4.37 \times 10^{+00}$ | 3.66×10^{-02} | 3.76×10^{-01} | 4.09×10^{-01} | 1.02×10^{-35} | 9.50×10^{-33} | 4.85×10^{-67} |
| f_{14} | $6.54 \times 10^{+01}$ | 0 | $7.35 \times 10^{+01}$ | 1.48×10^{-17} | 3.56×10^{-03} | 6.97×10^{-03} | 0 |
| f_{15} | 8.45×10^{-01} | 1.00×10^{-01} | 1.00×10^{-01} | 6.73×10^{-01} | 0 | 0 | 0 |
| f_{16} | 7.14×10^{-01} | 1.08×10^{-02} | 7.02×10^{-02} | 6.06×10^{-02} | 7.41×10^{-05} | 2.23×10^{-04} | 9.76×10^{-07} |
| f_{17} | $1.84 \times 10^{+02}$ | $8.71 \times 10^{+01}$ | $6.71 \times 10^{+01}$ | $4.72 \times 10^{+01}$ | 0 | 0 | 0 |
| f_{18} | $1.00 \times 10^{+00}$ | $1.00 \times 10^{+00}$ | $1.00 \times 10^{+00}$ | $1.00 \times 10^{+00}$ | 4.21×10^{-05} | 1.59×10^{-03} | 0 |
| f_{19} | $1.00 \times 10^{+00}$ | $1.00 \times 10^{+00}$ | $1.00 \times 10^{+00}$ | $1.00 \times 10^{+00}$ | 0 | 6.21×10^{-12} | 0 |
| f_{20} | 2.64×10^{-01} | 0 | 2.84×10^{-15} | 2.84×10^{-15} | $2.10 \times 10^{+00}$ | $3.92 \times 10^{+01}$ | 0 |
| f_{21} | 1.90×10^{-05} | 0 | 0 | 0 | 1.18×10^{-03} | 3.90×10^{-03} | 0 |
| f_{22} | 5.30×10^{-06} | 0 | 0 | 0 | 1.00×10^{-03} | 3.12×10^{-03} | 0 |
| f_{23} | 2.21×10^{-05} | 0 | 0 | 0 | 3.03×10^{-03} | 8.03×10^{-03} | 0 |
| f_{24} | $1.42 \times 10^{+02}$ | $7.37 \times 10^{+01}$ | $9.18 \times 10^{+01}$ | $6.97 \times 10^{+02}$ | $4.90 \times 10^{+01}$ | $4.90 \times 10^{+01}$ | $4.89 \times 10^{+01}$ |
| f_{25} | $9.12 \times 10^{+00}$ | 4.15×10^{-03} | $1.11 \times 10^{+01}$ | 6.25×10^{-01} | $1.07 \times 10^{+00}$ | $1.16 \times 10^{+00}$ | 8.09×10^{-01} |
| f_{26} | 1.48×10^{-01} | 1.39×10^{-32} | $8.99 \times 10^{+01}$ | $1.08 \times 10^{+01}$ | $4.91 \times 10^{+00}$ | $4.99 \times 10^{+00}$ | $4.99 \times 10^{+00}$ |

The optimal value is in bold.

(2) Friedman and Wilcoxon test: We used the 30-dimension and 50-dimension experiment results for a non-parametric test to obtain the algorithm mean rank. Tables 7 and 8 are the non-parametric test results in the 30-dimension and 50-dimension conditions, respectively. As seen, the experiment results of DPTCFA were almost the same in the case of $D = 50$ and $D = 30$, and they all achieved good results. The mean rank of the Friedman test and Wilcoxon test also confirmed the stability of DPTCFA.

Table 7. Friedman and Wilcoxon test results in different algorithms ($D = 30$).

| | FA | CFA | ApFA | RaFA | NaFA | MFANS | DPTCFA |
|----------|------|------|------|------|------|-------|-------------|
| Friedman | 5.69 | 3.48 | 5.38 | 4.83 | 3.29 | 3.62 | 1.71 |
| Wilcoxon | 5.58 | 3.15 | 5.35 | 5.04 | 3.08 | 3.62 | 2.19 |
| Rank | 7 | 5 | 2 | 6 | 4 | 3 | 1 |

The optimal result is in bold.

