biomimetics

MDPI

Article

A Variable Step Crow Search Algorithm and Its Application in Function Problems

Yuqi Fan, Huimin Yang, Yaping Wang *, Zunshan Xu and Daoxiang Lu

Key Laboratory of Advanced Manufacturing and Intelligent Technology, Ministry of Education, School of Mechanical and Power Engineering, Harbin University of Science and Technology, Harbin 150080, China
* Correspondence: wypbl@163.com

Abstract: Optimization algorithms are popular to solve different problems in many fields, and are inspired by natural principles, animal living habits, plant pollinations, chemistry principles, and physic principles. Optimization algorithm performances will directly impact on solving accuracy. The Crow Search Algorithm (CSA) is a simple and efficient algorithm inspired by the natural behaviors of crows. However, the flight length of CSA is a fixed value, which makes the algorithm fall into the local optimum, severely limiting the algorithm solving ability. To solve this problem, this paper proposes a Variable Step Crow Search Algorithm (VSCSA). The proposed algorithm uses the cosine function to enhance CSA searching abilities, which greatly improves both the solution quality of the population and the convergence speed. In the update phase, the VSCSA increases population diversities and enhances the global searching ability of the basic CSA. The experiment used 14 test functions,2017 CEC functions, and engineering application problems to compare VSCSA with different algorithms. The experiment results showed that VSCSA performs better in fitness values, iteration curves, box plots, searching paths, and the Wilcoxon test results, which indicates that VSCSA has strong competitiveness and sufficient superiority. The VSCSA has outstanding performances in various test functions and the searching accuracy has been greatly improved.

Keywords: crow search algorithm; optimization algorithm; test function

Citation: Fan, Y.; Yang, H.; Wang, Y.; Xu, Z.; Lu, D. A Variable Step Crow Search Algorithm and Its Application in Function Problems. *Biomimetics* **2023**, *8*, 395. https://doi.org/10.3390/biomimetics8050395

Academic Editors: Heming Jia, Laith Abualigah and Xuewen Xia

Received: 14 July 2023
Revised: 16 August 2023
Accepted: 24 August 2023
Published: 28 August 2023

1. Introduction

The optimization is to give existing solutions and parameters to present a satisfactory answer for a certain problem. For quite some time, people have conducted large research on various optimization problems. Newton and Leibniz founded calculus which can solve some optimization problems. Then, different mathematical concepts have been proposed, such as the steepest descent method and the linear programming solution method, which can be used in many fields [1–3].

For specific problems, traditional methods have produced specific optimization methods for different problems. However, most of these methods have specific requirements for the searching space which requires objective functions to be convex, continuously differentiable, and differentiable. These weaknesses of traditional optimization methods are limited in solving many practical problems [4–7]. These practical production problems have large-scale, non-linear, multi-extreme values, characteristics of multiple constraints, and non-convexities, making traditional optimization methods difficult to conduct mathematical modeling. Therefore, exploring information processing methods with intelligent features is valuable.

In practical applications, intelligent algorithms generally do not require problem special information, the constraint on the problem, the continuity, the differentiability, the convexity of the objective function, and the analytical expression. Intelligent algorithms have strong adaptability to uncertainty data in the calculation process. At present, intelligence algorithms mainly include African Vultures Optimization Algorithm (AVOA) [8], Beluga Whale Optimization (BWO) [9], Whale Optimization Algorithm (WOA) [10], Flow Direction Algorithm (FDA) [11], Grey Wolf Optimizer (GWO) [12], Harris Hawks Optimizer (HHO) [13], Sine Cosine Algorithm (SCA) [14], Spotted Hyena Optimizer (SHO) [15], Slime Mould Algorithm (SMA) [16], Symbiotic Organisms Search (SOS) [17], Wild Horse Optimizer (WHO) [18], Geometric Mean Optimizer (GMO) [19], Golden Jackal Optimization algorithm (GJO) [20], Coati Optimization Algorithm (COA) [21], Dandelion Optimizer (DO) [22], Remora Optimization Algorithm (ROA) [23], Great Wall Construction Algorithm (GWCA) [24], Generalized Normal Distribution Optimization (GNDO) [25], Pelican Optimization Algorithm (POA) [26], and so on [27–30].

