# Rafflesia Optimization Algorithm Applied in the Logistics Distribution Centers Location Problem

Jeng-Shyang Pan[1, 2], Zonglin Fu[1], Chia-Cheng Hu[3], Pei-Wei Tsai[4], Shu-Chuan Chu[1*]

[1] College of Computer Science and Engineering, Shandong University of Science and Technology, China
[2] Department of Information Management, Chaoyang University of Technology, Taiwan
[3] College of Artificial Intelligence, Yango University, China
[4] Department of Computer Science and Software Engineering, Swinburne University of Technology, Australia
jspan@cc.kuas.edu.tw, zonglin@sdust.edu.cn, cchu.chiachenghu@gmail.com, ptsai@swin.edu.au, scchu0803@gmail.com

## Abstract

Intelligent evolutionary algorithm is an important method to solve optimization problems. Most of their inspiration comes from the laws of nature and biology. This paper proposes a new intelligent evolutionary algorithm based on the life habits of Rafflesia, which is called Rafflesia Optimization Algorithm. It mainly consists of three stages: attracting insects, swallowing insects, and spreading seeds. In the first stage, the ROA algorithm performs the local search to find the optimal solution. In the second stage, it improves execution efficiency and solution accuracy by reducing the number of individuals. In the third stage, it performs the global search to jump out of the local optimal position. In the experimental part, this paper uses numerical functions (the CEC2013 benchmark function set) and practical application problems (the logistics distribution centers location problem) to test the performance of the ROA algorithm, and compares it with seven meta-heuristics algorithms. The experimental results prove the effectiveness and practicability of the ROA algorithm.

**Keywords:** Rafflesia optimization algorithm, Intelligent evolutionary algorithm, CEC2013, The logistics distribution centers location problem

## 1 Introduction

The development of science and technology has promoted the progress of society. At the same time, it brings many complex optimization problems. For example, transportation planning in mega-cities, logistics and transportation in multiple cities, high-precision medical testing, etc. As the problem size increases, traditional optimization techniques are gradually replaced by new techniques. Evolutionary computation is one of them. It is mainly inspired by phenomena and laws in the natural and biological worlds [1-2]. Moreover, it has stronger robustness, better self-learning and adaptability than traditional optimization algorithms [3].

Some classic optimization algorithms were first proposed. For example, Artificial Neural Network (ANN) algorithm [4], Simulated Annealing (SA) algorithm [5], Immune algorithm (IA) [6]. Early explorations provided a solid theoretical foundation for the subsequent research on many optimization algorithms. In 1992, Marco Dorigo proposed a discrete meta-heuristic intelligent optimization algorithm based on the foraging behavior of ants called Ant Colony Algorithm (ACO) [7]. After 1995, some continuous meta-heuristic intelligent optimization algorithms appeared one after another. Two classic algorithms were first proposed. One is the Particle Swarm Optimization (PSO) algorithm [8], which is inspired by the foraging behavior of birds. The other is the Differential Evolution (DE) algorithm [9], which is inspired by the biological evolution process. After these two algorithms were proposed, many researchers analyzed and improved them [10-11]. In 2006, D.Y. Sha and Cheng-Yu Hsu proposed a hybrid PSO algorithm, which used tabu search to improve the quality of the solution [12]. In 2008, A. Slowik and M. Bialko added an adaptive selection strategy of control parameters to the DE algorithm, and adopted it for neural network training [13].

In order to solve optimization problems in different fields, researchers have proposed many new optimization algorithms, such as Cat Swarm Optimization (CSO) algorithm [14], Firefly Algorithm (FA) [15], Fish Migration Optimization (FMO) algorithm [16], Bat Algorithm (BA) [17], Gray Wolf Optimization (GWO) algorithm [18], Moth-flame Optimization (MFO) algorithm [19], QUasi-Affine TRansformation Evolutionary (QUATRE) algorithm [20], Whale Optimization Algorithm (WOA) [21], and Butterfly Optimization Algorithm (BOA) [22], etc. There is also the Phasmatodea Population Evolution (PPE) algorithm which is recently proposed [23]. After these algorithms are proposed, they have successfully applied to path planning [24], feature selection [25-26], IIR system recognition [27], image segmentation [28-29], optimal power flow (OPF) [30], wireless sensor node positioning [31-32], engineering optimization [33-35] and other problems.

Although there are a large number of intelligent optimization algorithms, no one algorithm can solve all optimization problems. This phenomenon conforms to the Non-Free Lunch Theorem (NFL) [36]. An algorithm can show excellent results when it deals with a certain problem, nevertheless this does not mean that it has the same effect on other problems. The reason is that optimization problems in different fields have different coding methods. When they face the application problem, they may fall into the situation of local optimum or premature convergence [37-38]. Therefore, researchers continue to improve existing algorithms and

*Corresponding Author: Shu-Chuan Chu; E-mail: scchu0803@gmail.com
DOI: 10.53106/160792642022122307009

propose new algorithms [39-42] to solve optimization problems in different fields.

Based on the Non-Free Lunch Theorem (NFL), this paper proposes a new intelligent optimization algorithm, named the Rafflesia Optimization Algorithm (ROA). It is inspired by the characteristics of Rafflesia from flowering to seed propagation [43]. Rafflesia is a parasitic plant without roots and stems. It emits a carrion-like smell when it blooms to attract insects. Some insects that fly into the flower chamber come in contact with the anthers to help the Rafflesia complete pollination. Because Rafflesia has a unique flower chamber structure, some insects are trapped and die in it [44]. After the Rafflesia withers, it forms a large fruit containing tens of thousands of seeds. These seeds are brought to different places by various animals in different ways. Finally, a very small number of seeds survive in a suitable environment. Based on the above characteristics, we divide the ROA algorithm into three stages to implement, including, the stage of attracting insects, the stage of swallowing insects, and the stage of spreading seeds.

In the ROA algorithm, we use the method of replacing poor individuals and regularly deleting the worst individuals to reduce the impact of bad individuals on the population evolution. In addition, the method of individual update is not just dependent on moving to the optimal individual and moving randomly. We refer to the movement model of insects flapping their wings in nature. In the first stage of the ROA algorithm, a combination of autonomous flight and approaching the optimal individual is used to update the individual. In addition, in order to reduce the probability of falling into a local optimum, the search range of the third stage is a random range centered on the optimal individual. After that, we use the CEC2013 benchmark function set [45] to test the effect of the ROA algorithm on the numerical function. And its results are compared in the three dimensions of 10D, 30D and 50D with PSO, WOA, CSO, MFO, BA, FA, BOA. These results prove the convergence ability and optimization effect of the ROA algorithm. In addition, we also select two sets of cases of the logistics distribution centers location problem [46-47] to test the ability of the new algorithm to solve practical problems. Compared with other algorithms, the ROA algorithm has achieved good results in this application.

The rest of the paper is structured as follows: Section 2 introduces the preparation of the algorithm. Section 3 introduces the ROA algorithm in detail. Section 4 presents the test results and analyzes them. Section 5 describes the thoughts and results of the application of the ROA algorithm to the logistics distribution centers location problem. Finally, Section 6 summarizes the work of this paper and proposes future work directions.

## 2 Preliminaries

Rafflesia $F$ (the target Rafflesia) attracts insects by emitting scent (stage 1). Insects are affected by the natural environment when they fly to the Rafflesia $F$. Some insects die, and some insects fly to other flowers. At the same time, there are new insects attracted by Rafflesia $F$. Theoretically speaking, the distance between newly attracted insects and Rafflesia $F$ is equal or close to the distance between a certain insect in the population and Rafflesia $F$, which ensures that they are attracted to the same Rafflesia. In the first stage of the algorithm, we roughly divided insects into three categories. The first category is newly attracted insects. Their positions are updated using strategy 1. The second category is insects that no longer fly towards Rafflesia $F$. They are removed from the population. Their positions in the population is replaced by newly attracted insects. The third category is the insects that keep flying towards Rafflesia $F$. Their positions are updated using strategy 2. Below we first introduce the model rationale for the two strategies.

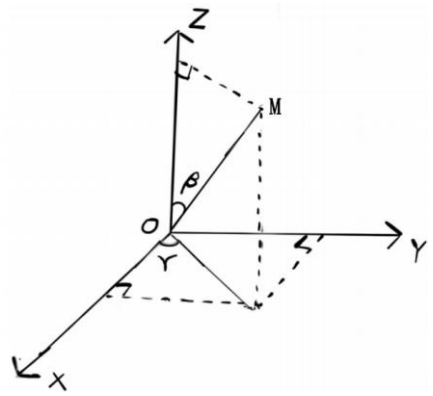



**Figure 1.** Example diagram in strategy 1

In strategy 1, we need to refer to the knowledge of coordinate solving in 3D space. Figure 1 is a standard three-dimensional coordinate system. As shown in Figure 1, there are point $O$ and point $M$ in the three-dimensional space. The coordinate of point $O$ is known as $(x_o, y_o, z_o)$. The modulus (length) of vector $\underset{OM}{\rightarrow}$ is $d'$. The angle between $\underset{OM}{\rightarrow}$ and the positive direction of the z-axis is $\beta'$. And the angle between the projection of $\underset{OM}{\rightarrow}$ on the $XOY$ plane and the positive direction of the x-axis is $\gamma'$. The coordinate $(x_M, y_M, z_M)$ of point $M$ is the unknown variable to be solved. Through analysis, it can be known that the coordinates of each dimension of point $M$ are equal to the projection length of $\underset{OM}{\rightarrow}$ on each coordinate axis. Therefore, the coordinates of point $M$ can be obtained by calculating the projection length of $\underset{OM}{\rightarrow}$ on each coordinate axis. The projection equations of $\underset{OM}{\rightarrow}$ on the $X$, $Y$, and $Z$ axes are:

$$\begin{cases} x_M = x_o + d' \times sin\beta' cos\gamma' \\ y_M = y_o + d' \times sin\beta' sin\gamma' \\ z_M = z_o + d' \times cos\beta' \end{cases}. \quad \textbf{(1)}$$

In strategy 2, the equation of insect flight speed is derived from the model in reference [48]. This model points out that when an insect flaps its wings, the movement of the wings can be decomposed into translation and rotation. The translation is described by the center coordinates of the wingtip. The rotation of the wing is characterized by the attack angle. The motion equation of the wing tip is:

$$\begin{cases} \xi = \frac{A}{2} cos(\omega_0 t + \theta) \\ \eta = \frac{B}{2} sin(\omega_1 t + \theta) \end{cases}. \quad \textbf{(2)}$$

The law of the attack angle is as follows:

$$\alpha = \alpha_0 (1 - sin(\omega_0 t + \theta + \varphi)). \quad \textbf{(3)}$$

Among them, $A$ is the amplitude of the wing during movement; $B$ is the lateral offset; $\omega_0$ represents the flapping frequency period; $\omega_1$ is the frequency period of the lateral flapping wing; $\theta$ represents the phase. The phase difference between translation and rotation is $\varphi$. $\alpha_0$ stands for the initial attack angle. *t* is the time.

