electronics

MDPI

Article

Multi-Strategy Fusion of Sine Cosine and Arithmetic Hybrid Optimization Algorithm

Lisang Liu [1,2], Hui Xu [1,2,*], Bin Wang [1,2] and Chengyang Ke [1,2]

[1] School of Electronic, Electrical Engineering and Physics, Fujian University of Technology, Fuzhou 350118, China
[2] National Demonstration Center for Experimental Electronic Information and Electrical Technology Education, Fujian University of Technology, Fuzhou 350118, China
* Correspondence: xuhui@smail.fjut.edu.cn

Abstract: The goal was to address the problems of slow convergence speed, low solution accuracy and insufficient performance in solving complex functions in the search process of an arithmetic optimization algorithm (AOA). A multi-strategy improved arithmetic optimization algorithm (SSCAAOA) is suggested in this study. By enhancing the population's initial distribution, optimizing the control parameters, integrating the positive cosine algorithm with improved parameters, and adding inertia weight coefficients and a population history information sharing mechanism to the PSO algorithm, the optimization accuracy and convergence speed of the AOA algorithm are improved. This increases the algorithm's ability to perform a global search and prevents it from hitting a local optimum. Simulations of SSCAAOA using other optimization algorithms are used to examine their efficacy on benchmark test functions and engineering challenges. The analysis of the experimental data reveals that, when compared to other comparative algorithms, the improved algorithm presented in this paper has a convergence speed and accuracy that are tens of orders of magnitude faster for the unimodal function and significantly better for the multimodal function. Practical engineering tests also demonstrate that the revised approach performs better.

Keywords: sine chaotic mapping; arithmetic optimization algorithm; mathematical optimizer acceleration function; sine cosine optimization algorithm

Citation: Liu, L.; Xu, H.; Wang, B.; Ke, C. Multi-Strategy Fusion of Sine Cosine and Arithmetic Hybrid Optimization Algorithm. *Electronics* **2023**, *12*, 1961. https://doi.org/10.3390/electronics12091961

Academic Editors: Padma Iyenghar and Elke Pulvermüller

Received: 26 February 2023
Revised: 17 April 2023
Accepted: 19 April 2023
Published: 23 April 2023

1. Introduction

With the development of social needs, people are facing more and more complex problems in various fields, and the scale of computation is increasing day by day. Traditional optimization methods are difficult to meet the computational demand; there are defects such as too many parameters affecting the algorithm, too much reliance on gradient information, and they are difficult to implement. Intelligent optimization algorithms have received a lot of attention from scholars because they have few parameters, are easy to implement, do not contain a gradient mechanism, and have advantages in solving optimization problems and search problems of high complexity [1–5]. Common intelligent optimization algorithms include the classical Particle swarm algorithm [6], the Genetic Algorithm [7], and the Ant colony algorithm [8]. In recent years, the whale optimization algorithm [9], the slime optimization algorithm [10], the floating optimization algorithm [11], etc. have been proposed. Many intelligent algorithms have been applied to various engineering optimization, parameter tuning, and other problems [12–16]. However, due to the complexity of practical applications, no algorithm can solve the problem perfectly, so there is a need to improve and optimize the algorithms so that they can better solve real-world problems.

In the optimization improvement of heuristic algorithms, the fusion of multiple algorithms for complementary advantages is a common approach nowadays. Alwajih et al. [17]

proposed a fusion algorithm based on the binary optimization whale algorithm and the Harris Hawk optimization algorithm for increasing data dimensionality and applied the algorithm to solve the feature selection problem. Khattab et al. [18] proposed a hybrid optimization based on CRO and BFS algorithm to solve the minimum vertex coverage problem. Shokouhifar et al. [19] proposed a hybrid optimization algorithm based on fuzzy algorithm and ant colony algorithm for solving the VNF-SPR problem with the fuzzy inference system as heuristic information. Zhang et al. [20] fused the arithmetic optimization algorithm and the skyhawk optimization algorithm and introduced energy parameters and a segmented line graph to optimize the parameters. To enhance the computational performance of the algorithm in complex models, Shokouhifar et al. [21] added heuristic information and a variable neighborhood search to the WOA to improve the search performance of the algorithm.

Article [22] introduces differential evolution to enhance the ability of arithmetic optimization algorithms to develop solutions and improve the convergence accuracy and was applied to multilevel thresholding segmentation of COVID-19 CT images. Article [23] hybridized the Hunger Games algorithm with the arithmetic optimization algorithm and tested it with 23 test functions, and the global search effect was somewhat improved. In article [24], the primitive functions are added to the mathematical model of the AOA algorithm, six perturbation functions are added to the parameters MOP and MOA, respectively, and then the six variants are compared to select the best solution. The best solution is applied to solve the economic load dispatching problem of the power system. The article [25] introduces the sigmoid function in the position update formula to balance the algorithm search mode, introduces moderate reverse learning and the gray wolf information feedback mechanism to improve the convergence accuracy of the algorithm, and finally performs simulation verification on the CEC2014 test function.

In this research, we combine the sine cosine algorithm (SCA) with the arithmetic optimization algorithm (AOA) on the basis of the aforementioned literature in an effort to boost the AOA's efficiency. Initiation is made easier with the introduction of a new sine chaotic mapping. In addition, the acceleration function of the mathematical optimizer (SMOA) is reconstructed. An alternative optimization method based on sine, cosine, and arithmetic operations is proposed (SSCAAOA). Standard test functions are used to compare the algorithm to the original algorithm and other optimization algorithms; the results show that the SSCAAOA effectively improves the algorithm's performance in finding the best and speeds up the convergence speed while also improving the convergence accuracy.

2. Arithmetic Optimization Algorithm (AOA)

The Algorithm for Arithmetic Optimization, AOA, was suggested by Abualigah et al. [26] in 2021 as a brand-new category of intelligent optimization algorithm. The approach is modeled after how arithmetic operators are used to solve mathematical problems, extending the dispersion of the operators and enhancing the global search through the use of multiplication and division operations. To improve local search accuracy, local convergence is conducted utilizing additive and subtractive processes. The AOA algorithm, which has been utilized in engineering applications, does not rely on derivation and offers the benefits of simplicity, few control factors, strong performance in finding the optimum, and improved stability. However, other scholars have also enhanced it because of its strong stochasticity, sluggish convergence time, easy to slip into local optimum, and other issues.

The optimal search process of the AOA algorithm is the same as that of most intelligent optimization algorithms, in which a set of randomly generated candidate solutions are first evaluated by the objective function under some optimization rules, and the optimal solution is gradually approximated by the algorithm in one iteration.

AOA implements a global search based on the distributional properties of arithmetic operators. There are three main phases: initialization, the exploration phase, and the development phase. As can be seen in Figure 1, the algorithm uses the distribution properties of the arithmetic operators to conduct a global search.

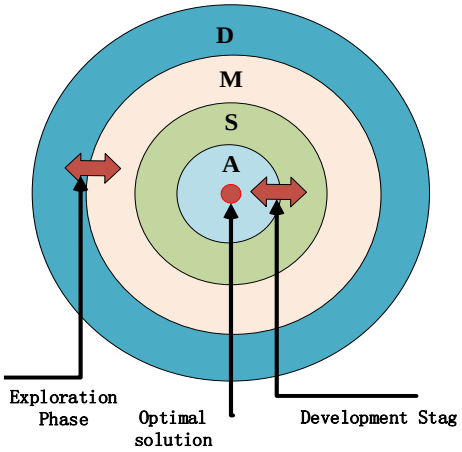

Figure 1. Arithmetic operator hierarchy.

2.1. Initialization

AOA's optimization procedure kicks off with a pool of potential solutions, denoted by the letter *X*. First, the population distribution is initialized randomly. The position vector *X* of defined individuals consists of n individuals of dimension *N*. The mathematical model is shown in Equation (1).

$$X = \begin{bmatrix} x_{1,1} & \cdots & \cdots & x_{1,j} & x_{1,n-1} & x_{1,n} \\ x_{2,1} & \cdots & \cdots & x_{2,j} & x_{2,n-1} & x_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & \cdots & x_{N-1,j} & \cdots & x_{N-1,n} \\ x_{N,1} & \cdots & \cdots & x_{N,j} & x_{N,n-1} & x_{N,n} \end{bmatrix} \tag{1}$$

Before running the algorithm, the search phase is selected, and the AOA selects the exploration or development phase by using the coefficients calculated by the mathematical optimizer acceleration function (MOA) in Equation (2). The random number r_1, $r_1 \in [0, 1]$ is taken first, and the algorithm performs a global search when $r_1 > MOA$, and a local search when $r_1 < MOA$.

$$MOA = Min + t \times (\frac{Max - Min}{T_{Max}}) \tag{2}$$

where t is the current number of iterations, Max and Min are the maximum and minimum values of the mathematical gas pedal function, which are taken as 1 and 0.2, respectively, in the original algorithm, and T_{Max} is the maximum number of iterations.