Table 8. The Friedman and Wilcoxon results in different algorithms ($D = 50$).

| | FA | CFA | ApFA | RaFA | NaFA | MFANS | DPTCFA |
|----------|------|------|------|------|------|-------|-------------|
| Friedman | 5.69 | 3.48 | 5.38 | 4.83 | 3.29 | 3.62 | 1.71 |
| Wilcoxon | 5.58 | 3.15 | 5.35 | 5.04 | 3.08 | 3.62 | 2.19 |
| Rank | 7 | 5 | 2 | 6 | 4 | 3 | 1 |

The optimal result is in bold.

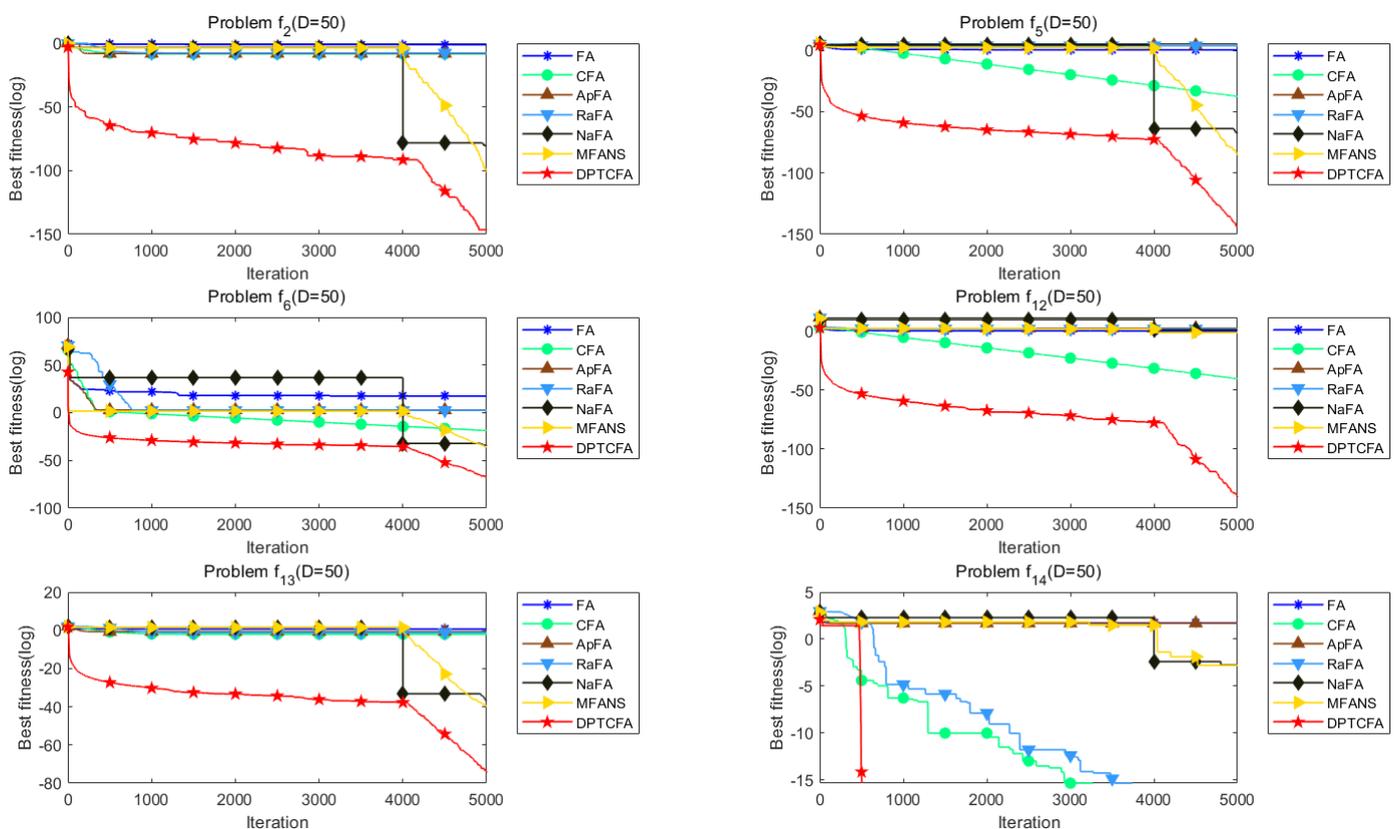


Figure 10. Search curves of DPTCFA and other FA variants ($D = 50$).