In the physical model, the movement process takes the derivative of time to get the movement speed. Therefore, the translational velocity of insects can be obtained by taking the derivative of time t in Equation (2).

$$\underset{v_1}{\rightarrow} = \begin{cases} \frac{d\xi}{dt} = -\frac{A}{2}\omega_0 sin(\omega_0 t + \theta) \\ \frac{d\eta}{dt} = \frac{B}{2}\omega_1 cos(\omega_1 t + \theta) \end{cases}. \quad \textbf{(4)}$$

The rotation speed can be obtained by taking the derivative of time $t$ in Equation (3). Its sign is related to the rotation of the coordinate system in the physical model.

$$\underset{v_2}{\rightarrow} = -\frac{d\alpha}{dt} = \alpha_0\omega_0 cos(\omega_0 t + \theta + \varphi). \quad \textbf{(5)}$$

After a period of flight, the insect finds the target Rafflesia F. During the pollination process, some insects are trapped and died in the flower chamber of the Rafflesia (stage 2). It is understood that a Rafflesia probably "swallows" more than a dozen insects in its lifetime. In the ROA algorithm, the number of individuals in the population is reduced by simulating the characteristics of the Rafflesia "swallowing" insects. And when the ROA algorithm executes a certain number of iterations, the number of individuals is reduced by one.

The seeds of Rafflesia $F$ are taken by animals to different places (stage 3). This process can be abstracted as a model that dots are scattered everywhere with the position of the Rafflesia $F$ as the center. In addition, among the seeds that are taken everywhere, only a handful of seeds can survive in a suitable environment. Therefore, the optimal solution at this stage is represented by the seed with the highest survival probability. There is no doubt that whether it is the insect that finds Rafflesia $F$ first in the first stage, or the seed with the highest survival probability in the third stage, their adaptability must be the best in their population.

# 3 Rafflesia Optimization Algorithm

## 3.1 The Stage of Attracting Insects

In this stage, individuals with poor fitness indicate insects that no longer fly towards the target Rafflesia, and new individuals indicate insects newly attracted to the target Rafflesia. Moreover, individuals with poor fitness are replaced by newly added individuals. In order to ensure the stability of the population, their numbers are set to 1/3 of the population size. Other individuals occupying 2/3 of the population size are individuals with better fitness. And they represent insects that keep flying towards the target Rafflesia. After that, strategy 1 is used to calculate the location of the newly added individual. Strategy 2 is used to update the positions of the remaining individuals in the population.

### 3.1.1 Strategy 1

Individuals in the population are in a multi-dimensional space, and it is currently impossible to construct a suitable multi-dimensional coordinate system in reality. Therefore, the position of the newly added individual is calculated with reference to the mathematical model in Section 2. Assumptions:

1. Each dimension of the newly added individual is abstracted into a three-dimensional space for calculation. This three-dimensional space is composed of $k1$, $k$, and $k2$. $k1$ and $k2$ are two arbitrary dimensions perpendicular to $k$ in a multi-dimensional space. The x-axis in Figure 1 represents the $k\text{-}th$ dimension. the y-axis and z-axis represent $k1$ and $k2$, respectively, as shown in Figure 2.

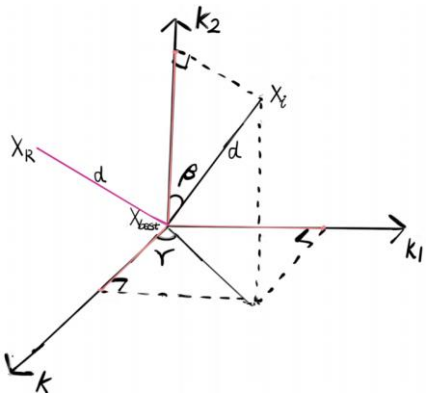



**Figure 2.** The model of the calculated dimensions

2. Point $O$ represents the optimal individual $X_{best}$, and point $M$ represents the newly added individual $X_i(i = 1,2,\dots,NP/3)$. Then the calculation method of the $k\text{-}th$ dimension of $X_i$ can be referred to $x_M$ in Equation (1). $NP$ is the population size.

3. The distance from $X_i$ to $X_{best}$ is equal to the distance from $X_R$ to $X_{best}$. $X_R$ is a random individual in the population.

Based on the above assumptions, the calculation equation for the position of the newly added individual is:

$$X_{i_k} = X_{best_k} + d \times sin\beta_k cos\gamma_k. \quad \textbf{(6)}$$

Where $\beta_k$ is the angle between $\underset{X_{best}X_i}{\longrightarrow}$ (a vector composed of $X_i$ and $X_{best}$) and the $k2$ dimension, and the value range is $(0, \pi/2)$. $\gamma_k$ represents the angle between the projection of $\underset{X_{best}X_i}{\longrightarrow}$ on the plane composed of $k$ and $k1$ dimensions and the $k$ dimension, and the value range is $(0, \pi)$. d is the distance between $X_i$ and $X_{best}$, calculated by Equation (7):

$$d = \sqrt{\sum_{k=1}^{D}(X_{R_k} - X_{best_k})^2}. \quad \textbf{(7)}$$

Equation (8) indicates that newly added individuals replace individuals with poor fitness values.

$$X_{worst_i} = X_i. \quad \textbf{(8)}$$

### 3.1.2 Strategy 2

In the ROA algorithm, the individual velocity update equation is derived from the insect flapping flight model in Section 2. The movement speed of the individual is equal to the superposition of translation speed and rotation speed. In Equation (4), the translational velocity $\underset{v_1}{\rightarrow}$ is expressed by two components. In this section, we perform vector synthesis on $\underset{v_1}{\rightarrow}$. And the ratio of the lateral flapping frequency period $\omega_1$ to the flapping frequency period $\omega_0$ is set to 1. After simplifying to get Equation (9).

$$\begin{aligned}\underset{v_1}{\rightarrow} &= \frac{1}{2}\sqrt{A^2\omega_0^2 sin^2(\omega_0 t+\theta)+B^2\omega_1^2 cos^2(\omega_1 t+\theta)} \\ &= \frac{\omega_0}{2}\sqrt{A^2 sin^2(\omega_0 t+\theta)+B^2 cos^2(\omega_1 t+\theta)}\end{aligned} \quad \textbf{(9)}$$

In addition, we use the rotation speed before the update to represent the initial attack angle $\alpha_0$, then Equation (5) can be transformed into:

$$\underset{v_2}{\rightarrow} = \underset{v_2}{\rightarrow}\omega_0 cos(\omega_0 t+\theta+\varphi). \quad \textbf{(10)}$$

In summary, the velocity equation of the individuals is:

$$\underset{v}{\rightarrow} = \underset{v_1}{\rightarrow} + \underset{v_2}{\rightarrow}. \quad \textbf{(11)}$$

In the ROA algorithm, the values of some parameters are obtained by referring to [48]. The value of $A$ is set to 2.5; the value of $B$ is 0.1; the values of $\omega_1$ and $\omega_0$ are both 0.025; the value range of $\theta$ is (0,1); $t$ is represented by each iteration of the algorithm, and its value is 1; the value of $\varphi$ is -0.78545; the initial value of $\underset{v_2}{\rightarrow}$ is within the range $(0, 2\pi)$.

The individual moves freely according to the updated speed. At the same time, it will be affected by the best individual. Therefore, the distance update equation of each individual movement is:

$$\underset{le}{\rightarrow} = C \times \underset{v}{\rightarrow} \times t + (X_{best} - X(t)) \times (1-C) \times rand. \quad \textbf{(12)}$$

In Equation (12), the first term represents the distance and direction that the individual moves in the free state. The second term is the distance and direction that the individual moves under the influence of the optimal individual. The actual moving distance and direction is the vector sum of these two items. Among them, $C$ is the influence factor. Its influence on the population evolution in a free state may be a promotion or a hindrance. Therefore, $C$ is a random number in $(-1, 1)$. In the second term, the influence of the optimal individual on the updated individual is only attractive, and there is no repulsion. Therefore, the impact factor of the second term is set to $(1-C)$, and its value is between 0 and 2. When the optimal individual has a greater impact on other individuals, the algorithm may fall into the local optimal prematurely. Therefore, we add a random number to the second term.

To sum up, the individual update equation is:

$$X(t) = X(t) + \underset{le}{\rightarrow}. \quad \textbf{(13)}$$

## 3.2 The Stage of Swallowing Insects

According to the evolutionary principle of "natural selection by nature, survival of the fittest", the population eliminates an individual with the worst fitness in this stage. And the second stage to be executed every certain number of iterations. In order to ensure the quality of the optimal solution, the experiment in Section 4.1 is used to determine the value of $NP_{reduced}$. $NP_{reduced}$ is the number of individuals that the ROA algorithm needs to reduce in total. In addition, the pseudo-code of the second stage is shown in Algorithm 1.

**Algorithm 1.** The second stage

**Input:** $iter$: current iteration number; $NP$: population size; $X_{worst}$: individual with worst fitness value; $pop$: population collection; $fit$: fitness value set;
**Output:** population and fitness value after deletion

```
if rem(iter,100) == 50 then
    pop (X_worst, :) = [ ];   //delete the position of X_worst
    fit (X_worst, :) = [ ];   //delete the fitness of X_worst
    NP= NP – 1; //reduce the population size by one
end if
```

## 3.3 The Stage of Spreading Seeds

After going through the first two stages, the algorithm enters the third stage. The position of the Rafflesia is represented by the initial optimal individual in this stage, that is, the optimal individual at the end of the first stage. Later, in the update process at this stage, the meaning of the best individual becomes the seed with the highest probability of survival. Other individuals randomly search for a more suitable environment for survival. The update equation of the individual at this stage is:

$$X(t)_k = X_{best_k} + rd \times exp\left(\frac{iter}{Max\_iter} - 1\right) \times sign(rand - 0.5). \quad \textbf{(14)}$$

Where, $rd$ is the distribution range of individuals, which is obtained by Equation (15). $k(k = 1,2,\ldots,D)$ is the population dimension; $iter$ represents the current number of iterations; Max_iter is the maximum number of iterations; $\exp\left(\frac{\text{iter}}{\text{Max\_iter}} - 1\right)$ stands for the influence factor that changes with the number of iterations; rand is a random number between 0 and 1. The value of $\text{sign}(\text{rand} - 0.5)$ is 1 or -1, and its purpose is to increase the diversity of solutions.

$$rd = rand \times (ub - lb) + lb. \quad \textbf{(15)}$$

Where, $ub$ is the maximum boundary, and $lb$ is the minimum boundary. The pseudo-code of the algorithm is shown in Algorithm 2.

