An improved firefly algorithm with dynamic self-adaptive adjustment

Yu Li¹, Yiran Zhao²*, Yue Shang³, Jingsen Liu³

¹ Institute of Management Science and Engineering, School of Business, Henan University, Kaifeng, Henan, China, ² School of Business, Henan University, Kaifeng, Henan, China, ³ Institute of Intelligent Network Systems, Software School, Henan University, Kaifeng, Henan, China

* 104753180951@vip.henu.edu.cn

Abstract

The firefly algorithm (FA) is proposed as a heuristic algorithm, inspired by natural phenomena. The FA has attracted a lot of attention due to its effectiveness in dealing with various global optimization problems. However, it could easily fall into a local optimal value or suffer from low accuracy when solving high-dimensional optimization problems. To improve the performance of the FA, this paper adds the self-adaptive logarithmic inertia weight to the updating formula of the FA, and proposes the introduction of a minimum attractiveness of a firefly, which greatly improves the convergence speed and balances the global exploration and local exploitation capabilities of FA. Additionally, a step-size decreasing factor is introduced to dynamically adjust the random step-size term. When the dimension of a search is high, the random step-size becomes very small. This strategy enables the FA to explore solution more accurately. This improved FA (LWFA) was evaluated with ten benchmark test functions under different dimensions (D = 10, 30, and 100) and with standard IEEE CEC 2010 benchmark functions. Simulation results show that the performance of improved FA is superior comparing to the standard FA and other algorithms, i.e., particle swarm optimization, the cuckoo search algorithm, the flower pollination algorithm, the sine cosine algorithm, and other modified FA. The LWFA also has high performance and optimal efficiency for a number of optimization problems.

1 Introduction

Inspired by various biological systems in nature, many scholars have proposed effective methods that simulate natural evolution to solve complex optimization problems. One of the earliest algorithms was a genetic algorithm proposed by Professor Holland [1]. Researchers then shifted to foraging behaviour of groups of animals, such as the ant colony optimization algorithm (ACO) that simulated the behaviour of ants [2]. Eber proposed particle swarm optimization (PSO) based on bird predation behaviour [3]. In recent years, new heuristic algorithms have been proposed. To name a few, Yang and BED proposed the cuckoo search algorithm (CS) algorithm based on the breeding and spawning of cuckoo [4]; The bat algorithm (BA) was proposed based on echolocation behaviour in bats [5]; the whale optimization algorithm (WOA)
was inspired by the hunting behaviour of humpback whales [6]; and there is also the grey wolf optimizer algorithm [7], the sine cosine algorithm (SCA) [8], the polar bear optimization algorithm (PBO) [9], etc. More swarm intelligent algorithms were created mainly because of the no free lunch (NFL) theorem [10], which states a single swarm intelligent optimization algorithm cannot solve all optimization problems. The key point of the NFL theorem is that it is meaningless to assess whether an algorithm is good without a real problem, and the performance of an algorithm must be verified by specific problems. Therefore, it is of great significance to study the performance of swarm intelligent algorithms in different application fields [11, 12].

The firefly algorithm (FA) was proposed in 2008 by Xinshe Yang, a Cambridge scholar. The algorithm is based on the luminous characteristics and attractive behaviour of individual fireflies [13]. Compared with other intelligent algorithms, the FA has the advantages of having a simple model, a clear concept, few parameters to adjust, and strong searchability. Like other algorithms, however, it could easily fall into local optimum, resulting in slow convergence speed and low convergence accuracy. As a result, many scholars have made improvements to the standard FA. Yang first introduced the Levy flight into the random part of the location updating formula of the FA and developed an FA with Levy flight characteristics [14]. Subsequently, Yang improved the quality of the FA by introducing chaos into the standard FA and increased the accuracy of the standard FA by dynamically adjusting its parameters [15]. Sharma introduced the inertia weight into the FA; this strategy can overcome the tendency of falling into local optima and can achieve a slow convergence for optimization problems [16]. Farahani and other scholars proposed a Gaussian distribution FA, which referred to an adaptive step size and improved the glow worm algorithm by improving the overall position of the FA population through Gaussian distribution [17]. An FA based on parameter adjustment is better than PSO in solving dynamic optimization problems. Sh. M. Farahani and other scholars introduced an automatic learning machine into the standard FA to adjust the algorithm’s parameters, so that the algorithm can adjust the parameter values at any time according to the environment [18]. Adiland and other scholars improved the self-adaptation of the search mechanism and parameters of individual fireflies and embedded chaotic mapping to solve a mechanical design optimization problem [19]. Carbas also used the FA to solve a steel construction design problem [20]. Tahereh and other scholars used fuzzy coefficients to adjust the parameters of the FA and balanced the local and global search capabilities of the FA [21]. A number of optimization problems of test functions and engineering optimization problems have been verified for performance of the improved FA [22]. The FA is widely used in many fields, especially in computer and engineering fields, such as routine optimization [23, 24], robot path planning [25] and image processing [26–28]. Many scholars around the world have conducted in-depth studies on the theory and application of the FA and are continuously expanding its application fields.

To improve the performance of the FA, this paper proposes an improved FA based on self-adaptive inertia weight logarithmic and dynamic step-size adjust factor (LWFA). Self-adaptive logarithmic inertia weight is also introduced in the updating formula. This strategy can effectively balance the exploration and exploitation capabilities and also improve the convergence speed of the algorithm. A step-size adjust factor is also added into the LWFA to randomly change the algorithm’s step size, preventing the FA from falling into a local optimum.

Section 2 discusses the standard FA. The proposed LWFA is introduced in Section 3. In Section 4, the complexity analysis and convergence analyses are conducted to calculate the stability and validity of the LWFA. In Section 5, ten benchmark optimization test functions and IEEE CEC2010 test functions are used to evaluate the performance of the proposed algorithm.
Experimental function curve results are shown in Section 5. Lastly, the work is summarized in Section 6.

2 Standard firefly algorithm

The firefly algorithm (FA) was proposed by a Cambridge scholar Xinshe Yang in 2008 [13]. The FA is a random search algorithm based on swarm intelligence that simulates the attraction mechanism between individual fireflies in nature. To idealize certain characteristics of fireflies when constructing mathematical models of FA, the following idealization criteria are used:

1. Fireflies, both male and female, are attracted only by light intensity between groups regardless of gender;
2. The attraction of fireflies to each other is proportional to the brightness of the light;
3. The brightness of fireflies is related to the objective function value that is to be optimized.