4.4. Strategy Evaluation

In this part, the corresponding work is mainly carried out from the following two aspects: **(1) Evaluation of the SN topology population optimization ability:** In DPTCFA, the SN and RN topology populations are used. The effectiveness of the RN topology was confirmed in [15]. Therefore, to ensure the SN topology population optimization

ability, we recorded the times of the SN and RN topology populations achieving the global optimal value in the population evolution process. **(2) Evaluation of the effectiveness of the staged balance mechanism:** In the proposed algorithm, the optimization is divided into two stages, i.e., the global and local optimization stages. Therefore, we maintained the global optimization stage the same, and verified the effectiveness of the NMSM in the local optimization stage.

(1) Evaluation of the SN topology population optimization ability

From the above description, the proposed algorithm DPTCFA was divided into two stages: global optimization and local optimization stages. Among them, the iteration numbers in the global optimization stage were 4000. In the global optimization stage, the SN and RN topology populations co-evolve. In order to evaluate the optimization ability of the SN topology population, Table 9 gives the statistical times of the SN and RN topology populations achieving the better value during the 4000 iterations in the global optimization stage. As seen, the SN topology population was better than or equal to the RN topology population on the 17 benchmark functions. Therefore, the global optimization ability of the SN topology population was stronger than the RN topology population. However, on the benchmark functions, such as f_1 – f_3 , f_8 , f_{10} , and f_{12} – f_{14} , the optimization ability of the SN topology population is bad. Thus, the dual-population topology coevolution mechanism effectively compensates for the defects of the SN and RN topology.

Table 9. The statistical times of two populations achieving the optimal value.

| Problems | SN Topology Population | RN Topology Population |
|----------|------------------------|------------------------|
| f_1 | 58 | 3942 |
| f_2 | 526 | 3474 |
| f_3 | 902 | 3098 |
| f_4 | 4000 | 3992 |
| f_5 | 2352 | 1468 |
| f_6 | 2440 | 1560 |
| f_7 | 3330 | 670 |
| f_8 | 18 | 3982 |
| f_9 | 4000 | 4000 |
| f_{10} | 1738 | 2262 |
| f_{11} | 4000 | 3990 |
| f_{12} | 734 | 4000 |
| f_{13} | 966 | 3034 |
| f_{14} | 592 | 3408 |
| f_{15} | 4000 | 3900 |
| f_{16} | 3954 | 46 |
| f_{17} | 3968 | 4000 |
| f_{18} | 4000 | 4000 |
| f_{19} | 4000 | 3390 |
| f_{20} | 4000 | 0 |
| f_{21} | 4000 | 0 |
| f_{22} | 4000 | 0 |
| f_{23} | 4000 | 3944 |
| f_{24} | 3994 | 6 |
| f_{25} | 4000 | 0 |
| f_{26} | 3998 | 2 |
| Total | 17 | 11 |

The optimal value is in bold.

(2) Evaluation of the effectiveness of the staged balance mechanism

For the evaluation of the staged balance mechanism’s effectiveness, since the global optimization ability has been evaluated in the previous work, it was only needed to evaluate the NMSM next. The comparison algorithm is listed as follows:

- ▶ Not contain NMSM (Not-NMSM).
- ▶ Contain NMSM (DPTCFA).

Table 10 shows the experiment results of the two algorithms ($D = 30$). Non-NMSM achieved better results in 16 benchmark functions; however, DPTCFA achieved better results in 23 benchmark functions. In terms of quantity, NMSM effectively improved the solution accuracy to achieve better results.