These algorithms have been achieved in various engineering fields [31–36]. For solving large-scale optimization problems, intelligent algorithms are significantly superior to traditional mathematical programming methods in terms of computational times and complexities.

Crow Search Algorithm (CSA) was proposed by Alireza Askarzadeh in 2016 [37]. Crows will hide their food and remember its hiding location for several months. At the same time, they will track other crows to steal food. Based on the living habits of crows in nature, the crow search algorithm has been proposed. From the algorithmic perspective, the overall flying area of the crow population is the searching space. The position of each crow represents the algorithm feasible solution and the location of crow hidden food represents the algorithm's objective function value. The best food position in the algorithm is the optimal solution in the searching space.

Shalini Shekhawat and Akash Saxena designed the Intelligent Crow Search Algorithm (ICSA) and used ICSA in the structural design problem, frequency wave synthesis problem, and Model Order Reduction [38]. Yilin Chen et al. introduced a robust adaptive hierarchical learning Crow Search Algorithm for feature selection [39]. Primitivo Díaz et al. introduced an improved Crow Search Algorithm Applied to Energy Problems [40]. Amrit Kaur Bhullar et al. proposed the enhanced crow search algorithm for AVR optimization [41]. Thippa Reddy Gadekallu et al. used CNN-CNS for handing gesture classification [42]. Malik Braik et al. designed a hybrid crow search algorithm for solving numerical and constrained global optimization problems [43]. Behrouz Samieiyan et al. applied Promoted Crow Search Algorithm (PCSA) to solve dimension reduction problems [44]. Qingbiao Guo et al. used an improved crow search algorithm for the parameter inversion of the probability integral method [45]. CSA has been applied in many fields.

In the basic CSA, crows update their positions by the fixed flight length in the searching space, wherein the fixed flight length will make the individual jump out of the fitness solution region, which can cause low searching accuracy. As a result, this paper proposes a variable step crow search algorithm (VSCSA). VSCSA uses Cosine function steps to update its positions. The rest of this paper is organized as follows: In Section 2, this paper gives the basic CSA. In Section 3, this paper proposes VSCSA. In Section 4, the function experiment results analysis is shown. In Section 5, the CEC2017 function experiment results analysis is shown. In Section 6, engineering application problems are shown. In Section 7, the conclusion is given.

2. Crow Search Algorithm

The crow is the general name of passerine corvus that is a large songbird which has a sturdy mouth and feet. Nostrils are circular and usually covered by feather whiskers. Crows like to live in groups and have strong clustering. They are forest and grassland birds with a steady gait. Except for a few species, they often gather in groups and nest, and wander in mixed groups during the autumn and winter seasons, flying and singing. Generally, the personality is fierce and full of aggressive habits. CSA is a metaheuristic algorithm based on crow intelligent behaviors. Crows will steal food by observing where the other birds hide their food, if a crow finds the thief, it will move to hiding places to avoid being a future victim. And crows use their own experiences to predict the pilferer's behavior. In CSA, the crow overall flight area is the searching space, and the position of each crow gives a feasible solution. The crow hidden food represents the quality of the algorithm function value.

The CSA Step is given in this section.

Step 1: Initialize the problem and adjustable parameters.

Set CSA size N, the maximum number of iterations $iter_{max}$, the flight length (*fl*), the awareness probability AP, and the searching dimension is d. The crow i at one iteration in the searching space is specified by a vector $x^{i,iter}(i = 1, \ldots, N;\ iter = 1, \ldots, iter_{max})$. The searching upper bound is $ub_i(i = 1, \ldots, N)$ and the searching lower bound is $lb_i(i = 1, \ldots, N)$,

Step 2: Initialize position and memory.

Each crow will save its hidden food location m during each iteration, which represents the best position the crow currently has because during the initial iteration of the algorithm, the crow is inexperienced. Therefore, the initial memory, which is the location where the crow first hides its food, is set as their initial position.

$$Crows = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_d^1 \\ x_1^2 & x_2^2 & \cdots & x_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^N & x_2^N & \cdots & x_d^N \end{bmatrix} \tag{1}$$

$$Memory = \begin{bmatrix} m_1^1 & m_2^1 & \cdots & m_d^1 \\ m_1^2 & m_2^2 & \cdots & m_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ m_1^N & m_2^N & \cdots & m_d^N \end{bmatrix} \tag{2}$$

Step 3: Evaluate the objective function.