2.1 Mathematical description and application

In the firefly algorithm, fireflies have a unique lighting mechanism and behavior, and the light they emit can only be perceived by other individual fireflies within a certain range for two reasons: the light intensity $I$ and the distance from the light source $r$ are in inverse proportion, and the light can be absorbed by air. The basic principle of the standard FA is that in a randomly distributed firefly population, fireflies with high brightness attract fireflies with low brightness toward them, and each firefly approach a firefly with high absolute brightness in a solution space, update its position, complete an iteration of positions, and find an optimal position to achieve optimization. The brightness of fireflies is related to the objective function value. If fireflies have higher brightness and better positions, they will attract more fireflies. The brightest fireflies move randomly because they cannot be attracted to any other fireflies. The relationship between fireflies and the objective function value should be established. The absolute brightness of the algorithm is expressed by the objective function value, that is, the absolute brightness $I_i$ of a firefly set at the location $x_t^i(x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iD})$ is equal to the objective function value at the location $x_t^i$, that is

$$I_i = f(x_t^i)$$

Supposing a brighter firefly attracts a firefly with low brightness, the relative brightness between the two fireflies is $I_{ij}(r_{ij}) = I_i e^{-\gamma r_{ij}^2}$. $I_i$ is expressed as the absolute brightness of firefly $i$; $\gamma$ is the light absorption coefficient, generally set as a constant; $r_{ij}$ is the cartesian distance from firefly $i$ to firefly $j$. That is

$$r_{ij} = ||x_j^i - x_t^i|| = \sqrt{\sum_{k=1}^{D} (x_{jk} - x_{tk})^2}$$

Supposing that the brightness between fireflies is proportional to the attraction, the attraction is $\beta_{ij}$,

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

$\beta_0$ is the largest attraction, and generally $\beta_0 = 1$.

The updating formula of the algorithm is as follows:

$$x_{i+1}^j = x_t^j + \beta_{ij}(r_{ij}) \times (x_t^j - x_t^i) + \alpha \delta_i$$
Where, \( t \) is the number of iterations, \( \delta_i \) is a random number derived from a uniform distribution, Gaussian distribution, or other distribution, and \( \alpha \) is a random term coefficient.

The flow chart of the standard firefly algorithm (FA) is shown in the Fig 1:

3 The proposed firefly algorithm

Due to the shortcomings of the standard FA, Yang optimized a test function with singularities with FA to achieve better optimization performance. The results show that the FA can effectively solve this kind of global optimization problem, and the FA was successfully applied to the global optimization problem of pressure piping design. However, the parameters in the standard FA are set in advance, which will lead to premature convergence of the algorithm, or the algorithm cannot converge due to improper parameter settings. Hence, the standard FA needs to be improved to achieve better optimization performance.

3.1 Parameter analysis of firefly algorithm

The standard FA updates positions based on the attraction between high-brightness fireflies and low-brightness fireflies. Since the visual range of fireflies is limited, low-brightness fireflies can only find mobile higher brightness fireflies within their visual range. It should be noted that the light absorption coefficient \( \gamma \) is an important parameter affecting the degree of
attraction between fireflies. When $\gamma \rightarrow 0$, the attraction $\beta \rightarrow \beta_0$. The attraction between fireflies will not change with a change in distance, that is to say, a flash firefly can be seen from anywhere. When $\gamma \rightarrow \infty$, $\beta \rightarrow 0$. When the attraction between fireflies is almost zero, individual fireflies move randomly, that is to say, individual fireflies are myopia, which is equivalent to flying in a fog environment; they cannot see each other. The attraction algorithm of fireflies is between these two extremes. When the light absorption coefficient gets a better value (usually 1.0), the distance between two fireflies is the distance that affects the attraction between the two fireflies. If the distance $r_{ij} \rightarrow \infty$ between two fireflies that are attracted to each other, the attraction $\beta \rightarrow 0$. Therefore, the basic conditions of the FA in the optimization process are as follows:

1. The distance between attracted low-brightness fireflies and actively high-brightness fireflies should not be too large;

2. There must be a difference in brightness between individual fireflies, with at least one firefly having a higher brightness to attract other fireflies for location updates. There is greater randomness in the distance between individual fireflies that are attracted to each other.

3.2 The improvement of firefly algorithm

3.2.1 The minimum attraction. By analysing the parameters of the FA, where the light absorption coefficient $\gamma$ is too large or where the light absorption coefficient $\gamma$ is a fixed value, the optimization interval become too large, which will easily result in the attraction between individual fireflies to approach zero and individual fireflies to lose attraction between each other. To avoid the distance between individual fireflies from being too far, the attraction between them may be almost zero. This paper introduces a minimum attraction to improve the standard FA to prevent individual fireflies from randomly moving. At this time,

$$\beta_0(r_{ij}) = \beta_{\min} + (\beta_0 - \beta_{\min})e^{-r_{ij}^2}$$  \hspace{1cm} (5)$$

$\beta_0 = 1.0, \beta_{\min} \in [0,1]$. So, even if the distance between fireflies is too far, $e^{-r_{ij}^2} \rightarrow 0$, the attraction between them can be the $\beta_{\min}$.

To validate the minimum attraction strategy can solve the shortcomings of the firefly algorithm, some test functions (D = 30) are selected to compare the FA with the algorithm which only obtain the minimum attraction strategy (LWFA-1). The comparison results of best value are shown in Table 1, and this paper also selects two benchmark functions for experiments and draws some conclusions. Fig 2 is the convergence curves of unimodal functions $f_2$. Fig 3 is the convergence curves of multimodal functions $f_7$. The population number (N) is 30, and the maximum number of iterations (T) is 1000. All algorithms ran 30 times independently on the test functions.