2.2. Exploration Phase

To help the population look for more possible solutions across the space, a high degree of dispersion can be achieved using multiplication (M) and division (D) operations during the exploration phase. This may cause the population to be over-dispersed and difficult to converge. However, after many iterations, the communication among the population will lead the population to a solution that is closer to the optimal solution. Taking the random number r_2, $r_2 \in [0, 1]$, the division strategy is executed when $r_2 \leq 0.5$ and the multiplication strategy is executed when $r_2 > 0.5$. The position update formula is as follows:

$$x_{i,j}(t+1) = \begin{cases} best(x_j) \div (MOP + \varepsilon) \times [(UB_j - LB_j) \times \mu + LB_j], & r_2 < 0.5, \\ best(x_j) \times MOP \times [(UB_j - LB_j) \times \mu + LB_j], & \text{otherwise}, \end{cases} \tag{3}$$

where $x_{i,j}(t+1)$ denotes the jth position from the ith solution of the current iteration, $best(x_j)$ denotes the optimal solution at the jth position, ε is a minimal number preventing the denominator from being 0, UB_j, LB_j denotes the upper and lower bounds of the jth position, respectively, and μ is a parameter that adjusts the search process and is set to 0.499, where MOP is the mathematical optimizer probability function with the following equation:

$$MOP(t) = 1 - (\frac{t}{T_{\max}})^{1/\alpha} \tag{4}$$

where α is a sensitive parameter to indicate the development accuracy during the iterative process.

2.3. Development Stage

In the development phase, additive and subtractive operators are applied. The additive operation (A) and subtractive operation (S) are less discrete but more intensive and can be further developed for the exploration phase to find out the solution set which is beneficial to approach the optimal solution quickly. The development phase is executed at $r_1 < MOA$, and then r_3 chooses which operator to execute. The following is the mathematical representation of this stage:

$$x_{ij}^{t+1} = \begin{cases} best(x_j) - MOP \times [(UB_j - LB_j) \times \mu + LB_j], & r_3 < 0.5, \\ best(x_j) + MOP \times [(UB_j - LB_j) \times \mu + LB_j], & \text{otherwise}, \end{cases} \tag{5}$$

where $r_3 \in [0, 1]$ is a random number.

3. Sine Cosine Algorithm (SCA)

Another relatively new intelligent optimization technique is the sine cosine algorithm (SCA), which was put forth by Australian researcher Mirjalili in 2016 [27]. The SCA algorithm is split into three phases: startup, exploration, and development, just as the AOA algorithm. The distinction is that the SCA method gradually approaches the optimal solution using the oscillatory properties of the sine and cosine functions by using the sine and cosine mathematical models as the optimization rules for the exploration and development stages.

The algorithm chooses which optimization rule to perform by a random parameter, r_4. The sine function model is chosen when $r_4 < 0.5$. Choose the cosine function model when $r_4 \geq 0.5$. The search phase of the method can be modulated by changing the amplitude using the adaptive parameter r_1; when $r_5 > 1$, the algorithm tends to global search and when $r_5 < 1$, using the periodicity of the sine and cosine function, the algorithm shifts its focus from a global to a local search.

The SCA algorithm position update equation is as follows:

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^t + r_5 \cdot \sin r_6 \cdot \left| r_7 p_{gj}^t - x_{ij}^t \right| & r_4 < 0.5, \\ x_{ij}^t + r_5 \cdot \cos r_6 \cdot \left| r_7 p_{gj}^t - x_{ij}^t \right| & \text{otherwise}, \end{cases} \tag{6}$$

where x_{ij}^t denotes the position of the ith individual in dimension j at the tth iteration, r_6, r_7, and r_4 are uniformly distributed random numbers, where $r_6 \in [0, 2\pi]$, $r_7 \in [0, 2]$, $r_4 \in [0, 1]$. The adaptive control parameters are given by:

$$r_5 = a \cdot (1 - \frac{t}{T_{\max}}) \tag{7}$$

where a is a constant, usually 2, $T_{\max}$ is the maximum number of iterations, and t is the current number of iterations.

4. Improved Arithmetic Optimization Algorithm (SSCAAOA)

4.1. Improved Sine Chaos Mapping Initialization

The accuracy and speed of convergence of the algorithm are somewhat affected by the population's initial distribution. If there are initial individuals in the vicinity of the optimal solution, then the algorithm will converge to the optimal solution quickly and with high accuracy. The method will easily enter a local optimum if the starting distribution is concentrated towards some local extremes, which also results in poor population variety and reduced exploration of other solutions. The original AOA algorithm uses random initialization, and the initial individuals are not uniformly distributed. This work introduces an improved chaotic mapping to conduct population initialization in order to increase the diversity of the initialized population. When compared to random initialization, chaotic mapping is a series of unpredictability produced by straightforward deterministic equations that has greater ergodicity, randomness, and population diversity, frequently outperforming pseudo-random numbers.

Many intelligent optimization strategies for replacing random number generators use Sine chaos mapping, but because the sequences generated by the traditional one-dimensional sine chaos mapping are not uniformly distributed over the phase space, an improved sine chaos mapping is proposed in the literature [28], and this improved method is used in this paper for population initialization. The equations of its system are as follows:

$$\begin{cases} d_{i+1} = \sin(\mu \pi d_i) \\ e_{i+1} = \sin(\mu \pi e_i) \\ w_{i+1} = d_{i+1} + e_{i+1} \mathrm{mod} 1 \end{cases} \tag{8}$$

where μ and w are the control parameters and iterative sequence values of the one-dimensional sine chaos mapping, respectively, and here $\mu = 0.99$.

Figure 2 displays the distribution and histogram before and after the improvement, with 1000 iterations and a distribution interval of [0, 1].

As shown in Figure 1, the distribution and histogram show that the improved sine chaos mapping initialization distribution has better uniformity and better chaos effect. It is possible for it to make the distribution of the initial solutions more uniform, maintain the diversity of the population, and prevent the population from falling into local extremes to some extent, which will improve the performance of the algorithm in finding the optimal answer.

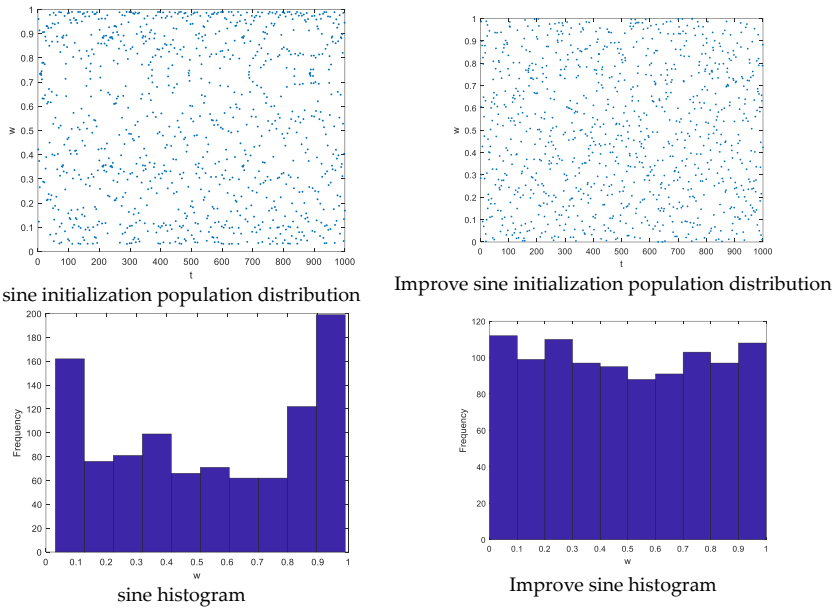

Figure 2. Sine initialization comparison chart.