**Algorithm 2.** Rafflesia optimization algorithm

**Input:** $Max\_iter$:the maximum number of iterations; $NP$:population size; $D$:population dimension;
**Output:** global optimal value

Initialize A, B, $\omega_0$, $\omega_1$, $\theta$, $t$, $\varphi$, $C$, $\underset{v_2}{\rightarrow}$.
Initialize population.
**for** $iter$ =1 to $Max\_iter$ **do**
    Calculate the fitness value of each individual.
    Sort the fitness value from poor to excellent, and find the current global optimal individual $X_{best}$.
    //the first stage
    **if** rem ($iter$, 100) < 50 **then**
        **for** $i$ = 1 to $NP$ **do**
            **if** $i \leq NP/3$ **then**
                Calculate the parameter $d$ by Equation (7).
                New individuals are obtained by Equation (6); the poor fitness individuals are updated by Equation (8).
            **else**
                **for** $j$ = 1 to $D$ **do**
                $\underset{v_1}{\rightarrow}$ and $\underset{v_2}{\rightarrow}$ are calculated by Equation (9) and Equation (10), respectively.
                Update the velocity $\underset{v}{\rightarrow}$ of each individual by Equation (11).
                Update the remaining individuals occupying 2/3 of the population size by Equation (12) and (13).
                **end for**
            **end if**
        **end for**
    **end if**
    //the second stage
    Delete the individuals with the worst fitness, see Algorithm 1.
    //the third stage
    **if** rem ($iter$, 100) > 50 **then**
        **for** $i$ = 1 to $NP$ **do**
            Use Equation (15) to calculate parameter $rd$.
            Use Equation (14) to update the population individuals.
        **end for**
    **end if**
**end for**

# 4 Experimental Results

In order to verify the optimization ability of the new algorithm, the CEC2013 benchmark function set is selected for performance testing. The CEC2013 test set contains 28 benchmark functions. F1-F5 are Unimodal functions, F6-F20 are Basic Multimodal functions, and F21-F28 are Composition functions. In Section 4.1, we determine the value of $NP_{reduce}$ through experiments. In Section 4.2, we compare the optimization ability and convergence speed of the ROA algorithm with other meta-heuristic algorithms. The other seven algorithms are PSO, WOA, CSO, MFO, BA, FA, BOA algorithms. In Section 4.3, the convergence effect of the ROA algorithm is analyzed from the convergence curve graph and the individual distribution graph. In Section 4.4, the optimization ability of the ROA algorithm and other optimization algorithms is further compared by Wilcoxon rank-sum test. It is worth noting that the ROA algorithm in Section 4.2-4.4 is the better version in Section 4.1. All experiments are run on MATLAB R2019a software in Windows 10 operating system.

## 4.1 Comparison of the ROA Algorithm with Different $NP_{reduce}$

In the experiment, except the value of $NP_{reduce}$ is different, the other parameters are the same. Each case runs 30 times. Table 1 records the results when the population size is 30 and the value of $NP_{reduce}$ is 5, 10, 15 and 20 respectively. 'Mean' is the average value of the difference between the optimal solution and the target optimal solution. 'Std' is the standard deviation. 'Win' represents the number of wins on 28 benchmark functions.

From the data in Table 1, it can be seen that when the value of $NP_{reduce}$ is 5 and 10, the result of the ROA algorithm at the optimal value is better than the latter two cases. But when the value of $NP_{reduce}$ is 10, the stability of the ROA algorithm is better than the other three cases. (The standard deviation partly reflects the stability of the algorithm.) In summary, when the value of $NP_{reduce}$ is 10, the ROA algorithm can strike a balance between the optimization ability and the degree of stability. Therefore, the number of individuals reduced is 1/3 of the initial population size. In addition, it is found in the experiment that the elimination of the worst individual has less impact on the population evolution.

## 4.2 Comparison with Other Algorithms

In order to ensure fairness, the number of iterations of each algorithm is uniformly set to 1000, and the initial population size is 30. Each algorithm runs 30 times independently. In the experiment, three different dimensions of 10D, 30D and 50D are selected to test the optimization performance of the algorithms. The parameters of the seven algorithms compared with the ROA algorithm are consistent with the original literature [8, 14-15, 17, 19, 21-22]. Table 2 to Table 4 summarizes the number of the optimal value, average value and standard deviation of the ROA algorithm in the three dimensions, which outperforms other algorithms. The optimal values, average values, and standard deviations of the eight algorithms are recorded on the website (https://github.com/Fuzonglin/ROA-Fig/tree/master).

**Table 1.** Comparison of the results with different $NP_{reduce}$

| Function name | 5 | | 10 | | 15 | | 20 | |
|---|---|---|---|---|---|---|---|---|
| | mean | std | mean | std | mean | std | mean | std |
| F1 | **8.05E-04** | **2.03E-04** | 9.55E-04 | 2.56E-04 | 1.12E-03 | 2.78E-04 | 1.30E-03 | 3.05E-04 |
| F2 | **1.19E+07** | **4.77E+06** | 1.56E+07 | 6.21E+06 | 1.37E+07 | 6.25E+06 | 1.49E+07 | 6.00E+06 |
| F3 | 4.53E+09 | 4.94E+09 | **4.17E+09** | **3.34E+09** | 4.38E+09 | 4.28E+09 | 4.73E+09 | 4.20E+09 |
| F4 | **6.32E+04** | 1.98E+04 | 6.63E+04 | **1.81E+04** | 7.12E+04 | 2.05E+04 | 7.56E+04 | 1.92E+04 |
| F5 | **2.08E-01** | 7.76E-01 | 3.75E-01 | **7.57E-01** | 2.57E+00 | 4.08E+00 | 5.03E+00 | 9.74E+00 |
| F6 | 6.63E+01 | 2.95E+01 | **5.21E+01** | **2.49E+01** | 6.42E+01 | 3.05E+01 | 6.80E+01 | 2.88E+01 |
| F7 | **1.91E+02** | 6.00E+01 | 1.95E+02 | 5.49E+01 | 2.04E+02 | **5.32E+01** | 2.00E+02 | 5.50E+01 |
| F8 | 2.10E+01 | **6.00E-02** | **2.10E+01** | 6.17E-02 | 2.11E+01 | 6.05E-02 | 2.10E+01 | 6.62E-02 |
| F9 | 3.74E+01 | 3.49E+00 | **3.61E+01** | **2.67E+00** | 3.66E+01 | 3.85E+00 | 3.63E+01 | 3.34E+00 |
| F10 | **3.09E+00** | **1.38E+00** | 3.78E+00 | 2.12E+00 | 4.65E+00 | 2.28E+00 | 5.84E+00 | 2.71E+00 |
| F11 | 4.31E+02 | 1.67E+02 | 4.54E+02 | **1.54E+02** | 4.52E+02 | 1.73E+02 | **4.25E+02** | 1.59E+02 |
| F12 | 4.55E+02 | 1.89E+02 | **4.09E+02** | **1.48E+02** | 4.11E+02 | 1.67E+02 | 4.76E+02 | 1.90E+02 |
| F13 | 3.67E+02 | **7.04E+01** | 3.64E+02 | 1.02E+02 | 3.74E+02 | 9.38E+01 | **3.58E+02** | 8.88E+01 |
| F14 | 4.83E+03 | 6.46E+02 | **4.82E+03** | **5.88E+02** | 5.07E+03 | 7.56E+02 | 4.90E+03 | 6.40E+02 |
| F15 | 5.29E+03 | 8.03E+02 | 5.25E+03 | **7.74E+02** | **5.00E+03** | 8.80E+02 | 5.27E+03 | 8.45E+02 |
| F16 | 1.84E+00 | 6.61E-01 | 1.64E+00 | 5.52E-01 | 1.65E+00 | **4.95E-01** | **1.63E+00** | 6.41E-01 |
| F17 | 4.87E+02 | 1.88E+02 | **4.23E+02** | **1.29E+02** | 4.51E+02 | 1.69E+02 | 5.31E+02 | 2.12E+02 |
| F18 | **4.17E+02** | 1.27E+02 | 4.46E+02 | 1.67E+02 | 4.28E+02 | 1.70E+02 | 4.27E+02 | **1.15E+02** |
| F19 | 3.57E+01 | **1.24E+01** | 3.68E+01 | 1.73E+01 | **3.27E+01** | 1.24E+01 | 3.63E+01 | 1.28E+01 |
| F20 | 1.48E+01 | **2.36E-01** | **1.47E+01** | 4.68E-01 | 1.48E+01 | 4.80E-01 | 1.48E+01 | 3.99E-01 |
| F21 | **2.89E+02** | 1.07E+02 | 3.28E+02 | 9.13E+01 | 2.96E+02 | **7.68E+01** | 3.12E+02 | 1.03E+02 |
| F22 | **6.05E+03** | 1.10E+03 | 6.26E+03 | **8.66E+02** | 6.21E+03 | 1.01E+03 | 6.34E+03 | 1.14E+03 |
| F23 | **5.99E+03** | 1.08E+03 | 6.53E+03 | 1.11E+03 | 6.53E+03 | **1.04E+03** | 6.48E+03 | 1.13E+03 |
| F24 | 3.11E+02 | 1.38E+01 | **3.09E+02** | **9.21E+00** | 3.12E+02 | 1.22E+01 | 3.18E+02 | 3.39E+01 |
| F25 | 3.29E+02 | 1.55E+01 | 3.29E+02 | 1.20E+01 | **3.28E+02** | 1.47E+01 | 3.33E+02 | **1.18E+01** |
| F26 | **3.36E+02** | 8.98E+01 | 3.66E+02 | 7.03E+01 | 3.75E+02 | 5.97E+01 | 3.83E+02 | **5.03E+01** |
| F27 | 1.30E+03 | 8.48E+01 | 1.32E+03 | **8.37E+01** | **1.28E+03** | 9.39E+01 | 1.31E+03 | 8.97E+01 |
| F28 | 1.02E+03 | 8.81E+02 | **9.00E+02** | **8.17E+02** | 1.14E+03 | 9.46E+02 | 1.02E+03 | 8.23E+02 |
| **win** | 11 | 7 | 10 | 14 | 4 | 4 | 3 | 3 |

Table 2 records the comparison results between the ROA algorithm and seven algorithms on optimal values. The optimal value represents the optimization ability of the algorithm. The 'win' in the tables indicates that the optimal value of the ROA algorithm on 28 functions is equal to or better than the number of other algorithms. It can be seen from these data that when the ROA algorithm is compared with the MFO algorithm, the winning numbers in the three dimensions are 8, 15, and 15, respectively. This shows that the optimization capability of the ROA algorithm is only worse than MFO algorithm in 10D. As the dimensionality increases, the optimization capability of the ROA algorithm approaches the MFO algorithm. Except for the MFO algorithm, the optimal value of the ROA algorithm is better than the other six algorithms in every dimension. It can even achieve an overwhelming victory over the FA algorithm and the BOA algorithm.

Table 3 records the comparison results between the ROA algorithm and the seven algorithms in terms of average values. It can be seen from the data in the tables that the average value of the ROA algorithm on 10D, 30D, and 50D can be equal to or better than the other seven algorithms in general. Compared with the WOA, BA, FA, BOA algorithms, the ROA algorithm can achieve absolute victory. Compared with the PSO algorithm and the CSO algorithm, the ROA algorithm can show a greater advantage in 30D, but the advantage in 10D and 50D is relatively small. Compared with the MFO algorithm, their optimizing capabilities for numerical functions are similar.