In the Table 1, the accuracy of the LWFA-1 is greatly reduced. Compared with FA, the LWFA-1 have better performance to solve optimization problems. The comparison results can

| Function | LWFA-1 | FA | Function | LWFA-1 | FA |
|----------|--------|----|----------|--------|----|
| $f_1$    | 0.1782 | 0.3122 | $f_6$    | 0.0556 | 1508.42 |
| $f_2$    | 3.3580E-04 | 103.77 | $f_7$    | 9.4238E-05 | 177.31 |
| $f_3$    | 60.6925 | 288.38 | $f_8$    | 5.163 | 446.17 |
| $f_4$    | 0.0075 | 1.1068 | $f_9$    | 0.0502 | 6.6831 |
| $f_5$    | 0.0050 | 11.782 | $f_{10}$ | 0.0666 | 190.87 |

https://doi.org/10.1371/journal.pone.0255951.t001
illustrate the minimum attraction strategy is effect in the LWFA-1. From the Figs 2 and 3, the convergence curve of the FA is basically a straight line and will not change with the increase of iteration times, which indicates that the FA cannot jump out of the local optimal solution. By contrast, converge curve of LWFA-1 is smoother. Therefore, this strategy can help the original firefly algorithm jump out of the local optimal solution and improve the convergence accuracy of the FA.

3.2.2 Self-Adaptive inertia weight based on logarithmic decrement. The setting of inertia weight $w$ can balance the global search ability and local optimization ability to some extent in the iterative process of FA. Initially, the setting of inertia weight is generally linear inertia weight. Based on the motion rule of a firefly in the FA, when a low-brightness firefly is attracted by a brighter individual firefly, the early motion amplitudes of the individual firefly is larger with strong global searchability so that it can enter the process of local optimization at a faster speed. An increase in iteration time helps to bring a firefly closer to the optimal value, and its moving speed should be decreased to improve its local optimization ability and should converge quickly to avoid oscillation state in the vicinity of the optimal value, thus improving the optimization ability of the FA.

As shown in Fig 4, compared with the linear decreasing inertia weight [29], the inertia weight of sinusoidal adjustment strategy [30], and the Gaussian decreasing inertia weight [31], the logarithmic decreasing self-adaptive inertia weight can be rapidly reduced in the early stages of iteration so that the local search stage can be quickly started and the search speed can be increased. Similarly, compared with the Gaussian decreasing strategy of inertia weight, in the late stages of iteration, the value of inertia weight changes more slowly, and individual fireflies can perform local searches to reduce the occurrence of oscillation. However, Gaussian
decreasing inertia weight is almost unchanged after 500 iterations. If a firefly at this time falls into local optimum, it cannot easily jump out of the local optimum position. Therefore, the logarithmic self-adaptive inertia weight strategy is more suitable for individual firefly movement in the FA.

Fig 3. Convergence curves of multimodal functions $f_7$. https://doi.org/10.1371/journal.pone.0255951.g003

Fig 4. Inertia weight curves for different strategies. https://doi.org/10.1371/journal.pone.0255951.g004
In this paper, logarithmic decreasing self-adaptive inertia weight \([32]\) is used to meet the above requirements. The formula of logarithmic decreasing self-adaptive inertia weight is as follows:

\[
  w_t = w_1 - b \times (w_1 - w_2) \times \log_T(t)
\]  

\(b\) is a logarithmic adjustment factor, \(w_1\) is an initial value of inertia weight, \(w_2\) is a final value of inertia weight, \(t\) is the current number of iterations, and \(T\) is the highest number of iterations.

The variation range of the inertial weight coefficient may have an impact on the result of function optimization. In this paper, by reading literature and simulation experiments, the parameter is set to \(b = 1.0\), \(w_1 = 0.9\), \(w_2 = 0.4\). In this case, the optimize of the firefly algorithm is better.

To prove the inertial weight can also solve the shortcomings of the firefly algorithm, some test functions \((D = 30)\) are selected to compare the FA with the algorithm which only obtain the inertial weight \((\text{LWFA-2})\). The comparison results of best value are shown in Table 2, and this paper also selects two benchmark functions for experiments and draws some conclusions. Fig 5 is the convergence curves of unimodal functions \(f_2\). Fig 6 is the convergence curves of multimodal functions \(f_7\). The parameter Settings are the same as in Section 3.2.1.

In the Table 2, compared with FA, the LWFA-2 have better performance to solve optimization problems. From the Figs 5 and 6, the converge curve of LWFA-2 is smoother. Therefore, this strategy also can help the original firefly algorithm jump out of the local optimal solution and improve the convergence accuracy of the FA.

3.2.3 Self-adaptive and dynamic step-size. The random step in the updating formula of the standard FA is generally a random number vector of Gaussian distribution, uniform distribution, or other distribution. Yang introduced a Levy flight into the random part of the updating formula of the FA, which to a certain extent reduces the possibility of individual fireflies falling into local optimal and improves the search performance of the standard FA. At the same time, an increase in the search dimension of an individual firefly decreases the optimization accuracy of the standard FA. For high-dimensional test functions, random turbulence will easily occur and the convergence speed will decrease, and the algorithm cannot converge to an optimal value. Inspired by references \([17, 33]\), this paper introduces the step adjustment factor \(c\).

\[
  c = \theta^D \times T \times e^{(-t/T)}
\]

\(T\) is the largest iteration, \(t\) is the current iteration, \(D\) is the dimension of an individual firefly, and \(\theta \in [0,1]\). In this paper, \(\theta = 0.1\). In the operation of the algorithm, the random step size decreases with an increase in the search dimension of an individual firefly or iteration number. This can ensure that the individual firefly’s random step size change is small in a high-dimensional environment, and accurate explosion can be in a small range to find the optimal value.
Fig 5. Convergence curves of unimodal functions $f_2$.

https://doi.org/10.1371/journal.pone.0255951.g005

Fig 6. Convergence curves of multimodal functions $f_7$.

https://doi.org/10.1371/journal.pone.0255951.g006
with high precision. To prove that the dynamic step-size strategy can improve the effectiveness of the FA in different dimensions, three dimensions (D = 10/30/100) are used for comparative experiments. This paper selects multimodal function $f_2$ and unimodal function $f_7$ for simulation test on the three dimensions; the population number (N) is 30, and the maximum number of iterations (T) is 1000. Experimental results show that the step adjustment factor can greatly improve the optimization accuracy of the FA. The experimental results are shown in Fig 7.

### 3.3 The procedures for the realization of LWFA

1. In the initial stage, set the following parameters $\gamma$, $\beta_0$, $\beta_{\min}$ and $\alpha$, firefly population number $N$, and maximum iteration number $T$, and randomly generate the initial position $x_i^t (i = 1, 2, \ldots, n)$ of fireflies in the optimization interval;

2. Substitute the position vector $x_i^t (i = 1, 2, \ldots, n)$ of fireflies into the objective function $f(x)$ to obtain the initialization brightness $I_i$ of fireflies, and compare and analyze results to obtain the current global optimal brightness $I_{\text{best}}$ and individual optimal positions.