4.2. Improved Math Optimizer Acceleration Function (SMOA)

When trying to identify the algorithm with the optimum performance, intelligent optimization algorithms frequently run into the issue of balancing global and local searches. The original AOA technique has a critical parameter for the symmetric search phase based on the value of the mathematical optimizer acceleration function (MOA). The stronger the MOA, the greater the probability of greater than the random number r_1, and the algorithm's current local search capability; the stronger the MOA, the greater the probability of less than the random number r_1, and the algorithm's current global search capability. Based on this property, this paper redesigned a new SMOA using the sine function, whose mathematical model is as follows:

$$SMOA = (Max - Min) \times \sin\left(\frac{\pi t}{2T_{\max}}\right)^2 + Min \tag{9}$$

As shown in Figure 3, the MOA in the original algorithm grows linearly and uniformly throughout the search process. However, the intelligent optimization algorithm needs to focus more on the global search in the early iterations, traversing more spaces in a short time so that more feasible solutions can be searched. Later in the iteration, the algorithm needs to focus more on the local search, so that the algorithm can converge better in the field of feasible solutions. The uniformly increasing MOA struggles to match the actual circumstances of algorithm optimization and struggles to strike a good balance between the local and global search. The new SMOA reconstructed in this paper, however, grows slowly in the early iteration and can maintain a lower value, which has a higher probability of being smaller than the random number and conducts a sufficient global search. In the last iteration, it keeps a greater value for a considerable amount of time and is more likely to be higher than the random number r_1, which improves the probability of local search and accelerates the convergence speed.

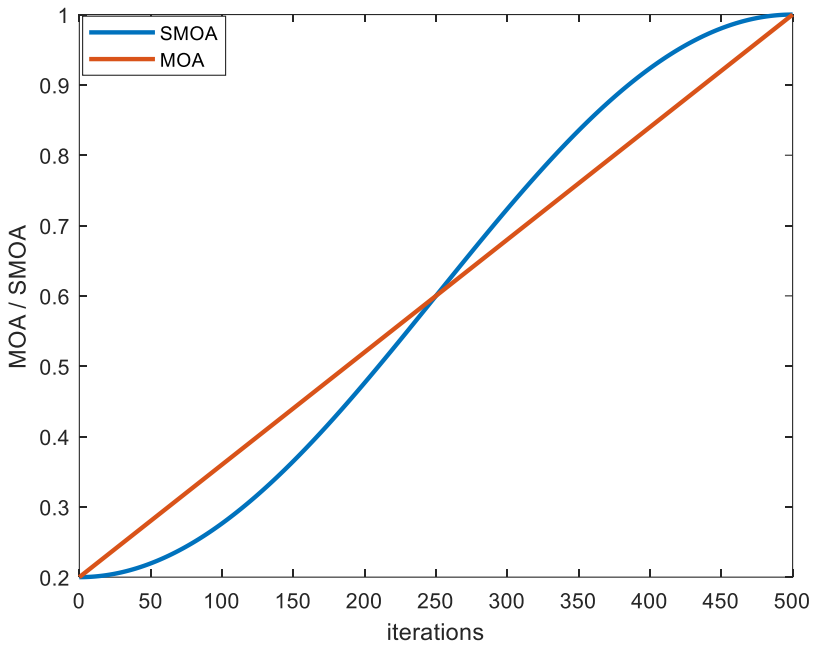

Figure 3. Comparison chart of MOA and SMOA iterations.

4.3. Improved Adaptive Control Parameters

In the standard AOA algorithm, μ is an important sensitive parameter that plays the role of adjusting the search step and coordinating the search process. From Equations (3) and (5), μ is a constant value and takes the value of 0.499. In this study, a nonlinear function is incorporated to improve the algorithm's ability to perform global searches, and μ is

configured as a nonlinear function that gets smaller as iterations increase. This keeps a big value throughout the initial iterative phase, which increases the step size, allows for quick searches for the best solution across the entire global range, and enhances the capability of global searches. In the late iterative period when the population is concentrated in the neighborhood range of the optimal solution, the step size decreases rapidly, which ensures the local search accuracy while converging quickly and improves the exploitation capability. The mathematical model is as follows:

$$\mu = \frac{1}{2} \times (1.1 - (\frac{t}{T_{\max}})^2) \tag{10}$$

4.4. Fusion Sine Cosine Algorithm

This paper introduces the sine and cosine search strategy in the development stage of the AOA algorithm and directly replaces the addition and subtraction operator strategy with the sine and cosine strategy. This is completed to fix the problem of the AOA algorithm's slow convergence and poor search performance in its late iterations. The periodic estimate of the optimal solution using the sine and cosine functions is more stable, more accurate, and can get closer to the global optimal solution faster than the addition and subtraction operator.

As can be seen from Sections 1 and 2, the standard AOA algorithm is developed by comparing the values of random numbers and MOA to perform a search, and the global and local searches are switched randomly. The location update strategy of an individual is chosen by the size of the random value and guided by the best individual of the current population, and this location update mechanism is based on the current population iterative update without drawing on historical information. Therefore, inspired by the PSO algorithm, this paper considers introducing the mechanism of interaction of historical information of individuals and populations in the particle swarm algorithm in the exploration phase.

In the PSO algorithm, the idea of inertia weight w is brought up during the sigmoid search phase [6,29]. Here, w uses an inverted S-shaped function curve based on the sigmoid activation function, which allows the algorithm to draw on the historical information of the previous generation during the iterative process, and w is updated as follows:

$$w = 1 - \frac{1}{1 + e^{5-0.02t}} \tag{11}$$

The fusion algorithm uses the following formula to calculate position updates:
Exploration phase:

$$x_{ij}^{t+1} = \begin{cases} best(x_j) \div (MOP + \varepsilon) \times [(UB_j - LB_j) \times \mu + LB_j] + (best(x_j) - x_{ij}^t), r_2 < 0.5, \\ best(x_j) \times MOP \times [(UB_j - LB_j) \times \mu + LB_j] \quad + (best(x_j) - x_{ij}^t), \quad \text{otherwise}, \end{cases} \tag{12}$$

Development phase:

$$x_{ij}^{t+1} = \begin{cases} wx_{ij}^t + \mu \cdot \sin r_6 \cdot \left| r_7 best(x_j) - x_{ij}^t \right| & r_4 < 0.5, \\ wx_{ij}^t + \mu \cdot \cos r_6 \cdot \left| r_7 best(x_j) - x_{ij}^t \right| & \text{otherwise}, \end{cases} \tag{13}$$

As can be seen from Section 2, the parameter r_5 is an important parameter in the SCA algorithm. It not only contributes to the algorithm's overall stability, but also influences its eventual convergence and precision. If r_5 converges slowly, the algorithm's search efficiency for the neighborhood of the best answer will be reduced, and then the final convergence and accuracy will be affected. If r_5 converges too fast, there will not be enough disruption, and there will not be any means to do enough local exploration to identify the best answer from its immediate surroundings, despite the fact that the neighborhood of the optimal solution has been found in the previous global search, and this drawback is particularly

prominent in multi-peaked problems. Therefore, in order to make the sine and cosine search perform the exploitation operation and to maintain better convergence and accuracy at a later stage, the parameter r_5 is treated nonlinearly and keeps r_5 at a small value. As can be seen in Equation (10), r_5 and the parameter are handled in the same manner in order to simplify the procedure. This was completed to cut down on the amount of work that needed to be conducted.

