

Improved Arithmetic Optimization Algorithm with Multi-Strategy Fusion Mechanism and Its Application in Engineering Design

Yu Liu¹, Minge Chen^{1*}, Ran Yin¹, Jianwei Li¹, Yafei Zhao¹, Xiaohua Zhang^{2#}

¹School of Mathematics and Data Science, Shaanxi University of Science & Technology, Xi'an, China

²School of Materials Science and Engineering, Shaanxi University of Science & Technology, Xi'an, China

Email: *chenminge@sust.edu.cn, #zhangcity@126.com

How to cite this paper: Liu, Y., Chen, M.G., Yin, R., Li, J.W., Zhao, Y.F. and Zhang, X.H. (2024) Improved Arithmetic Optimization Algorithm with Multi-Strategy Fusion Mechanism and Its Application in Engineering Design. *Journal of Applied Mathematics and Physics*, 12, 2212-2253.
<https://doi.org/10.4236/jamp.2024.126134>

Received: May 16, 2024

Accepted: June 24, 2024

Published: June 27, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This article addresses the issues of falling into local optima and insufficient exploration capability in the Arithmetic Optimization Algorithm (AOA), proposing an improved Arithmetic Optimization Algorithm with a multi-strategy mechanism (BSFAOA). This algorithm introduces three strategies within the standard AOA framework: an adaptive balance factor *SMOA* based on sine functions, a search strategy combining Spiral Search and Brownian Motion, and a hybrid perturbation strategy based on Whale Fall Mechanism and Polynomial Differential Learning. The BSFAOA algorithm is analyzed in depth on the well-known 23 benchmark functions, CEC2019 test functions, and four real optimization problems. The experimental results demonstrate that the BSFAOA algorithm can better balance the exploration and exploitation capabilities, significantly enhancing the stability, convergence mode, and search efficiency of the AOA algorithm.

Keywords

Arithmetic Optimization Algorithm, Adaptive Balance Factor, Spiral Search, Brownian Motion, Whale Fall Mechanism

1. Introduction

Optimization problem [1] refers to the issue of adjusting a system or process to better meet the requirements and achieve the optimal results under specific conditions, which is widely distributed in various fields such as medicine, engineering science, finance [2] [3] [4], and so on. With the popularity of high-speed and large-capacity computers, scientific researchers have been able to abstract prac-

tical problems into function optimization and combinatorial optimization issues for solving by establishing mathematical models, but due to their nonlinear and nonconvex characteristics, the results of solving them by using traditional optimization algorithms are not ideal [5]. Meta-heuristic algorithms are a class of algorithms inspired by complex phenomena in nature and thus form their respective different optimization theories, which can take into account the global exploration and local development capabilities, and provide fast, efficient, and accurate solutions for all kinds of complex optimization problems.

Classical meta-heuristic algorithms include Particle Swarm Optimization [6] (PSO), Gray Wolf Optimization [7] (GWO), Differential Evolution [8] (DE), etc. Algorithms proposed in recent years include Exchange Market Algorithm [9] (EMA), covariance matrix adaptation evolution strategy [10] (CMA-ES), Beluga Whale Optimization [11] (BWO), Goose Algorithm [12] (GOOSE), Ray Optimization [13] (RO), Sine Cosine Algorithm [14] (SCA), Golden Ratio Optimization Method [15] (GROM). The Arithmetic Optimization Algorithm [16] (AOA) is a novel meta-heuristic algorithm proposed by Abualigah *et al.* in 2021, which updates the solution by establishing mathematical models based on the characteristics of the main arithmetic operators (multiplication, division, subtraction, and addition) in mathematics.

The AOA updates candidate solutions utilizing four formulas: multiplication and division operators for exploring, and addition and subtraction operators for exploiting. This algorithm is capable of searching for the optimal solution precisely and has the advantages of good portability, few parameters, and fast execution efficiency. Since the AOA was proposed, it has been extensively applied in various fields with good results. For example, Reddy *et al.* [17] adopted the arithmetic optimization algorithm with improved math optimizer accelerated Function (IMOA-AOA) to optimize and adjust the parameters of deep learning methods, and proposed a deep structured architectures classification and segmentation scheme for brain tumor, which has high segmentation and classification accuracy. Barua *et al.* [18] proposed the Lévy arithmetic Algorithm (LAA) by combining the Lévy random step size with the arithmetic optimization algorithm and applied it to the economic load distribution of renewable energy integrated microgrids, improving search capability and minimizing computational requirements. In addition, AOA has also been applied to image processing, text clustering, feature selection, engineering design [19] [20] [21] [22], medical diagnosis, energy management, etc., and has achieved good results. Experimental results in the literature [16] indicate that the AOA algorithm exhibits strong competitiveness over the GWO [7] and PSO [6] algorithms in terms of convergence speed and quality of solutions.

However, according to the “No Free Lunch Theorem [23]”, no single meta-heuristic algorithm can address all optimization problems, and scholars have proposed enhancements to the original AOA algorithm and applied it to diverse fields. In 2021, Wang *et al.* [24] introduced an Adaptive Parallel AOA with a novel parallel communication strategy to enhance the performance of the algo-

rithm, aiming at the problem of robot planning to find the collection-free optimal motion path in obstacle environments. In 2022, Abualigah *et al.* [20] combined Opposite-Based Learning (OBL) and Levy Flight Distribution (LFD) with AOA to overcome the limitations of traditional AOA. In 2022, Zheng *et al.* [25] introduced a forced switching mechanism and utilized Random Mathematical Optimization Probability (RMOP) to enhance the Arithmetic Optimization Algorithm, helping the AOA escape local optima. In 2023, Yıldız *et al.* [26] proposed a new hybrid optimization algorithm, AOA-NM, which incorporates the Nelder-Mead local search methodology into the basic AOA framework to promote the AOA search exploration-exploitation behavior improvement to overcome the shortcomings of local optimal traps. In 2023, Gölcük *et al.* [27] proposed an improved arithmetic optimization algorithm (AOA) to train artificial neural networks in dynamic environments, and the performance of the algorithm was tested on dynamic classification problems, which exemplified the superiority of the proposed algorithm. In 2024, Barua *et al.* [18] used the Lévy stochastic step size to enhance the Arithmetic Optimization Algorithm, which improves search capability and minimizes computational requirements to improve results.

Despite the aforementioned literature has made some advancements in improving the optimization capabilities and applicability of AOA, challenges remain. The value of the math optimizer accelerated Function *MOA* plays a critical role in the iterative solution process of the AOA algorithm, the structure of *MOA* is simple and linear, which can easily cause the algorithm to fall into local optimum. During the exploration and development stages of the algorithm, the concise iterative formulas and restricted search step size, make the algorithm unable to fully search the solution area and affect the solution accuracy. In addition, the algorithm is not perturbed by the optimal individual after each iteration update, which may lead to the premature convergence phenomenon of the algorithm if the optimal individual falls into the local extreme value space.

Therefore, to address the defects of the AOA algorithm, this article proposes an improved Arithmetic Optimization Algorithm (BSFAOA) with a multi-strategy mechanism. The algorithm introduces an adaptive balance factor *SMOA* based on sine functions, to balance the exploration and exploitation capabilities of the AOA algorithm; employs a spiral search factor to expand the step size of the global search, and enhances the accuracy of the local search by utilizing the Brownian motion, to improve the optimization search speed and accuracy of the algorithm; and the hybrid perturbation strategy of whale fall mechanism and polynomial difference learning is used to disturb the population position, which helps the algorithm jump out of the local optimal position. The effectiveness of the BSFAOA is verified by comparing it with other improved algorithms, existing algorithms and single strategy improved algorithms on 23 well-known benchmark functions and CEC2019 test functions, respectively. The experimental results demonstrate that BSFAOA outperforms in terms of optimization search accuracy and stability. Furthermore, the BSFAOA is further demonstrated to

have superior performance in addressing practical application problems by solving four optimization problems. In summary, the main contributions of this study are summarized as follows:

- 1) The BSFAOA improves AOA in three aspects to help the algorithm avoid falling into local extremes and enhance the convergence performance.
- 2) The performance of BSFAOA is analyzed using CEC 2005 and CEC2019 functions.
- 3) The BSFAOA-SVR regression prediction model was established to verify the applicability of BSFAOA, and three engineering constraint optimization problems were evaluated by BSFAOA.
- 4) Evaluation indexes such as minimum value, mean value, standard deviation, Wilcoxon rank, and algorithm comprehensive ranking were used to evaluate the performance of BSFAOA, and compared with other improved algorithms and existing meta-heuristic optimization algorithms.

2. Arithmetic Optimization Algorithm

The exploration phase of the AOA corresponds to division (“÷”) and multiplication (“×”), while the development phase corresponds to subtraction (“-”) and addition (“+”). The mechanism of exploration and development of AOA is shown in **Figure 1**. The math optimizer accelerated Function MOA is defined and calculated using Equation (1).

$$MOA(k) = MOA_M + k \times \frac{MOA_M - MOA_m}{K}. \quad (1)$$

The Arithmetic Optimization Algorithm (AOA)

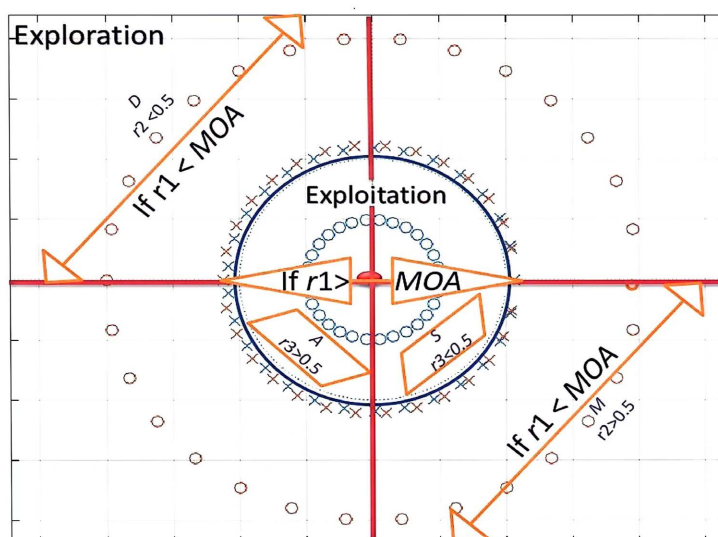


Figure 1. The search phases of the AOA.

where k and K denote the current iteration and the maximum number of iterations, respectively. MOA_m and MOA_M are the minimum and maximum

values respectively, $MOA_M = 1$, $MOA_m = 0.2$. r_1 is a random number. If $r_1 > MOA$, the exploration phase is entered, otherwise the mining phase is introduced.

The updated expression for the solution in the AOA exploration phase is as follows:

$$x(k+1) = \begin{cases} x_{best}(k) \div (MOP + \varepsilon) \times ((ub - lb) \times \mu + lb), & r_2 < 0.5, \\ x_{best}(k) \times MOP \times ((ub - lb) \times \mu + lb), & r_2 \geq 0.5. \end{cases} \quad (2)$$

$$MOP(k) = 1 - \left(\frac{k}{K}\right)^{\frac{1}{\alpha}}. \quad (3)$$

where $x(k+1)$ represents the solution at the $k+1$ iteration, $x_{best}(k)$ is the optimal solution obtained at the k iteration, and ε is the minimum value. μ is set to 0.5, and MOP , representing the probability of mathematical optimization, α is fixed at 5.

The update expression during the development phase is as follows:

$$x(k+1) = \begin{cases} x_{best}(k) - MOP \times ((ub - lb) \times \mu + lb), & r_3 < 0.5, \\ x_{best}(k) + MOP \times ((ub - lb) \times \mu + lb), & r_3 \geq 0.5. \end{cases} \quad (4)$$

3. Improvements to the Arithmetic Optimization Algorithm BSFAOA through a Multi-Strategy Mechanism

The BSFAOA is an improved arithmetic optimization algorithm with a multi-strategy mechanism that introduces an adaptive balance factor $SMOA$ based on sinusoidal function, a search strategy that combines spiral search and Brownian motion, and an adaptive hybrid perturbation strategy with whale fall mechanism and polynomial differential learning to improve the original AOA. The proposed algorithm is described in detail in the following subsections.

3.1. Adaptive Balancing Factor $SMOA$ Strategy Based on Sine Function

In the fundamental AOA algorithm, the value of MOA determines the balance between global exploration and local development. When the MOA value is small, the algorithm tends to favor global exploration; when it is large, the algorithm tends to favor local exploitation, this balance is crucial for the optimization performance of the algorithm. As the number of iterations increases, the MOA value increases linearly from 0.2 to 1, enabling the algorithm to gradually transition from initial global exploration to local exploitation. Insufficient exploration of the solution space in the early stages of iteration may cause the algorithm to fall into a local optimum from which it is difficult to escape.

To better the balance between global exploration and local development in algorithms, and reduce the probability of falling into local optimum, this article adopts the sine function to improve the MOA of the original AOA algorithm. Since the MOA in the original algorithm is linear, the linear growth of MOA

cannot accurately reflect the actual iterative process, while the sine function can transform the change of *MOA* into a nonlinear so that it is more relevant to the actual iterative process of the algorithm, so this article introduces an adaptive balance factor *SMOA* based on the sinusoidal function, the mathematical expression is shown in Equation (5) as follows.

$$SMOA(k) = \frac{1}{2} - \frac{1}{2} \times \sin\left(\frac{\pi}{2} \times \frac{k}{K} - \frac{5\pi}{2}\right). \quad (5)$$

where k and K denote the current iteration and the maximum iteration.

The comparison curve between the original *MOA* and the *SMOA* is shown in **Figure 2**, the *SMOA* shows a nonlinear decreasing trend at the beginning of the iteration, the global exploration ability of the algorithm is enhanced, and a large number of individuals can be involved in the exploration process, and with the number of iterations increases, the *SMOA* starts to grow nonlinear, at this time, the higher the probability of local development, and the more the local exploitation ability of the algorithm is stronger, it can prevent from falling into the local extremes, speed up the optimization search speed while improving the convergence accuracy of the algorithm.

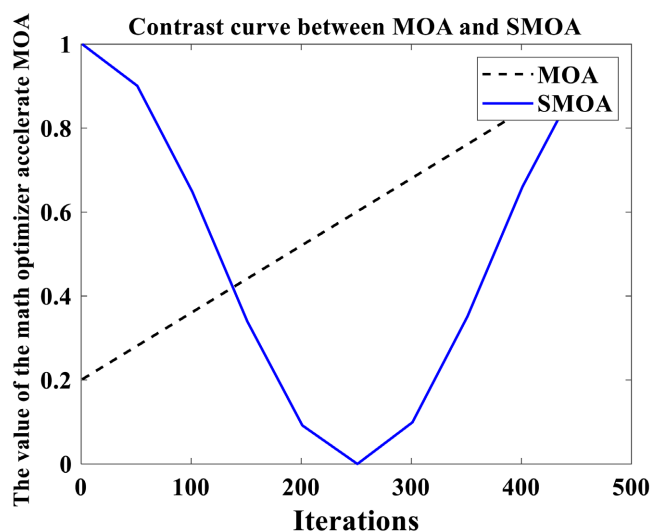


Figure 2. Comparison between original *MOA* and improved *SMOA*.

3.2. Search Strategy Combining Spiral Search and Brownian Motion

1) Spiral Search

In the iterative process of the whale algorithm [28], individual humpback whales use a spiral search strategy to update their positions with their prey, which not only enhances the convergence speed and search accuracy of the algorithm but also increases the diversity among individuals. This strategy can cover the search space faster by expanding the search step, making it easier for the algorithm to find the optimal solution, thereby effectively enhancing the global search performance of the algorithm.

The basic AOA algorithm has a more limited search step size and cannot adequately find the global optimal solution. To overcome this limitation, this article adds a spiral search factor to the global search phase of the AOA algorithm, which expands the capability of the algorithm to explore the unknown, makes the algorithm carry out a broader and more effective search within the solution space region, and continuously adjusts the position of the individuals along the spiral path to improve the efficiency and stability of the search. The global search update formula with spiral exploration factor is as follows:

$$x(k+1) = \begin{cases} D \times x_{best}(k) \div (MOP + \varepsilon) \times ((ub - lb) \times \mu + lb), & r_2 < 0.5, \\ D \times x_{best}(k) \times MOP \times ((ub - lb) \times \mu + lb), & r_2 \geq 0.5. \end{cases} \quad (6)$$

$$z = e^{\pi \times \cos\left(1 - \frac{k}{K}\right)} \quad (7)$$

$$D = e^{z \cdot l} \cos(2\pi l) \quad (8)$$

where z is the constant defining the logarithmic spiral, l is a random number in $[-1, 1]$, $r \in [0, 1]$ is a random number, x_{best} is the current optimal individual position.