3. Update the phase of the fireflies’ locations. The attraction between individual fireflies $\beta_{ij}$.

4. Update the formula according to the position of fireflies. A firefly with high initialization brightness attracts a firefly with low initialization brightness to finalize position updating. The updated formula after improvement is as follows:

$$x_i^{t+1} = w_i x_i^t + \beta_{ij} (r_{ij}) \times (x_j^t - x_i^t) + \alpha \delta_{ij}$$

$$\delta_{ij} = c \ast \text{rand}(1, d)$$

$$c = \theta^D \ast T \ast e^{-t/T}$$

---

**Fig 7.** The convergence curves of the function in different dimensions.

https://doi.org/10.1371/journal.pone.0255951.g007
5. After updating its position, a firefly generates a new position vector that replaces the old position vector in an objective function, completes the brightness update, and reorders its current brightness to obtain the current global optimal value.

6. Update $w$, $c$ and $t$. Generally, the stop condition for the iteration in the FA is that the number of iterations reaches a predefined number or that the global optimal result is sought to achieve the required accuracy.

The flow chart of the LWFA is shown in the Fig 8:
4 Theoretical analysis of improved firefly algorithm

4.1 The complexity analysis

The complexity of a swarm intelligence algorithm reflects the performance of the algorithm from other different aspects. If the algorithm needs an infinite amount of time to find a global optimal solution, the uptime availability is not included. Therefore, the computational complexity of the algorithm is very significant and crucial. Suppose the population size of the SCA is \( N \), the maximum number of iterations is \( T \), and the dimension of the decision variable is \( D \).

In the standard FA, the computational complexity of the initialization phase is \( O(N) \), the positions update of fireflies is \( O(T\cdot N \cdot D) \), and checking for fireflies positions outside the search space is \( O(T\cdot N) \). Hence, the computational complexity of the FA for the global search for an optimal position is \( O(N) + O(T\cdot N \cdot D) + O(T\cdot N) \).

Similarly, the computation complexity of the improved FA in the initialization phase is \( O(N) \). The improvement of the minimum attraction is \( O(T\cdot N) \), inertia weight based on logarithmic decrement \( w(t) \) is \( O(T\cdot N) \), and self-adaptive and dynamic step-size is \( O(T\cdot N \cdot D) \). The particle positions update is also \( O(T\cdot N \cdot D) \), and checking for particle positions outside the search space is \( O(T\cdot N) \). Therefore, the computational complexity of the LWFA to achieve an optimal particle position is \( O(N) + O(3\cdot T\cdot N) + O(2^2 \cdot T\cdot N \cdot D) \).

Although the computation complexity of the improved FA (LWFA) is higher than that of the standard FA, both of them are in the same order of magnitude.

4.2 The convergence analysis

Convergence analysis is a key factor to evaluate the convergence of swarm intelligence algorithms. As the number of iterations increases, errors between optimal results and theoretical optimal values become minute; eventually, swarm intelligence algorithms approach a fixed value. In the FA, the position update phase of fireflies determines whether the algorithm can converge to a global optimal. Many scholars have used different methods to analyze the convergence of swarm intelligence algorithms. In this paper, the LWFA is analyzed using a second-order nonhomogeneous differential equation.

Using the updating mechanism of the LWFA, we can easily find that the position update of the FA is conducted on a dimension-by-dimensional basis, and each dimension is independent. For the convenience of analysis, this paper simplifies the algorithm by choosing a one-dimensional analysis. From the updating formula (8), weight \( w \) is the constant coefficient \( a \), \( \beta_{ij} \) is the constant coefficient \( b \), and \( \alpha \delta_{ij} \) is the constant coefficient \( d \); \( x_j(t) \) is the current optimal position of a firefly, denoted as \( g_b \), and the optimal location of the next iteration is denoted as \( p_b \). Thus, the updating formula (8) is reduced as follows:

\[
x(t + 1) = ax(t) + b(g_b - x(t)) + d \tag{10}
\]

\[
x(t + 2) = ax(t + 1) + b(p_b - x(t + 1)) + d \tag{11}
\]

Formulations (10–11) can be obtained as:

\[
x(t + 2) + (1 + r)x(t + 1) + rx(t) = bg_b + bp_b + 2d \tag{12}
\]

\[
r = b - a \tag{13}
\]

Its characteristic equation is \( \lambda^2 + (1 + r)\lambda + r = 0 \), \( \Delta = (1+r)^2 - 4r = (r-1)^2 \geq 0 \).

Therefore, the convergence process of LWFA needs to consider the following two conditions:
1) When \( \Delta = 0 \), the characteristic equation has two same real roots, \( \lambda_1 = \lambda_2 = \frac{-(1+i)\sqrt{\Delta}}{2} = -1 \), and \( x(t) = (A_0 + A_1 t)\lambda^t \); \( A_0, A_1 \) are undetermined coefficients, the result is calculated as:

\[
\begin{align*}
A_0 &= x(0) \\
A_1 &= (r - 1)x(0) - bg_a - d
\end{align*}
\]

2) When \( \Delta > 0 \), the characteristic equation has two different real roots, \( \lambda_1 = \lambda_2 = \frac{-(1+i)\pm\sqrt{\Delta}}{2} \), and \( x(t) = A_0 + A_1 \lambda_1^t + A_2 \lambda_2^t \); \( A_0, A_1, A_2 \) are undetermined coefficients, and the result is calculated as follows:

\[
\begin{align*}
A_0 &= x(0) - A_1 - A_2 \\
A_1 &= \frac{\lambda_2 x(0) - (1 + \lambda_2)x(1) + x(2)}{(\lambda_2 - \lambda_1)(1 - \lambda_1)} \\
A_2 &= \frac{\lambda_1 x(0) - (1 + \lambda_1)x(1) + x(2)}{(\lambda_2 - \lambda_1)(1 - \lambda_2)}
\end{align*}
\]

Based on the convergence analysis, if the LWFA converges iteratively, the following two conditions must be met [34].