Table 4 records the comparison results of the ROA algorithm and the seven algorithms on the standard deviation. From the data in Table 4, it is found that the stability of the ROA algorithm is completely better than the MFO algorithm in three dimensions. Compared with the CSO algorithm and

the BOA algorithm, the ROA algorithm has poor stability. Compared with the FA algorithm and the WOA algorithm, the stability of the ROA algorithm can achieve absolute victory in all three dimensions. Its effect is close to the PSO algorithm. In addition, combining the data in Table 2 to Table 4, we find that the ROA algorithm performs better on unimodal functions than multimodal functions and composition functions.

**Table 2.** Comparison of the optimal value of ROA and other algorithms in different function types and dimensions

| Best | Function type | PSO | WOA | CSO | MFO | BA | FA | BOA |
|---|---|---|---|---|---|---|---|---|
| 10D | Unimodal | 4 | 5 | 4 | 3 | 3 | 5 | 5 |
| | Multimodal | 12 | 14 | 14 | 3 | 13 | 15 | 15 |
| | Composition | 5 | 4 | 6 | 2 | 7 | 8 | 6 |
| | **Win** | **21** | **23** | **24** | **8** | **23** | **28** | **26** |
| 30D | Unimodal | 5 | 5 | 5 | 5 | 4 | 5 | 5 |
| | Multimodal | 13 | 15 | 14 | 7 | 12 | 15 | 15 |
| | Composition | 5 | 6 | 7 | 3 | 5 | 2 | 8 |
| | **Win** | **23** | **26** | **26** | **15** | **21** | **28** | **28** |
| 50D | Unimodal | 5 | 4 | 4 | 5 | 2 | 5 | 5 |
| | Multimodal | 11 | 14 | 13 | 7 | 12 | 15 | 14 |
| | Composition | 5 | 6 | 7 | 3 | 4 | 8 | 8 |
| | **Win** | **21** | **24** | **24** | **15** | **18** | **28** | **27** |

**Table 3.** Comparison of the average value of ROA and other algorithms in different function types and dimensions

| Mean | Function type | PSO | WOA | CSO | MFO | BA | FA | BOA |
|---|---|---|---|---|---|---|---|---|
| 10D | Unimodal | 4 | 5 | 4 | 5 | 3 | 5 | 4 |
| | Multimodal | 5 | 13 | 9 | 8 | 13 | 15 | 14 |
| | Composition | 5 | 6 | 3 | 1 | 7 | 8 | 5 |
| | **Win** | **14** | **24** | **16** | **14** | **23** | **28** | **23** |
| 30D | Unimodal | 5 | 5 | 4 | 5 | 3 | 5 | 5 |
| | Multimodal | 10 | 15 | 12 | 9 | 13 | 15 | 15 |
| | Composition | 4 | 8 | 5 | 2 | 4 | 2 | 8 |
| | **Win** | **19** | **28** | **21** | **16** | **20** | **28** | **28** |
| 50D | Unimodal | 5 | 4 | 4 | 5 | 2 | 5 | 4 |
| | Multimodal | 7 | 13 | 8 | 6 | 11 | 15 | 14 |
| | Composition | 5 | 7 | 4 | 3 | 6 | 8 | 7 |
| | **Win** | **17** | **24** | **16** | **14** | **19** | **28** | **25** |

**Table 4.** Comparison of the standard deviation of ROA and other algorithms in different function types and dimensions

| Std | Function type | PSO | WOA | CSO | MFO | BA | FA | BOA |
|---|---|---|---|---|---|---|---|---|
| 10D | Unimodal | 5 | 5 | 3 | 5 | 2 | 5 | 4 |
| | Multimodal | 7 | 12 | 3 | 7 | 9 | 11 | 6 |
| | Composition | 3 | 5 | 1 | 4 | 5 | 3 | 2 |
| | **Win** | **15** | **22** | **7** | **16** | **16** | **19** | **12** |
| 30D | Unimodal | 5 | 5 | 4 | 5 | 2 | 5 | 4 |
| | Multimodal | 8 | 8 | 4 | 10 | 11 | 10 | 4 |
| | Composition | 3 | 6 | 1 | 5 | 4 | 4 | 3 |
| | **Win** | **16** | **19** | **9** | **20** | **17** | **19** | **11** |
| 50D | Unimodal | 5 | 4 | 4 | 5 | 1 | 5 | 4 |
| | Multimodal | 6 | 5 | 4 | 8 | 4 | 7 | 5 |
| | Composition | 3 | 5 | 0 | 2 | 4 | 4 | 4 |
| | **Win** | **14** | **14** | **8** | **15** | **9** | **16** | **13** |

## 4.3 Convergence Analysis

To evaluate the performance of the algorithm, it is necessary to analyze not only the numerical results but also the convergence of the algorithm. First, we extract the individual distribution diagrams of the ROA algorithm when it runs on F1 and F6, as shown in Figure 3 and Figure 4. Secondly, in order to compare the convergence speed of the ROA algorithm with other algorithms, we randomly select the convergence curves of some benchmark functions from unimodal, multimodal, and composite functions to display. Due to space constraints, only the convergence curves of some functions are show in the paper. The complete experimental charts can be viewed on the website (https://github.com/Fuzonglin/ROA-Fig/tree/master).

Figure 3 and Figure 4 show the function images of F1 and F6, the change process of individual search and the algorithm convergence curve. (b)-(f) are the individual distribution of ROA algorithm at the initial stage of iteration. (b) shows the initial distribution of individuals. With the iterative update, it is manifest from (b)-(f) that the individuals gradually gather to the optimal position, and the search range is gradually reduced. (g) represents the situation when the algorithm executes the third stage. At this point, the ROA algorithm performs the global search to find a better solution. It can be seen from Equation (14) that as the number of iterations increases, the value of $\exp\left(\frac{\text{iter}}{\text{Max\_iter}}-1\right)$ gradually increases from $1/e$ to $1$. Then the search range gradually expands from $1/e[lb,ub]$ to $[lb,ub]$. This means that the ROA algorithm still has the opportunity to jump out of the local optimal value in the later stage. (h)-(k) show that the ROA algorithm is performing a new round of updates. (l) is the convergence curve of the ROA algorithm on the test function.

In Figure 5, the horizontal axis is the number of iterations of the algorithm, and the vertical axis is the fitness value. (a) and (b) are the convergence curves of eight algorithms on unimodal functions F1 and F5. It can be seen from these two figures that the ROA algorithm converges faster on the unimodal function than other algorithms. When the number of iterations reaches about 110, the convergence speed of the ROA algorithm becomes slow. But its final convergence result is better than most algorithms. From the convergence curve (c) of multimodal function F15, it can be seen that the ROA algorithm has a fast convergence speed as a whole, and the final convergence value is also the best. In the convergence curve (d) of F18, the initial convergence ability of the ROA algorithm is inferior to the WOA algorithm and the CSO algorithm. After 60 iterations, the ROA algorithm converges faster, and it shows a better convergence effect than other algorithms. In the convergence curves (e) and (f) of F21 and F28, the ROA algorithm shows strong convergence performance in the first 150 iterations, which makes its final result equal to or better than the other seven algorithms. To sum up, the ROA algorithm has a strong convergence ability, so that it can achieve better optimal values than other algorithms on most functions.