2) Brownian motion

The local development capability of arithmetic optimization algorithms is insufficient, leading to the algorithm easily getting trapped in local optima, thus reducing the convergence speed of the algorithm. Brownian motion, as a stochastic process, is widely used to simulate the foraging behavior of animals, the endless and irregular motion of particles, the fluctuating behavior of stock prices, and so on. The step size of a standard Brownian motion is obtained by a probability density function based on a normal distribution with zero mean and one variance [29]. The action trajectory of Brownian motion is concentrated in a certain region, and it can utilize the dynamic and uniform step size to explore some potential regions in the population space and to realize wide trajectory motion in the control region. Its expression is shown in Equation (9):

$$f_{BM}(x; 0, 1) = \frac{1}{\sqrt{2\pi}} \times e^{-\frac{x^2}{2}} \quad (9)$$

Figure 3 shows the trajectory diagram of Brownian motion. In this article, Brownian motion is applied to improve the local development formula of the arithmetic optimization algorithm, enhancing the accuracy of local search. Using the characteristics of Brownian motion, the individuals of the population in the arithmetic optimization algorithm, successfully converge to a potential region, which provides a greater possibility of obtaining the global optimal solution, thereby enhancing the convergence accuracy and optimization-seeking efficiency of the algorithm. The local development update formula with Brownian motion is shown in Equation (10):

$$x(k+1) = \begin{cases} x_{best}(k) - BM(k) \times MOP \times ((ub - lb) \times \mu + lb), & r_3 < 0.5, \\ x_{best}(k) + BM(k) \times MOP \times ((ub - lb) \times \mu + lb), & r_3 \geq 0.5. \end{cases} \quad (10)$$

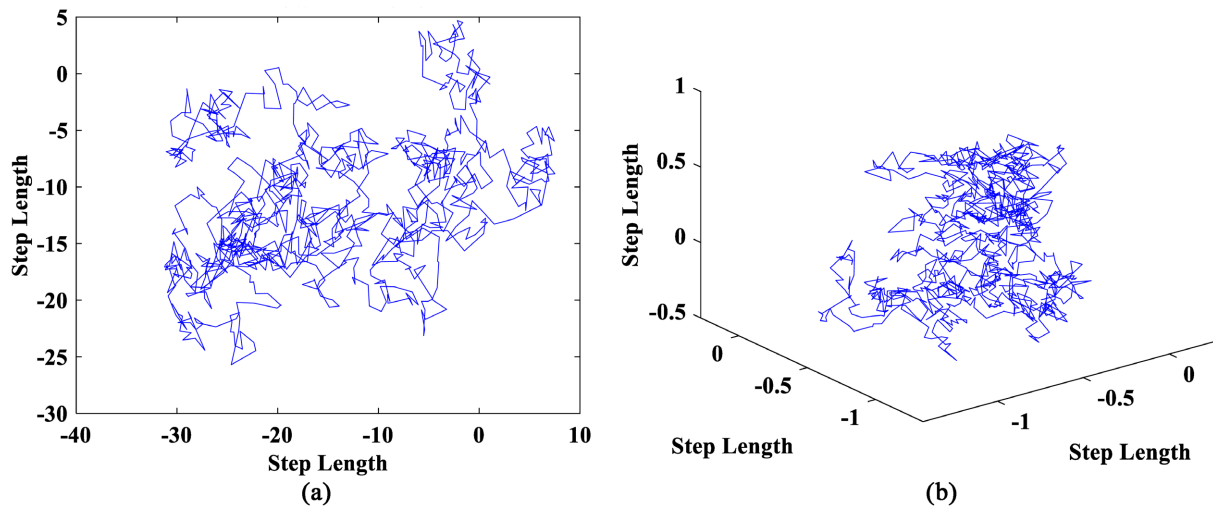


Figure 3. Brownian motion. (a) 2-Dim Brownian motion; (b) 3-Dim Brownian motion.

where $BM(k)$ denotes a vector containing random numbers that are based on a Gaussian distribution representing Brownian motion.

3.3. Adaptive Hybrid Perturbation Strategy Based on Whale Fall Mechanism and Polynomial Differential Learning

To accelerate the speed of algorithm optimization and enhance the ability of the algorithm to escape local optima, this article proposes an adaptive hybrid perturbation strategy that perturbs the population of individuals after each iteration.

1) Whale Fall Mechanism

Inspired by the whale fall phase of the Beluga Whale Optimization [11], a few of the threatened beluga whales do not survive and fall into the seabed during beluga migration and foraging. To simulate the behavior of beluga whales falling in each iteration and to ensure that the population size remains constant, the current position of the beluga whales and the step size of the whale fall descent are used to establish the position update formula, which is represented in the mathematical model [11] as:

$$C_2 = \left(0.2 - \frac{k}{T}\right) \times N. \tag{11}$$

$$X_{step} = (ub - lb) e^{-C_2 \frac{k}{K}}. \tag{12}$$

$$X_i^{k+1} = r_4 X_i^k - r_5 X_r^k + r_6 X_{step}. \tag{13}$$

where r_4 , r_5 , and r_6 are random numbers between (0, 1), X_{step} is the step size of whale fall, C_2 is a step factor related to population size, and X_r^k is a randomly selected individual in the population.

In the basic AOA algorithm, the update of the optimal individual depends on the update of the population at each iteration. That is, after each iteration, the individual with the current best fitness replaces the optimal individual, and the algorithm does not actively perturb the optimal individual. If the optimal indi-

vidual falls into the local extremum space, it will lead to the algorithm is difficult to jump out of the local optimum, and the phenomenon of premature convergence occurs. Introducing the whale fall mechanism into the arithmetic optimization algorithm makes the evolution direction of the algorithm clearer, no longer blind, and is more conducive to improving the global exploration capability of the algorithm, and can explore the global optimal solution more quickly.

2) Polynomial Differential Learning

Differential Evolution [8] was proposed by Storn in 1997 and comprises three main steps: mutation, crossover, and selection, the most critical operation is mutation within this algorithm, and there are various types of strategies for mutation, including DE/rand/1, DE/best/1, DE/best/2, and other 10 types [8]. To disperse the positions of individuals within the population, this article combines the DE/rand/1 and DE/best/1 methods of mutation strategies in Differential Evolution, called the Polynomial Differential Learning Strategy, to enhance the information communication and transfer feedback mechanism between populations, thereby to improve the algorithm's global optimization optimality seeking ability and convergence precision. The Polynomial Differential Strategy is an updated iterative approach centered around the optimal individual and scaled by a polynomial coefficient p whose result is the original vectors, and the mathematical expression is shown in Equation (14):

$$X(k+1)_{new} = X_{best} + p \times (p \times X_{best} - X(k)). \quad (14)$$

where k represents the number of iterations, $X(k)$ denotes the position of an individual, $p \in [-0.5, 0.5]$ indicates a random probability, and X_{best} is the position of the currently optimal individual.

3) Mixed Perturbation Strategy

Relying solely on a single perturbation strategy can speed up the optimization process but also risks confining the algorithm to local optima. To address this, the present study integrates the whale fall mechanism and polynomial differential learning into the AOA, employing a hybrid mutation perturbation on the population positions with r_7 . As demonstrated in Equation (15), when $r_7 > 0.5$, the whale fall mechanism is applied for perturbation to enhance the exploration capability of the algorithm; when $r_7 < 0.5$, polynomial differential learning is utilized for perturbation to enhance the search precision. The two strategies complement each other, which prompts the algorithm to get rid of the local extreme value space and has better adaptability in solving various optimization problems.

$$X_i^{k+1} = \begin{cases} r_4 X_i^k - r_5 X_r^k + r_6 X_{step}, & r_7 < 0.5, \\ bestX_i^k + p \times (p \times bestX_i^k - X_i^k), & r_7 \geq 0.5. \end{cases} \quad (15)$$

Here, r_7 represents a random number between 0 and 1, and $bestX_i^k$ denotes the position of the current optimal individual.

The specific implementation process of the algorithm BSFAOA is illustrated in **Figure 4**. The pseudocode of BSFAOA is presented as **Algorithm 1**.

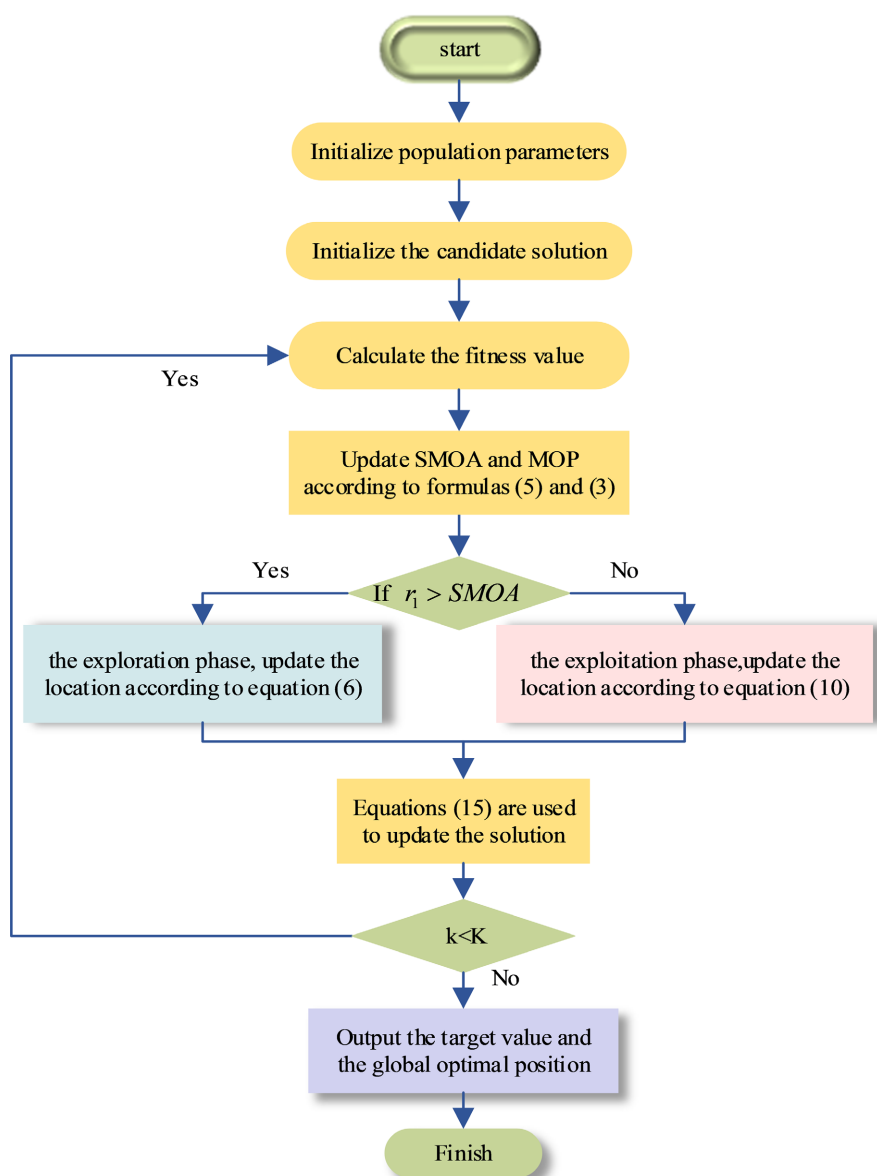


Figure 4. Flowchart of the proposed BSFAOA.

Computational complexity is an indispensable component in evaluating the performance merits of algorithms. The computational complexity of the BSFAOA algorithm primarily depends on three processes: initialization, iterative determination of the optimal solution, and updating the search agent positions. Assuming a population size N , dimension D , and the maximum number of iterations K , the time complexity of the initial phase of BSFAOA is $O(N \times D)$. During the iteration process, the time complexity for N search agents to update their D -dimensional position vectors in K iterations is $O(K \times N \times D)$, and the time for N search agents to discover the optimal solution in each iteration is $O(K \times N)$, and the maximum computational complexity of the hybrid perturbation formula is $O(K \times N \times D)$. Therefore, the overall computational complexity of the BSFAOA proposed in this article is $O(N \times (K + KD + 1)) + O(K \times N \times D)$.

Algorithm 1. Pseudocode of the BSFAOA algorithm.

```

1:   Input related parameter  $N, K, \alpha, \mu$ .
2:   Randomly initialize the location of the solution  $X$ .
3:   While  $k < K$  do
4:     Calculate the fitness values for each solution.
5:     Identify the solution with the minimum fitness value (optimal solution).
6:     Update the  $SMOA$  value using formula (5).
7:     Update the  $MOP$  value using formula (3).
8:     for  $i = 1: N$  do
9:       for  $j = 1: Dim$  do
10:        Generate a random value between 0 and 1 ( $r_1, r_2$  and  $r_3$ ).
11:        if  $r_1 > SMOA$  then
12:          Update the position of the individual using equation (6).
13:        else
14:          Update the position of the individual using equation (10).
15:        end if
16:        Update the position of the individual  $x$  using the formula (15) and calculate its fitness value.
17:      end for
18:    end for
19:     $k = k + 1$ 
20:  end While
21:  Return  $x_{best}$ .

```

$= O(N \times (K + 2KD + 1))$. In comparison, the computational complexity of the original AOA is $O(N \times (K + KD + 1))$. In subsequent sections, the performance of the BSFAOA in handling optimization problems will be verified and confirmed using benchmark functions and real optimization challenges.

4. Experiment Testing and Results Discussion

In this section, to evaluate the performance of the proposed BSFAOA and validate its effectiveness, 23 classical functions from CEC2005 (unimodal (F1 - F7), multimodal (F8 - F13), and fixed-dimension multimodal (F14 - F23)) and 10 CEC2019 test functions are used. The results of the BSFAOA on the 23 test functions are compared with those of other improved algorithms and other well-known optimization algorithms, and the results of the BSFAOA on the CEC2019 test functions are compared with those of algorithms improved using a single strategy.

4.1. Benchmark Testing Function

Table 1 lists 23 test functions with different characteristics, among which: F1 -

F7 are high-dimensional unimodal test functions to evaluate the development precision of the algorithms; F8 - F13 are high-dimensional multimodal test functions to check whether the algorithms are able to jump out of the local points and find the global optimal solution; and F14 - F23 are fixed low-dimensional test functions to assess the stability of the algorithms. **Table 2** provides detailed information on 10 CEC2019 test functions.

4.2. Experiment Environment

Experimental environment: Windows 10 operating system, Intel (R) Core (TM)

Table 1. Benchmark functions.

Function	Name	Dimension	Range	Theoretical optimal value
F1	Sphere	30	[-100, 100]	0
F2	Schwefel2.22	30	[-10, 10]	0
F3	Schwefel1.2	30	[-100, 100]	0
F4	Schwefel2.21	30	[-100, 100]	0
F5	Rosenbrock	30	[-30, 30]	0
F6	Step	30	[-100, 100]	0
F7	Quartic	30	[-1.28, 1.28]	0
F8	Schwefel	30	[-500, 500]	12,569.5
F9	Rastrigin	30	[-5.12, 5.12]	0
F10	Ackley	30	[-32, 32]	0
F11	Griewank	30	[-600, 600]	0
F12	Penalized	30	[-50, 50]	0
F13	Penalized2	30	[-50, 50]	0
F14	Foxholes	2	[-65, 65]	-1
F15	Kowalik	4	[-5, 5]	0.0003
F16	Six-Hump Camel Back	2	[-5, 5]	-1.0316
F17	Branin	2	[-5, 0] \cup [10, 15]	0.398
F18	Gold Stein Price	2	[-2, 2]	3
F19	Hartman3	3	[0, 1]	-3.86
F20	Hartman6	6	[0, 1]	-3.32
F21	Shekel5	4	[0, 10]	-10.1532
F22	Shekel7	4	[0, 10]	-10.4028
F23	Shekel10	4	[0, 10]	-10.5363

Table 2. Properties of IEEE CEC-2019 Benchmarks.

Function	Dimension	Range	Theoretical optimal value
CEC01	9	[-8192, 8192]	1
CEC02	16	[-16,384, 16,384]	1
CEC03	18	[-4, 4]	1
CEC04	10	[-100, 100]	1
CEC05	10	[-100, 100]	1
CEC06	10	[-100, 100]	1
CEC07	10	[-100, 100]	1
CEC08	10	[-100, 100]	1
CEC09	10	[-100, 100]	1
CEC10	10	[-100, 100]	1

i7-12700F CPU @2.10 GHz processor, 16 GB of running memory, MATLAB2018a running software; the experimental parameters of all algorithms are consistent, due to the algorithm itself has randomness, in the experiment was run 30 times and averaged, and the population size was set to $N = 30$ and the maximum number of iterations $K = 500$. The experimental data was statistically calculated in the form of minimum (min), average (avg), and standard deviation (std), rank sum test results (p-value) and algorithm performance rankings (rank), and the overall ranking of the algorithms was obtained based on the performance statistics of the algorithms in their local development accuracy, global exploration capabilities and stability (Overall Ranking) to evaluate the performance of the algorithms. **Table 3** details the specific parameter settings for each algorithm, which are consistent with the parameters presented in the original publications of the algorithms. The SAOA1 algorithm incorporates an adaptive balance factor based on the sine function into the

AOA algorithm, the BSAOA2 algorithm integrates a search strategy combining spiral search with Brownian motion into the AOA algorithm, and the FAOA3 algorithm adds a whale fall mechanism and a polynomial differential learning-based adaptive hybrid perturbation strategy to the AOA algorithm. The BSFAOA algorithm combines all three strategies within the AOA framework.