1. If \( t \to \infty \), \( x(0) \) has its maximum and tend to finite value.
2. \( \| \lambda_1 \| < 1 \) and \( \| \lambda_2 \| < 1 \)

The calculated results are as follows:

When \( \Delta = 0 \), the convergent domain is \( r = 1 \);

When \( \Delta > 0 \), the convergent domain is \( \frac{-(1+i)\pm\sqrt{\Delta}}{2} < 1 \Rightarrow -1 < b - a \leq 1 \), Thus,

\[
\begin{align*}
b - a &\leq 1 \\
b - a &> -1
\end{align*}
\]

According to Eq (5), (6) and (8), the \( b \in (0,1) \), \( a \in (-1,2) \Rightarrow \) (t) \( \in (-1,2) \).

In the whole iteration process of the improved algorithm, the inertia weight region is between 0.4 and 0.9. Therefore, the range of inertia weight conforms to the convergent domain (-1,2), thus the LWFA can converge to a global optimal solution in the iteration.

5 Application of improved firefly algorithm in function optimization

To evaluate whether the improved FA has advantages in various applications, two sets of benchmark test functions, ten classical benchmark functions with different dimensions (\( D = 10, 30, \) and \( 100 \)) and standard IEEE CEC2010 functions have been used. In this paper, PSO, CS SCA, FPA, FA, the other improved FA and the LWFA are respectively used to conduct simulation experiments, and the experimental results are compared. Based on the experimental results, the improved FA in this paper shows good optimization performance in terms of optimization accuracy and convergence speed.

5.1 The benchmark test functions set 1

In this section, ten classical benchmark functions are used as test functions to evaluate the performance of the improved FA algorithm [35]. These functions represent different optimization problems. The specific test functions are as follows:
1. Schaffer function:

\[
    f_1(x) = 0.5 + \frac{\sin^2 \left( \sum_{i=1}^{D} x_i^2 \right) - 0.5}{[1 + 0.001 \times \sum_{i=1}^{D} x_i^2]^2}, \quad x \in [-10, 10]
\] (14)

This function’s theoretical optimal value is 0, and the optimal position is at (0, . . . , 0). This function is continuous, differentiable, non-separable, non-scalable, and unimodal.

2. Sphere function:

\[
    f_2(x) = \sum_{i=1}^{D} x_i^2, \quad x \in [-100, 100]
\] (15)

This function’s minimum value is 0. This function is a classical high-dimensional unimodal function. Its optimal position point (0, . . . , 0) is in the center of the brim, and the relative exploration area is very limited.

3. Rastrigin function:

\[
    f_3(x) = \sum_{i=1}^{D} [x_i^2 - 10 \cos(2\pi x_i) + 10], \quad x \in [-5.12, 5.12]
\] (16)

This function obtains its optimum value 0 at (0, . . . , 0). This function is a multimodal function, which has ten minimum values in solutions. The peak shape of the function fluctuates in volatility. It is rather difficult to explore global search.

4. Griewank function:

\[
    f_4(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1, \quad x \in [-100, 100]
\] (17)

This function obtains its minimum value 0 at (0, . . . , 0). This function is continuous, differentiable, non-separable, scalable, and multimodal.

5. Ackley function:

\[
    f_5(x) = -20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^{D} \cos(2\pi x_i) \right) + 20 + e, \quad x \in [-35, 35]
\] (18)

This function obtains its minimum value 0 at (0, . . . , 0). This function is continuous, differentiable, non-separable, scalable, and multimodal. It has several local optimal values in its search space.

6. Sum-Squares function:

\[
    f_6(x) = \sum_{i=1}^{D} i x_i^2, \quad x \in [-10, 10]
\] (19)

This function’s theoretical optimal value is 0, and its optimal position point is at (0, . . . , 0). This function is continuous, differentiable, separable, scalable, and unimodal.
7. Zakharov function:
\[
f_7(x) = \sum_{i=1}^{D} x_i^2 + \left( \frac{1}{2} \sum_{i=1}^{D} i x_i \right)^2 + \left( \frac{1}{2} \sum_{i=1}^{D} i x_i \right)^4, \quad x \in [-5, 10]
\]
This function obtains its minimum value 0 at (0, ..., 0). This function is continuous, differentiable, non-separable, scalable, and multimodal.

8. Schwefel’s problem 1.2 function:
\[
f_8(x) = \sum_{i=1}^{D} \left( \sum_{j=1}^{i} x_j \right)^2, \quad x \in [-10, 10]
\]
This function obtains its minimum value 0 at (0, ..., 0). This function is continuous, differentiable, non-separable, scalable, and unimodal.

9. Schwefel’s problem 2.21 function:
\[
f_9(x) = \max_i \{|x_i|, 1 \leq i \leq D \}, \quad x \in [-100, 100]
\]
This function gets the minimum value 0 at (0, ..., 0). This function is continuous, non-differentiable, separable, scalable, and unimodal.

10. Schwefel’s problem 2.22 function:
\[
f_{10}(x) = \sum_{i=1}^{D} |x_i| + \prod_{i=1}^{D} |x_i|, \quad x \in [-10, 10]
\]
This function obtains its minimum value 0 at (0, ..., 0). This function is continuous, differentiable, non-separable, scalable, and unimodal.

5.2 Comparison with other improved firefly algorithms on set 1
In this section, some improved firefly algorithms are selected to prove the optimization ability of LWFA. The results of other improved FA are extracted from the original literatures. The comparison results of LWFA, VVSFA [36], IWFA [37], and CLFA [38] are shown in the Table 3.

From the Table 3, the average value is contained. In the same dimension \(D = 100\), the accuracy of LWFA is obviously higher than other improved algorithms. When solve the high dimension optimization problems, compared with other FA, the LWFA can overcome the curse of dimensionality. Therefore, the LWFA is competitive in solving optimization problems.