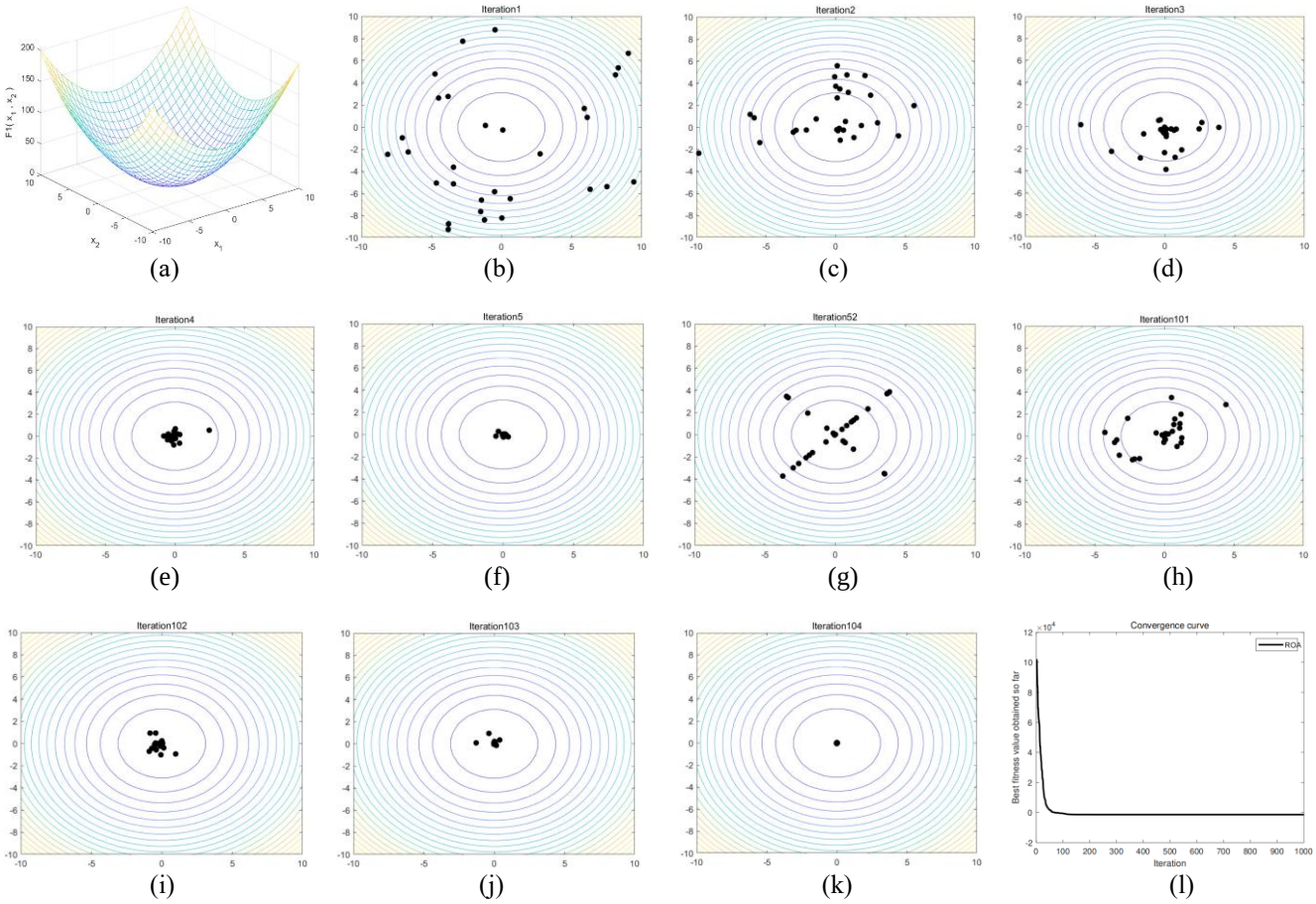



**Figure 3.** The individual distribution result of the ROA algorithm on F1

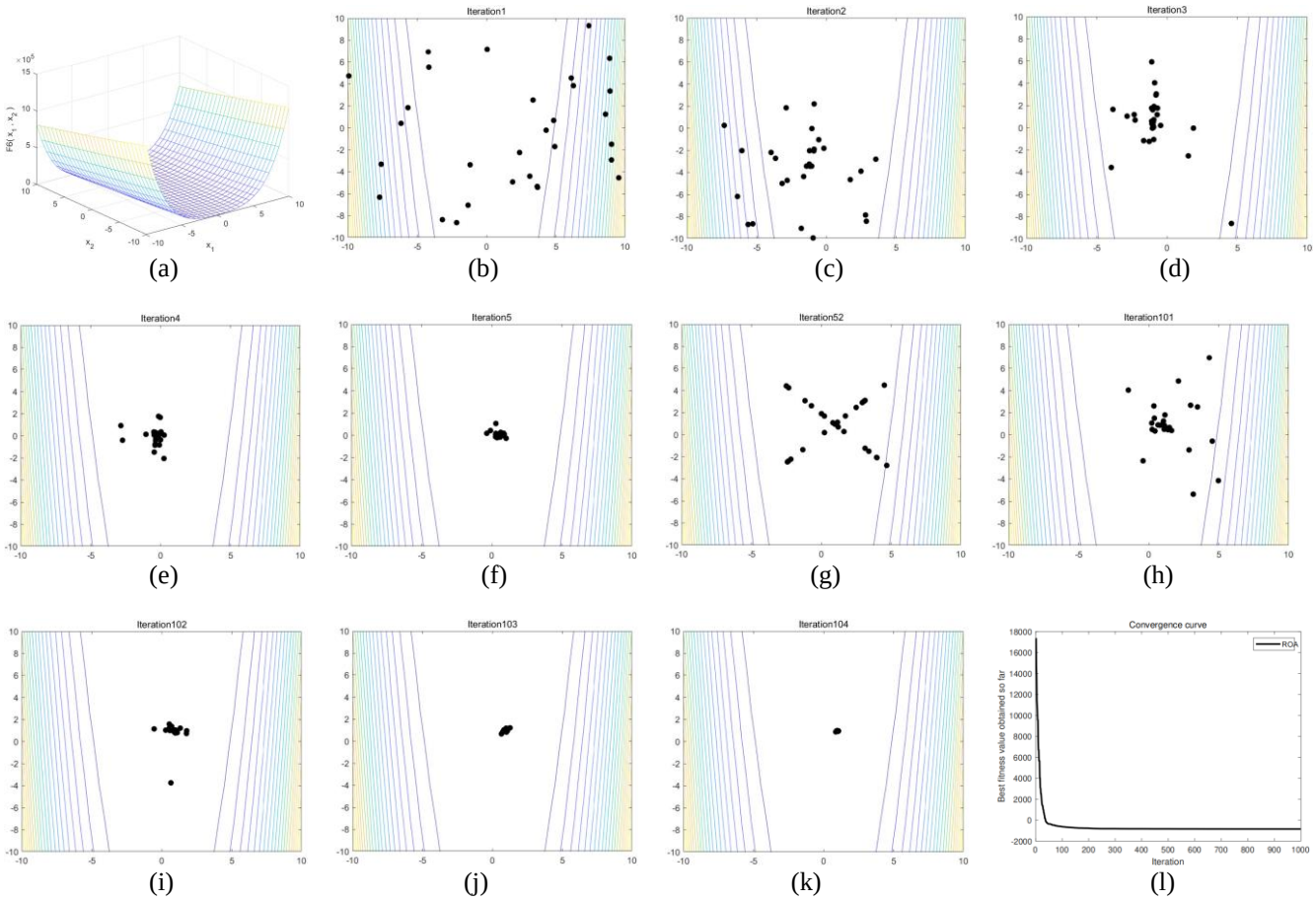



**Figure 4.** The individual distribution result of the ROA algorithm on F6

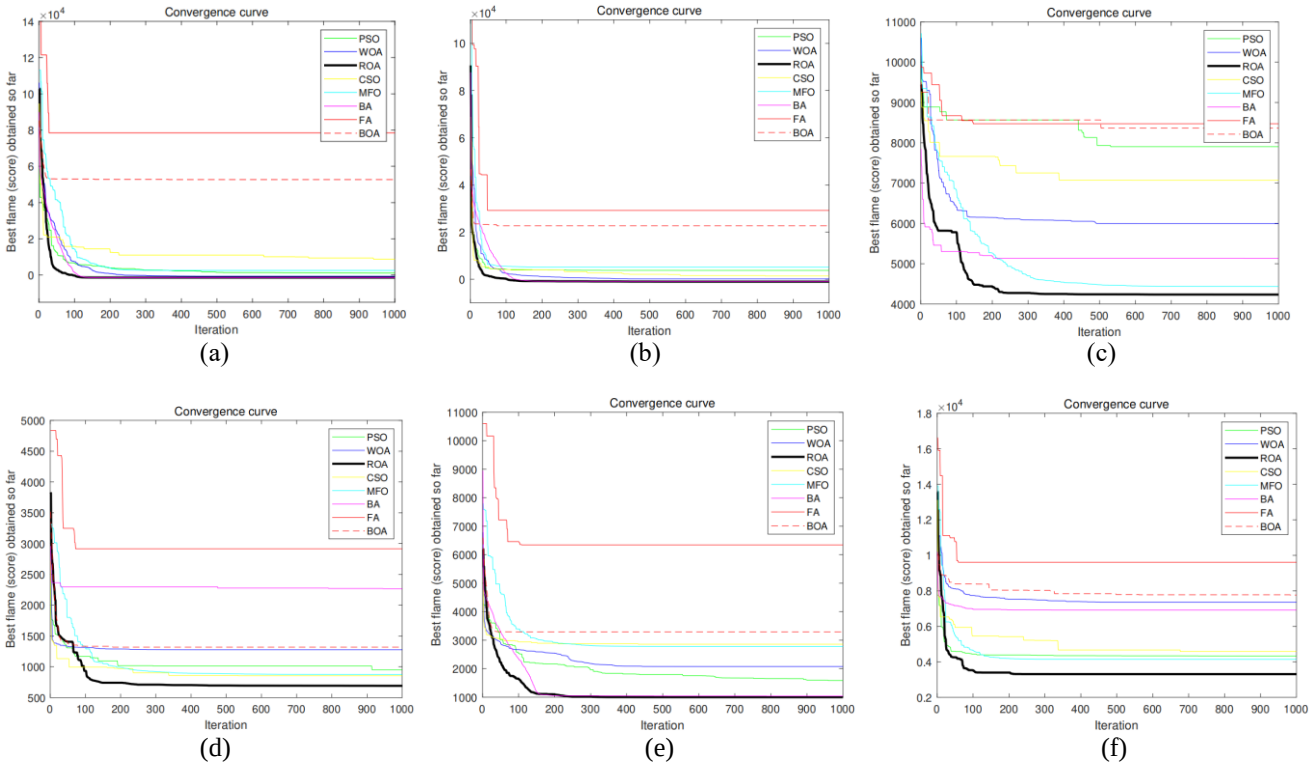



**Figure 5.** Convergence test results in 30D (F1, F5, F15, F18, F21, F28)

**Table 5.** The results of the Wilcoxon rank-sum test (the significance level is 0.05)

| Function | PSO | | WOA | | CSO | | MFO | | BA | | FA | | BOA | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| F1 | 1.51E-11 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 | 6.64E-11 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 |
| F2 | 1.51E-11 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 | 2.98E-09 | 1 | 0.999999 | 0 | 1.51E-11 | 1 | 1.51E-11 | 1 |
| F3 | 1.51E-11 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 | 1 | 0 | 1.51E-11 | 1 | 1.51E-11 | 1 |
| F4 | 0.000721 | 1 | 4.61E-05 | 1 | 1 | 0 | 1.6E-09 | 1 | 8.65E-07 | 1 | 1.51E-11 | 1 | 0.763327 | 0 |
| F5 | 0.51E-11 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 | 5.05E-09 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 |
| F6 | 1.51E-11 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 | 6.64E-11 | 1 | 0.998165 | 0 | 1.51E-11 | 1 | 1.51E-11 | 1 |
| F7 | 1.14E-05 | 1 | 1.01E-08 | 1 | 1.76E-07 | 1 | 0.030726 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 |
| F8 | 0.77679 | 0 | 0.990816 | 0 | 1 | 0 | 0.657839 | 0 | 0.299845 | 0 | 8.88E-11 | 1 | 0.305004 | 0 |
| F9 | 0.999951 | 0 | 0.074724 | 0 | 4.15E-06 | 1 | 0.999241 | 0 | 0.057681 | 0 | 2.04E-11 | 1 | 3.37E-06 | 1 |
| F10 | 1.51E-11 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 | 0.966066 | 0 | 1.51E-11 | 1 | 1.51E-11 | 1 |
| F11 | 0.995466 | 0 | 0.004734 | 1 | 0.983063 | 0 | 1 | 0 | 1.08E-10 | 1 | 1.51E-11 | 1 | 2.25E-11 | 1 |
| F12 | 0.999349 | 0 | 2.99E-05 | 1 | 0.772352 | 0 | 0.996191 | 0 | 1.91E-10 | 1 | 1.51E-11 | 1 | 2.04E-11 | 1 |
| F13 | 0.185539 | 0 | 4.96E-11 | 1 | 2.47E-05 | 1 | 0.011622 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 |
| F14 | 6.01E-09 | 1 | 1.51E-07 | 1 | 1.51E-11 | 1 | 0.993084 | 0 | 0.000428 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 |
| F15 | 1.01E-08 | 1 | 6.93E-07 | 1 | 3.96E-10 | 1 | 0.201769 | 0 | 0.739928 | 0 | 1.51E-11 | 1 | 2.31E-10 | 1 |
| F16 | 1.02E-09 | 1 | 0.001669 | 1 | 0.00125 | 1 | 0.767864 | 0 | 1.08E-06 | 1 | 1.84E-11 | 1 | 2.75E-11 | 1 |
| F17 | 0.331367 | 0 | 3.01E-08 | 1 | 0.455854 | 0 | 0.870974 | 0 | 1.51E-11 | 1 | 1.51E-11 | 1 | 6.64E-11 | 1 |
| F18 | 0.155594 | 0 | 1.19E-07 | 1 | 0.003809 | 1 | 0.342161 | 0 | 1.51E-11 | 1 | 1.51E-11 | 1 | 2.49E-11 | 1 |
| F19 | 1.19E-08 | 1 | 3.03E-11 | 1 | 0.933273 | 0 | 1.51E-11 | 1 | 0.001835 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 |
| F20 | 0.999999 | 0 | 0.02471 | 1 | 0.111286 | 0 | 0.944188 | 0 | 0.000427 | 1 | 0.086881 | 0 | 0.003856 | 1 |
| F21 | 1.51E-11 | 1 | 1.51E-11 | 1 | 2.55E-08 | 1 | 2.79E-10 | 1 | 0.02506 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 |
| F22 | 0.04667 | 1 | 1.48E-05 | 1 | 1.3E-10 | 1 | 0.999994 | 0 | 0.177736 | 0 | 1.51E-11 | 1 | 1.51E-11 | 1 |
| F23 | 4.1E-07 | 1 | 4.76E-06 | 1 | 3.03E-11 | 1 | 0.907117 | 0 | 0.245891 | 0 | 1.51E-11 | 1 | 3.69E-11 | 1 |
| F24 | 0.918812 | 0 | 0.00106 | 1 | 0.999777 | 0 | 1 | 0 | 1.19E-10 | 1 | 1.51E-11 | 1 | 1.67E-11 | 1 |
| F25 | 0.998623 | 0 | 0.008477 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1.51E-11 | 1 | 1.22E-09 | 1 |
| F26 | 0.61344 | 0 | 0.003334 | 1 | 1 | 0 | 0.999777 | 0 | 0.003049 | 1 | 0.000189 | 1 | 0.960109 | 0 |
| F27 | 0.999995 | 0 | 0.006366 | 1 | 2.99E-05 | 1 | 1 | 0 | 4.53E-08 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 |
| F28 | 1.19E-10 | 1 | 2.49E-11 | 1 | 1.51E-11 | 1 | 1.42E-08 | 1 | 1.84E-11 | 1 | 1.51E-11 | 1 | 1.51E-11 | 1 |
| Win | 16 | | 26 | | 18 | | 12 | | 18 | | 27 | | 25 | |