4.3. Comparison of the BSFAOA with Other Improved Algorithms on CEC2005 Benchmark Functions

To assess the optimization performance of the BSFAOA algorithm and explore the effectiveness of a single strategy for the AOA improvement, this section compares the BSFAOA with AOA, SAOA1, BSAOA2, FAOA3 and other improved algorithms, MSCSO [30], ISSA [31], and other literature-improved versions like IAOA [32] and HSMAAOA [33], in 23 benchmark Comparison experiments are conducted on the test functions to verify the performance of BSFAOA

Table 3. Setting of algorithm parameters.

Algorithms	Parameters	Parameter values
MSCSO [30]	s_M	2
ISSA [31]	λ	0 - 1
AOA [16]	Math optimizer accelerated Function	Linear reduction from 1 to 0.2
	Sensitive parameter	$\alpha = 5$
	Control parameter	$\mu = 0.499$
IAOA [32]	Math optimizer accelerated Function	Linear reduction from 1 to 0.2
	Sensitive parameter	$\alpha = 5$
	Control parameter	$\mu = 0.499$
HSMABOA [33]	The proportion of randomly distributed slime molds in the total population	$z = 0.03$
	Sensitive parameter	$\alpha = 5$
	Control parameter	$\mu = 0.499$
SAOA1	Control parameter	$\mu = 0.499$
	Sensitive parameter	$\alpha = 5$
BSAOA2 and FAOA3	Math optimizer accelerated Function	Linear reduction from 1 to 0.2
	Sensitive parameter	$\alpha = 5$
	Control parameter	$\mu = 0.499$
BSFAOA	Sensitive parameter	$\alpha = 5$
	Control parameter	$\mu = 0.499$
GWO [7]	Convergence parameter (a)	Linear reduction from 2 to 0
SMA [34]	The proportion of randomly distributed slime molds in the total population	$z = 0.03$
HHO [35]	β	1.5
	Strength of breakaway during prey escape (J)	[0, 2]
SOA [36]	Control Parameter (A)	Linear reduction from 2 to 0
WOA [28]	Linear convergence factor (a)	Decreased from 2 to 0
PSO [6]	Inertia weight	Linear decrease from 0.9 to 0.1
	learning factor	$c_1 = 1.5, c_2 = 1.5$
SSA [37]	Leader position update probability	$c_3 = 0.5$

algorithm in terms of convergence speed, the capacity to balance exploration and exploitation, and the accuracy of optimization search. **Table 4** presents the comparison results between BSFAOA and other improved algorithms on 23 test functions. The bold data in the table are the best means (or standard deviations) of the nine algorithms.

Table 4. Comparison of the BSFAOA and other improved algorithms on 23 test functions.

Function	Metrics	BSFAOA	AOA	SAOA1	BSAOA2	FAOA3	MSCSO	ISSA	IAOA	HSMAAOA
F1	min	0	2.28E-292	0	4.99E-12	7.43E-257	0	0	4.91E-165	0
	std	0	9.63E-49	0	8.63E-07	0	0	0	2.69E-145	0
	avg	0	1.76E-49	0	1.24E-06	2.18E-244	0	0	5.10E-146	0
	p-value		1.21E-12	1.00E+00	1.21E-12	1.21E-12	1.00E+00	1.00E+00	1.21E-12	1.00E+00
	rank	1	8	1	9	6	1	1	7	1
F2	min	0	0	0	7.63E-08	1.12E-131	0	0	3.71E-85	0
	std	0	0	0	3.06E-04	1.51E-123	0	0	2.60E-73	0
	avg	0	0	0	2.71E-04	3.08E-124	0	0	4.76E-74	0
	p-value		1.00E+00	1.00E+00	1.21E-12	1.21E-12	1.00E+00	1.00E+00	1.21E-12	1.00E+00
	rank	1	1	1	9	7	1	1	8	1
F3	min	0	2.07E-163	0	2.31E-08	7.03E-265	0	0	2.24E-174	0
	std	0	3.03E-03	0	8.07E-05	0.00E+00	0	0	1.51E-146	0
	avg	0	9.95E-04	0	8.75E-05	8.90E-246	0	0	4.04E-147	0
	p-value		1.21E-12	1.00E+00	1.21E-12	1.21E-12	1.00E+00	1.00E+00	1.21E-12	1.00E+00
	rank	1	9	1	8	6	1	1	7	1
F4	min	0	2.54E-100	0.00E+00	6.81E-07	8.25E-131	0	0	8.67E-85	0
	std	0	1.83E-02	4.87E-04	1.11E-03	6.82E-125	0	0	2.00E-75	0
	avg	0	1.31E-02	8.89E-05	1.41E-03	2.92E-125	0	0	6.06E-76	0
	p-value		1.21E-12	8.15E-02	1.21E-12	1.21E-12	1.00E+00	1.00E+00	1.21E-12	1.00E+00
	rank	1	9	7	8	5	1	1	6	1
F5	min	4.36E-06	2.74E+01	2.56E+01	2.64E+01	5.37E-06	2.44E+01	2.13E-03	2.51E+01	1.71E-03
	std	4.06E-05	3.44E-01	5.62E-01	2.24E-01	5.06E-05	1.35E+00	6.47E+00	7.25E-01	5.13E-01
	avg	7.15E-05	2.84E+01	2.75E+01	2.69E+01	7.61E-05	2.68E+01	1.76E+00	2.66E+01	3.90E-01
	p-value		3.02E-11	3.02E-11	3.02E-11	8.42E-01	3.02E-11	3.02E-11	3.77E-04	3.02E-11
	rank	1	6	8	4	2	8	6	5	3
F6	min	7.07E-07	2.37E+00	1.27E-06	3.38E-03	5.95E-01	3.55E-06	4.22E-08	1.14E+00	1.39E-04
	std	7.94E-07	3.08E-01	2.14E-03	6.53E-02	3.18E-01	4.04E-01	4.02E-07	2.41E-01	1.94E-02
	avg	2.12E-06	2.87E+00	1.06E-03	7.61E-02	1.31E+00	4.67E-01	3.81E-07	1.55E+00	1.59E-02
	p-value		3.02E-11	5.46E-06	3.02E-11	3.02E-11	3.69E-11	1.61E-10	3.02E-11	3.02E-11
	rank	2	9	3	5	7	7	1	6	4
F7	min	8.01E-07	1.09E-06	1.56E-06	6.73E-07	3.64E-07	3.78E-07	4.99E-06	1.18E-06	5.10E-06
	std	4.10E-05	3.69E-05	5.27E-05	2.64E-05	2.27E-05	3.85E-05	1.87E-04	1.66E-05	4.26E-05
	avg	4.48E-05	3.05E-05	4.30E-05	2.96E-05	2.89E-05	4.36E-05	2.11E-04	1.96E-05	4.67E-05
	p-value		3.27E-02	2.46E-01	8.77E-02	7.48E-02	8.65E-01	5.60E-07	2.75E-03	8.19E-01

Continued

	rank	6	4	6	3	2	5	9	1	8
F8	min	-1.26E+04	-6.45E+03	-9.26E+03	-9.43E+03	-1.26E+04	-9.22E+03	-9.80E+03	-8.15E+03	-1.26E+04
	std	2.92E-06	4.18E+02	4.34E+02	6.94E+02	1.43E+00	6.20E+02	8.88E+02	6.72E+02	2.00E-01
	avg	-1.26E+04	-5.69E+03	-8.27E+03	-8.05E+03	-1.26E+04	-7.95E+03	-8.46E+03	-6.64E+03	-1.26E+04
	p-value		3.02E-11	3.02E-11	3.02E-11	4.18E-09	3.02E-11	3.02E-11	3.02E-11	3.02E-11
	rank	1	5	4	8	3	5	5	9	2
F9	min	0	0	0	0	0	0	0	0	0
	std	0	0	0	4.61E-07	1.04E-14	0	0	0	0
	avg	0	0	0	3.85E-07	1.89E-15	0	0	0	0
	p-value		1.00E+00	1.00E+00	1.66E-11	3.34E-01	1.00E+00	1.00E+00	1.00E+00	1.00E+00
	rank	1	1	1	9	8	1	1	1	1
F10	min	8.88E-16	8.88E-16	8.88E-16	2.25E-07	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	std	0	0	0	1.35E-04	1.14E-15	0	0	0	0
	avg	8.88E-16	8.88E-16	8.88E-16	2.28E-04	4.56E-15	8.88E-16	8.88E-16	8.88E-16	8.88E-16
	p-value		1.00E+00	1.00E+00	1.21E-12	2.57E-13	1.00E+00	1.00E+00	1.00E+00	1.00E+00
	rank	1	1	1	9	8	1	1	1	1
F11	min	0	2.66E-03	4.82E-06	1.32E-06	0	0	0	0	0
	std	0	8.63E-02	2.73E-06	3.07E-06	0	0	0	0	0
	avg	0	1.11E-01	9.41E-06	6.98E-06	0	0	0	0	0
	p-value		1.21E-12	1.21E-12	1.21E-12	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
	rank	1	9	7	7	1	1	1	1	1
F12	min	2.17E-07	3.17E-01	3.45E-03	4.78E-03	3.44E-01	3.27E-07	1.43E-09	5.09E-01	1.92E-11
	std	9.46E-05	6.19E-02	2.03E-02	5.78E-03	8.27E-02	1.50E-02	1.09E-08	4.03E-02	1.25E-04
	avg	5.33E-05	4.30E-01	1.22E-02	1.17E-02	5.25E-01	1.76E-02	1.21E-08	5.85E-01	1.04E-04
	p-value		3.02E-11	3.02E-11	3.02E-11	3.02E-11	8.35E-08	3.02E-11	3.02E-11	1.33E-01
	rank	2	7	5	4	9	5	1	8	3
F13	min	1.78E-08	2.61E+00	1.23E+00	3.45E-01	2.65E-09	2.00E-01	1.54E-09	2.35E+00	5.43E-06
	std	9.92E-08	1.06E-01	3.03E-01	2.91E-01	8.42E-07	6.31E-01	1.81E-02	1.57E-01	4.07E-03
	avg	1.56E-07	2.81E+00	1.89E+00	9.08E-01	4.82E-07	1.31E+00	8.76E-03	2.91E+00	3.28E-03
	p-value		3.02E-11	3.02E-11	3.02E-11	4.55E-01	3.02E-11	1.87E-05	3.02E-11	3.02E-11
	rank	1	6	7	5	2	7	4	7	3
F14	min	0.9980	0.9980	0.9980	2.9821	1.9920	0.9980	0.9980	0.9980	0.9980
	std	4.85E+00	4.18E+00	4.51E+00	3.45E+00	4.19E+00	1.89E+00	1.98E+00	5.03E+00	1.41E-12
	avg	7.9892	8.7744	8.0923	10.9633	8.7421	2.2494	1.6203	8.6305	0.9980
	p-value		8.23E-02	3.87E-01	2.94E-04	1.02E-01	1.08E-03	4.34E-10	1.29E-01	1.10E-06

Continued

	rank	4	6	4	6	6	2	2	9	1
F15	min	0.0003	0.0004	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003
	std	2.69E-05	1.01E-02	2.68E-02	2.04E-04	2.40E-05	1.67E-04	4.85E-04	1.32E-03	6.75E-05
	avg	0.0003	0.0059	0.0173	0.0004	0.0003	0.0003	0.0006	0.0008	0.0004
	p-value		7.39E-11	6.70E-11	2.84E-01	3.39E-02	2.78E-07	8.07E-01	1.89E-04	8.15E-05
	rank	2	8	9	5	1	3	6	7	3
F16	min	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	std	2.42E-12	7.36E-08	2.54E-11	2.13E-12	1.57E-11	3.28E-12	6.65E-16	1.70E-11	6.27E-11
	avg	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	p-value		3.02E-11	6.72E-10	1.45E-01	1.46E-10	3.59E-01	2.36E-12	2.37E-10	7.30E-04
	rank	3	9	7	2	5	3	1	6	8
F17	min	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979
	std	1.07E-12	3.68E-08	1.81E-11	8.72E-13	2.43E-11	2.27E-10	0.00E+00	2.64E-11	1.33E-08
	avg	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979
	p-value		3.02E-11	1.39E-10	6.25E-02	6.07E-11	2.92E-09	1.21E-12	5.07E-10	3.02E-11
	rank	3	9	4	2	5	7	1	5	8
F18	min	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	std	1.10E+01	1.16E+01	1.73E+01	1.65E+01	1.76E+01	9.08E-08	1.14E-15	2.84E-10	1.04E-13
	avg	8.4000	9.3001	10.2000	8.4000	11.1000	3.0000	3.0000	3.0000	3.0000
	p-value		4.12E-06	4.08E-05	9.82E-01	1.43E-05	1.11E-04	2.33E-11	1.24E-03	1.46E-10
	rank	5	6	8	6	9	4	1	3	2
F19	min	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005
	std	2.26E-16	2.26E-16	2.26E-16	2.26E-16	2.26E-16	2.26E-16	2.26E-16	2.26E-16	2.26E-16
	avg	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005
	p-value		1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00
	rank	1	1	1	1	1	1	1	1	1
F20	min	-3.3220	-3.2528	-3.3220	-3.3220	-3.3220	-3.3220	-3.3220	-3.3220	-3.3220
	std	5.13E-02	1.08E-01	5.70E-02	5.94E-02	5.83E-02	5.99E-02	5.35E-02	6.05E-02	5.93E-02
	avg	-3.2941	-3.1051	-3.2824	-3.2743	-3.2784	-3.2705	-3.2903	-3.2663	-3.2506
	p-value		2.37E-10	1.63E-02	2.12E-01	2.61E-02	2.61E-02	5.90E-05	2.38E-03	5.26E-04
	rank	1	9	3	5	4	6	2	8	6
F21	min	-10.1532	-6.7026	-10.1531	-10.1531	-10.1532	-10.1532	-10.1532	-10.1532	-10.1531
	std	1.04E-04	9.93E-01	3.11E+00	2.02E+00	1.59E-04	7.66E-07	2.04E+00	2.57E+00	3.03E-04
	avg	-10.1530	-3.8330	-8.0494	-5.8255	-10.1528	-10.1532	-9.0308	-7.9440	-10.1528
	p-value		3.02E-11	5.46E-09	3.50E-09	1.49E-06	3.02E-11	2.57E-01	6.00E-01	1.37E-03

Continued

	rank	2	6	8	6	3	1	5	8	4
F22	min	-10.4029	-8.0482	-10.4029	-10.4029	-10.4029	-10.4029	-10.4029	-10.4028	-10.4029
	std	9.88E-05	1.54E+00	3.41E+00	2.70E+00	1.61E-04	8.62E-07	2.55E+00	2.70E+00	3.41E-04
	avg	-10.4028	-3.8614	-8.1202	-6.4964	-10.4025	-10.4029	-8.3501	-7.7452	-10.4025
	p-value		3.02E-11	1.25E-07	9.51E-06	1.19E-06	3.02E-11	2.69E-02	5.57E-03	6.67E-03
	rank	2	6	7	7	3	1	5	7	4
F23	min	-10.5364	-6.8689	-10.5363	-10.5364	-10.5364	-10.5364	-10.5364	-10.5364	-10.5364
	std	1.08E-04	1.55E+00	3.52E+00	2.54E+00	2.18E-04	7.14E-07	2.39E+00	2.52E+00	3.29E-04
	avg	-10.5362	-4.1504	-7.8071	-6.4803	-10.5360	-10.5364	-9.0526	-6.7507	-10.5360
	p-value		3.02E-11	1.69E-09	3.26E-07	4.35E-05	3.02E-11	3.86E-01	3.83E-06	1.03E-02
	rank	2	6	8	9	3	1	5	6	3
Overall Ranking		1	9	5	8	6	4	2	7	2

Observation of **Table 4** shows that for most of the test functions, the algorithms SAOA1, BSAOA2, and FAOA3, with a single improvement strategy, outperform AOA in terms of mean, variance, and best values, especially BSFAOA showing superior performance in convergence speed, mean, variance, and best values. Across 23 benchmark test functions, except for F7, BSFAOA achieves better mean and standard deviation compared to AOA; for the optimization functions F1-F4, F6, F8, F11-F12, F14, F16-F17 and F20, SAOA1 performs better mean and variance than AOA; and for the optimization functions F3-F7, F11-F13, F15-F17 and F20, BSAOA2 has better mean and variance than AOA; except for F2, F9, F10, F12 and F18, FAOA3 outperforms AOA in mean and standard deviation across the remaining 18 benchmark functions. Among the other improved algorithms of the literature, the results of BSFAOA perform worse than those of HSMAAOA and ISSA results in the fixed-dimensional multimodal test functions F14 and F18, and most of the results in the rest of the test functions are better than those of HSMAAOA, ISSA, and other improved algorithms. Particularly in F13, the BSFAOA demonstrates high optimization accuracy, as F13 is a Penalized2 function with a global minimum and high optimization difficulty, which indicates that the BSFAOA can effectively jump out of the local minima and find the global optimal solution. The bold data in **Table 3** shows that the BSFAOA achieves the minimum value on all 14 test functions, highlighting the significant advantage of the proposed BSFAOA over comparative algorithms.