Table 3. Comparison results of LWFA and other improved FA with \(D = 100\).

| Function | VVSFA  | IWFA  | CLFA  | LWFA   |
|----------|--------|-------|-------|--------|
| \(f_2\)  | 84.843 | 1.0028| 7.5021| 5.42E-201|
| \(f_3\)  | 470.286| 177.324| 561.675| 0      |
| \(f_4\)  | 0.74506| 0.02078| 0.16899| 0      |
| \(f_5\)  | 4.84312| 0.84394| 3.60037| 0      |
| \(f_6\)  | 3853.70| 51.513 | 364.380| 2.04E-199|
| \(f_8\)  | 3828.88| 50.2529| 374.359| 2.02E-101|
| \(f_{10}\)| 64.584 | 1.64239| 9.0614 | 5.81E-100|

https://doi.org/10.1371/journal.pone.0255951.t003
5.3 Comparison with other state-of-art algorithms on set 1

For the ten test functions, $f_2, f_6, f_8, f_9,$ and $f_{10}$ are unimodal functions, and $f_1, f_3, f_4, f_5,$ and $f_7$ are multimodal functions. The dimensions of the test functions are set to 10, 30, and 100, respectively. The test results are obtained using PSO, CS algorithm, SCA, FPA, FA, and LWFA.

The experimental environment is MATLAB 2014a, the operating system is windows 7 flagship, 4.00 GB of running memory and the processor is Intel (R) Core (TM) i3 – 2350M CPU @ 2.30 GHz. The parameters of the six algorithms PSO [33], the CS [34], the FPA [35], the SCA [8], the FA [36], and the LWFA are as shown in Table 4.

To ensure fairness and comparability in the simulation experiments, the same experimental parameters are defined for all the algorithms. Each algorithm will run 30 times to solve each test function, and the test results will be recorded; the optimal solution of the objective function value, the worst solution, the average value, and standard deviation are obtained in this dimension, as shown in Tables 5–7.

Based on Tables 5–7, when $D = 10/30/100$, the LWFA significantly improves convergence accuracy. For either unimodal functions or multimodal functions that have multiple extreme points, as the dimension increases, the LWFA obtains better optimal performance. When $D = 100$, the accuracy of LWFA is more than 100 magnitudes higher than when $D = 10$. The LWFA can also solve optimization problems with a small standard deviation, indicating that the improved algorithm has greater robustness to solve high-dimension problems.

By comparing Tables 5–7, we can find that when $D = 10$, the LWFA can obtain optimal value 0 for functions $f_1, f_3,$ and $f_4$, while the optimization accuracy of the other four algorithms (PSO, the CS, FPA, and FA) is not high. For functions $f_2, f_5-f_{10}$ the LWFA proposed in this paper is more accurate than PSO, the CS, and FA in terms of optimal values and standard deviation. The LWFA has higher convergence accuracy than the SCA except for the functions $f_2, f_5-f_6$. When $D = 30$, the standard CS is used to solve $f_6$ and $f_{10}$, and it failed many times in the convergence experiments. Moreover, when $D = 100$, PSO is unable to solve the convergence problem for the function $f_7$. The CS also loses convergence for functions $f_3, f_6, f_8,$ and $f_{10}$. When $D = 100$, the LWFA can quickly reach the optimal value 0 for functions $f_1, f_3, f_4,$ and $f_7$. For other functions, the optimization accuracy and stability of PSO, the CS, FPA, and FA is lower than the performance of LWFA proposed in this paper. Particularly, the performance of the SCA decreases rapidly as the dimension increases, but the LWFA has a faster convergence speed and higher accuracy in 30 high-dimensional independent runs ($D = 100$).

5.4 Experiments and results analysis on set 2

In this section, to further analyze the performance of the improved FA, IEEE CEC2010 functions [39] are used in simulation experiments. PSO, the SCA, the FPA, and the standard FA are used to evaluate the performance of the LWFA. The simulation environment is similar to

| Algorithm | Parameters values |
|-----------|-------------------|
| PSO       | $c_1 = c_2 = 1.494, \ w \in [0.4, 0.9]$, $V_{max} = 1$, $V_{min} = -1$, $N = 30$, $T = 1000$ |
| FPA       | $P = 0.2$, $N = 30$, $T = 1000$ |
| SCA       | $a = 2$, $N = 30$, $T = 1000$ |
| CS        | $P_A = 0.25$, $N = 30$, $T = 1000$ |
| FA        | $\alpha = 1.0$, $\beta_0 = 1.0$, $\gamma = 1.0$, $N = 30$, $T = 1000$ |
| LWFA      | $\alpha = 1.0$, $\beta_0 = 1.0$, $\beta_{max} = 0.2$, $\gamma = 1.0$, $N = 30$, $T = 1000$ |

https://doi.org/10.1371/journal.pone.0255951.t004
that of set 1. The parameters of the five algorithms are also shown in Table 4. To realize significant solutions, each function is run 30 times independently, and all the simulation experiments are tested under the same conditions. The solutions of all the algorithms are shown in Table 8. Based on Table 8, it can be easily analyzed that the LWFA outperformed the standard FA. Compared with the other algorithms (PSO, the FPA, FA, and SCA), the improved FA achieved superior optimization performance.

5.5 Convergence curve analysis

To clearly show the convergence speed and accuracy of the LWFA, this paper will plot the convergence curves of the LWFA and the five other algorithms (PSO, the CS, FPA, SCA, and FA) with D = 10 on ten benchmark functions, as shown in Fig 9. When the dimension D = 30/100, the convergence curve of function $f_2$, $f_5$, and $f_6$ are shown in Fig 10. In these graphs, the
horizontal axes represent the iteration $T = 1000$, and the robustness of each algorithm is demonstrated in the vertical axes.

Fig 9 shows the performance of the LWFA. The other algorithm (PSO, the CS, FPA, and FA) have similar solutions based on the convergence curves. Compared with the SCA, the LWFA can obtain a higher accuracy for the function $f_2$, $f_5$, and $f_6$. However, the performance of the SCA drops rapidly as the dimension increases. Fig 10 shows that the LWFA obtains a higher optimal value than the SCA.

Overall, based on the above comparison results above, we can see that the improved FA proposed in this paper is better than the other algorithms in the same dimension. When the comparison dimension is 30/100, the convergence rate of the LWFA is faster than the other algorithms in the convergence graph of the same function and has a vertical decline state. In different dimensions, the optimization accuracy of PSO, the CS, FPA, SCA, and FA will decrease with an increase in dimension. However, the optimization accuracy of the LWFA
also increases gradually and shows good optimization performance for each function with an increase in its dimensions.