Compute one crow position.

Step 4: Generate a new position.

Crow i will generate a new position. In this case, two states will happen:

State 1: Crow i will approach crow j.

State 2: Crow j will go to another position.

States 1 and 2 can be expressed as follows:

$$x^{i,iter+1} = \begin{cases} x^{i,iter} + r_i \times fl^{i,iter} \times \left(m^{j,iter} - x^{i,iter}\right) & r_{j.iter} \geq AP^{j.iter} \\ a \quad random \quad position & otherwise \end{cases} \tag{3}$$

where r_i is a random number in the range of [0, 1] and $fl^{i,iter}$ denotes the flight length of crow i at iteration iter. AP denotes the awareness probability.

Step 5: Check the feasibility of new positions.

Check the new position feasibility of each crow.

Step 6: Evaluate fitness functions of new positions.

Calculate all feasible solutions. The function value for the new position of each crow will be calculated.

Step 7: Update memory

The crows update their memory as follows:

$$m^{i,iter+1} = \begin{cases} x^{i,iter+1} & f\left(x^{i,iter+1}\right) \quad is \quad better \quad than \quad f\left(m^{i,iter}\right) \\ m^{i,iter} & otherwise \end{cases} \tag{4}$$

Compare all fitness function values. If there is a better fitness function value of the new position, the memory will be updated.

Step 8: Check termination criterion.

Calculate $iter = iter + 1$. Stop if the termination criterion is met $iter = iter_{max}$. If not, Steps 4–7 are repeated until the $iter_{max}$ is reached.

3. Variable Step Crow Search Algorithm

In the original CSA, crows constantly update their positions in the searching space, but their flight length fl is fixed, and the solutions to the search problem are diverse. When initializing a population, individuals often cannot directly locate the optimal solution and approach the region where the optimal solution is located without prior exploration experience in the searching space as a guide. Therefore, the searching process should be carried out in multiple different directions to expand the searching scope and thereby increase the probability of approaching the area. In addition, individuals in the population often wish to visit unexplored areas when exploring the searching space, thereby increasing the breadth of the search. And from the CSA position update formula, it can be seen that the crow population mainly updates its position by moving towards a fixed flight length. Therefore, as the species iteration continues, the crow population will gradually cluster and the population diversity will gradually decrease, which can easily lead to a single searching direction and form too many local optima which is not conducive to the algorithm's small-scale search in the later stage. To solve this problem, this article proposes a variable step crow search algorithm (VSCSA).

The cosine function is a Periodic function with a minimum positive period of 2π. When the independent variable is an integer 2kπ (k is an integer), the function has a maximum value of 1. When the independent variable is (2k + 1) π (k is an integer), the function has a minimum value of −1. The cosine function is an even function, and its image is symmetric about the *y*-axis.

$$x_{new}^{i,iter+1} = x_{new}^{i,iter} + |\cos(r_i)| \times \left(m^{j,iter} - x_{new}^{i,iter}\right) \quad r_{j.iter} \geq AP^{j.iter} \tag{5}$$

When crow *j* knows that crow *i* is following it, then as a result, crow *j* will go to another position by the searching upper bound.

$$x_{new}^{i,iter+1} = 0.5 \times \left(ub_i \times rs_i + m^{j,iter}\right) \tag{6}$$

New states 1 and 2 can be expressed as follows:

$$x_{new}^{i,iter+1} = \begin{cases} x_{new}^{i,iter} + |\cos(r_i)| \times \left(m^{j,iter} - x_{new}^{i,iter}\right) & r_{j.iter} \geq AP^{j.iter} \\ 0.5 \times \left(ub_i \times rs_i + m^{j,iter}\right) & otherwise \end{cases} \tag{7}$$

where rs_i is a random number in the range of [−1, 1] and $ub_i (i = 1, \ldots, N)$ is the searching upper bound.