In order to compare the nine algorithms more intuitively, **Figure 5** illustrates the convergence curves of each algorithm for some of the test functions. As shown in **Figure 5**, the BSFAOA excels in both convergence speed and accuracy, finding the optimal solution more rapidly. Overall, the BSFAOA boosts not only global search capability and prevents the algorithm from getting trapped in local

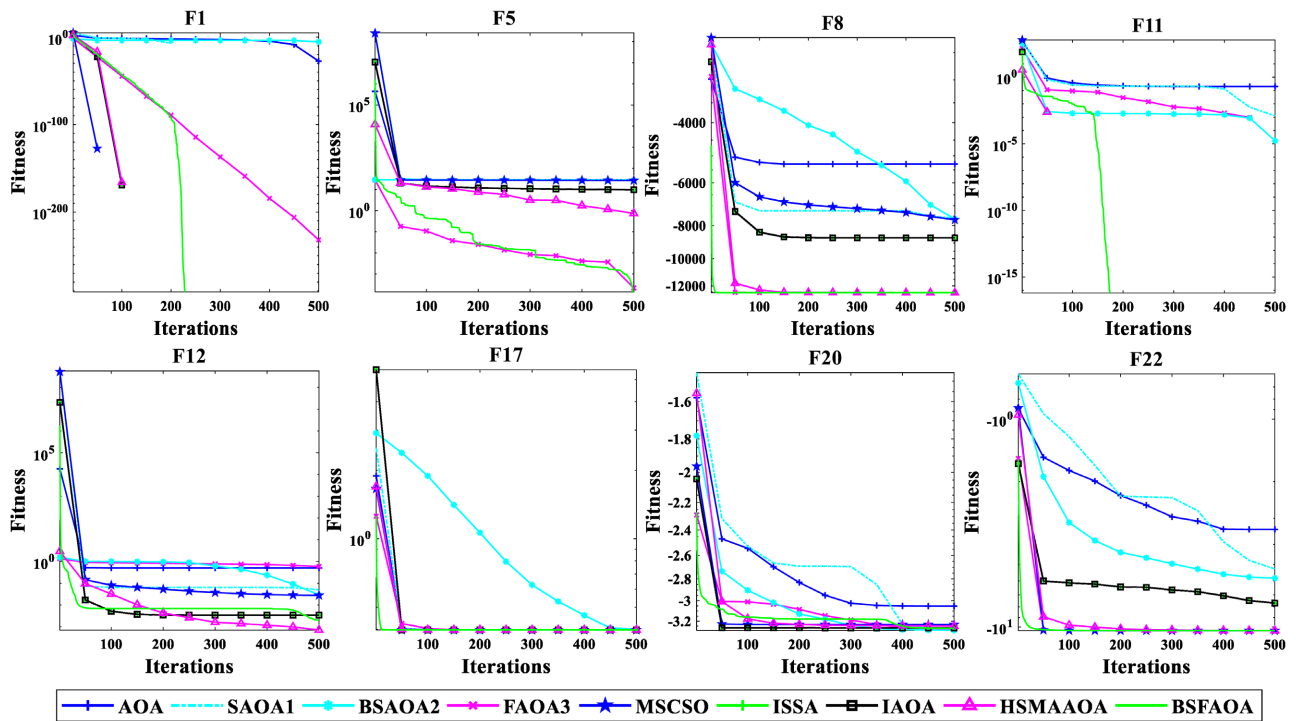


Figure 5. Optimization curves of nine algorithms on some test functions.

optima, but also local search capability is still strong and the accuracy of the algorithm is significantly enhanced. Figure 6 depicts the fitness box plots of the best individuals found in the final generation of each algorithm, and it can be seen that in most cases, the distribution of the objectives of the BSFAOA algorithm is more concentrated than that of the other improved algorithms, which demonstrates its excellent robustness in handling these test problems. The experimental results show that the adaptive balance factor *SMOA* based on sinusoidal function, the search strategy combining spiral search and Brownian motion, and the hybrid perturbation strategy with whale fall mechanism and polynomial difference learning provide the improved AOA algorithm with extremely high efficiency.

4.4. Comparison of the BSFAOA with Other Intelligent Algorithms on CEC2005 Benchmark Functions

To show the optimization performance of BSFAOA more clearly, this article compares the proposed BSFAOA with AOA [16], GWO [7], SMA [34], HHO [35], SOA [36] WOA [28], PSO [6] and SSA [37] algorithms on 23 test functions in terms of best values, average values, and standard deviations, and the specific results are shown in Table 5, where the bold data are the optimal mean (or standard deviation) of the nine algorithms. Their convergence curves on some of the functions are depicted in Figure 7. From Figure 7, the BSFAOA exhibits a faster optimization rate and best fitness values and has an obvious convergence advantage. Despite not finding the optimal value on F15, the BSFAOA demonstrated a superior optimization speed compared to the original AOA algorithm,

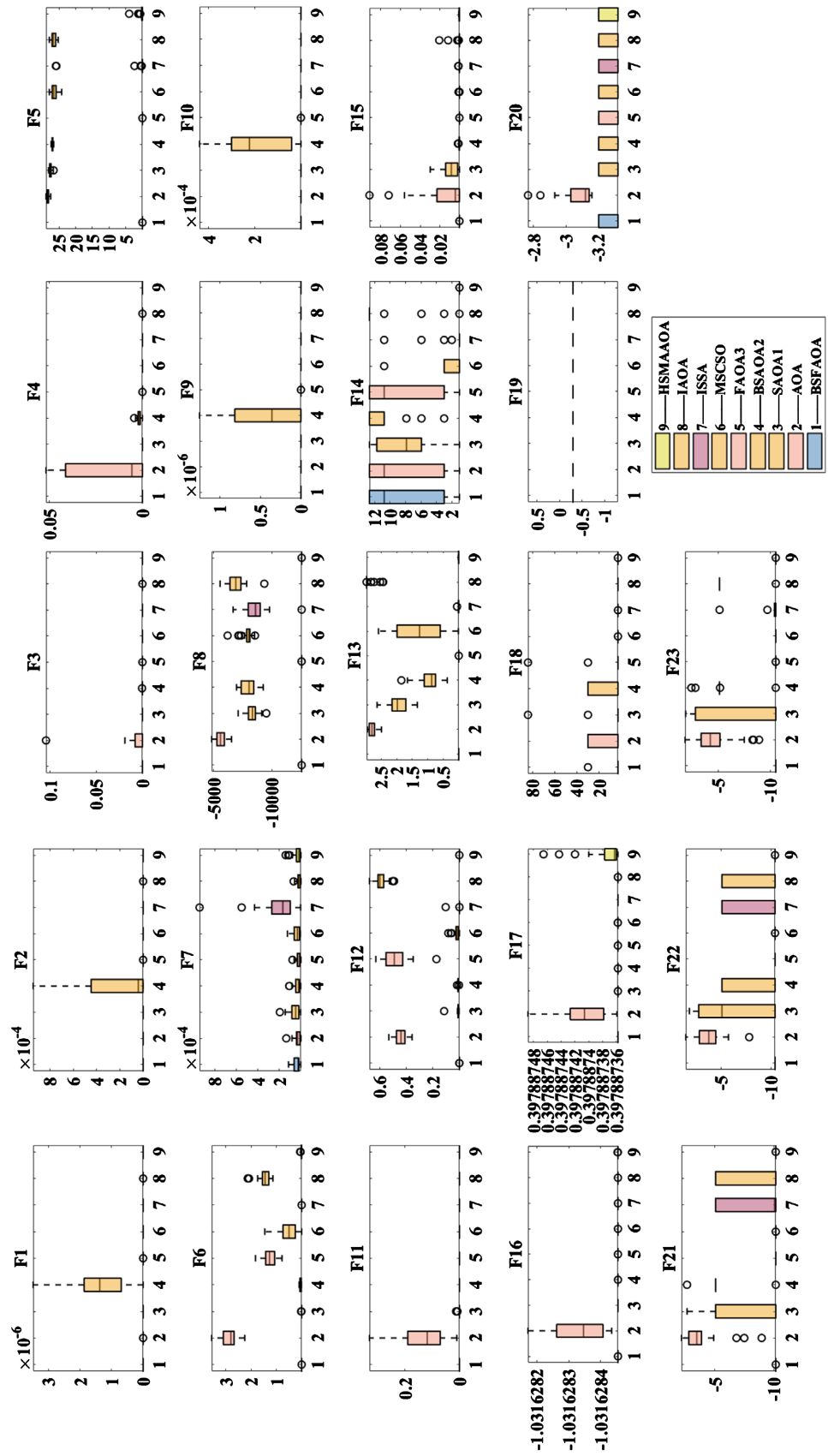


Figure 6. Comparison results of nine algorithms on 23 test functions in box plots.

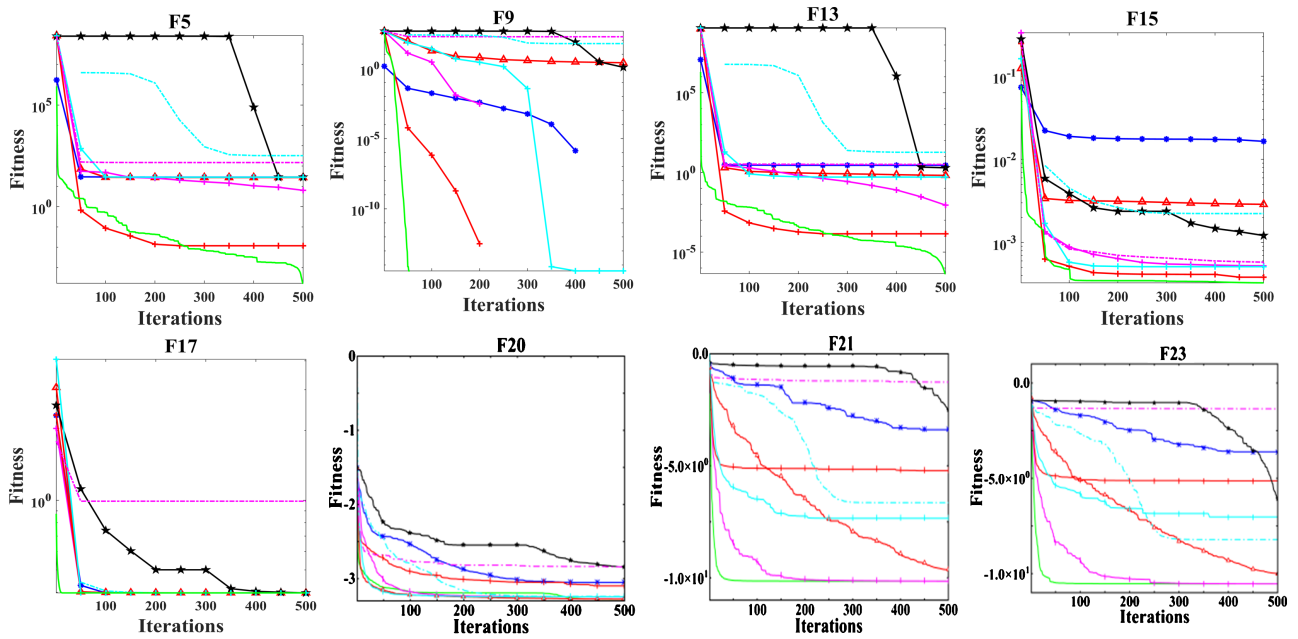


Figure 7. Convergence curve of each algorithm on some test functions.

and the extreme value converged to has been infinitely close to the theoretical optimal value of 0.3979, whose curve is smooth and below the other eight algorithms, which verifies the effectiveness of the improvement strategy, which enables the algorithm to maintain a good diversity of the populations and global convergence, while improving the optimization speed and accuracy.

According to the data in **Table 5**, the BSFAOA is ranked first among all nine algorithms, demonstrating a significant convergence advantage on the unimodal and multimodal test functions F1 - F13, with its optimization precision surpassing that of other comparative algorithms in most cases. Particularly notable is its performance on F11, where the BSFAOA consistently converges to the theoretical optimum, and many local extremes in F11 are continuously distributed, and it plays a role in testing the capability of the algorithm to jump out of local extreme values. The bolded data in **Table 5** indicate that, except for F6, F14, and F18, BSFAOA achieved the most outstanding results in the other 20 test functions, highlighting the superiority of the algorithm BSFAOA proposed in this study. In addition, **Figure 8** presents the box plots of the nine algorithms for 23 test functions, showcasing that in most instances, the objective distribution of the BSFAOA algorithm is narrower than that of other comparative algorithms, which indicates that it has good stability in solving these test problems and that the accuracy of the combined convergence speed and solution is superior to that of the other algorithms.

4.5. Comparison of the BSFAOA and Algorithm Improved by a Single Strategy on the CEC2019 Test Functions

To ensure the fairness of the experiment, all experimental conditions for each algorithm remained consistent with Section 3.2. **Table 6** presents the optimal

Table 5. Comparison of the BSFAOA and other intelligent algorithms on 23 test functions.

Function	Metrics	BSFAOA	AOA	GWO	SMA	HHO	SOA	WOA	PSO	SSA
F1	min	0	0	4.17E-73	0	1.60E-219	8.49E-35	1.09E-187	8.85E-01	5.84E-09
	std	0	0	2.77E-70	0	0.00E+00	7.03E-29	0.00E+00	9.59E-02	1.78E-09
	avg	0	1.31E-175	1.52E-70	0	4.88E-194	1.79E-29	9.53E-172	1.13E+00	8.60E-09
	p-value		1.31E-07	1.21E-12	1.00E+00	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
	rank	1	4	6	1	3	7	5	9	8
F2	min	0	0	5.97E-42	0	2.64E-110	9.05E-21	4.63E-119	4.27E+00	1.39E-04
	std	0	0	6.40E-41	0	6.41E-98	7.47E-19	6.54E-108	3.67E-01	8.58E-01
	avg	0	0	5.85E-41	3.84E-209	1.63E-98	5.01E-19	1.33E-108	5.08E+00	5.89E-01
	p-value		1.00E+00	1.21E-12	1.95E-09	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
	rank	1	1	6	3	5	7	4	8	8
F3	min	0	0.00E+00	1.12E-26	0	2.12E-189	4.10E-20	3.78E+02	9.09E-01	4.04E+00
	std	0	2.60E-03	1.25E-19	0	8.97E-154	2.64E-16	7.12E+03	2.37E-01	2.71E+01
	avg	0	6.49E-04	3.34E-20	0	1.64E-154	1.09E-16	1.21E+04	1.51E+00	3.71E+01
	p-value		6.25E-10	1.21E-12	1.00E+00	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
	rank	1	6	4	1	3	5	9	7	8
F4	min	0	3.29E-212	1.23E-19	0	1.63E-106	1.01E-11	1.68E-02	3.04E-01	8.01E-01
	std	0	1.92E-02	2.90E-17	0	2.00E-97	2.04E-08	2.61E+01	1.88E-02	2.42E+00
	avg	0	1.31E-02	1.96E-17	3.84E-228	5.80E-98	5.62E-09	2.53E+01	3.74E-01	3.56E+00
	p-value		1.21E-12	1.21E-12	6.25E-10	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
	rank	1	6	4	2	3	5	9	6	8
F5	min	8.59E-07	2.66E+01	2.53E+01	1.16E-06	5.08E-06	2.72E+01	2.60E+01	1.20E+02	2.30E+01
	std	1.10E-05	4.39E-01	6.42E-01	2.18E-01	1.45E-03	6.18E-01	2.79E-01	1.09E+01	2.38E+02
	avg	1.84E-05	2.81E+01	2.66E+01	2.40E-01	1.16E-03	2.80E+01	2.66E+01	1.45E+02	1.28E+02
	p-value		3.02E-11	3.02E-11	5.07E-10	5.00E-09	3.02E-11	3.02E-11	3.02E-11	3.02E-11
	rank	1	5	5	3	2	5	4	8	8
F6	min	2.82E-07	1.93E+00	5.76E-06	6.46E-07	2.23E-09	1.49E+00	1.39E-03	3.27E+00	4.28E-09
	std	1.35E-07	2.42E-01	2.48E-01	3.33E-04	1.94E-05	4.94E-01	1.57E-03	5.19E-01	2.32E-09
	avg	5.08E-07	2.47E+00	3.33E-01	6.37E-04	1.67E-05	2.90E+00	3.84E-03	4.80E+00	9.02E-09
	p-value		3.02E-11	3.02E-11	4.50E-11	6.01E-08	3.02E-11	3.02E-11	3.02E-11	3.02E-11
	rank	2	6	6	4	3	8	5	9	1
F7	min	3.96E-07	4.08E-07	1.14E-04	5.10E-06	7.93E-07	5.61E-05	8.50E-05	9.52E-01	1.97E-02
	std	2.11E-05	2.84E-05	2.72E-04	4.66E-05	5.11E-05	5.51E-04	6.72E-04	5.55E-01	2.23E-02
	avg	2.14E-05	2.32E-05	5.48E-04	6.05E-05	4.85E-05	7.18E-04	8.65E-04	2.27E+00	4.62E-02
	p-value		7.84E-01	3.02E-11	8.66E-05	9.88E-03	3.69E-11	3.34E-11	3.02E-11	3.02E-11