### 4.4 Wilcoxon Rank-sum Test

Table 5 records the results of the Wilcoxon rank-sum test performed by the ROA algorithm and the seven algorithms. The significance level is 0.05. The first column under each algorithm in the table represents the inspection value. The '1' in the second column indicates that the performance of the ROA algorithm is better than this algorithm. '0' indicates that it is worse than this algorithm. As can be seen from the test results in the table, the number of the ROA algorithm is superior to the PSO, WOA, CSO, MFO, BA, FA and BOA algorithms are 16, 26, 18, 12, 18, 27 and 25, respectively. This displays that the ROA algorithm is slightly worse than the MFO algorithm in the test. However, it is better than the other six algorithms.

## 5 Application in the Logistics Distribution Centers Location Problem

### 5.1 Mathematical Model

The logistics distribution centers location problem can be described as: selecting $M$ locations from $N$ logistics center candidate points as the distribution center. They are responsible for providing goods to all cities. The farther the distribution center is from the city, the higher the cost. Therefore, the goal of the problem is to have the lowest total cost when the demand in all cities is met. Assumptions:

(1) The distribution center can meet the demand for goods in all cities, that is, the demand for each city is less than or equal to the total amount of goods in the distribution center.

(2) The goods in each city can only be provided by one distribution center.

(3) Cities beyond the service range of the distribution center will not be delivered.

(4) Other expenses such as factory to distribution center are not considered.

Based on the above assumptions, we establish a mathematical model of the logistics distribution centers location problem with multiple constraints. First, the objective function is:

$$minF = \sum_{i\epsilon N} \sum_{j\epsilon M_i} \omega_i \, d_{ij} z_{ij}. \quad \textbf{(16)}$$

Where, $F$ is the total cost of completing the task. $N = \{1,2,\ldots,n\}$ is the number set of all cities. $M = \{1,2,\ldots,g\}$ is the collection of all distribution centers. $M_i$ is the set of candidate distribution centers whose distance to city $i$ is less than $s$; $i \in N, M_i \in M$. $\omega_i$ represents the number of goods required by the i-th city. $d_{ij}$ represents the distance from city $i$ to the nearest distribution center $j$. $z_{ij}$ represents the distribution relationship between the city and the distribution center, and the value is 0 or 1. When $z_{ij} = 1$, it means that the goods of city $i$ are provided by the distribution center $j$, otherwise $z_{ij} = 0$.

Second, the constraints are:

$$\sum_{j\epsilon M_i} z_{ij} = 1, \;\; i\epsilon N. \quad \textbf{(17)}$$

Equation (17) indicates that the goods of city $i$ can only be provided by the distribution center $j$ closest to it, which ensures that each city is delivered by only one distribution center.

$$z_{ij} \leq h_j, \quad i \epsilon N, j \epsilon M_j. \tag{18}$$

Equation (18) indicates that there are no cities in the locations that the distribution center cannot radiate, which ensures that all cities can be distributed. The value of $h_j$ is 0 or 1. When $h_j = 1$, it means that location $j$ is selected as the distribution center.

$$\sum_{j \epsilon M_i} h_j = p. \tag{19}$$

Equation (19) indicates that there are $p$ locations selected as the distribution center.

$$d_{ij} \leq s. \tag{20}$$

Equation (20) indicates that the distance from city $i$ to distribution center $j$ is less than $s$. $s$ is the radiation range of distribution center $j$.

## 5.2 Coding Ideas and Solving Steps

The ROA algorithm is a continuous optimization algorithm, and the logistics distribution centers location problem is a discrete problem. Therefore, the ROA algorithm needs to be discretized when it solves the logistics center location problem. Therefore, before solving the problem, we chose a set of coding schemes to establish the mapping relationship from continuous algorithms to discrete problems. The coding idea in the solution process is: take the position of each individual as a set of feasible solutions to the problem. The population size $NP$ corresponds to the number of feasible solutions. And the algorithm dimension $D$ corresponds to the number of cities $N$. Each individual is a $2 \times D$ vector, one of which is denoted as $X\_r$ and the other is denoted as $X$. Each value in $X\_r$ is a real number, and the data in $X$ corresponds to the order of elements in $X\_r$. For each time location selection, the city number corresponding to rank $1-p$ in $X$ is selected as the distribution center. For examole, two distribution centers are selected from six locations. Among them, the element in $X\_r$ of one individual is [0.1,1.2, -0.4, -2.1,0.01,3.3], and the element in $X$ can be obtained as [4,5,2,1,3,6]. Then the distribution center cities given by this individual are City 3 and City 4.

The specific solution steps of the ROA algorithm on the logistics distribution centers location problem are as follows:

Step 1: Set the parameters of the ROA algorithm, such as the population size $NP$, the maximum number of iterations $Max\_iter$, etc.;

Step 2: Use the $rand()$ function to generate $X\_r$ with a quantity of $NP$, which is the initial population, and obtain $X$ according to the above coding idea;

Step 3: Calculate fitness according to Equation (16) and find $X\_r$ and $X$ of the best individual;

Step 4: The $X\_r$ of each individual is updated iteratively according to the three stages of the algorithm in Section 3;

Step 5: If the current number of iterations reaches the maximum number of iterations, output the city number corresponding to rank $1-p$ in $X$ of the current best individual as the distribution center, and end the algorithm; otherwise, go to step 3.

## 5.3 Simulation Experiment

In order to verify the ability of the ROA algorithm to solve the logistics distribution centers location problem, two sets of test cases are selected for the simulation experiment. One test case is 31g-6n, which means that 6 cities from 31 cities are selected as the distribution center, and the other test case is 100g-20n. Table 6 and Table 7 record the city coordinates and demand of the test case. Among them, 'No.' represents the city number, 'coordinate' is the city coordinate, and 'need' represents the demand of the city. After that, the results of the two sets of experiments are compared and analyzed with the data of other algorithms on the logistics distribution centers location problem. In order to ensure that the experimental results are comparable, all algorithms are set to the same test conditions. The maximum number of iterations is 100, the population size is 50, and they run 10 times independently. The experimental results of the two sets of test cases are recorded in Table 8 and Table 9, respectively. It includes the optimal value, average value, standard deviation, running time, and distribution center location of each algorithm in 10 runs. In addition, Figure 6 and Figure 7 show randomly selected convergence curves and distribution center location maps.

**Table 6.** The location of cities and their demand in 31g-6n

| No. | Coordinates | Need | No. | Coordinates | Need |
|---|---|---|---|---|---|
| 1 | (1304, 2312) | 20 | 17 | (3918, 2179) | 90 |
| 2 | (3639, 1315) | 90 | 18 | (4061, 2370) | 70 |
| 3 | (4177, 2244) | 90 | 19 | (3780, 2212) | 100 |
| 4 | (3712, 1399) | 60 | 20 | (3676, 2578) | 50 |
| 5 | (3488, 1535) | 70 | 21 | (4029, 2838) | 50 |
| 6 | (3326, 1556) | 70 | 22 | (4263, 2931) | 50 |
| 7 | (3238, 1229) | 40 | 23 | (3429, 1908) | 80 |
| 8 | (4196, 1044) | 90 | 24 | (3507, 2376) | 70 |
| 9 | (4312, 790) | 90 | 25 | (3394, 2643) | 80 |
| 10 | (4386, 570) | 70 | 26 | (3439, 3201) | 40 |
| 11 | (3007, 1970) | 60 | 27 | (2935, 3240) | 40 |
| 12 | (2562, 1756) | 40 | 28 | (3140, 3550) | 60 |
| 13 | (2788, 1491) | 40 | 29 | (2545, 2357) | 70 |
| 14 | (2381, 1676) | 40 | 30 | (2778, 2826) | 50 |
| 15 | (1332, 695) | 20 | 31 | (2370, 2975) | 30 |
| 16 | (3715, 1678) | 80 | | | |