Table 10. Comparison of results obtained by DPTCFA and non-NMSM ($D = 30$).

| Problems | Not-NMSM | DPTCFA |
|----------|--|---|
| f_1 | 2.91×10^{-84} | 1.13×10^{-144} |
| f_2 | 2.82×10^{-88} | 9.75×10^{-159} |
| f_3 | 1.52×10^{-78} | 9.84×10^{-132} |
| f_4 | 0 | 0 |
| f_5 | 5.95×10^{-77} | 1.43×10^{-141} |
| f_6 | 1.50×10^{-39} | 3.27×10^{-69} |
| f_7 | 0 | 0 |
| f_8 | 3.91×10^{-02} | 2.43×10^{-02} |
| f_9 | 0 | 0 |
| f_{10} | 0 | 0 |
| f_{11} | 0 | 0 |
| f_{12} | 5.99×10^{-87} | 7.69×10^{-141} |
| f_{13} | 3.53×10^{-41} | 7.34×10^{-71} |
| f_{14} | 0 | 0 |
| f_{15} | 0 | 0 |
| f_{16} | 2.23×10^{-06} | 1.73×10^{-06} |
| f_{17} | 0 | 0 |
| f_{18} | $1.00 \times 10^{+00}$ | 8.55×10^{-05} |
| f_{19} | 0 | 0 |
| f_{20} | 2.84×10^{-14} | 1.26×10^{-10} |
| f_{21} | 0 | 0 |
| f_{22} | 0 | 0 |
| f_{23} | 0 | 0 |
| f_{24} | $2.89 \times 10^{+01}$ | $2.89 \times 10^{+01}$ |
| f_{25} | 4.94×10^{-01} | 8.53×10^{-01} |
| f_{26} | $2.81 \times 10^{+00}$ | $2.99 \times 10^{+00}$ |
| Total | 16 | 23 |

The optimal value is in bold.

5. Conclusions

In this paper, an enhanced FA with dual-population topology coevolution (DPTCFA) is proposed. In DPTCFA, a two-neighborhood topology is mainly used for optimization; it conforms to the evolutionary law of fireflies in nature, and enhances the population diversity and the global search ability. In addition, the main contributions of this paper are summarized as follows: First, the chaos map strategy is used in the SN topology population to reduce the parameter sensitivity. A new distance strategy for calculating attractiveness is designed based on the dimension difference in the RN topology population. Second, the diversity neighborhood enhanced search strategy is improved for the firefly position update. Third, NMSM is adopted to enhance the algorithm exploitation, so as to obtain a balance between the algorithm exploration and exploitation. In order to verify the performance of DPTCFA, 26 benchmark functions are used to test this algorithm and other FA variants. The experimental results show that the effect of DPTCFA is significantly better than other algorithms. Finally, ablation experiments were performed for the evaluation of the SN topology population optimization ability and NMSM. The experiment results verify the effectiveness of DPTCFA. Furthermore, this paper also compares other state-of-the-art algorithms other than FAs, such as PSO, DE, and their variants, and the experiment

results show that the performance of the proposed DPTCFA algorithm far exceeds these algorithms, making the conclusion more persuasive.

However, there are still some shortcomings in this work: (1) Compared with ApFA, RaFA, and NaFA, DPTCFA has a longer running time, but it is much better than FA and CFA. (2) On most functions, the effect of DPTCFA is better than the other algorithms, but on the composite functions f_{25} and f_{26} , CFA achieves the best results. The above shortcomings will be the focus of follow-up work, and the application of theoretical research (DPTCFA) to practical optimization problems is of great significance.