Continued

	rank	1	2	5	3	3	6	7	9	8
F8	min	-12569.49	-6797.62	-7376.99	-12569.48	-12569.49	-7455.07	-12569.43	-3563.56	-9036.13
	std	4.75E-07	4.11E+02	5.70E+02	3.17E-02	1.05E+02	6.39E+02	1.33E+03	4.04E+02	6.64E+02
	avg	-12569.49	-6196.55	-6331.20	-12569.44	-12550.21	-5537.53	-11546.24	-2589.07	-7772.77
	p-value		3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
	rank	1	4	4	2	3	9	6	6	6
F9	min	0	0	0	0	0	0	0	1.38E+02	1.79E+01
	std	0	0	1	0	0	1.83E+00	2.08E-14	1.24E+01	1.80E+01
	avg	0	0	0	0	0	3.35E-01	3.79E-15	1.65E+02	5.30E+01
	p-value		1.00E+00	3.34E-01	1.00E+00	1.00E+00	8.15E-02	3.34E-01	1.21E-12	1.21E-12
	rank	1	1	6	1	1	7	5	8	8
F10	min	8.88E-16	8.88E-16	7.99E-15	8.88E-16	8.88E-16	2.00E+01	8.88E-16	1.71E+00	1.78E-05
	std	0	0	2.41E-15	0	0	1.96E-03	2.35E-15	9.60E-02	9.74E-01
	avg	8.88E-16	8.88E-16	1.36E-14	8.88E-16	8.88E-16	2.00E+01	4.09E-15	1.96E+00	1.63E+00
	p-value		1.00E+00	3.70E-13	1.00E+00	1.00E+00	1.21E-12	1.01E-08	1.21E-12	1.21E-12
	rank	1	1	6	1	1	7	5	7	7
F11	min	0	3.21E-04	0	0	0	0	0	3.59E-02	2.16E-08
	std	0	5.76E-02	4.53E-03	0.00E+00	0.00E+00	5.99E-03	1.04E-02	4.02E-03	9.02E-03
	avg	0	6.43E-02	1.11E-03	0.00E+00	0.00E+00	1.09E-03	2.73E-03	4.29E-02	7.22E-03
	p-value		1.21E-12	1.61E-01	1.00E+00	1.00E+00	3.34E-01	1.61E-01	1.21E-12	1.21E-12
	rank	1	9	4	1	1	4	7	6	7
F12	min	1.84E-09	2.57E-01	1.16E-06	1.20E-06	5.80E-11	1.07E-01	1.86E-04	7.65E-01	3.72E-03
	std	1.55E-09	3.21E-02	1.50E-02	4.69E-04	1.07E-06	1.34E-01	5.53E-03	5.87E-02	2.62E+00
	avg	4.14E-09	3.13E-01	2.38E-02	3.98E-04	6.43E-07	2.49E-01	2.42E-03	9.30E-01	3.85E+00
	p-value		3.02E-11	3.02E-11	3.02E-11	1.86E-09	3.02E-11	3.02E-11	3.02E-11	3.02E-11
	rank	1	6	5	3	2	7	4	8	9
F13	min	1.39E-09	2.56E+00	2.02E-05	4.25E-08	3.11E-12	1.48E+00	1.79E-03	2.59E+00	4.63E-10
	std	2.17E-08	1.07E-01	1.75E-01	2.14E-04	1.06E-05	1.81E-01	6.15E-02	1.66E-01	8.65E-03
	avg	2.76E-08	2.78E+00	3.20E-01	4.04E-04	7.28E-06	1.88E+00	3.60E-02	3.10E+00	9.12E-03
	p-value		3.02E-11	3.02E-11	4.50E-11	2.02E-08	3.02E-11	3.02E-11	3.02E-11	7.96E-03
	rank	1	6	6	3	2	8	5	8	4
F14	min	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980	0.9980	2.8104	0.9980
	std	4.99E+00	4.41E+00	3.99E+00	4.37E-14	1.81E-01	8.07E-01	2.48E+00	2.78E+00	1.82E-16
	avg	7.5668	9.0278	3.8057	0.9980	1.0311	1.3948	2.1782	11.0323	0.9980
	p-value		1.07E-02	1.81E-01	6.30E-05	6.72E-05	7.66E-04	4.20E-03	1.85E-05	8.39E-12

Continued

	rank	8	8	6	2	3	4	5	7	1
F15	min	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0004	0.0003
	std	2.45E-04	2.88E-02	9.35E-03	1.54E-04	1.69E-04	2.35E-04	3.95E-04	4.77E-05	3.34E-04
	avg	0.0004	0.0115	0.0063	0.0004	0.0004	0.0012	0.0006	0.0004	0.0009
	p-value		4.20E-10	4.23E-03	3.59E-05	9.03E-04	2.44E-09	3.26E-07	8.48E-09	1.20E-08
	rank	4	9	8	1	2	5	6	2	6
F16	min	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316
	std	3.92E-13	8.27E-08	4.65E-09	1.05E-11	5.65E-13	4.05E-07	9.11E-12	9.86E-03	7.48E-15
	avg	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0252	-1.0316
	p-value		3.02E-11	3.02E-11	4.43E-03	8.73E-07	3.02E-11	1.76E-01	3.02E-11	4.00E-11
	rank	2	7	6	5	2	8	4	9	1
F17	min	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979
	std	1.28E-13	3.66E-08	1.37E-07	6.24E-10	1.17E-08	1.43E-05	6.17E-07	6.85E-01	2.80E-15
	avg	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.3979	0.8381	0.3979
	p-value		3.01E-11	3.01E-11	3.01E-11	4.61E-10	3.01E-11	3.01E-11	3.01E-11	3.61E-10
	rank	2	5	6	3	4	8	7	9	1
F18	min	3.00	3.00	3.00	3.00	3.00	3.00	3.00	3.55	3.00
	std	1.69E+01	1.02E+01	1.48E+01	2.67E-12	1.72E-09	6.54E-06	4.38E-06	1.06E+01	1.05E-13
	avg	9.30	7.50	5.70	3.00	3.00	3.00	3.00	22.18	3.00
	p-value		8.29E-06	6.74E-06	1.99E-02	2.06E-01	9.51E-06	9.51E-06	2.03E-07	3.08E-08
	rank	9	6	7	2	3	5	4	8	1
F19	min	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005	-1.9446	-0.3005
	std	2.26E-16	2.26E-16	2.26E-16	2.26E-16	2.26E-16	2.26E-16	2.26E-16	4.81E-01	2.26E-16
	avg	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005	-0.3005	-1.2892	-0.3005
	p-value		1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.00E+00	1.91E-07	1.00E+00
	rank	1	1	1	1	1	1	1	9	1
F20	min	-3.3220	-3.2453	-3.3220	-3.3220	-3.2981	-3.1327	-3.3220	-3.2136	-3.3220
	std	5.83E-02	6.86E-02	6.72E-02	4.51E-02	6.78E-02	2.11E-01	7.31E-02	1.86E-01	4.56E-02
	avg	-3.2784	-3.1183	-3.2534	-3.2229	-3.1893	-3.0063	-3.2616	-2.9803	-3.2219
	p-value		2.61E-10	8.15E-05	8.24E-02	2.68E-06	3.02E-11	1.04E-04	2.61E-10	1.37E-03
	rank	1	7	3	2	6	8	5	8	3
F21	min	-10.1532	-5.7616	-10.1531	-10.1532	-10.0751	-10.1382	-10.1532	-4.7208	-10.1532
	std	3.81E-05	9.47E-01	1.92E+00	3.75E-05	9.17E-01	4.24E+00	1.93E+00	9.15E-01	2.45E+00
	avg	-10.1531	-3.8788	-9.3093	-10.1531	-5.2223	-3.5728	-9.3032	-1.5866	-8.7205
	p-value		3.02E-11	1.61E-10	5.69E-01	3.02E-11	3.02E-11	9.26E-09	3.02E-11	1.95E-03
	rank									

Continued

	rank	2	6	3	1	4	9	5	6	8
F22	min	-10.4029	-8.7123	-10.4029	-10.4029	-10.4018	-10.3969	-10.4029	-6.3696	-10.4029
	std	3.56E-05	1.37E+00	1.34E+00	3.47E-05	1.34E+00	4.18E+00	2.22E+00	1.09E+00	1.34E+00
	avg	-10.4029	-4.4152	-10.0496	-10.4029	-5.4406	-7.3740	-9.3862	-1.6075	-10.0500
	p-value		3.02E-11	9.92E-11	9.35E-01	3.02E-11	3.02E-11	1.19E-06	3.02E-11	8.48E-09
	rank	1	8	3	1	6	8	6	5	3
F23	min	-10.5364	-9.5513	-10.5363	-10.5364	-5.1285	-10.5352	-10.5364	-5.7021	-10.5364
	std	3.35E-05	2.12E+00	1.46E-04	3.06E-05	2.38E-04	3.34E+00	3.02E+00	1.09E+00	2.58E+00
	avg	-10.5364	-4.5326	-10.5361	-10.5364	-5.1282	-8.9279	-8.4775	-1.7736	-9.2943
	p-value		3.02E-11	3.82E-10	3.11E-01	3.02E-11	3.02E-11	2.39E-08	3.02E-11	6.77E-05
	rank	2	6	3	1	4	6	6	6	4
Overall Ranking	1	6	4	2	3	8	7	9	5	

values, means, standard deviations, rank sum test results, and rankings of all compared algorithms. The bold data in the table represent the best means (or standard deviations). Based on the data in **Table 6**, it can be seen that both BSFAOA and FAOA3 have improved by 4 orders of magnitude compared to the AOA in terms of mean accuracy and standard deviation accuracy on CEC02; based on the bolded data in **Table 6**, it can be seen that the optimal means and standard deviations are distributed in FAOA3, BSAOA2, SAOA1, and BSFAOA; in terms of the final ranking (Overall Ranking) results, the BSFAOA is ranked first, followed by FAOA3, BSAOA2, SAOA1, and AOA. **Figure 9** illustrates the convergence curves of the algorithms on the CEC2019 test functions, and the convergence curves of SAOA1, BSAOA2, and FAOA3 on the vast majority of the functions are located underneath the AOA curves, with the optimality search accuracy and convergence speed better than the basic AOA, and the BSFAOA algorithm has better search accuracy and convergence speed than SAOA1, BSAOA2, FAOA3 and AOA algorithms. Overall, when combining the three improvement strategies, the algorithm optimization accuracy is significantly better than the AOA improved by a single strategy, demonstrating the superiority of the BSFAOA and verifying the effectiveness of the single improvement strategy in this study. In addition, **Figure 10** displays box plots of the solutions of the five algorithms for the CEC2019 test functions, showing that in most cases, the target distribution of the BSFAOA algorithm is smaller than that of the other compared algorithms, indicating good stability performance of the BSFAOA algorithm in solving these test problems.

5. Application of the BSFAOA in Optimization Problems

In this section, to further evaluate the applicability and effectiveness of the BSFAOA algorithm, four practical optimization problems are solved, including

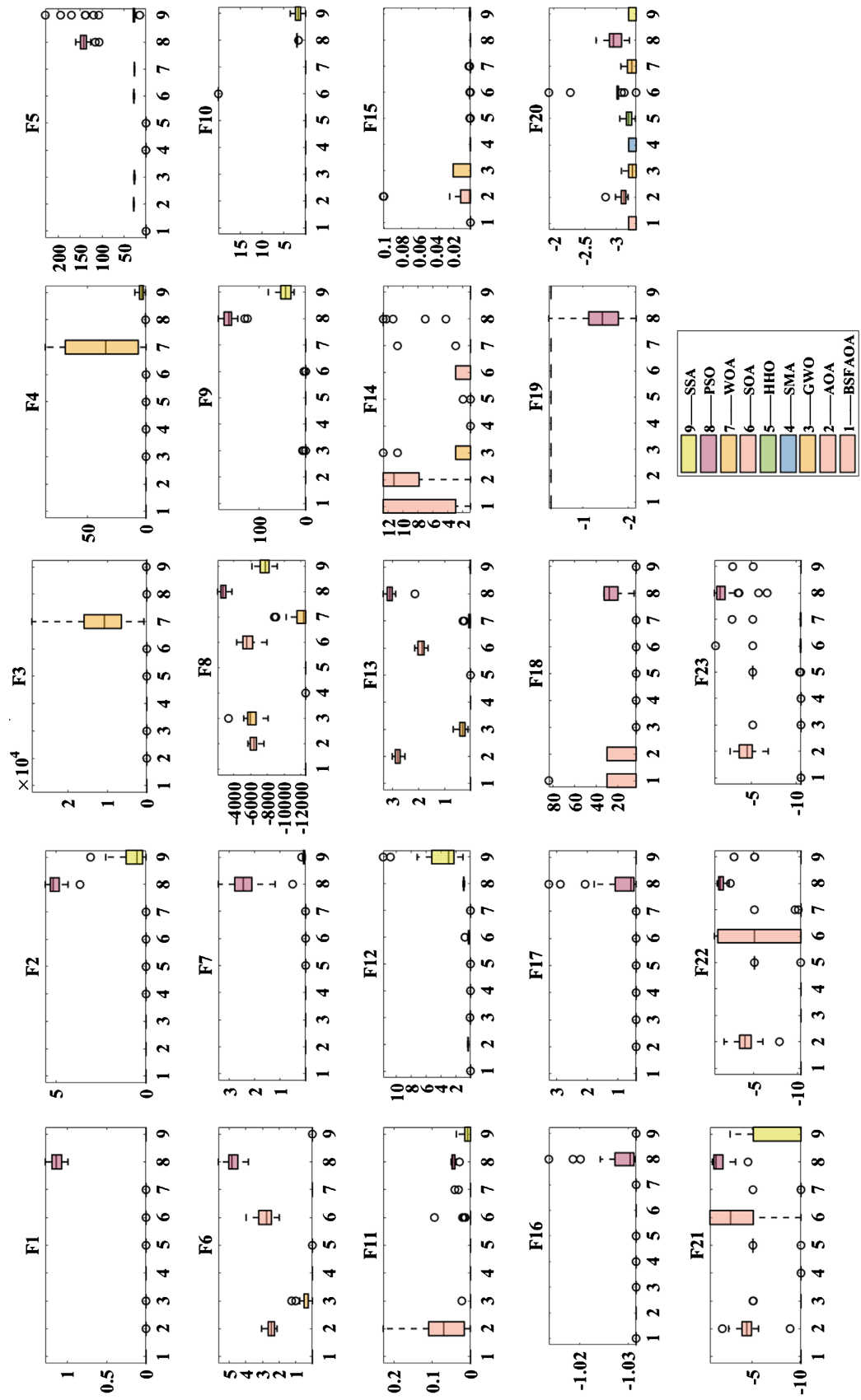


Figure 8. Comparison results of various algorithms on 23 test functions in box plots.

Table 6. Comparison of the BSFAOA and algorithm improvement through a Single Strategy on CEC2019 test functions.

Function	Metrics	BSFAOA	AOA	SAOA1	BSAOA2	FAOA3
F1	min	1	1	1	1	1
	std	0	7.8255E+06	4.2207E+07	9.4961E+05	4.9364E-03
	avg	1	2.1390E+06	1.1528E+07	1.7394E+05	1.0009E+00
	p-value		5.7720E-11	1.6572E-11	2.9343E-05	8.1523E-02
	rank	1	4	5	3	2
F2	min	4.2280E+00	6.2028E+03	4.8495E+03	4.9442E+00	4.3429E+00
	std	2.2492E-01	2.7325E+03	3.6390E+03	4.0192E+03	1.5338E-01
	avg	4.6132E+00	1.1253E+04	1.1593E+04	6.1898E+03	4.6869E+00
	p-value		3.0199E-11	3.0199E-11	4.5043E-11	1.6238E-01
	rank	1	3	5	4	1
F3	min	1.4094E+00	8.5860E+00	6.0652E+00	8.1168E+00	3.9969E+00
	std	1.8194E+00	9.1089E-01	1.2096E+00	8.3688E-01	1.8102E+00
	avg	3.6565E+00	1.0478E+01	9.3890E+00	1.0103E+01	7.6745E+00
	p-value		3.0199E-11	5.4941E-11	3.0199E-11	7.1186E-09
	rank	2	5	2	1	2
F4	min	9.9549E+00	3.7812E+01	3.4829E+01	3.4695E+01	2.7649E+01
	std	1.5403E+01	1.4025E+01	2.3770E+01	1.2516E+01	1.3647E+01
	avg	5.1899E+01	5.7585E+01	7.4310E+01	5.4002E+01	5.2856E+01
	p-value		1.8090E-01	2.2539E-04	6.9522E-01	8.7663E-01
	rank	3	4	5	1	1
F5	min	1.2520E+00	2.8210E+01	4.8294E+01	5.3399E+00	9.4895E+00
	std	1.2314E+00	2.7515E+01	2.6463E+01	9.4855E+00	2.0547E+01
	avg	2.8399E+00	8.1705E+01	9.7859E+01	1.5953E+01	5.0384E+01
	p-value		3.0199E-11	3.0199E-11	3.3384E-11	3.0199E-11
	rank	1	4	4	2	3
F6	min	3.6019E+00	7.7205E+00	9.7787E+00	4.8677E+00	5.7422E+00
	std	1.7044E+00	1.4729E+00	1.3381E+00	1.4903E+00	1.3863E+00
	avg	7.5802E+00	1.0923E+01	1.3066E+01	8.1354E+00	9.1194E+00
	p-value		9.2603E-09	5.4941E-11	2.5805E-01	6.9125E-04
	rank	2	5	2	2	1
F7	min	3.4246E+02	9.5548E+02	1.1311E+03	6.9239E+02	8.1495E+02
	std	2.2735E+02	2.6501E+02	2.6013E+02	2.5500E+02	2.2152E+02
	avg	8.9551E+02	1.4007E+03	1.6648E+03	1.2136E+03	1.2305E+03
	p-value		7.7725E-09	1.0937E-10	1.9963E-05	1.3853E-06

Continued

	rank	1	4	4	3	2
F8	min	2.8912E+00	4.0315E+00	4.0935E+00	3.5822E+00	3.8130E+00
	std	4.2845E-01	3.0349E-01	3.5273E-01	4.1266E-01	2.8676E-01
	avg	4.1588E+00	4.7997E+00	5.1976E+00	4.5692E+00	4.5009E+00
	p-value		1.0666E-07	6.1210E-10	1.7836E-04	9.5207E-04
	rank	2	2	5	4	1
F9	min	1.1340E+00	1.4951E+00	1.5904E+00	1.2357E+00	1.3157E+00
	std	1.2209E-01	6.0739E-01	4.4410E-01	9.3915E-02	7.8217E-01
	avg	1.3334E+00	3.0440E+00	3.1834E+00	1.4008E+00	2.4541E+00
	p-value		3.6897E-11	3.3384E-11	2.6077E-02	1.8500E-08
	rank	1	3	3	1	3
F10	min	2.0712E+01	2.0950E+01	2.0990E+01	2.0621E+01	2.1053E+01
	std	5.2041E-02	5.5023E-02	4.1116E-03	1.0071E-01	4.4784E-02
	avg	2.0983E+01	2.1124E+01	2.0995E+01	2.1088E+01	2.1118E+01
	p-value		5.0723E-10	1.1199E-01	5.5727E-10	3.0199E-11
	rank	2	5	1	4	3
Overall Ranking		1	5	4	3	2

the optimization of SVR hyperparameters, the welded beam design problem, the pressure vessel design problem, and the multiple-disc clutch brake design problem.