4.5. Improved Algorithm Flow Chart

The flowchart of the SSCAAOA algorithm is shown in Figure 4:

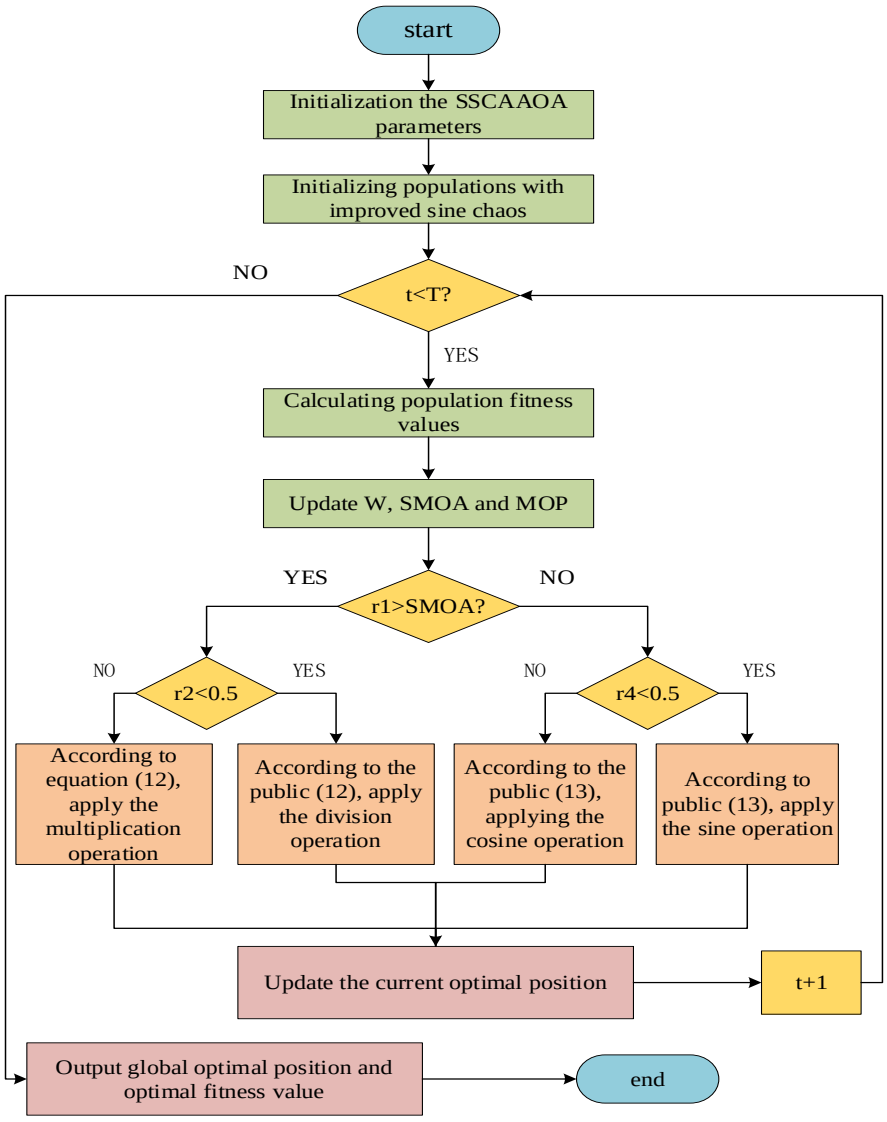

Figure 4. Flowchart of the proposed SSCAAOA.

5. Computational Complexity Analysis

The computational complexity of an algorithm is a key performance indicator. SSCAAOA's computational complexity is mostly determined by the time and effort required for population initialization and population position update. In this research, we investigate the complexity of SSCAAOA in the same way as the AOA algorithm is analyzed in the literature [17].

The algorithm's parameters are already set, considering that N is the number of people in the population, D is the size of the search space, and T is the most iterations that can be

completed. Then, from the literature [17], the standard AOA algorithm time complexity is $O(N \times (TD + 1))$.

Analysis of SSCAAOA time complexity according to the algorithm flow in Section 3: In this paper, the original random initialization is replaced by the improved sine chaos initialization with the same parameter initialization time, hence, the time of initialization is denoted by $O(N \times D)$. The position updating process is conducted using the multiplication and division operator of AOA and the positive cosine strategy of SCA, and the time required for the adaptive control parameters μ, SMOA and inertia weight w is introduced as t_1, t_2, t_3, then the time complexity is $O(T + N \times D \times T + N \times D)$ and updating the optimal solution and position is $O(1)$.

In conclusion, SSCAAOA has the same temporal complexity as the classic AOA algorithm, $O(N \times D \times T)$ which is the same as the standard AOA algorithm.

6. Simulation Experiments and Results Analysis

6.1. Test Environment and Parameter Settings

The 64-bit version of the Windows 10 operating system, an Intel Core i5-6200U processor running at 2.4 GHz, 12 GB of RAM, and the algorithm simulation and programming software MATLAB R2018b make up the experimental simulation environment for this paper's experiments.

As stated in Table 1, ten benchmark test functions that each have their own unique set of features were chosen in order to test the effectiveness of the SSCAAOA algorithm that is discussed in this work. This was completed in order to verify that the algorithm is effective. The convergence and growth capabilities of the method are evaluated using $f_1 \sim f_5$ unimodal functions; Functions $f_6 \sim f_{10}$ are multimodal and are used to measure the algorithm's search capability, whereas functions $f_9 \sim f_{10}$ are fixed, low-dimensional tests used to ensure the method strikes a good balance between the two.

Table 1. Test function.

F	Function	Dim	Range	Best
f_1	Sphere Model	30/100	$[-100, 100]$	0
f_2	Schwefel's problem 2.22	30/100	$[-10, 10]$	0
f_3	Schwefel's problem 1.2	30/100	$[-10, 10]$	0
f_4	Schwefel's problem 2.21	30/100	$[-100, 100]$	0
f_5	Quartic Function	30/100	$[-1.28, 1.28]$	0
f_6	Generalized Rastrigin's Function	30/100	$[-5.12, 5.12]$	0
f_7	Ackley's Function	30/100	$[-32, 32]$	0
f_8	Generalized Griewank Function	30/100	$[-600, 600]$	0
f_9	Kowalik's Function	4	$[-5, 5]$	0.003
f_{10}	Goldstein-Price Function	2	$[-2, 2]$	3

Six other algorithms are chosen to be compared with the proposed SSCAAOA in this research to better verify the performance and advancement of the algorithms in this paper, including WOA [9], CFAWOA [27], AOA, SCA [29], SMSCABOA [30], and GWO [31] The best performance has been found using these techniques, and this has been validated. All algorithms had a population size of N = 30, a maximum number of iterations $T_{\max}$ of 500, and their individual parameters were adjusted as stated in Table 2 to guarantee a level playing field. Thirty separate iterations of each method were conducted, and their performance was analyzed using the mean, standard deviation, and Wilcoxon rank sum test. The Wilcoxon rank sum test determines whether two algorithms are statistically different based on their mean value (which reflects the algorithms' convergence speed and accuracy) and their standard deviation (which reflects the algorithms' stability).

Table 2. Experimental parameter settings of each algorithm.

Algorithm	Parameter Setting
SSCAAOA	$\alpha = 5$; $Max = 1$; $Min = 0.2$;
WOA	$a_1 = [2, 0]$; $a_2 = [-2, 0]$; $b = 1$;
CFAWOA	$a_1 = [2, 0]$; $a_2 = [-2, 0]$; $b = 1$;
AOA	$\mu = 0.499$; $\alpha = 5$; $Max = 1$; $Min = 0.2$;
SCA	$m = 2$;
SMSCABOA	$a = 2$; $c = 0.01$; $p = 0.8$; $limit = 60$;
GWO	$a = [2, 0]$;

6.2. Algorithm Performance Analysis

Table 3 shows the test results. Mean is the average, Std is the standard deviation, and bolded data represent this paper's algorithm findings. In solving the unimodal test functions $f_1 \sim f_5$, The SSCAAOA algorithm manages to reach the theoretically best value in $f_1 \sim f_4$ while maintaining the lowest possible standard deviation. Despite the fact that f_5 does not converge to the theoretically optimal value, the accuracy and stability of the solution it produces are superior to those produced by other comparable algorithms. The enhanced sine chaos initialization that was provided by the method in this work is able to improve the exploitation capability of the algorithm, as can be observed, and the positive-cosine position update formula also makes the convergence accuracy better. Both of these improvements were made possible by the algorithm in this paper. The algorithm SSCAAOA also has a decent performance in the multimodal test functions $f_6 \sim f_{10}$, which were previously mentioned, in which both f_6 and f_8 reach the theoretical optimal value. For f_7, several algorithms did not find the optimal value, but compared with several other compared algorithms, AOA and SSCAAOA have better convergence accuracy and stability, and the accuracy is improved by more than ten orders of magnitude over SMSCABOA and the original SCA. Two fixed low-dimensional test functions. Several algorithms fail to achieve the theoretical optimal value in the f_9 test function, but the mean and standard deviation of the algorithm presented in this study are only smaller than those of SMSCABOA, which is not very different from SCA. In f_{10}, except for AOA and CFAWOA, the mean values of all algorithms are close to the theoretical optimum. However, the SSCAAOA algorithm has the lowest standard deviation, indicating its superior stability. The results of the above investigation demonstrate that SSCAAOA excels at both unimodal and multimodal tasks, boasts superior divergence and stability, and can effectively strike a balance between global and local searches.