### 6 Conclusions

In this paper, the biological principle and mathematical description of the standard FA are presented, and the improvement of the FA and the research of its application fields are summarized. To achieve minimum attraction, an FA based on self-adaptive logarithmic inertia weight is proposed. The step adjustment factor is added to the random step term to dynamically adjust the random step, thus greatly improving the optimization performance of the FA. To evaluate the performance of the LWFA, multiple simulation experiments were conducted. In
the first experiment set, ten benchmark test functions and IEEE CEC2010 standard test functions are used to compare the performance of state-of-the-art algorithms with that of the LWFA. The comparisons with PSO, the FPA, SCA, and CS show that the improved FA increases the accuracy of the solutions and significantly enhances the robustness of the solutions for the test functions, particularly for high-dimensional test functions. The

Table 8. The result of CEC2010 standard functions for different algorithms (D = 30).

| function | Index | PSO       | FPA         | SCA         | FA           | LWFA        |
|----------|-------|-----------|-------------|-------------|--------------|-------------|
| f1       | Mean  | 1.1758E+08| 2.8482E+07 | 3.2023E+08 | 3.2758E+09  | 1.2335E+09 |
|          | St.dev| 1.2266E+08| 9.8650E+06 | 8.3177E+07 | 6.6525E+08  | 2.9624E+08 |
| f2       | Mean  | 194.4268  | 270.7726    | 313.9159    | 429.0046     | 487.6912    |
|          | St.dev| 22.5446   | 24.3295     | 18.1015     | 22.3716      | 18.0347     |
| f3       | Mean  | 6.9845    | 11.5825     | 8.4616      | 9.9132       | 20.6573     |
|          | St.dev| 5.0723    | 0.8924      | 0.4262      | 0.1842       | 0.1956      |
| f4       | Mean  | 1.9000E+09| 1.1098E+08 | 1.6428E+09 | 8.6480E+10   | 5.2756E+09 |
|          | St.dev| 2.5837E+09| 3.4122E+07 | 6.3352E+08 | 1.9106E+11   | 1.4965E+09 |
| f5       | Mean  | 843.4820  | 341.8182    | 2.1540E+03  | 1.3410E+04   | 5.1709E+03 |
|          | St.dev| 159.9266  | 28.5857     | 1.3582E+03  | 9.2452E+04   | 1.3000E+03 |
| f6       | Mean  | 1.6696E+06| 29.0407     | 8.9861E+04  | 3.2047E+06   | 5.8600E+05 |
|          | St.dev| 1.9389E+06| 7.4758      | 6.9757E+04  | 3.1767E+06   | 4.1916E+05 |
| f7       | Mean  | 4.1934E+07| 3.5154E+04  | 2.3353E+05  | 7.4838E+07   | 3.3784E+06 |
|          | St.dev| 1.0738E+05| 8.0394E+03  | 7.2418E+04  | 8.5399E+07   | 3.5215E+06 |
| f8       | Mean  | 4.5953E+07| 1.7543E+05  | 3.2598E+06  | 7.9379E+11   | 1.1232E+07 |
|          | St.dev| 2.7810E+04| 1.4545E+04  | 2.3914E+04  | 5.7614E+04   | 8.4959E+04 |
| f9       | Mean  | 112.8304  | 158.8623    | 179.4504    | 303.8515     | 268.3395    |
|          | St.dev| 34.0315   | 16.8684     | 19.1803     | 29.1816      | 20.2220     |
| f10      | Mean  | 40.7407   | 28.7450     | 34.2656     | 64.0681      | 51.6848     |
|          | St.dev| 5.4358    | 2.0124      | 3.8885      | 4.3817       | 2.6954      |
| f11      | Mean  | 2.7810E+04| 1.4545E+04  | 2.3914E+04  | 5.7614E+04   | 8.4959E+04 |
|          | St.dev| 1.8096E+03| 3.9676E+03  | 2.2166E+03  | 4.5165E+03   | 8.8299E+03 |
| f12      | Mean  | 4.4534E+07| 4.1246E+05  | 4.8125E+06  | 4.4521E+09   | 1.3288E+08 |
|          | St.dev| 7.5115E+08| 3.1061E+05  | 1.8795E+06  | 2.0723E+09   | 1.2952E+08 |
| f13      | Mean  | 4.1494E+07| 6.0881E+07  | 2.5674E+08  | 1.5656E+09   | 9.1115E+08 |
|          | St.dev| 5.4354E+07| 8.4243E+07  | 9.2009E+07  | 2.0959E+08   | 1.0834E+08 |
| f14      | Mean  | 46.9529   | 92.8979     | 99.1639     | 166.1725     | 173.0482    |
|          | St.dev| 10.6700   | 9.1652      | 12.0043     | 18.0469      | 11.9303     |
| f15      | Mean  | 46.6296   | 44.6271     | 46.9118     | 71.5064      | 73.7265     |
|          | St.dev| 8.8237    | 4.1806      | 3.2170      | 3.9258       | 2.6880      |
| f16      | Mean  | 5.2609E+04| 1.4721E+04  | 2.9294E+04  | 1.0348E+05   | 9.4555E+04 |
|          | St.dev| 1.4283E+04| 4.0175E+03  | 5.4200E+03  | 1.0450E+04   | 1.2981E+04 |
| f17      | Mean  | 1.0972E+10| N/A         | 3.5760E+08  | 2.5780E+10   | 7.0092E+09 |
|          | St.dev| 2.8496E+09| N/A         | 1.5551E+08  | 6.6432E+09   | 2.2750E+09 |
| f18      | Mean  | 3.1396E+04| 1.6728E+04  | 3.4017E+04  | 1.1001E+05   | 1.1440E+05 |
|          | St.dev| 7.3101E+03| 5.7618E+03  | 6.6369E+03  | 2.1627E+04   | 1.9785E+04 |
| f19      | Mean  | 7.1139E+09| 2.4342E+09  | 2.3806E+09  | 2.2325E+10   | 4.6233E+10 |
|          | St.dev| 8.4790E+08| 1.0194E+09  | 7.0705E+08  | 4.1391E+09   | 7.4252E+09 |

https://doi.org/10.1371/journal.pone.0255951.t008
The experimental results in this study have convincingly shown and confirmed that the LWFA achieves high performance in solving optimization problems. The application of the improved FA to real-world engineering problems and multi-objective optimization will be done in future work.