**Table 7.** The location of cities and their corresponding demand in 100g-20n

| No. | Coordinates | Need | No. | Coordinates | Need | No. | Coordinates | Need | No. | Coordinates | Need |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | (2168, 2179) | 71 | 26 | (4442, 2034) | 46 | 51 | (2740, 2139) | 44 | 76 | (2138, 3819) | 60 |
| 2 | (1500, 1395) | 66 | 27 | (1082, 1446) | 65 | 52 | (1694, 2526) | 79 | 77 | (1244, 2861) | 83 |
| 3 | (2675, 1475) | 47 | 28 | (1599, 4104) | 47 | 53 | (1522, 1696) | 60 | 78 | (3498, 2137) | 91 |
| 4 | (3502, 3923) | 48 | 29 | (3426, 3754) | 50 | 54 | (1363, 4040) | 50 | 79 | (2578, 1482) | 99 |
| 5 | (3013, 1040) | 37 | 30 | (1578, 1812) | 20 | 55 | (3189, 2936) | 14 | 80 | (1848, 2564) | 43 |
| 6 | (2750, 3501) | 74 | 31 | (4345, 1676) | 27 | 56 | (1401, 2169) | 78 | 81 | (3239, 2417) | 96 |
| 7 | (2383, 2198) | 84 | 32 | (4122, 2665) | 96 | 57 | (3099, 2347) | 96 | 82 | (1985, 2971) | 57 |
| 8 | (2219, 2626) | 20 | 33 | (3668, 2895) | 29 | 58 | (1298, 4064) | 17 | 83 | (1703, 2230) | 31 |
| 9 | (4180, 3866) | 43 | 34 | (2131, 2356) | 81 | 59 | (2565, 1600) | 52 | 84 | (2821, 3424) | 86 |
| 10 | (3085, 3800) | 68 | 35 | (2986, 3390) | 96 | 60 | (4272, 1279) | 80 | 85 | (2044, 3315) | 29 |
| 11 | (3501, 1767) | 24 | 36 | (2522, 1135) | 78 | 61 | (3844, 2553) | 53 | 86 | (3697, 2507) | 31 |
| 12 | (4245, 4278) | 22 | 37 | (4284, 3054) | 58 | 62 | (3792, 2904) | 35 | 87 | (1121, 3147) | 91 |
| 13 | (3232, 3162) | 72 | 38 | (1163, 1566) | 61 | 63 | (2391, 3567) | 51 | 88 | (1880, 1036) | 83 |
| 14 | (2871, 2000) | 64 | 39 | (2156, 1713) | 11 | 64 | (4002, 4139) | 60 | 89 | (2567, 2572) | 95 |
| 15 | (2093, 1463) | 30 | 40 | (2714, 2575) | 20 | 65 | (4265, 2973) | 81 | 90 | (3219, 3936) | 88 |
| 16 | (1118, 3403) | 73 | 41 | (2122, 1471) | 43 | 66 | (2203, 4165) | 15 | 91 | (2993, 3593) | 90 |
| 17 | (2500, 2796) | 42 | 42 | (3189, 2209) | 65 | 67 | (3609, 4450) | 20 | 92 | (2732, 3115) | 58 |
| 18 | (1768, 1415) | 11 | 43 | (2986, 1562) | 95 | 68 | (1862, 3003) | 23 | 93 | (4079, 3310) | 61 |
| 19 | (2872, 1926) | 58 | 44 | (3722, 4010) | 57 | 69 | (2560, 4245) | 54 | 94 | (4255, 2294) | 88 |
| 20 | (2247, 1096) | 45 | 45 | (4442, 3702) | 89 | 70 | (1770, 1329) | 89 | 95 | (4053, 1510) | 10 |
| 21 | (1091, 3950) | 19 | 46 | (3834, 2899) | 32 | 71 | (2619, 4427) | 93 | 96 | (1535, 3579) | 17 |
| 22 | (3679, 1844) | 94 | 47 | (1295, 1526) | 42 | 72 | (3358, 3829) | 22 | 97 | (2063, 2177) | 41 |
| 23 | (1445, 2711) | 40 | 48 | (1475, 2326) | 89 | 73 | (3717, 2180) | 25 | 98 | (2808, 4053) | 89 |
| 24 | (1244, 2138) | 20 | 49 | (3206, 3427) | 91 | 74 | (3383, 2515) | 59 | 99 | (3223, 3318) | 86 |
| 25 | (2374, 4382) | 77 | 50 | (2703, 3493) | 47 | 75 | (3380, 3464) | 56 | 100 | (3310, 2113) | 21 |

**Table 8.** The results of the ROA algorithm and other algorithm in 31g-6n

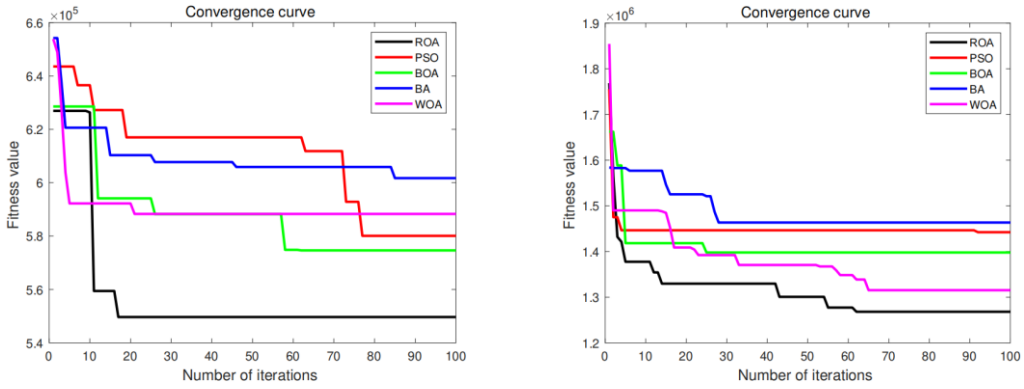
| 31g-6n | ROA | PSO | BOA | BA | WOA |
|---|---|---|---|---|---|
| Best | **549648.31** | 563036.71 | **549648.31** | 563036.71 | **549648.31** |
| Mean | **564573.66** | 582676.38 | 580592.38 | 579595.59 | 575244.31 |
| Std | 11738.9 | 12065 | 22483.7 | **11142.24** | 15080.2 |
| Time | **30.5259** | 32.2373 | 65.8789 | 33.7039 | 32.9595 |
| Location | **12-27-17-9-20-5** | 17-27-24-12-9-5 | **9-27-12-5-17-20** | 5-27-9-17-12-24 | **12-27-17-5-20-9** |

**Table 9.** The results of the ROA algorithm and other algorithm in 100g-20n

| 100g-20n | ROA | PSO | BOA | BA | WOA |
|---|---|---|---|---|---|
| Best | **1235674.83** | 1385014.92 | 1322936.94 | 1405859.74 | 1280610.1 |
| Mean | **1298202.3** | 1437103.9 | 1390539.5 | 1433594.5 | 1343057.7 |
| Std | 30896.9 | **28190** | 42950.1 | 25495.2 | 32527.7 |
| Time | **112.527** | 121.081 | 226.16 | 123.684 | 120.711 |
| Location | **14-34-58-60-88** | 1-60-65-91-7-69 | 18-28-87-99-9 | 94-66-31-70-14 | 79-47-16-81-90 |
| | **-85-71-57-99-84** | -45-63-17-88-23 | -7-57-91-3-29 | -61-79-71-17-2 | -18-7-64-11-25 |
| | **-9-26-16-90-73** | -3-59-78-72-99 | -76-98-60-82-73 | -50-4-60-87-78 | -45-32-43-88-31 |
| | **-48-3-65-61-47** | -26-58-47-56 | -56-65-27-62-26 | -34-9-35-90-48 | -82-99-54-84-56 |

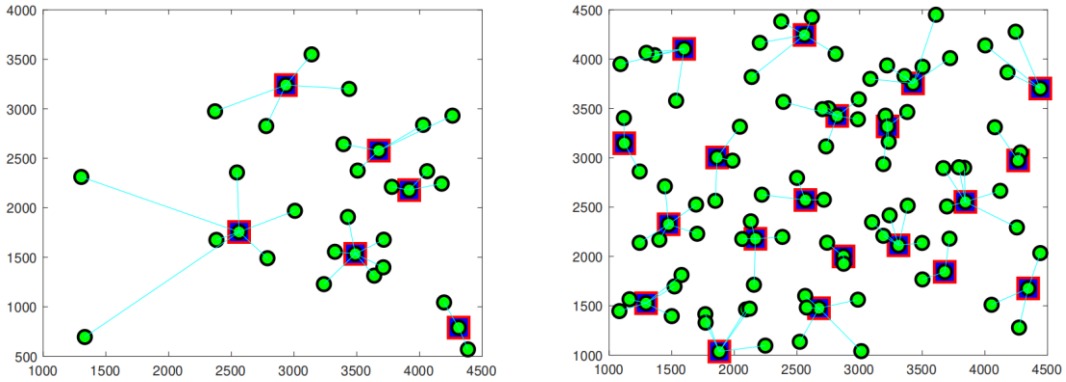
In the data in Table 8 and Table 9, 'best' represents the minimum value of the objective function. 'mean' is the average value of the optimal value obtained by running 10 times. 'std' is the standard deviation. 'time' is the total time for each algorithm to run 10 times. 'location' is the city serial number selected by each algorithm as the distribution center. It can be seen from Table 8 that in the test case of 31g-6n, the BOA, WOA and ROA algorithms can obtain the optimal value of 549648.313, and the corresponding distribution center location is 5-9-12-17-20-27. From the perspective of average value and time, the ROA algorithm is superior to other algorithms. Because it can get the best average in the shortest time. But its standard deviation is slightly worse than the BA algorithm as a whole. Comparing the data in Table 8 and Table 9, it can be concluded that as the number of cities increases, the ROA algorithm can still calculate the most accurate optimal value and average value in the shortest time. However, its stability is worse than the PSO and BA algorithm. The location of the distribution center obtained by the ROA algorithm in the 100g-20n case is 3-9-11-16-26-34-47-48-57-58-60-61-65-71-73-84-85-88-90-99. It can be seen from Figure 6 that the ROA algorithm has a faster convergence rate in the first 20 iterations. Compared with other algorithms, its final convergence value achieved an overwhelming victory. It shows that the ROA algorithm has greater advantages in solution accuracy and convergence ability.

On the whole, the ROA algorithm can efficiently solve the logistics distribution centers location problem. Yet the stability of the ROA algorithm needs to be further improved, which has a direct relationship with the accuracy of the average value. The more stable the algorithm, the more accurate the solution will be. Therefore, studying how to improve the stability of the ROA algorithm will become the focus of the next step.




(a) 31g-6n (b)100g-20n

**Figure 6.** Convergence curve of the 31g-6n and the 100g-20n



(a) 31g-6n (b) 100g-20n

**Figure 7.** Distribution center location maps of the 31g-6n and the 100g-20n

## 6 Conclusion

By simulating the growth characteristics of Rafflesia, we propose a new intelligent optimization algorithm, named the Rafflesia Optimization Algorithm (ROA). The three stages of the ROA algorithm simulate the processes of Rafflesia attracting insects, swallowing insects, and spreading seeds. In order to verify the performance of the new algorithm, it is tested on the CEC2013 benchmark function set and compared with seven meta-heuristics algorithms. In addition, it is also applied to the logistics distribution centers location problem to test the ability to solve actual problems. The test results of the two parts have proved that the new algorithm has strong optimization ability and competitiveness. And it is an effective algorithm for solving optimization problems. However, it is discovered during the test that there is still room for further optimization in the stability of the ROA algorithm. In the future, we will improve the optimization capabilities of the ROA algorithm by introducing different optimization strategies.

## References

[1] R. S. Parpinelli, H. S. Lopes, New inspirations in swarm intelligence: a survey, *International Journal of Bio-Inspired Computation*, Vol. 3, No. 1, pp. 1-16, February, 2011.

[2] X.-S. Yang, Nature-inspired optimization algorithms: Challenges and open problems, *Journal of Computational Science*, Vol. 46, Article No. 101104, October, 2020.

[3] A. R. Simpson, G. C. Dandy, L. J. Murphy, Genetic algorithms compared to other techniques for pipe optimization, *Journal of water resources planning and management*, Vol. 120, No. 4, pp. 423-443, July, 1994.

[4] J. J. Hopfield, Artificial neural networks, *IEEE Circuits and Devices Magazine*, Vol. 4, No. 5, pp. 3-10, September, 1988.