5.1. Application of SVR Hyperparameter Optimization

In constructing the Support Vector Regression (SVR) model, the Radial Basis Function (RBF) kernel was selected. Due to the significant influence of the penalty factor C and the RBF kernel function parameter g on the SVR fitting accuracy, the sum of SVR is optimized using the AOA, PSO, and BSFAOA algorithms, which in turn constructs the SVR regression prediction model. The dataset used is derived from the literature [38] and contains 254 samples, each containing 7 features. The initial parameters of the algorithm are set as follows: the maximum number of iterations is 500, the population size is 30, the upper bound of the parameter values $[-8, 8]$, and the lower bound of the parameter values $[-8, 8]$. The mean square error of tenfold cross-validation is used to calculate the fitness value, and the smallest value is the optimal fitness value, according to which the corresponding optimal individual positions are output as the values of C and g . The tenfold cross-validation method provides the best compromise between the computational cost and the effective parameter estimates, and iterative optimization is carried out according to the algorithmic process. The construction process of the BSFAOA-SVR model is depicted in **Figure 11**. The model performance is evaluated by R^2 , mean absolute error

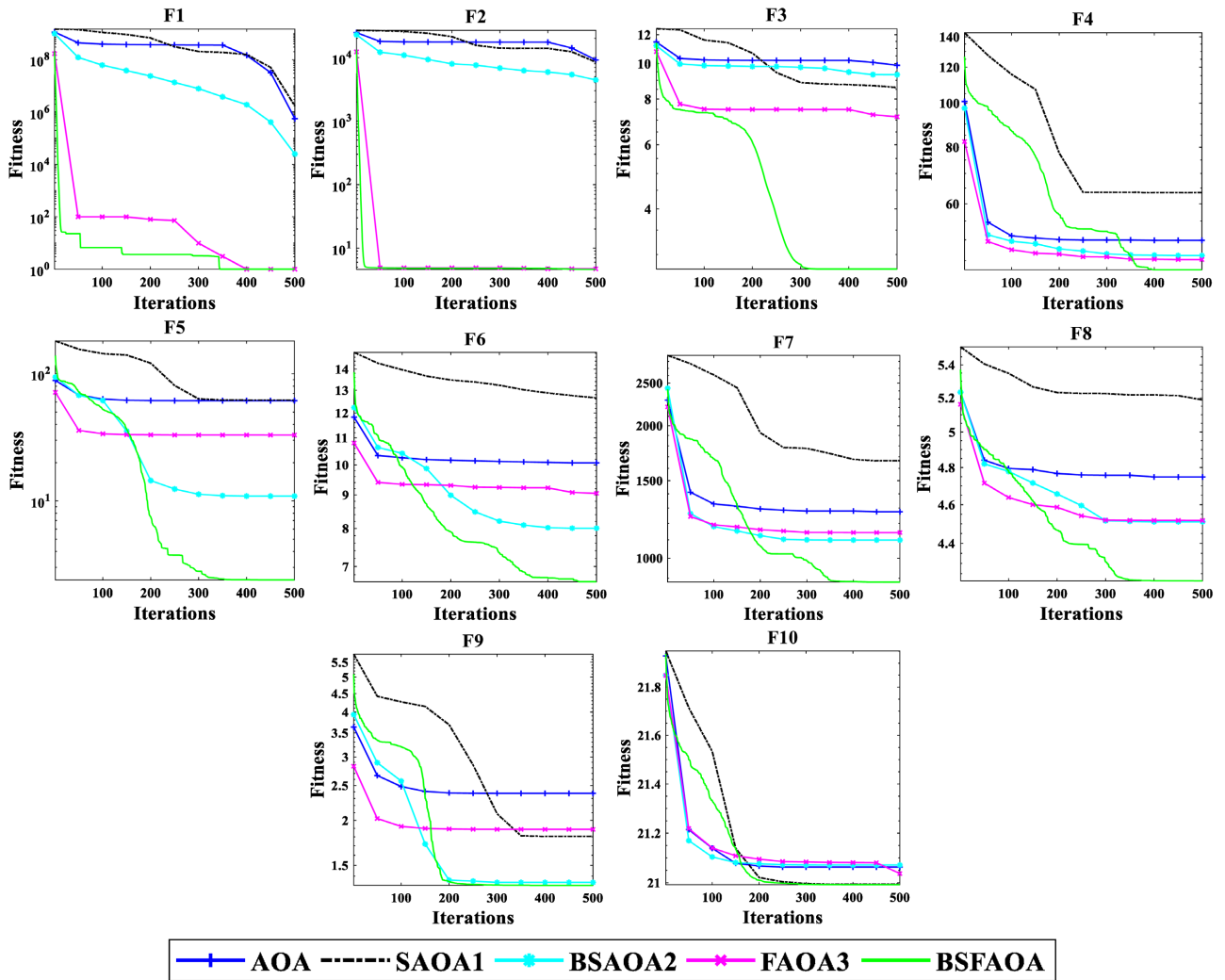


Figure 9. Convergence curve of each algorithm on CEC2019 test functions.

MAE, mean absolute percentage error *MAPE*, and root mean square error *RMSE* on the test set, the smaller the values of *MAE*, *MAPE* and *RMSE*, the closer the value of R^2 is to 1, indicates that the better the modeling effect, which are computed by the formulas of:

$$MAE = \frac{1}{j} \sum_{i=1}^j |f_i - y_i| \tag{16}$$

$$MAPE = \frac{1}{j} \sum_{i=1}^j \left| \frac{f_i - y_i}{y_i} \right| \tag{17}$$

$$RMSE = \sqrt{\frac{1}{j} \sum_{i=1}^j (f_i - y_i)^2} \tag{18}$$

$$R^2 = 1 - \frac{\sum_{i=1}^j (f_i - y_i)^2}{\sum_{i=1}^j (\bar{f}_i - y_i)^2} \tag{19}$$

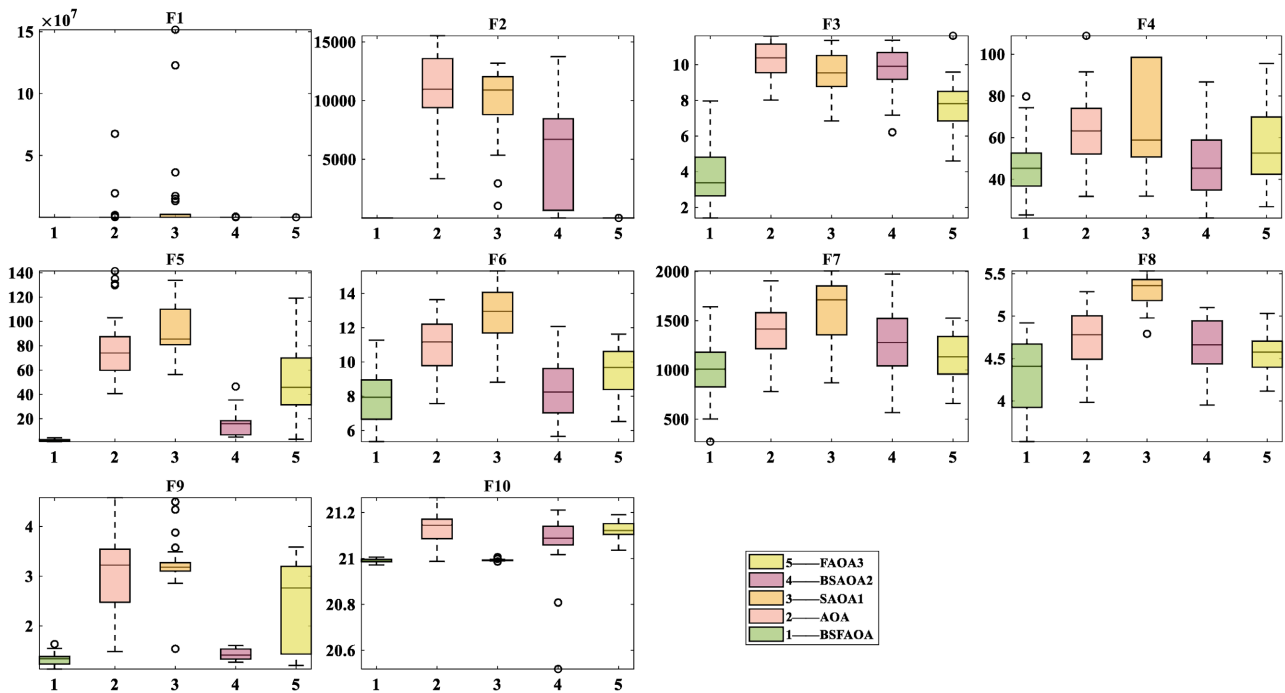


Figure 10. Comparison results of five algorithms on CEC2019 test functions in box plots.

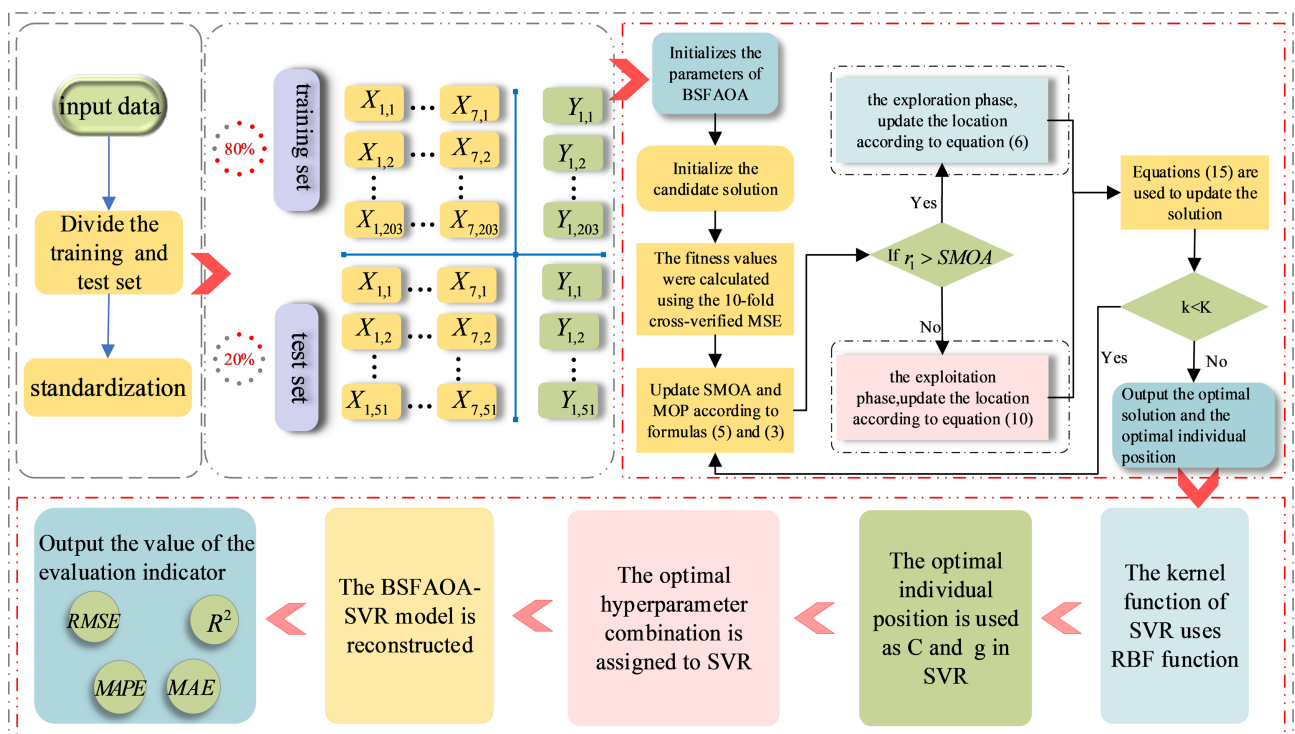


Figure 11. The construction process of BSFAOA-SVR model.

where f_i represents the predicted value for the i sample, y_i denotes the actual value for the i sample, and j indicates the number of samples.

The comparison of the experimental results of different SVR models is presented in Table 7, from which it can be seen that compared with the rest of the

SVR models, the BSFAOA-SVR regression prediction model has the smallest *MAE*, *RMSE* and *MAPE*, and the highest R^2 . After the BSFAOA algorithm optimizes the hyperparameters $C = 1.5398$ and $g = 0.5608$ of SVR, the test set R^2 of the optimized model is improved to 0.8591, *MAE* is reduced to 1.3937, *RMSE* is reduced to 1.8940, and *MAPE* is reduced to 0.1455, which further proves that the improvement strategy added in this article enhances the performance of the original AOA, leading to enhanced model performance and better prediction stability, which indicates that the BSFAOA is resultful for optimizing SVR.

The original SVR model and the BSFAOA-SVR model are constructed separately, and the convergence performance of the optimization process of the BSFAOA-SVR model is depicted in **Figure 12(a)**. The comparison of predicted and actual values for the training set is shown in **Figure 12(b)** and for the test set in **Figure 12(c)**, from which it can be observed that the BSFAOA-SVR model

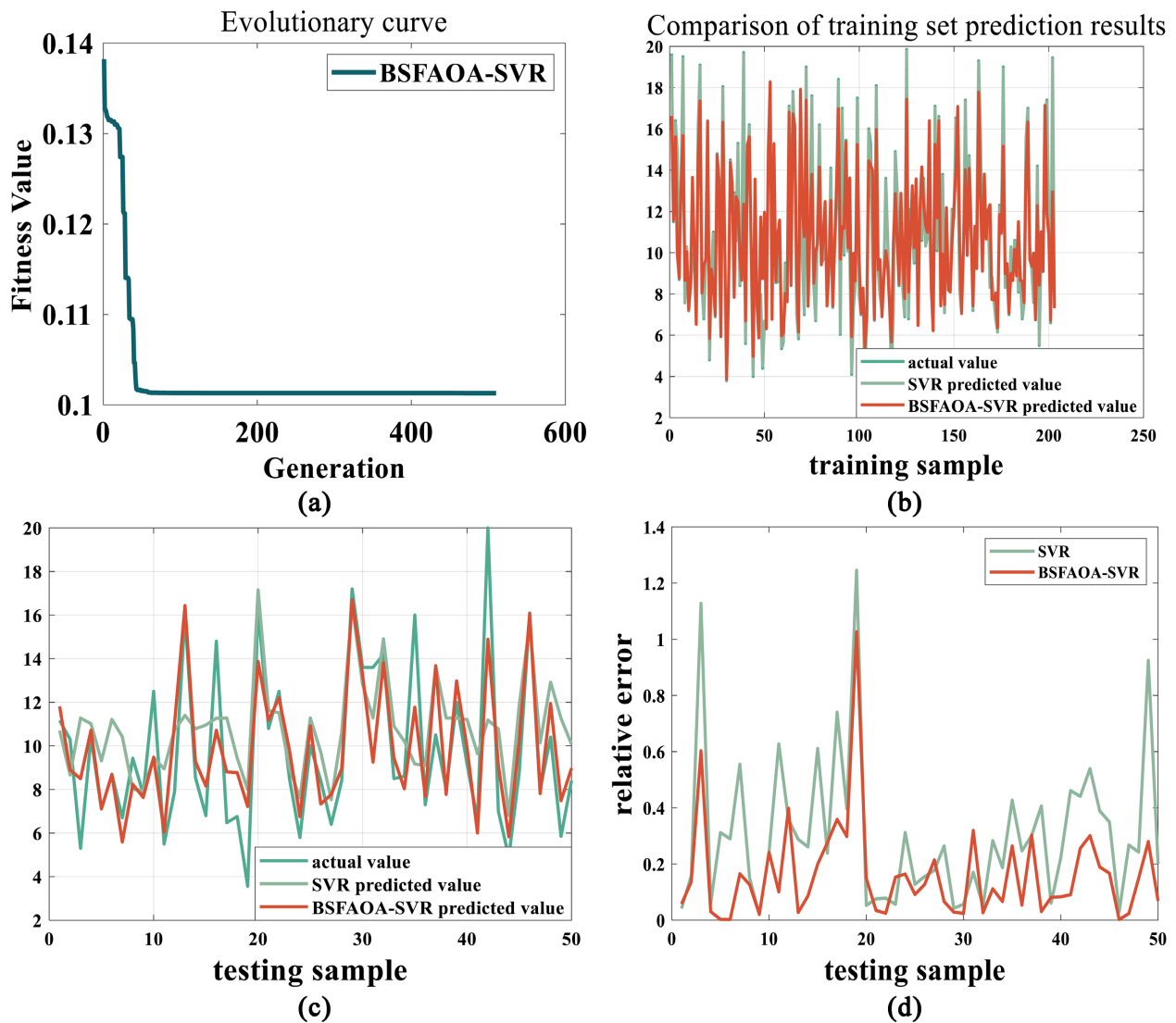


Figure 12. Operation result of the BSFAOA-SVR model. (a) Convergence process of BSFAOA-SVR; (b) Comparison of prediction results in the train set; (c) Comparison of prediction results on the test; (d) Comparison of relative errors in the test.