Author Contributions: Conceptualization, W.L. (Wei Li); Data curation, W.L. (Wangdong Li); Formal analysis, Y.H.; Funding acquisition, W.L. (Wei Li); Resources, W.L. (Wei Li); Software, W.L. (Wangdong Li); Validation, Y.H.; Writing—original draft, W.L. (Wangdong Li); Writing—review & editing, W.L. (Wei Li). All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant Nos. 62066019, 61903089), the Natural Science Foundation of Jiangxi Province (Grant Nos. 20202BABL202020, 20202BAB202014), and the National Key Research and Development Program of China (Grant No. 2020YFB1713700).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Abdel-Basset, M.; Mohamed, R.; Chakraborty, R.K.; Ryan, M.J. IEGA: An improved elitism-based genetic algorithm for task scheduling problem in fog computing. *Int. J. Intell. Syst.* **2021**, *36*, 4592–4631. [\[CrossRef\]](#)
2. Abdollahzadeh, B.; Soleimani Gharehchopogh, F.; Mirjalili, S. Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. *Int. J. Intell. Syst.* **2021**, *36*, 5887–5958. [\[CrossRef\]](#)
3. Yang, X.S. Nature-inspired optimization algorithms: Challenges and open problems. *J. Comput. Sci.* **2020**, *46*, 101104. [\[CrossRef\]](#)
4. Mosayebi, M.; Sodhi, M.; Wettergren, T.A. The traveling salesman problem with job-times (tsjp). *Comput. Oper. Res.* **2021**, *129*, 105226. [\[CrossRef\]](#)
5. Gao, W.; Tang, Q.; Ye, B.; Yang, Y.; Yao, J. An enhanced heuristic ant colony optimization for mobile robot path planning. *Soft Comput.* **2020**, *24*, 6139–6150. [\[CrossRef\]](#)
6. Tang, H.; Chen, R.; Li, Y.; Peng, Z.; Guo, S.; Du, Y. Flexible job-shop scheduling with tolerated time interval and limited starting time interval based on hybrid discrete PSO-SA: An application from a casting workshop. *Appl. Soft Comput.* **2019**, *78*, 176–194. [\[CrossRef\]](#)
7. Wang, F.; Li, Y.; Zhou, A.; Tang, K. An estimation of distribution algorithm for mixed-variable newsvendor problems. *IEEE Trans. Evol. Comput.* **2019**, *24*, 479–493. [\[CrossRef\]](#)
8. Li, W.; Meng, X.; Huang, Y.; Fu, Z.H. Multipopulation cooperative particle swarm optimization with a mixed mutation strategy. *Inf. Sci.* **2020**, *529*, 179–196. [\[CrossRef\]](#)
9. Li, W.; Xia, L.; Huang, Y.; Mahmoodi, S. An ant colony optimization algorithm with adaptive greedy strategy to optimize path problems. *J. Ambient. Intell. Humaniz. Comput.* **2021**, *13*, 1557–1571. [\[CrossRef\]](#)
10. Yang, X.S. Firefly algorithms for multimodal optimization. In *International Symposium on Stochastic Algorithms*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 169–178.
11. Mohiz, M.J.; Baloch, N.K.; Hussain, F.; Saleem, S.; Zikria, Y.B.; Yu, H. Application Mapping Using Cuckoo Search Optimization With Lévy Flight for NoC-Based System. *IEEE Access* **2021**, *9*, 141778–141789. [\[CrossRef\]](#)
12. Xu, X.; Hao, J.; Zheng, Y. Multi-objective artificial bee colony algorithm for multi-stage resource leveling problem in sharing logistics network. *Comput. Ind. Eng.* **2020**, *142*, 106338. [\[CrossRef\]](#)
13. Yang, X.S. *Nature-Inspired Metaheuristic Algorithms*; Luniver Press: Frome, UK, 2010.
14. Fister, I., Jr.; Yang, X.S.; Fister, I.; Brest, J. Memetic firefly algorithm for combinatorial optimization. *arXiv* **2012**, arXiv:1204.5165.
15. Gandomi, A.H.; Yang, X.S.; Talatahari, S.; Alavi, A.H. Firefly algorithm with chaos. *Commun. Nonlinear Sci. Numer. Simul.* **2013**, *18*, 89–98. [\[CrossRef\]](#)
16. Wang, H.; Wang, W.; Sun, H.; Rahnamayan, S. Firefly algorithm with random attraction. *Int. J. Bio-Inspired Comput.* **2016**, *8*, 33–41. [\[CrossRef\]](#)
17. Wang, H.; Wang, W.; Zhou, X.; Sun, H.; Zhao, J.; Yu, X.; Cui, Z. Firefly algorithm with neighborhood attraction. *Inf. Sci.* **2017**, *382*, 374–387. [\[CrossRef\]](#)
18. Yu, G. A modified firefly algorithm based on neighborhood search. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e6066. [\[CrossRef\]](#)