[5] P. J. Van Laarhoven, E. H. Aarts, *Simulated annealing: Theory and applications*, Springer, 1987.

[6] L. N. De Castro, F. J. Von Zuben, Learning and optimization using the clonal selection principle, *IEEE transactions on evolutionary computation*, Vol. 6, No. 3, pp. 239-251, June, 2002.

[7] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, *IEEE computational intelligence magazine*, Vol. 1, No. 4, pp. 28-39, November, 2006.

[8] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of ICNN'95-international conference on neural networks*, Perth, WA, Australia, 1995, Vol. 4, pp. 1942-1948.

[9] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, *Journal of global optimization*, Vol. 11, No. 4, pp. 341-359, December, 1997.

[10] H. Yin, J. Qiao, P. Fu, X. Xia, Face feature selection with binary particle swarm optimization and support vector machine, *Journal of Information Hiding and Multimedia Signal Processing*, Vol. 5, No. 4, pp. 731-739, October, 2014.

[11] W. Li, B. Sun, Y. Huang, S. Mahmoodi, Adaptive particle swarm optimization using scale-free network topology, *Journal of Network Intelligence*, Vol. 6, No. 3, pp. 500-517, August, 2021.

[12] D. Sha, C.-Y. Hsu, A hybrid particle swarm optimization for job shop scheduling problem, *Computers & Industrial Engineering*, Vol. 51, No. 4, pp. 791-808, December, 2006.

[13] A. Slowik, M. Bialko, Training of artificial neural networks using differential evolution algorithm, *2008 conference on human system interactions, IEEE*, Krakow, Poland, 2008, pp. 60-65.

[14] S.-C. Chu, P.-W. Tsai, J.-S. Pan, Cat swarm optimization, *Pacific Rim international conference on artificial intelligence, Springer*, Guilin, China, 2006, pp. 854-858.

[15] X.-S. Yang, Firefly algorithm, stochastic test functions and design optimisation, *International journal of bio-inspired computation*, Vol. 2, No. 2, pp. 78-84, March, 2010.

[16] J.-S. Pan, P.-W. Tsai, Y.-B. Liao, Fish migration optimization based on the fishy biology, *2010 Fourth International Conference on Genetic and Evolutionary Computing, IEEE*, Shenzhen, China, 2010, pp. 783-786.

[17] X.-S. Yang, A. H. Gandomi, Bat algorithm: a novel approach for global engineering optimization, *Engineering computations*, Vol. 29, No. 5, pp. 464-483, July, 2012.

[18] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Advances in engineering software*, Vol. 69, pp. 46-61, March, 2014.

[19] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowledge-based systems*, Vol. 89, pp. 228-249, November, 2015.

[20] Z. Meng, J.-S. Pan, H. Xu, Quasi-affine transformation evolutionary (quatre) algorithm: A cooperative swarm based algorithm for global optimization, *Knowledge-Based Systems*, Vol. 109, pp. 104-121, October, 2016.

[21] S. Mirjalili, A. Lewis, The whale optimization algorithm, *Advances in engineering software*, Vol. 95, pp. 51-67, May, 2016.

[22] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, *Soft Computing*, Vol. 23, No. 3, pp. 715-734, February, 2019.

[23] P.-C. Song, S.-C. Chu, J.-S. Pan, H. Yang, Simplified phasmatodea population evolution algorithm for optimization, *Complex & Intelligent Systems*, pp. 1-19, June, 2021.

[24] P.-C. Song, J.-S. Pan, S.-C. Chu, A parallel compact cuckoo search algorithm for three-dimensional path planning, *Applied Soft Computing*, Vol. 94, Article No. 106443, September, 2020.

[25] K. Kira, L. A. Rendell, The feature selection problem: Traditional methods and a new algorithm, *AAAI'92: Proceedings of the tenth national conference on Artificial intelligence*, San Jose, CA, USA, 1992, pp. 129-134.

[26] G. Fu, C. Sun, Y. Tan, G. Zhang, Y. Jin, A surrogate-assisted evolutionary algorithm with random feature selection for large-scale expensive problems, *International Conference on Parallel Problem Solving from Nature*, Leiden, The Netherlands, 2020, pp. 125-139.

[27] G. Panda, P. M. Pradhan, B. Majhi, Iir system identification using cat swarm optimization, *Expert Systems with Applications*, Vol. 38, No. 10, pp. 12671-12683, September, 2011.

[28] M. Abd El Aziz, A. A. Ewees, A. E. Hassanien, Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation, *Expert Systems with Applications*, Vol. 83, pp. 242-256, October, 2017.

[29] R. M. Haralick, L. G. Shapiro, Image segmentation techniques, *Computer vision, graphics, and image processing*, Vol. 29, No. 1, pp. 100-132, January, 1985.

[30] A. A. El-Fergany, H. M. Hasanien, Single and multi-objective optimal power flow using grey wolf optimizer and differential evolution algorithms, *Electric Power Components and Systems*, Vol. 43, No. 13, pp. 1548-1559, July, 2015.

[31] S. Arora, S. Singh, Node localization in wireless sensor networks using butterfly optimization algorithm, *Arabian Journal for Science & Engineering (Springer Science & Business Media BV)*, Vol. 42, No. 8, pp. 3325-3335, August, 2017.

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

[33] G. I. Sayed, A. E. Hassanien, A hybrid sa-mfo algorithm for function optimization and engineering design problems, *Complex & Intelligent Systems*, Vol. 4, No. 3, pp. 195-212, October, 2018.

[34] M. Gen, R. Cheng, Genetic algorithms and engineering optimization, *John Wiley & Sons*, 1999.

[35] S. Dereli, R. Köker, Strengthening the pso algorithm with a new technique inspired by the golf game and solving the complex engineering problem, *Complex & Intelligent Systems*, Vol. 7, No. 3, pp. 1515-1526, June, 2021.

[36] D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, *IEEE transactions on evolutionary computation*, Vol. 1, No. 1, pp. 67-82, April, 1997.

[37] G. Rudolph, Self-adaptive mutations may lead to premature convergence, *IEEE Transactions on Evolutionary Computation*, Vol. 5, No. 4, pp. 410-414, August, 2001.

[38] G. Squillero, A. Tonda, Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization, *Information Sciences*, Vol. 329, pp. 782-799, February, 2016.

[39] A.-Q. Tian, S.-C. Chu, J.-S. Pan, H. Cui, W.-M. Zheng, A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station, *Sustainability*, Vol. 12, No. 3, Article No. 767, February, 2020.

[40] J.-S. Pan, N. Liu, S.-C. Chu, T. Lai, An efficient surrogate-assisted hybrid optimization algorithm for expensive optimization problems, *Information Sciences*, Vol. 561, pp. 304-325, June, 2021.

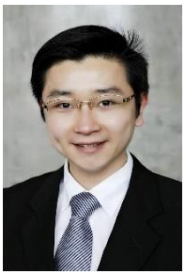
[41] D. Wei, Z. Wang, L. Si, C. Tan, Preaching-inspired swarm intelligence algorithm and its applications, *Knowledge-Based Systems*, Vol. 211, Article No. 106552, January, 2021.
[42] G. G. Wang, Y. Tan, Improving metaheuristic algorithms with information feedback models, *IEEE transactions on cybernetics*, Vol. 49, No. 2, pp. 542-555, February, 2019.
[43] R. Brown, *An account of a new genus of plants, named Rafflesia*, Taylor, 1821.
[44] R. S. Beaman, P. J. Decker, J. H. Beaman, Pollination of rafflesia (rafflesiaceae), *American Journal of Botany*, Vol. 75, No. 8, pp. 1148-1162, August, 1988.
[45] J. Liang, B. Qu, P. Suganthan, A. G. Hernández-Díaz, *Problem definitions and evaluation criteria for the cec 2013 special session on realparameter optimization*, Technical Report 201212 Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, January, 2013.
[46] J. Li, Y.-H. Yang, H. Lei, G.-G. Wang, Solving logistics distribution center location with improved cuckoo search algorithm, *International Journal of Computational Intelligence Systems*, Vol. 14, No. 1, pp. 676-692, 2021.
[47] W. Hu, An improved flower pollination algorithm for optimization of intelligent logistics distribution center, *Advances in Production Engineering and Management*, Vol. 14, No. 2, pp. 177-188, June, 2019.
[48] M.-L. Cheng, W.-B. Miao, C.-S. Zhong, Numerical simulation of insect flight, *Applied Mathematics and Mechanics*, Vol. 27, No. 5, pp. 533-538, May, 2006.

## Biographies

**Jeng-Shyang Pan** received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology in 1986, the M.S. degree in communication engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in electrical engineering from the University of Edinburgh, U.K., in 1996. He is currently the Professor of Shandong University of Science and Technology. He is the IET Fellow, U.K., and has been the Vice Chair of the IEEE Tainan Section and Tainan Chapter Chair of IEEE Signal Processing Society. His current research interest includes the information hiding, artificial intelligence and wireless sensor networks.

**Zonglin Fu** received the B.S. degree from the Shandong University of Science and Technology, Qingdao, China, in 2020. She is currently pursuing the master's degree with the Shandong University of Science and Technology, Qingdao, China. Her recent research interests include swarm intelligence and transportation problems.

**Chia-Cheng Hu** received the M.S. degree in Engineer Science from National Cheng Kung University in 1995. He received his Ph.D. in department of Computer Science and Information Engineering, National Taiwan University, Taiwan in 2005. He is currently a Professor with the College of Artificial Intelligence, Yango University.



**Pei-Wei Tsai** received his B.S., M.S. and PhD in Electronic Engineering in 2005, 2007, and 2012, respectively, in National Kaohsiung University of Applied Sciences in Taiwan. He was one of the invited speakers in 2008 in the Global COE Program at Tohoku University in Japan and an invited speaker in 2021 International Symposium on Novel and Sustainable Technology (2021 ISNST) in Taiwan. He was a visiting scholar in 2011 in the Center of Excellence in Information Assurance (CoEIA) at King Saud University in Saudi Arabia. He was the Program Committee Co-chair of the 13th International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2017) in 2017 and the Enabling and HCI Technologies (HCI) track co-chair of the 36th IEEE International Conference on Consumer Electronics (ICCE) in 2018. He is a technical committee of the Application-Specific CE for Smart Cities in IEEE Consumer Technology Society. Currently, he is a lecturer at Swinburne University of Technology in Australia. His research interests include intelligent optimization, machine learning, and data analytics.

**Shu-Chuan Chu** received the Ph.D. degree from the School of Computer Science, Engineering and Mathematics, Flinders University, Australia, in 2004. She joined Flinders University, in December 2009, after nine years at Cheng Shiu University, Taiwan. She has been a Research Fellow with the College of Science and Engineering, Flinders University, since December 2009. She has been a Research Fellow, with PhD advisor, at the College of Computer Science and Engineering, Shandong University of Science and Technology, since September 2019. Her research interests mainly include swarm intelligence, intelligent computing, and data mining.