Table 7. Comparison of different SVR model experimental results.

Model	<i>MAE</i>	<i>RMSE</i>	<i>MAPE</i>	<i>R</i> ²
SVR	2.4911	3.0485	0.2400	0.6146
Reference [38]	-	2.54	-	0.7638
PSO-SVR	1.5344	1.9886	0.1574	0.8427
AOA-SVR	1.4153	1.9722	0.1456	0.8488
BSFAOA-SVR	1.3937	1.8940	0.1455	0.8591

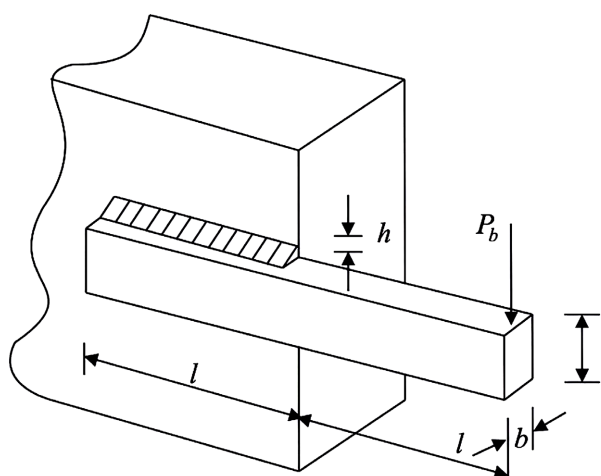
Note: In **Table 7**, “-” indicates that the data does not exist.

illustrates superior fitting effects compared to the traditional SVR model. **Figure 12(d)** presents the comparison of the relative error of the test set between the original SVR model and the BSFAOA-SVR model, revealing that the BSFAOA-SVR model demonstrates lower relative errors and more effective predictive performance.

In summary, applying the BSFAOA algorithm to optimize the key parameters of the SVR model makes up for the drawbacks of the blindness of parameter selection in the training process, enhancing the predictive accuracy of the regression model. Experimental results demonstrate that the algorithm in this article possesses commendable predictive precision and practicality.

5.2. Welded Beam Design

As exhibited in **Figure 13**, the welded beam structure consists of the beam “a” and the welded “b” connected to the member. The objective of the welded beam design problem is to determine the optimal design variables to minimize the total manufacturing cost of the welded beam.

**Figure 13.** Welding beam design problems.

The optimization constraints of this problem include shear stress (τ), bending stress in the beam (θ), buckling load on the rod (P_c), deflection at the end

of the beam (δ), and boundary conditions. To address this issue, it is necessary to explore potential combinations of design parameters for the welded beam structure, such as weld thickness (h), clamping rod length (l), rod height (t), and rod thickness (b). These parameters are represented by the vector $X = [x_1 \ x_2 \ x_3 \ x_4]$, where x_1, x_2, x_3, x_4 respectively denote h, l, t, b . The mathematical description of the problem can be defined as follows:

Variables:

$$\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4] = [h \ l \ t \ b].$$

Objective function:

$$f(\mathbf{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2).$$

Constraints conditions:

$$g_1(\mathbf{x}) = \tau(\mathbf{x}) - \tau_{\max} \leq 0, \quad g_2(\mathbf{x}) = \sigma(\mathbf{x}) - \sigma_{\max} \leq 0,$$

$$g_3(\mathbf{x}) = \delta(\mathbf{x}) - \delta_{\max} \leq 0, \quad g_4(\mathbf{x}) = x_1 - x_4 \leq 0,$$

$$g_5(\mathbf{x}) = P - P_c(\mathbf{x}) \leq 0, \quad g_6(\mathbf{x}) = 0.0125 - x_1 \leq 0,$$

$$g_7 = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0.$$

Variables range:

$$0.1 \leq x_i \leq 2, \quad i = 1, 4; \quad 0.1 \leq x_j \leq 10, \quad j = 2, 3.$$

Where:

$$\tau(\mathbf{x}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}, \quad \tau' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J}, \quad M = P\left(L + \frac{x_2}{2}\right),$$

$$R = \sqrt{\frac{x_2^2}{2} + \left(\frac{x_1 + x_3}{2}\right)^2}, \quad J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\},$$

$$\sigma(\mathbf{x}) = \frac{6PL}{x_4x_3^2}, \quad \delta(\mathbf{x}) = \frac{6PL^3}{Ex_3^2x_4}, \quad P_c(\mathbf{x}) = \frac{4.013E\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right).$$

Note that, $P = 6000$ lb, $L = 14$ in., $\delta_{\max} = 0.25$ in., $E = 30 \times 10^6$ psi, $G = 12 \times 10^6$ psi, $\tau_{\max} = 13600$ psi, $\sigma_{\max} = 30000$ psi.

To ensure the consistency of the algorithm operation, the number of populations is set to 30, the maximum number of iterations is 500, and each algorithm is run independently 30 times and averaged. The welded beam design problem was solved by BSFAOA and compared with TSO [39], GOOSE [12], BA [40], and BOA [41]. The statistical results of different algorithms for the welding beam design problem are displayed in **Table 8**, where bold values indicate the best results. **Figure 14** illustrates the convergence curves of the BSFAOA with the comparison algorithms for solving the design problem of welded beams. Further, **Table 9** details the minimum cost and the corresponding optimal solution for each algorithm, and it can be seen that BSFAOA obtains the best value of cost and the optimal solution obtained is better than the other algorithms.

5.3. Pressure Vessel Design

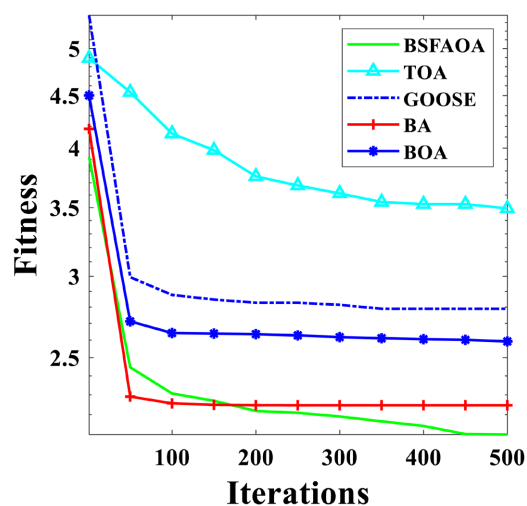
Pressure vessel design is a classic engineering challenge aimed at minimizing the

Table 8. Statistical results of welded beam design problems in different algorithms.

Algorithms	Best	Std	Avg	Median	Worst
BSFAOA	1.7694	0.1633	2.0297	1.9908	2.3356
TSO [39]	2.1073	0.84265	3.533	3.4283	5.6105
GOOSE [12]	1.8542	0.73191	2.8441	2.7566	5.3954
BA [40]	1.7293	0.34394	2.2209	2.2641	2.8199
BOA [41]	2.0468	0.25507	2.662	2.7169	3.1736

Table 9. Results of welded beam design problems.

Algorithms	x_1	x_2	x_3	x_4	Optimal Cost
BSFAOA	0.23566	3.3048	8.245	0.2479	2.0297
TSO [39]	0.80762	7.7072	0.39406	1.3005	3.5330
GOOSE [12]	0.65184	7.285	9.4765	0.5262	2.8441
BA [40]	0.1	7.6903	9.4655	0.1976	2.2209
BOA [41]	0.23188	4.0155	5.6977	0.5191	2.6620

**Figure 14.** Convergence curves of the welded beam design problem in different algorithms.

total cost of the vessel, which includes the costs of materials, forming, and welding. As illustrated in **Figure 15**, the vessel is capped at both ends, with the head end featuring a hemispherical seal. This optimization involves four decision variables: the thickness of the vessel wall (T_s), the thickness of the hemispherical head (T_h), the inner radius (R), and the length of the cylindrical section (L). The mathematical model is as follows:

Variable

$$\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4] = [T_s, T_h, R, L].$$

Objective function:

$$f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3.$$

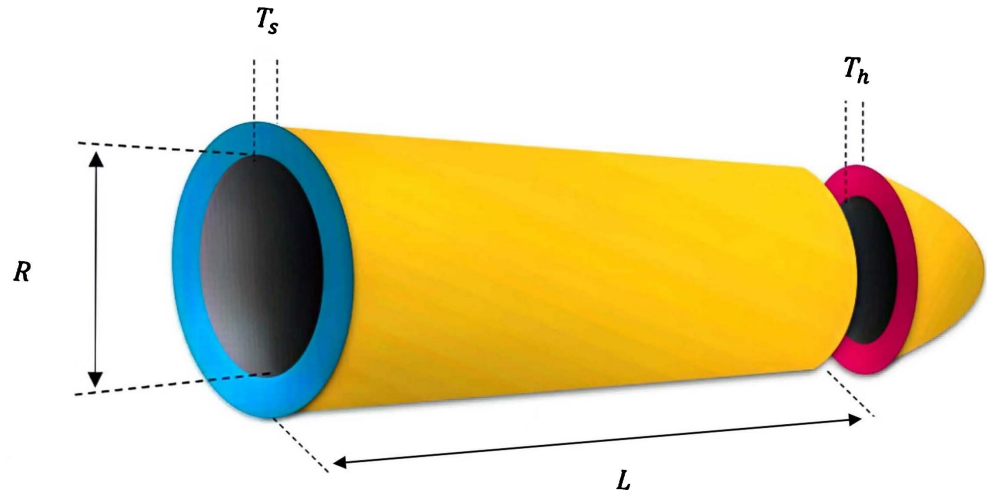


Figure 15. Pressure vessel design problems.

Constraints conditions:

$$g_1(x) = -x_1 + 0.0913x_3 \leq 0, \quad g_2(x) = -x_2 + 0.00954x_3 \leq 0,$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4\pi x_3^3}{3} + 1296000 \leq 0, \quad g_4(x) = x_4 - 240 \leq 0.$$

Variable range

$$0 \leq x_i \leq 100, i = 1, 2; 10 \leq x_i \leq 200, i = 3, 4.$$

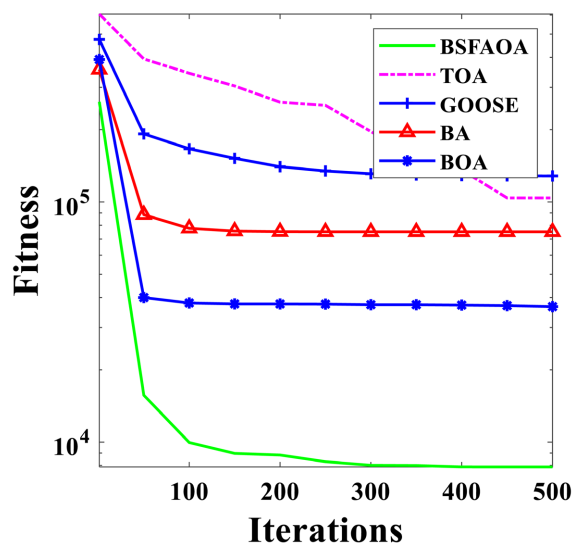
To ensure consistency in algorithm performance, a population size of 30 and a maximum of 500 iterations are set, with each algorithm running independently 30 times to derive an average value. **Table 10** compares the statistical results of the BSFAOA algorithm with those from TSO [39], GOOSE [12], BA [40] and BOA [41] in solving the pressure vessel design problem, highlighting the best values in bold. The BSFAOA algorithm demonstrates superior performance and a smaller standard deviation compared to the other algorithms, indicating its greater stability. **Figure 16** illustrates the convergence curves of the BSFAOA and other algorithms in addressing the pressure vessel design challenge. **Table 11** further presents the minimum cost and the corresponding optimal solutions for each algorithm. For the objective function, the optimal values achieved by the BSFAOA algorithm are as follows: T_s at 0.944063, T_h at 0.5344369, R at 48.69901, and L at 109.0643.

Table 10. Statistical results of pressure vessel design problems in different algorithms.

Algorithms	Best	Std	Avg	Median	Worst
BSFAOA	6149.1733	856.7829	8014.84	8158.9677	9557.4131
TSO [39]	10,284.2101	51,234.3999	77,970.082	70,005.4797	192,159.672
GOOSE [12]	7497.9613	167,073.9719	119,524.7076	37,363.322	782,745.3838
BA [40]	7436.3318	69,610.9165	84,564.1814	54,618.4261	260,142.4444
BOA [41]	7282.3882	18,561.9821	29,000.54	23,607.263	89,714.6831

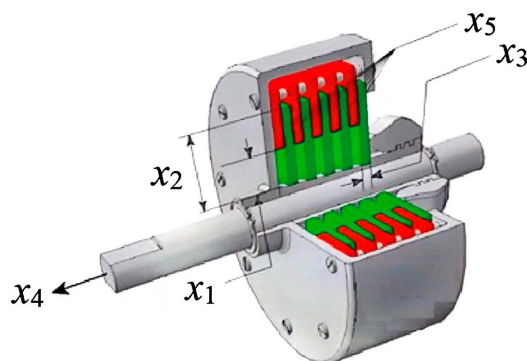
Table 11. Results of pressure vessel design problems.

Algorithms	x_1	x_2	x_3	x_4	Optimal Cost
BSFAOA	0.944063	0.5344369	48.69901	109.0643	8014.8477
TSO [39]	16.5123	16.5123	41.6902	41.6902	77970.082
GOOSE [12]	2.60942	30.6539	51.1913	96.3097	119524.7076
BA [40]	1.353537	3.234005	70.13143	177.4731	84564.1814
BOA [41]	1.37569	5.32675	58.4035	43.6665	29000.54

**Figure 16.** Convergence curves of pressure vessel design problems in different algorithms.

5.4. Multiple Disk clutch Brake Design Problems

The primary objective of this problem is to minimize the mass of a multiple-disk clutch brake. This problem involves 5 integer decision variables: inner radius (r_i), outer radius (r_o), disc thickness (t), force factor (F), and several friction surfaces (Z), with structural parameters as shown in **Figure 17**. There are 9 non-linear constraints in this problem. The definition of the multiple disk clutch brake design problem is as follows:

**Figure 17.** Multiple disk clutch brake design problems.

Variables:

$$\mathbf{x} = [x_1, x_2, x_3, x_4, x_5] = [r_i, r_o, t, F, Z].$$

Objective function:

$$f(\mathbf{x}) = \pi(x_2^2 - x_1^2)x_3(x_5 + 1)_p.$$

Constraints conditions:

$$\begin{aligned} g_1(\mathbf{x}) &= -p_{\max} + p_{rz} \leq 0, \quad g_2(\mathbf{x}) = p_{rz}v_{sr} - v_{sr,\max} p_{\max} \leq 0, \\ g_3(\mathbf{x}) &= \Delta R + x_1 - x_2 \leq 0, \quad g_4(\mathbf{x}) = -L_{\max} + (x_5 + 1)(x_3 + \delta) \leq 0, \\ g_5(\mathbf{x}) &= sM_s - M_h \leq 0, \quad g_6(\mathbf{x}) = T \geq 0, \\ g_7(\mathbf{x}) &= -v_{sr,\max} + v_{sr} \leq 0, \quad g_8(\mathbf{x}) = T - T_{\max} \leq 0. \end{aligned}$$

Variables range:

$$60 \leq x_1 \leq 80, 90 \leq x_2 \leq 110, 1 \leq x_3 \leq 3, 0 \leq x_4 \leq 1000, 2 \leq x_5 \leq 9.$$

where

$$\begin{aligned} M_h &= \frac{2}{3} \mu x_4 x_5 \frac{x_2^3 - x_1^3}{x_2^2 - x_1^2} \text{ N} \cdot \text{mm}, \quad \omega = \frac{\pi n}{30} \text{ rad/s}, \quad A = \pi(x_2^2 - x_1^2) \text{ mm}^2, \\ p_{rz} &= \frac{x_4}{A} \text{ N/mm}^2, \quad v_{sr} = \frac{\pi R_{sr} n}{30} \text{ mm/s}, \quad R_{sr} = \frac{2}{3} \frac{x_2^3 - x_1^3}{x_2^2 - x_1^2} \text{ mm}, \quad T = \frac{I_z \omega}{M_h + M_f}. \end{aligned}$$

Note that, $\Delta R = 20 \text{ mm}$, $L_{\max} = 30 \text{ mm}$, $\mu = 0.6$, $v_{sr,\max} = 10 \text{ m/s}$, $\delta = 0.5 \text{ mm}$, $s = 1.5$, $T_{\max} = 15 \text{ s}$, $n = 250 \text{ rpm}$, $T_z = 55 \text{ kg} \cdot \text{m}^2$, $M_s = 40 \text{ N} \cdot \text{m}$, $M_f = 2 \text{ N} \cdot \text{m}$, and $p_{\max} = 1$.