19. Wang, F.; Liao, F.; Li, Y.; Wang, H. A new prediction strategy for dynamic multi-objective optimization using Gaussian Mixture Model. *Inf. Sci.* **2021**, *580*, 331–351. [[CrossRef](#)]
20. Gu, Q.; Chen, S.; Jiang, S.; Xiong, N. Improved strength Pareto evolutionary algorithm based on reference direction and coordinated selection strategy. *Int. J. Intell. Syst.* **2021**, *36*, 4693–4722. [[CrossRef](#)]
21. Yang, X.S. Multiobjective firefly algorithm for continuous optimization. *Eng. Comput.* **2013**, *29*, 175–184. [[CrossRef](#)]
22. Lv, L.; Zhao, J.; Wang, J.; Fan, T. Multi-objective firefly algorithm based on compensation factor and elite learning. *Future Gener. Comput. Syst.* **2019**, *91*, 37–47. [[CrossRef](#)]
23. Marichelvam, M.K.; Prabaharan, T.; Yang, X.S. A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems. *IEEE Trans. Evol. Comput.* **2013**, *18*, 301–305. [[CrossRef](#)]
24. Karthikeyan, S.; Asokan, P.; Nickolas, S.; Page, T. A hybrid discrete firefly algorithm for solving multi-objective flexible job shop scheduling problems. *Int. J.-Bio-Inspired Comput.* **2015**, *7*, 386–401. [[CrossRef](#)]
25. Li, W.; Sun, B.; Huang, Y.; Mahmoodi, S. Adaptive particle swarm optimization using scale-free network topology. *J. Netw. Intell.* **2021**, *6*, 500–517.
26. Li, X. Niching without niching parameters: Particle swarm optimization using a ring topology. *IEEE Trans. Evol. Comput.* **2009**, *14*, 150–169.
27. Galántai, A. A convergence analysis of the Nelder–Mead simplex method. *Acta Polytech. Hung* **2021**, *18*, 93–105. [[CrossRef](#)]
28. Yang, X.S. Firefly algorithm, stochastic test functions and design optimisation. *Int. J. Bio-Inspired Comput.* **2010**, *2*, 78–84. [[CrossRef](#)]
29. Li, W.; Sun, B.; Huang, Y.; Mahmoodi, S. Adaptive complex network topology with fitness distance correlation framework for particle swarm optimization. *Int. J. Intell. Syst.* **2021**. [[CrossRef](#)]
30. Barabási, A.L. Scale-free networks: A decade and beyond. *Science* **2009**, *325*, 412–413. [[CrossRef](#)]
31. Wang, H.; Wang, W.; Cui, L.; Sun, H.; Zhao, J.; Wang, Y.; Xue, Y. A hybrid multi-objective firefly algorithm for big data optimization. *Appl. Soft Comput.* **2018**, *69*, 806–815. [[CrossRef](#)]
32. Xue, Y.; Xue, B.; Zhang, M. Self-adaptive particle swarm optimization for large-scale feature selection in classification. *ACM Trans. Knowl. Discov. Data (TKDD)* **2019**, *13*, 1–27. [[CrossRef](#)]
33. Xue, Y.; Tang, T.; Pang, W.; Liu, A.X. Self-adaptive parameter and strategy based particle swarm optimization for large-scale feature selection problems with multiple classifiers. *Appl. Soft Comput.* **2020**, *88*, 106031 [[CrossRef](#)]
34. Yu, G.; Wang, H.; Zhou, H.; Zhao, S.; Wang, Y. An efficient firefly algorithm based on modified search strategy and neighborhood attraction. *Int. J. Intell. Syst.* **2021**, *36*, 4346–4363. [[CrossRef](#)]
35. Nelder, J.A.; Mead, R. A simplex method for function minimization. *Comput. J.* **1965**, *7*, 308–313. [[CrossRef](#)]
36. Wang, H.; Sun, H.; Li, C.; Rahnamayan, S.; Pan, J.S. Diversity enhanced particle swarm optimization with neighborhood search. *Inf. Sci.* **2013**, *223*, 119–135. [[CrossRef](#)]
37. Jamil, M.; Yang, X.S. A literature survey of benchmark functions for global optimisation problems. *Int. J. Math. Model. Numer. Optim.* **2013**, *4*, 150–194. [[CrossRef](#)]
38. García, S.; Fernández, A.; Luengo, J.; Herrera, F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Inf. Sci.* **2010**, *180*, 2044–2064. [[CrossRef](#)]
39. Wang, H.; Zhou, X.; Sun, H.; Yu, X.; Zhao, J.; Zhang, H.; Cui, L. Firefly algorithm with adaptive control parameters. *Soft Comput.* **2017**, *21*, 5091–5102. [[CrossRef](#)]