Table 3. Test results of benchmark functions of each algorithm.

Function	Metric	WOA	CFAWOA	SCA	SMSCABOA	GWO	AOA	SSCAAOA
f_1	Mean	2.45×10^{-74}	2.54×10^{-168}	7.39×10^{-03}	3.44×10^{-15}	1.30×10^{-27}	2.54×10^{-168}	**0**
	Std	8.64×10^{-74}	0	3.29×10^{-02}	8.66×10^{-15}	2.47×10^{-27}	2.07×10^{-25}	**0**
f_2	Mean	1.35×10^{-50}	2.72×10^{-106}	4.77×10^{-09}	3.25×10^{-10}	7.57×10^{-17}	0	**0**
	Std	4.17×10^{-50}	1.33×10^{-105}	2.13×10^{-08}	7.07×10^{-10}	5.04×10^{-17}	0	**0**
f_3	Mean	4.18×10^{4}	1.42×10^{-104}	1.20×10^{-02}	2.87×10^{-06}	7.55×10^{-06}	7.08×10^{-3}	**0**
	Std	1.48×10^{4}	4.02×10^{-104}	1.35×10^{-02}	1.55×10^{-5}	1.63×10^{-5}	1.20×10^{-2}	**0**
f_4	Mean	3.56×10^{1}	3.13×10^{-81}	1.30×10^{-3}	1.06×10^{-7}	1.20×10^{-7}	7.66×10^{-59}	**0**
	Std	3.56×10^{1}	1.66×10^{-80}	3.09×10^{-3}	1.20×10^{-7}	7.48×10^{-7}	3.98×10^{-58}	**0**
f_5	Mean	4.03×10^{-3}	1.10×10^{-4}	7.74×10^{-2}	1.70×10^{-3}	2.02×10^{-3}	6.77×10^{-5}	$\mathbf{3.52 \times 10^{-5}}$
	Std	3.77×10^{-3}	8.97×10^{-5}	4.77×10^{-2}	6.29×10^{-4}	8.53×10^{-4}	6.13×10^{-05}	$\mathbf{2.61 \times 10^{-5}}$
f_6	Mean	3.79×10^{-15}	0	3.41×10	5.92×10^{-7}	1.92×10^{0}	0	**0**
	Std	2.04×10^{-14}	0	3.17×10	1.18×10^{-6}	2.50×10^{0}	0	**0**
f_7	Mean	4.44×10^{-15}	2.66×10^{-15}	1.41×10^{1}	7.10×10^{-5}	9.73×10^{-14}	8.88×10^{-16}	$\mathbf{8.88 \times 10^{-16}}$
	Std	2.25×10^{-15}	1.78×10^{-15}	9.16×10	7.99×10^{-5}	1.98×10^{-14}	0	**0**

Table 3. *Cont.*

Function	Metric	WOA	CFAWOA	SCA	SMSCABOA	GWO	AOA	SSCAAOA
f_8	Mean	0	0	8.12×10^{-2}	1.60×10^{-7}	4.30×10^{-3}	1.37×10^{-1}	**0**
	Std	0	0	3.01×10^{-1}	2.43×10^{-7}	4.30×10^{-3}	8.12×10^{-2}	**0**
f_9	Mean	7.03×10^{-4}	6.84×10^{-4}	1.09×10^{-3}	3.42×10^{-4}	6.40×10^{-3}	1.85×10^{-2}	$\mathbf{6.41 \times 10^{-4}}$
	Std	4.81×10^{-4}	2.25×10^{-4}	3.77×10^{-4}	6.16×10^{-5}	9.14×10^{-3}	2.82×10^{-2}	$\mathbf{1.63 \times 10^{-4}}$
f_{10}	Mean	3.00×10^{0}	1.40×10^{-4}	3.00×10^{0}	3.00×10^{0}	3.00×10^{0}	7.50×10^{0}	$\mathbf{3.00 \times 10^{0}}$
	Std	1.40×10^{-4}	4.53×10^{-5}	9.21×10^{-5}	4.82×10^{-4}	6.16×10^{-5}	1.01×10^{1}	**0**

6.3. Convergence Analysis

This study plots the iterative convergence process of the test functions, as seen in the following Figure 5 This allows for a more intuitive comparison of the performance of each algorithm, as well as an analysis of the convergence speed and the optimization search process. Due to the limited space, only the iterative convergence graphs of the six benchmark test functions are selected for analysis. The convergence impact produced by CFAWOA is significantly stronger than that produced by a number of other algorithms. When applied to the unimodal test functions f_1 and f_3, the convergence impact of AOA and WOA is superior, with the exception of the algorithm presented in this study. However, several algorithms converge slowly and with low accuracy, while the algorithm in this paper, SSCAAOA, converges quickly and only requires 80–120 iterations to reach the theoretical optimum. It is because the sine cosine position update is incorporated, which allows for faster convergence, and the step size is reduced at the optimum thanks to the adaptive control parameters, yielding improved convergence accuracy. For f_5, the theoretical best value is not found by SSCAAOA, but it has the highest convergence accuracy and the fastest convergence speed. Although the convergence accuracy is the same, as can be shown from the multimodal test functions f_6 and f_7, SSCAAOA converges far more quickly than CFAWOA, WOA, and AOA, and converges almost in a straight line down, which is due to the sine chaos initialization. It broadens the range of populations and raises understanding standards, allowing the algorithm to swiftly reach the ideal value despite a large number of local minima. The convergence graph of the fixed low-dimensional test function f_9 shows that SSCAAOA has several jumps out of the local extremes, indicating that SMOA has a good balance of global and local search.

In conclusion, SSCAAOA balances the global and local search more effectively than other algorithms and the original AOA. It also has faster convergence rates compared to other algorithms and the original AOA. This keeps the original algorithm's superior local exploitation ability while increasing convergence speed and boosting the ability to escape local extrema. It has been proven that the algorithm is superior and efficient.

6.4. Wilcoxon Rank Sum Test

The above analysis demonstrates the algorithm's superiority, but a thorough evaluation of its individual runs and the differences between them and the comparison algorithm would require more statistical tools than just the mean and standard deviation. This work uses the Wilcoxon rank sum test to ensure that these differences are statistically significant. Based on the test results in Table 3, a rank sum test is performed with a significance level of 5%. If the $p < 0.05$ indicates that there is a significant difference between the two algorithms, the performance of the two algorithms differs significantly; otherwise, there is little difference. In this paper, we sample from the test results of the six comparison algorithms and run a total of ten standard test functions to determine whether or not there is a statistically significant difference between the results obtained by these six comparison algorithms and SSCAAOA when the population size is set to $N = 30$, the dimension is set to $D = 30$, and each algorithm is run 30 times on its own. Table 4 displays the outcomes of the tests. If the table shows *N/A* for both methods, it means that the experimental data are identical, and the algorithms perform similarly. A p-value of less than 0.05 suggests that there is a significant difference between the algorithms. Compared with the original AOA,

SSCAAOA has better performance on f_1, f_3, f_4, f_8, f_9 and f_{10}; the majority of the values are less than 0.05. SSCAAOA performs better than WOA in all tested functions except for f_8 and f_9. SSCAAOA performs better than CFAWOA for all tested functions except f_6, f_8 and f_9, which are not significant. For GWO, there are significant differences in all other test functions except f_9; Compared with SSCAAOA, the SMSCABOA and SCA algorithms have significant differences in 10 test functions.