Fig 9. Convergence curve of functions $f_1$-$f_{10}$ in $D = 10$.

https://doi.org/10.1371/journal.pone.0255951.g009
Fig 10. Contrast convergence curve of LWFA and SCA in D = 30/100.

https://doi.org/10.1371/journal.pone.0255951.g010

Author Contributions
Conceputalization: Yu Li.
Data curation: Yiran Zhao, Yue Shang.
Funding acquisition: Jingsen Liu.
Investigation: Yiran Zhao.
Methodology: Yue Shang, Jingsen Liu.
Project administration: Jingsen Liu.
Software: Jingsen Liu.
Supervision: Yu Li.
Writing – original draft: Yiran Zhao.
Writing – review & editing: Yu Li.

References
1. Holland John H. (1973). Genetic Algorithms and the Optimal Allocation of Trials. Siam Journal on Computing, 2(2):88–105.
2. Dorigo M, Maniezzo V, Colomi A. (2002). Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems Man and Cybernetics*, 26(1):29–41.
3. Kennedy J, Eberhart R C. (2002). Particle swarm optimization. *International conference on networks*, pp.1942–1948.
4. Yang X, Deb S. (2009). Cuckoo Search via Lévy flights. *Nature and biologically inspired computing*, pp.210–214.
5. Pelta D A, Krasnogor N. (2009). Nature-inspired cooperative strategies for optimization. *Nature inspired cooperative strategies for optimization*, pp.982–989.
6. Mirjalili S, Lewis A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 2016, 95(95):51–67.
7. Mirjalili S, Mirjalili S M and Lewis A. (2014) Grey wolf optimizer. *Advances in Engineering Software*, 69:46–61.
8. Polap D, Niak M W. (2017). Polar Bear Optimization Algorithm: Meta-Heuristic with Fast Population Movement and Dynamic Birth and Death Mechanism. *Symmetry*, 9(10):203–223.
9. Wolpert D H, Macready W G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, pp.67–82.
10. Husnain G, Anwar S. (2021). An intelligent cluster optimization algorithm based on Whale Optimization Algorithm for VANETs (WOACNET). *PLOS ONE*, 16(4): e0250271. https://doi.org/10.1371/journal.pone.0250271 PMID: 33882105
11. Farahani S M, Abshouri A A, Nasiri B, et al. (2011). A Gaussian Firefly Algorithm. *International Journal of Machine Learning and Computing*, pp.448–453. https://doi.org/10.7763/IJMLC.2011.V1.67
12. Farahani S M, Abshouri A A, Nasiri B, et al. (2012), Some Hybrid models to Improve Firefly Algorithm Performance. *International journal of artificial intelligence*, pp.97–117.
13. Baykasoglu A, Ozsoydan F B. (2015). Adaptive firefly algorithm with chaos for mechanical design optimization problems. *Applied Soft Computing*, pp.152–164.
14. Carbas S. (2020). Enhanced Firefly Algorithm for Optimum Steel Construction Design. *Applications of Firefly Algorithm and its Variants*. Springer, pp.119–146.
15. Chandra wati T B, Sari R F. (2018). A Review of Firefly Algorithm for Path Planning, Vehicle Routing and Traveling Salesman Problems. *International conference on electrical engineering and informatics*, pp.30–35.
16. Rajmohan S, Natarajan R. (2019). Group influence based improved firefly algorithm for Design Space Exploration of Datapath resource allocation. *Applied Intelligence*, 49(6):2084–2100.
17. Altabeeb A M, Mohsen A M and Ghallab A. (2019). An improved hybrid firefly algorithm for capacitated vehicle routing problem. *Applied Soft Computing*, pp.105728.
18. Fan T, Wang J, Feng M, et al. (2019). Application of multi-objective firefly algorithm based on archive learning in robot path planning. *International Journal of Intelligent Information and Database Systems*, 12(3):199–211.
19. Mondal A, Dey N and Ashour S A. (2019). Digital Image Processing based Cuckoo Search Algorithm and its Variants: A Comprehensive Review. *Applications of, Cuckoo Search Algorithm and its Variants*. 
28. Rajagopalan A, Modale D R, Senthilkumar R. (2020). Optimal Scheduling of Tasks in Cloud Computing Using Hybrid Firefly-Genetic Algorithm. *Advances in Decision Sciences, Image Processing, Security and Computer Vision Springer*, pp.678–687.

29. Wang F, Ye C. (2016). Location of emergency rescue center in urban areas based on improved firefly algorithm. *Industrial engineering and management*, pp.21–24.

30. Jiang C, Zhao S, Shen S, et al. (2012). Particle swarm optimization algorithm with sinusoidal changing inertia weight. *Computer Engineering and Applications*, 48(8):40–42.

31. Zhang X, Wang P, Xing J, et al. (2012). Particle swarm optimization algorithms with decreasing inertia weight based on Gaussian function. *Application Research of Computers*, 29(10): 3710–3712.

32. Wenzhi D, Xinle Y. (2015). Particle swarm optimization algorithm based on inertia weight logarithmic decreasing. *Computer engineering and applications*, pp.99.

33. Hassanzadeh T, Meybodi M R, and Shahramirad M. (2017). A New Fuzzy Firefly Algorithm with Adaptive Parameters. *International Journal of Computational Intelligence and Applications*, 16(03):1750017.

34. Liu J, Xing Y and Li Y. (2018). A Gravitational Search Algorithm with Adaptive Mixed Mutation for Function Optimization. *International journal of performability engineering*, 14(4):681–690.

35. Jamil M, Yang X. (2013). A literature survey of benchmark functions for global optimization problems. *International Journal of Mathematical Modelling & Numerical Optimisation*, 4(2):150–194.

36. Yu S, Zhu S, Ma Y, et al. (2015). A variable step size firefly algorithm for numerical optimization. *Applied Mathematics & Computation*, 263(C):214–220.

37. Zhu Q G, Xiao Y K, Chen W D, Ni C X, Chen Y. (2016). Research on the improved mobile robot localization approach based on firefly algorithm. *Chinese Journal of Scientific Instrument*, 37(2):323–329.

38. Kaveh A, Javadi S M. (2019). Chaos-based firefly algorithms for optimization of cyclically large-size braced steel domes with multiple frequency constraints. *Computers and Structures*, 214:28–39.

39. Tang K, Yao X, Suganthan P, et al. (2010). Benchmark Functions for the CEC 2010 Special Session and Competition on Large Scale Global Optimization. *University of Science and Technology of China (USTC), School of Computer Science and Technology, Nature Inspired Computation and Applications Laboratory (NICAL), Hefei, Anhui, China. Tech. Rep.*