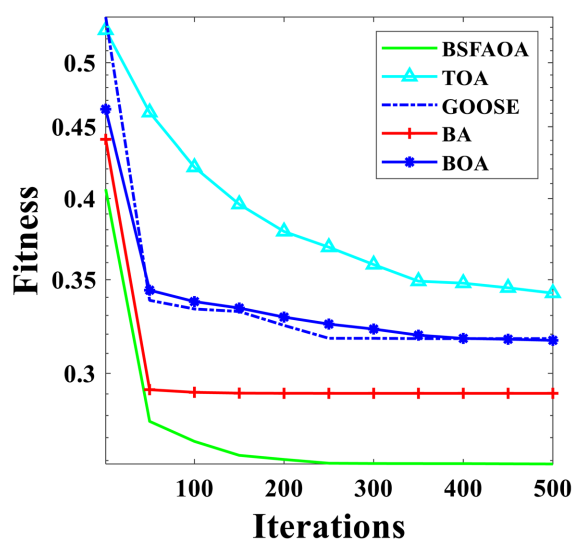
To reduce the randomness of algorithm results, the population size is set to 30, the maximum number of iterations is 500, each algorithm runs independently 30 times, and the average is taken. The proposed BSFAOA algorithm for minimizing the quality of the multiple disk clutch brake problem is compared with other optimization algorithms, including TSO [39], GOOSE [12], BA [40], and BOA [41] with the results listed in Table 12, where bold data indicates the best value. Table 13 further compares the minimum quality and design variables obtained by the BSFAOA algorithm and other optimization algorithms. Figure 18 shows the convergence curves of the BSFAOA and the comparative algorithms in solving the multiple disk clutch brake. The results indicate that this algorithm provides better solutions than other algorithms and performs well in the problem of multiple disk clutch brakes.

Table 12. Statistical results of multiple disk clutch brake design problems in different algorithms.

Algorithms	Best	Std	Avg	Median	Worst
BSFAOA	0.24177	0.011691	0.25699	0.25657	0.29061
TSO [39]	0.27066	0.037675	0.34909	0.34372	0.42138
GOOSE [12]	0.24294	0.05471	0.30759	0.29521	0.46535
BA [40]	0.23903	0.050293	0.29354	0.2823	0.42889
BOA [41]	0.25642	0.038578	0.32965	0.3248	0.41181

Table 13. Results of multiple disk clutch brake design problems.

Algorithms	x_1	x_2	x_3	x_4	x_5	Optimal Cost
BSFAOA	71.5355	91.5367	1.00572	96.7566	2.00783	0.25699
TSO [39]	77.99517	91.47936	2.802325	360.312	2.534466	0.34909
GOOSE [12]	61.78032	92.48533	1.840281	724.5309	4.5654	0.30759
BA [40]	76.25687	96.69942	1	641.4063	2	0.29354
BOA [41]	68.50653	90.02777	1.505628	585.3709	2.065188	0.32965

**Figure 18.** Convergence curves of multiple disk clutch brake design problems in different algorithms.

6. Conclusions

This article presents an enhanced arithmetic optimization algorithm (BSFAOA) with a multi-strategy fusion mechanism. It introduces the adaptive balance factor *SMOA* based on the sine function, a search strategy combining spiral search and Brownian motion, and a hybrid disturbance strategy based on the whale fall mechanism and polynomial differential learning to enhance the original AOA algorithm. The adaptive balance factor *SMOA* balances the exploration and exploitation capabilities of the algorithm, enhancing its convergence accuracy. The search strategy combining spiral search and Brownian motion improves the optimization efficiency the algorithm. The hybrid disturbance strategy of the whale fall mechanism and polynomial differential learning prevents the algorithm from getting trapped in local optima.

To comprehensively evaluate the effectiveness of BSFAOA, it was compared with other improved algorithms, existing intelligent optimization algorithms, and algorithms improved through single strategies on two sets of benchmark functions, CEC2005 and CEC2019 benchmark functions. Statistical tests were conducted on the BSFAOA. The results indicate that the BSFAOA is highly compet-

itive, showing significant improvements in both convergence speed and accuracy. To demonstrate the effectiveness of BSFAOA in finding high-quality solutions that optimize the objective function, and to further verify its applicability in practice, the problems are solved by including the optimization of SVR hyperparameters, the welded beam design problem, the pressure vessel design problem, and the multiple-disc clutch brake design problem.

Acknowledgements

This work was financially supported by the Natural Science Foundation of China (52273315), the Education Bureau of Shaanxi Province (21JT003), the Key Research and Development Program of Shaanxi Province (2024GX-YBXM-324), Tsinghua University New Ceramics and Fine Technology State Key Laboratory Open Project (KF202212) and Jingdezhen Science and Technology Project (20192GYZD008-17).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Ghasemi, M., Bagherifard, K., Parvin, H., Nejatian, S. and Pho, K. (2021) Multi-Objective Whale Optimization Algorithm and Multi-Objective Grey Wolf Optimizer for Solving Next Release Problem with Developing Fairness and Uncertainty Quality Indicators. *Applied Intelligence*, **51**, 5358-5387. <https://doi.org/10.1007/s10489-020-02018-2>
- [2] Qiao, W., Moayedi, H. and Foong, L.K. (2020) Nature-inspired Hybrid Techniques of IWO, DA, ES, GA, and ICA, Validated through a K-Fold Validation Process Predicting Monthly Natural Gas Consumption. *Energy and Buildings*, **217**, Article ID: 110023. <https://doi.org/10.1016/j.enbuild.2020.110023>
- [3] Wang, M. and Chen, H. (2020) Chaotic Multi-Swarm Whale Optimizer Boosted Support Vector Machine for Medical Diagnosis. *Applied Soft Computing*, **88**, Article ID: 105946. <https://doi.org/10.1016/j.asoc.2019.105946>
- [4] Wang, S., Xiang, J., Zhong, Y. and Zhou, Y. (2018) Convolutional Neural Network-Based Hidden Markov Models for Rolling Element Bearing Fault Identification. *Knowledge-Based Systems*, **144**, 65-76. <https://doi.org/10.1016/j.knsys.2017.12.027>
- [5] Zhang, J., Xiao, M., Gao, L. and Pan, Q. (2018) Queuing Search Algorithm: A Novel Metaheuristic Algorithm for Solving Engineering Optimization Problems. *Applied Mathematical Modelling*, **63**, 464-490. <https://doi.org/10.1016/j.apm.2018.06.036>
- [6] Kennedy, J. and Eberhart, R. (1995) Particle Swarm Optimization. *Proceedings of ICNN'95—International Conference on Neural Networks*, Perth, 27 November-1 December 1995, 1942-1948.
- [7] Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014) Grey Wolf Optimizer. *Advances in Engineering Software*, **69**, 46-61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- [8] Storn, R. and Price, K. (1997) Differential Evolution—A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, **11**, 341-359. <https://doi.org/10.1023/a:1008202821328>

- [9] Ghorbani, N. and Babaei, E. (2014) Exchange Market Algorithm. *Applied Soft Computing*, **19**, 177-187. <https://doi.org/10.1016/j.asoc.2014.02.006>
- [10] Hansen, N. and Kern, S. (2004) Evaluating the CMA Evolution Strategy on Multimodal Test Functions. *Parallel Problem Solving from Nature-PPSN*, **8**, 282-291.
- [11] Zhong, C., Li, G. and Meng, Z. (2022) Beluga Whale Optimization: A Novel Nature-Inspired Metaheuristic Algorithm. *Knowledge-Based Systems*, **251**, Article ID: 109215. <https://doi.org/10.1016/j.knsys.2022.109215>
- [12] Hamad, R.K. and Rashid, T.A. (2024) GOOSE Algorithm: A Powerful Optimization Tool for Real-World Engineering Challenges and Beyond. *Evolving Systems*. <https://doi.org/10.1007/s12530-023-09553-6>
- [13] Kaveh, A. and Khayatizad, M. (2012) A New Meta-Heuristic Method: Ray Optimization. *Computers & Structures*, **112**, 283-294. <https://doi.org/10.1016/j.compstruc.2012.09.003>
- [14] Mirjalili, S. (2016) SCA: A Sine Cosine Algorithm for Solving Optimization Problems. *Knowledge-Based Systems*, **96**, 120-133. <https://doi.org/10.1016/j.knsys.2015.12.022>
- [15] Nematollahi, A.F., Rahiminejad, A. and Vahidi, B. (2019) A Novel Meta-Heuristic Optimization Method Based on Golden Ratio in Nature. *Soft Computing*, **24**, 1117-1151. <https://doi.org/10.1007/s00500-019-03949-w>
- [16] Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M. and Gandomi, A.H. (2021) The Arithmetic Optimization Algorithm. *Computer Methods in Applied Mechanics and Engineering*, **376**, Article ID: 113609. <https://doi.org/10.1016/j.cma.2020.113609>
- [17] Sreenivasa Reddy, B. and Sathish, A. (2024) A Multiscale Atrous Convolution-Based Adaptive ResUNet3+ with Attention-Based Ensemble Convolution Networks for Brain Tumour Segmentation and Classification Using Heuristic Improvement. *Bio-medical Signal Processing and Control*, **91**, Article ID: 105900. <https://doi.org/10.1016/j.bspc.2023.105900>
- [18] Barua, S. and Merabet, A. (2024) Lévy Arithmetic Algorithm: An Enhanced Metaheuristic Algorithm and Its Application to Engineering Optimization. *Expert Systems with Applications*, **241**, Article ID: 122335. <https://doi.org/10.1016/j.eswa.2023.122335>
- [19] Das, A., Namtirtha, A. and Dutta, A. (2023) Lévy-Cauchy Arithmetic Optimization Algorithm Combined with Rough K-Means for Image Segmentation. *Applied Soft Computing*, **140**, Article ID: 110268. <https://doi.org/10.1016/j.asoc.2023.110268>
- [20] Abualigah, L., Almotairi, K.H., Al-qaness, M.A.A., Ewees, A.A., Yousri, D., Elaziz, M.A., et al. (2022) Efficient Text Document Clustering Approach Using Multi-Search Arithmetic Optimization Algorithm. *Knowledge-Based Systems*, **248**, Article ID: 108833. <https://doi.org/10.1016/j.knsys.2022.108833>
- [21] Alzaqebah, M. and Ahmed, E.A.E. (2023) Accelerated Fuzzy Min-Max Neural Network and Arithmetic Optimization Algorithm for Optimizing Hyper-Boxes and Feature Selection. *Neural Computing and Applications*, **36**, 1553-1568. <https://doi.org/10.1007/s00521-023-09131-6>
- [22] Hu, G., Zhong, J., Du, B. and Wei, G. (2022) An Enhanced Hybrid Arithmetic Optimization Algorithm for Engineering Applications. *Computer Methods in Applied Mechanics and Engineering*, **394**, Article ID: 114901. <https://doi.org/10.1016/j.cma.2022.114901>
- [23] Wolpert, D.H. and Macready, W.G. (1997) No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, **1**, 67-82.

- <https://doi.org/10.1109/4235.585893>
- [24] Wang, R., Wang, W., Xu, L., Pan, J. and Chu, S. (2021) An Adaptive Parallel Arithmetic Optimization Algorithm for Robot Path Planning. *Journal of Advanced Transportation*, **2021**, Article ID: 3606895. <https://doi.org/10.1155/2021/3606895>
- [25] Zheng, R., Jia, H., Abualigah, L., Liu, Q. and Wang, S. (2022) An Improved Arithmetic Optimization Algorithm with Forced Switching Mechanism for Global Optimization Problems. *Mathematical Biosciences and Engineering*, **19**, 473-512. <https://doi.org/10.3934/mbe.2022023>
- [26] Yıldız, B.S., Kumar, S., Panagant, N., Mehta, P., Sait, S.M., Yildiz, A.R., et al. (2023) A Novel Hybrid Arithmetic Optimization Algorithm for Solving Constrained Optimization Problems. *Knowledge-Based Systems*, **271**, Article ID: 110554. <https://doi.org/10.1016/j.knosys.2023.110554>
- [27] Gölcük, İ., Ozsoydan, F.B. and Durmaz, E.D. (2023) An Improved Arithmetic Optimization Algorithm for Training Feedforward Neural Networks under Dynamic Environments. *Knowledge-Based Systems*, **263**, Article ID: 110274. <https://doi.org/10.1016/j.knosys.2023.110274>
- [28] Mirjalili, S. and Lewis, A. (2016) The Whale Optimization Algorithm. *Advances in Engineering Software*, **95**, 51-67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- [29] Faramarzi, A., Heidarinejad, M., Mirjalili, S. and Gandomi, A.H. (2020) Marine Predators Algorithm: A Nature-Inspired Metaheuristic. *Expert Systems with Applications*, **152**, Article ID: 113377. <https://doi.org/10.1016/j.eswa.2020.113377>
- [30] Wu, D., Rao, H., Wen, C., Jia, H., Liu, Q. and Abualigah, L. (2022) Modified Sand Cat Swarm Optimization Algorithm for Solving Constrained Engineering Optimization Problems. *Mathematics*, **10**, Article No. 4350. <https://doi.org/10.3390/math10224350>
- [31] Song, W., Liu, S., Wang, X. and Wu, W. (2020). An Improved Sparrow Search Algorithm. 2020 *IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCLOUD/SocialCom/SustainCom)*, Exeter, 17-19 December 2020, 537-543. <https://doi.org/10.1109/ispa-bdcloud-socialcom-sustaincom51426.2020.00093>
- [32] Zhang, M., Yang, J., Ma, R., Du, Q. and Rodriguez, D. (2022) RETRACTED: Prediction of Small-Scale Piles by Considering Lateral Deflection Based on Elman Neural Network-Improved Arithmetic Optimizer Algorithm. *ISA Transactions*, **127**, 473-486. <https://doi.org/10.1016/j.isatra.2021.08.036>
- [33] Jia, H.M., Liu, Y.X., Liu, Q.X., Wang, S. and Zheng, R. (2022) Hybrid Algorithm of Slime Mould Algorithm and Arithmetic Optimization Algorithm Based on Random Opposition-Based Learning. *Journal of Frontiers of Computer Science and Technology*, **16**, 1182-1192.
- [34] Li, S., Chen, H., Wang, M., Heidari, A.A. and Mirjalili, S. (2020) Slime Mould Algorithm: A New Method for Stochastic Optimization. *Future Generation Computer Systems*, **111**, 300-323. <https://doi.org/10.1016/j.future.2020.03.055>
- [35] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H. (2019) Harris Hawks Optimization: Algorithm and Applications. *Future Generation Computer Systems*, **97**, 849-872. <https://doi.org/10.1016/j.future.2019.02.028>
- [36] Dhiman, G. and Kumar, V. (2019) Seagull Optimization Algorithm: Theory and Its Applications for Large-Scale Industrial Engineering Problems. *Knowledge-Based Systems*, **165**, 169-196. <https://doi.org/10.1016/j.knosys.2018.11.024>
- [37] Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H. and Mirjalili, S.M.

-
- (2017) Salp Swarm Algorithm: A Bio-Inspired Optimizer for Engineering Design Problems. *Advances in Engineering Software*, **114**, 163-191. <https://doi.org/10.1016/j.advengsoft.2017.07.002>
- [38] Qin, J., Liu, Z., Ma, M. and Li, Y. (2021) Machine Learning Approaches for Permittivity Prediction and Rational Design of Microwave Dielectric Ceramics. *Journal of Materiomics*, **7**, 1284-1293. <https://doi.org/10.1016/j.jmat.2021.02.012>
- [39] Xie, L., Han, T., Zhou, H., Zhang, Z., Han, B. and Tang, A. (2021) Tuna Swarm Optimization: A Novel Swarm-Based Metaheuristic Algorithm for Global Optimization. *Computational Intelligence and Neuroscience*, **2021**, Article ID: 9210050. <https://doi.org/10.1155/2021/9210050>
- [40] Yang, X.S. (2010) A New Metaheuristic Bat-Inspired Algorithm. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G. and Krasnogor, N., Eds., *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Springer, 65-74.
- [41] Arora, S. and Singh, S. (2018) Butterfly Optimization Algorithm: A Novel Approach for Global Optimization. *Soft Computing*, **23**, 715-734. <https://doi.org/10.1007/s00500-018-3102-4>