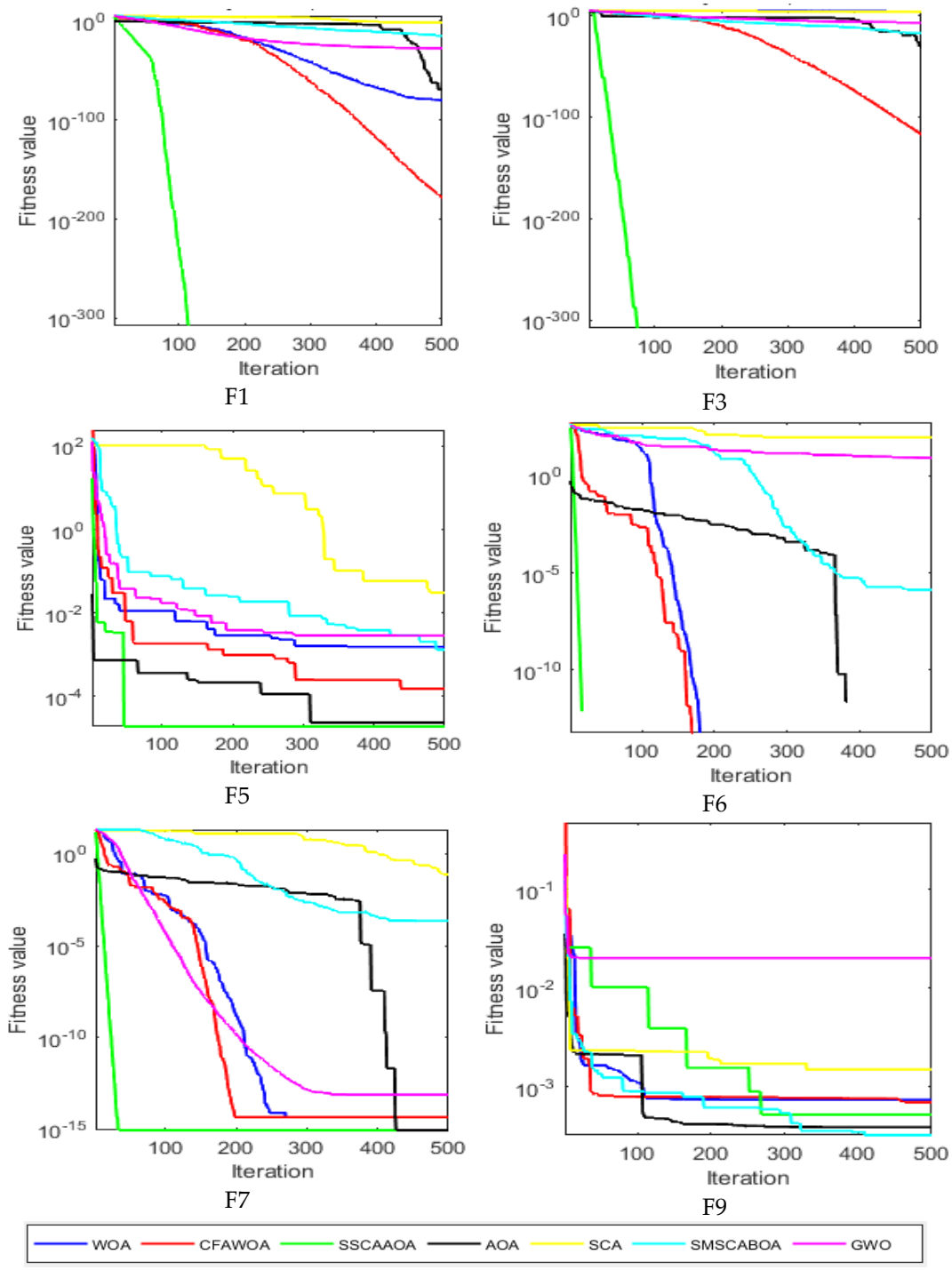

Figure 5. The convergence profile of the comparative methods.

Table 4. Wilcoxon rank sum test results.

Function	WOA	CFAWOA	AOA	SCA	SMSCABOA	GWO
f_1	6.39×10^{-5}	6.39×10^{-5}	6.39×10^{-5}	6.39×10^{-5}	6.39×10^{-5}	6.39×10^{-5}
f_2	1.21×10^{-12}	1.21×10^{-12}	**N/A**	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}
f_3	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}
f_4	1.21×10^{-12}	1.21×10^{-12}	4.09×10^{-4}	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}
f_5	3.02×10^{-11}	5.97×10^{-5}	2.24×10^{-2}	3.02×10^{-11}	3.02×10^{-11}	3.02×10^{-11}
f_6	3.34×10^{-1}	**N/A**	**N/A**	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}
f_7	9.84×10^{-10}	9.65×10^{-6}	**N/A**	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}
f_8	**N/A**	**N/A**	1.21×10^{-12}	1.21×10^{-12}	1.21×10^{-12}	5.58×10^{-3}
f_9	1.45×10^{-1}	6.52×10^{-1}	4.84×10^{-2}	8.88×10^{-6}	3.82×10^{-10}	1.18×10^{-1}
f_{10}	5.58×10^{-3}	4.19×10^{-2}	2.14×10^{-2}	6.51×10^{-4}	2.16×10^{-6}	2.15×10^{-2}

When the results of the preceding performance study and curve convergence analysis are combined, it becomes clear that SSCAAOA's overall performance has been improved from that of the original AOA and SCA algorithms, albeit to varying degrees. Overall, the performance of this algorithm is better than that of other existing improved algorithms such as CFAWOA and SMSCABOA, even if there is little to no difference between them in terms of any given function. This algorithm outperforms competing methods in terms of convergence accuracy, convergence speed, and stability.

7. Solving Path Planning Problems

Path planning is an important step for a mobile robot to be able to accomplish autonomous navigation. According to one or more optimization criteria (e.g., least work cost, shortest trip distance, shortest travel time, etc.), it refers to a mobile robot determining an optimal or nearly optimal path in the motion space from the beginning state to the goal state that can avoid obstacles. This is the simplest description of the path planning problem, but in practical applications, the complexity increases a lot. For example, the mechanical loss of the robot, the uncertainty of the obstacles, the matching of the planning speed of the algorithm and the robot motion speed, the smoothness of the planned route, etc. Due to the complexity, numerous limitations, and several objectives of the path planning problem, researchers frequently approach it as an optimization problem to solve and consider the job requirements as constraints. In order to keep the algorithm from being stuck in a local optimum, the authors of the study [32] propose using an adaptive parallel AOA with a parallel communication strategy to solve the robot path planning problem. Zhang et al. [33] fused the genetic algorithm and the firefly algorithm, and used crossover and mutation in GA to mutate the position when the FA algorithm falls into local optimum. In the final step, the enhanced algorithm is applied to the obstacle course planning issue.

In the path planning problem, the raster method is a more commonly used environment modeling method, which can describe the real environment more completely, and it is also easier to model when the environment changes. However, due to the constraints of the modeling method itself, it is also necessary to combine it with the planning algorithm when performing path planning in order to meet the demand.

The path planned by the mobile robot in this section is the encoding of candidate solutions, and each candidate solution initialized corresponds to a potential path. The shortest path is employed as the goal function in the raster technique with the assumption of safe obstacle avoidance.

The fitness function is shown in Equation (14):

$$L = \sum_{i+1}^{n} \sqrt{(x_{(i+1)} - x_i)^2 + (y_{(i+1)} - y_i)^2} \tag{14}$$

The formula reflects the sum of the shortest distance between two adjacent sites. L is the ultimate planned path length, and n is the number of nodes passed through.

7.1. Node Optimization

When a mobile robot plans its path in a raster map, the final path is a collection of rasters that start from the starting point and connect one adjacent raster to the end point. It is also this planning method that makes more nodes in the path, which often results in larger corners and more inflection points. This situation not only makes the path not optimal, but also increases the mechanical loss of the robot due to more inflection points and frequent steering. In the literature [34], an LPS planning method is proposed, based on the principle of the shortest straight line between two points, which first connects the starting point and the end point when performing path planning, and then performs secondary planning for the part of the path that crosses the obstacle with the help of other traditional path planning algorithms. The purpose of fast planning and improving the quality of the path is achieved. According to the literature [35], the sparrow search algorithm was used as the foundation for a multi-metric, comprehensive assessment approach, and the node optimization strategy was added to the fitness function. Optimizing nodes is one of the necessary steps when using raster maps for robot path planning. Reducing unnecessary turning points can not only make the paths shorter, but also smoother in complex environments. Based on the above-mentioned literature, this section introduces a node-quadratic planning method for secondary optimization of paths after path planning using SSCAAOA.

The method consists of two main stages, namely obstacle detection and connecting paths, as follows:

Step 1: Three nodes are chosen in order starting from the origin.
Step 2: Get the distance in coordinates between the first and third nodes.
Step 3: Check the raster map to see if there are any potential obstructions in the area of interest.
Step 4: If there is not an obstruction, get rid of the second node in the chosen node. If there is, start at the next node and keep on until the optimization reaches the end point.

7.2. Path Planning Experiments

This part runs simulation experiments to compare the original AOA and the enhanced SSCAAOA algorithm in this work in order to confirm the viability of the SSCAAOA method on the path planning problem and the quality of the secondary optimized paths. To ensure the authenticity of the experimental data, 30 independent experiments are conducted for each algorithm, and the experimental data are homogenized. This part creates a 20 × 20 basic obstacle environment and a 20 × 20 difficult obstacle environment, respectively, to demonstrate the algorithms' flexibility and efficacy. Yellow is the starting point and green is the ending point, and the specific experimental results are as follows:

(1) Experimental environment 1

As can be seen from Figure 6, the starting point is (1,1) and the end point is set to (20,20) in a less obstructed and more dispersed environment. Although the path directions planned by the two algorithms are roughly the same, the SSCAAOA algorithm's projected path is substantially smoother and has only five clear inflection points, while the original AOA algorithm has 14 inflection points, and the secondary optimization introduced in this section of the surface plays a significant role in path smoothing. The enhanced algorithm in this chapter outperforms the original AOA algorithm in terms of the shortest path, the worst path, and average value, as shown in Table 5, and the ideal path is also significantly shorter thanks to the decreased number of inflection points.

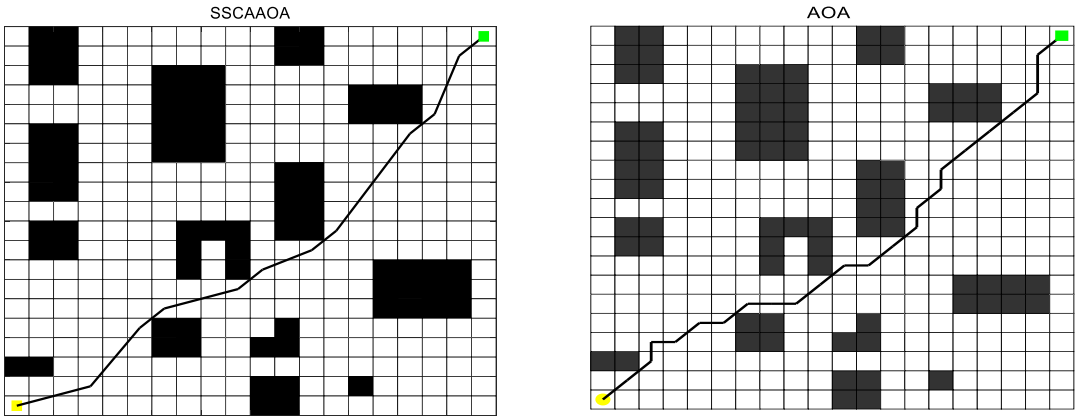

Figure 6. Experiment 1 planning results. Yellow is the starting point and green is the ending point.

Table 5. Environment 1 simulation experimental results.

Algorithm	Shortest Path	Longest Path	Average Path	Average Inflection Point
AOA	29.21	34.97	31.28	13.31
SSCAAOA	27.68	28.80	28.19	4.44

(2) Experimental environment 2

As shown in Figure 7, a more complex environment model is set up in the experimental environment 2, where concave and continuously distributed obstacles are set up in the environment, and the planning difficulty is increased by placing obstacles in front of both the starting point and the destination; this calls for a more robust algorithm to determine the best route. The improved SSCAAOA algorithm in this chapter can find feasible paths every time with 90% accuracy of optimal or suboptimal paths with relatively few inflection points and smoother paths under the guarantee of obstacle avoidance. The original AOA algorithm, on the other hand, did not find feasible paths even to the end of the iterations at some times, and finally increased the number of iterations to 300 to complete the task. The final planned paths were significantly more tortuous, with larger turning angles and more turns. Table 6 shows that as the environment becomes more complex, the benefits of the optimized algorithm become more apparent. Search performance is enhanced, accuracy in locating the optimal path is guaranteed, and smoothness is enhanced following the second iteration of the optimized algorithm.

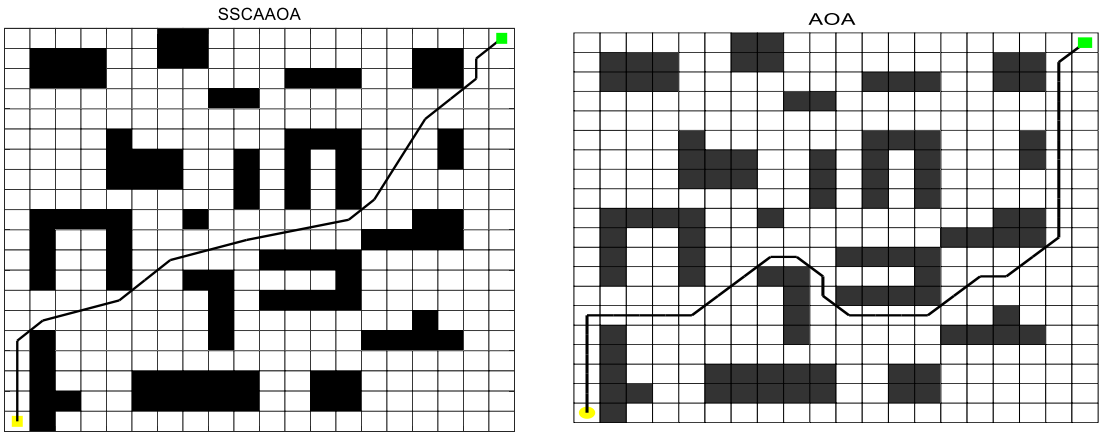

Figure 7. Experiment 2 planning results. Yellow is the starting point and green is the ending point.

Table 6. Environment 2 simulation experimental results.

Algorithm	Shortest Path	Longest Path	Average Path	Average Inflection Point
AOA	33.55	54.62	39.75	18.06
SSCAAOA	29.81	34.25	31.06	9.56

8. Conclusions and Future Work

This study proposes a new optimization method that combines the benefits of AOA and SCA in order to address the issues with the classic AOA algorithm's slow convergence speed, low solution accuracy, and propensity for falling into the local extreme value problem. This study puts forward a variety of improvement strategies for different problem defects, and the work completed in this paper mainly includes the following points:

- The population is first seeded using a modified sine chaotic mapping. Compared with the standard AOA's random initialization strategy, the chaotic nature can make the population more uniformly distributed in the search space, which makes the algorithm find the global optimal solution most easily. A new mathematical optimizer acceleration function, SMOA, was redesigned to enhance the algorithm's ability to balance global and local searches in order to maximize the algorithm's search performance. We improve the sensitive parameter in the standard AOA by replacing the fixed value in the original algorithm with a convex function to obtain an adaptive adjustment parameter, so that the step size can be adjusted adaptively with the increase in iterations in the optimization search process to avoid the problem of missing the optimal solution due to too large step size in the later stage, which leads to low accuracy.
- By including the SCA throughout development, we are able to solve the issue of late-stage AOA algorithm's poor convergence speed and inadequate convergence accuracy.
- On the basis of the above, the particle swarm algorithm idea is borrowed, and the influence of historical information on the population is added in the exploration phase to improve the situation that AOA is too dependent on the current optimal solution. The inverse S-shaped inertia weights based on the improved sigmoid activation function are introduced in the sigmoid development phase, allowing the development phase particles to draw on the information of the previous generation, and the use of parameters in the standard SCA integrally strengthens the local development capability of the sigmoid strategy.
- A total of 10 standard test functions are selected to test the algorithm for experimental simulation and compare the performance with some existing optimization algorithms. As displayed through the data analysis, the algorithm presented in this study improves upon prior methods in terms of convergence speed, accuracy, and stability, and this holds true for both unimodal and multimodal functions. The efficiency of the optimization search has been enhanced to varying degrees compared to the initial AOA and SCA.
- Last but not least, the enhanced technique in this study is used to resolve path planning issues as well as engineering issues. The experimental results show that, in comparison to the original algorithm, the improved algorithm not only successfully solves path planning problems in various environments but also produces relatively smooth planned paths, demonstrating the improved algorithm's effectiveness in solving engineering problems and further validating its performance.

Based on the foregoing analysis, it is clear that the improved algorithm presented in this paper has some advantages. However, the results of the multimodal function test show that the improved algorithm presented in this paper does not significantly differ from the comparison algorithm in solving the multimodal function; it also has some limitations and may fall into local extremes when there are multiple extremes. In the future, more engineering problems will need to be applied to SSCAAOA before its efficacy can be fully evaluated. Therefore, the algorithm in this paper can be further improved and enhanced.

Author Contributions: All of the authors contributed extensively to the work. H.X. proposed the key ideas; H.X. analyzed the key contents using a simulation and wrote the manuscript; L.L. obtained the financial support for the project leading to this publication; B.W. and C.K. modified the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was financially supported by the Natural Science Foundation of Fujian Province, Grant/Award (Grant no. 2022H6005, 2022J01952), in part by the National Natural Science Foundation of China (Grant no. 61973085).

Data Availability Statement: Data available on request due to restrictions eg privacy or ethical. The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to thank the anonymous reviewers for their valuable comments.

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

References

1. Hashim, F.A.; Hussain, K.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W. Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems. *Appl. Intell.* **2021**, *51*, 1531–1551. [CrossRef]
2. Faramarzi, A.; Heidarinejad, M.; Stephens, B.; Stephens, B.; Mirjalili, S. Equilibrium optimizer: A novel optimization algorithm. *Knowl. Based Syst.* **2020**, *191*, 105190. [CrossRef]
3. Abualigah, L.; Yousri, D.; Abd Elaziz, M.; Ewees, A.A.; Al-Qaness, M.A.; Gandomi, A.H. Aquila optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [CrossRef]
4. Zeng, N.; Wang, Z.; Liu, W.; Zhang, H.; Hone, K.; Liu, X. A dynamic neighborhood-based switching particle swarm optimization algorithm. *IEEE Trans. Cybern.* **2020**, *52*, 9290–9301. [CrossRef]
5. Pan, J.S.; Zhang, L.G.; Wang, R.B.; Snášel, V.; Chu, S.C. Gannet optimization algorithm: A new metaheuristic algorithm for solving engineering optimization problems. *Math. Comput. Simul.* **2022**, *202*, 343–373. [CrossRef]
6. Eberhart, R.; Kennedy, J. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
7. Deng, W.; Zhang, X.; Zhou, Y.; Liu, Y.; Zhou, X.; Chen, H.; Zhao, H. An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. *Inf. Sci.* **2022**, *585*, 441–453. [CrossRef]
8. Colorni, A.; Dorigo, M.; Maniezzo, V. *An Investigation of Some Properties of an "Ant Algorithm"*; Ppsn. 1992; Elsevier: Amsterdam, The Netherlands, 1992.
9. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
10. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Systems* **2020**, *111*, 300–323. [CrossRef]
11. Zervoudakis, K.; Tsafarakis, S. A mayfly optimization algorithm. *Comput. Ind. Eng.* **2020**, *145*, 106559. [CrossRef]
12. Yan, L.-J.; Li, Z.-B.; Wei, J.-H.; Du, X. A New Hybrid Optimization Algorithm and Its Application in Job Shop Scheduling. *ACTA Autom. Sin.* **2008**, *34*, 604–608. [CrossRef]
13. Miao, C.; Chen, G.; Yan, C.; Wu, Y. Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm. *Comput. Ind. Eng.* **2021**, *156*, 107230. [CrossRef]
14. Micev, M.; Ćalasan, M.; Ali, Z.M.; Hasanien, H.M.; Aleem, S.H.A. Optimal design of automatic voltage regulation controller using hybrid simulated annealing–Manta ray foraging optimization algorithm. *Ain Shams Eng. J.* **2021**, *12*, 641–657. [CrossRef]
15. de Lacerda, M.G.P.; de Araujo Pessoa, L.F.; de Lima Neto, F.B.; Ludermir, T.B.; Kuchen, H. A systematic literature review on general parameter control for evolutionary and swarm-based algorithms. *Swarm Evol. Comput.* **2021**, *60*, 100777. [CrossRef]
16. Gharehchopogh, F.S.; Gholizadeh, H. A comprehensive survey: Whale Optimization Algorithm and its applications. *Swarm Evol. Comput.* **2019**, *48*, 1–24. [CrossRef]
17. Alwajih, R.; Abdulkadir, S.J.; Al Hussian, H.; Aziz, N.; Al-Tashi, Q.; Mirjalili, S.; Alqushaibi, A. Hybrid binary whale with harris hawks for feature selection. *Neural Comput. Appl.* **2022**, *34*, 19377–19395. [CrossRef]
18. Khattab, H.; Mahafzah, B.A.; Sharieh, A. A hybrid algorithm based on modified chemical reaction optimization and best-first search algorithm for solving minimum vertex cover problem. *Neural Comput. Appl.* **2022**, *34*, 15513–15541. [CrossRef]
19. Shokouhifar, M. FH-ACO: Fuzzy heuristic-based ant colony optimization for joint virtual network function placement and routing. *Appl. Soft Comput.* **2021**, *107*, 107401. [CrossRef]
20. Zhang, Y.J.; Yan, Y.X.; Zhao, J.; Gao, J.M. AOAAO: The hybrid algorithm of arithmetic optimization algorithm with aquila optimizer. *IEEE Access* **2022**, *10*, 10907–10933. [CrossRef]
21. Shokouhifar, M.; Sohrabi, M.; Rabbani, M.; Molana, S.M.H.; Werner, F. Sustainable Phosphorus Fertilizer Supply Chain Management to Improve Crop Yield and P Use Efficiency Using an Ensemble Heuristic–Metaheuristic Optimization Algorithm. *Agronomy* **2023**, *13*, 565. [CrossRef]
22. Abualigah, L.; Diabat, A.; Sumari, P.; Gandomi, A.H. A novel evolutionary arithmetic optimization algorithm for multilevel thresholding segmentation of covid-19 ct images. *Processes* **2021**, *9*, 1155. [CrossRef]

23. Abualigah, L.; Pandit, A.K. Hybrid arithmetic optimization algorithm with hunger games search for global optimization. *Multimed. Tools Appl.* **2022**, *81*, 28755–28778.
24. Hao, W.K.; Wang, J.S.; Li, X.D.; Wang, M.; Zhang, M. Arithmetic optimization algorithm based on elementary function disturbance for solving economic load dispatch problem in power system. *Appl. Intell.* **2022**, *52*, 11846–11872. [CrossRef]
25. Yang, W.Z.; He, Q. Multi-head reverse series arithmetic optimization algorithm with activation mechanism. *Appl. Res. Comput.* **2022**, *39*, 151–156. [CrossRef]
26. Abualigah, L.; Diabat, A.; Mirjalili, S.; Abd Elaziz, M.; Gandomi, A.H. The arithmetic optimization algorithm. *Comput. Methods Appl. Mech. Eng.* **2021**, *376*, 113609. [CrossRef]
27. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* **2016**, *96*, 120–133. [CrossRef]
28. Jiang, D.H.; Liu, L.D.; Chen, Y.P.; Wang, X.Y.; Sun, K. Visual image encryption algorithm based on fractional-order Chen hy-perchaotic system and compression-awareness [J/OL]. *J. Chin. Mini-Micro Comput. Syst.* **2022**, *43*, 2387–2393.
29. Tu, C.; Chen, G.; Liu, C. Research on chaotic feedback adaptive whale optimization algorithm. *Stat. Curation* **2019**, *35*, 1720. [CrossRef]
30. Gao, W.X.; Liu, S.; Xiao, Z.Y.; Yu, J.F. Butterfly optimization algorithm for global optimization. *Comput. Appl. Res.* **2020**, *37*, 2966–2970. [CrossRef]
31. Rezaei, H.; Bozorg-Haddad, O.; Chu, X. Grey wolf optimization (GWO) algorithm. In *Advanced Optimization by Nature-Inspired Algorithms*; Springer: Singapore, 2018; pp. 81–91.
32. Wang, R.B.; Wang, W.F.; Xu, L.; Pan, J.-S.; Chu, S.-C. An adaptive parallel arithmetic optimization algorithm for robot path planning. *J. Adv. Transp.* **2021**, *2021*, 1–22. [CrossRef]
33. Zhang, T.W.; Xu, G.H.; Zhan, X.S.; Han, T. A new hybrid algorithm for path planning of mobile robot. *J. Supercomput.* **2022**, *78*, 4158–4181. [CrossRef]
34. Fareh, R.; Baziyad, M.; Rabie, T.; Bettayeb, M. Enhancing Path Quality of Real-Time Path Planning Algorithms for Mobile Robots: A Sequential Linear Paths Approach. *IEEE Access* **2020**, *8*, 167090–167104. [CrossRef]
35. Zhang, Z.; He, R.; Yang, K. A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm. *Adv. Manuf.* **2022**, *10*, 114–130. [CrossRef]