VSCSA improves the population diversity and the changing pattern search guidance method during the evolution process. In the early searching stage, the population diversity is relatively high. Cosine steps serve as a guide for population evolution, which can avoid blind individual searching and population diversity rapid decay. This meets the requirement that the algorithm should conduct a large-scale exploration as much as possible during the initial iteration. In the later searching stages, the proposed algorithm will shift from a global exploration to a local development, which can avoid the divergence in search directions. When the population falls into the local optima, the proposed algorithm can use individuals generated by cosine steps as searching guides to effectively increase population diversities and jump out of different local optima areas. Therefore, the proposed strategy reflects the adaptive interaction between population diversities and multiple search guided individuals, the changes in the population searching steps reflect different stages of evolution, and different searching guided methods can be adaptively selected. In turn, different guided methods can alter the diversity of the population, expand the algorithm searching range, and strengthen the algorithm searching precision.

The VSCSA Flowchart can be presented in Figure 1 as follows:

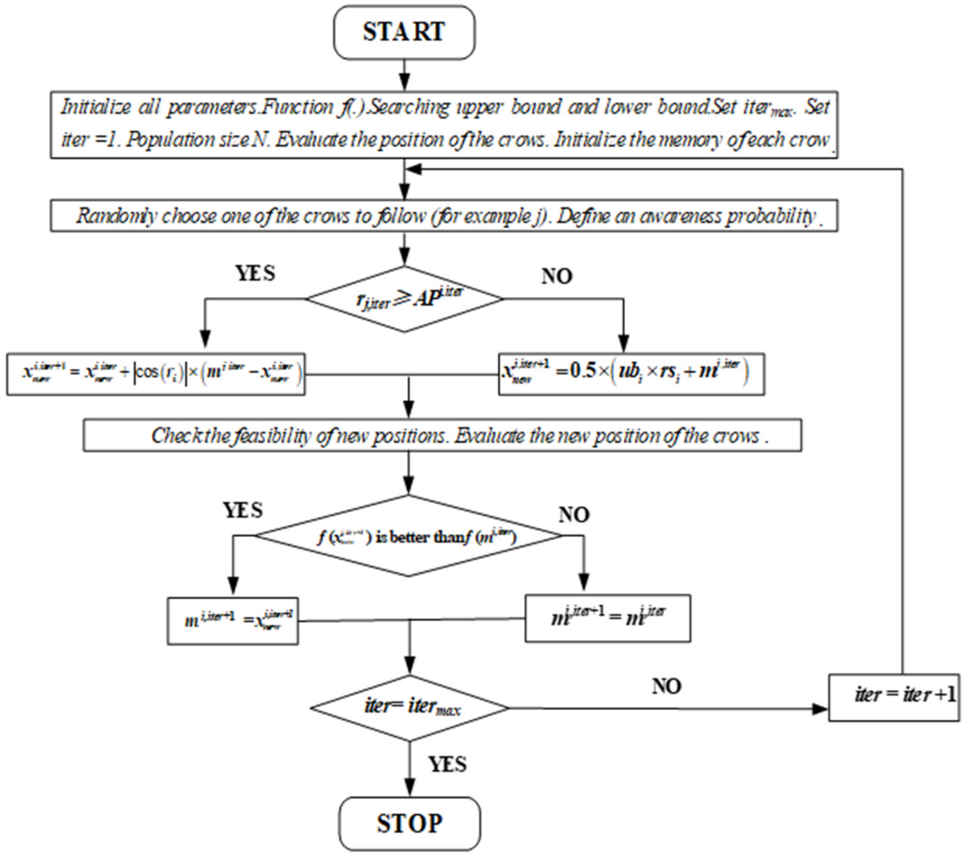

Figure 1. The VSCSA Flowchart.

The VSCSA pseudo code can be summarized in Algorithm 1.

Algorithm 1: VSCSA.

Input: *Function f(.). Searching upper bound and lower bound. Set* $iter_{max}$. *Set iter = 1. Population size N. Evaluate the position of the crows. Initialize the memory of each crow.*
While ($iter < iter_{max}$**)**
For *i = 1:N*
Randomly choose one of the crows to follow (for example j).
Define an awareness probability.
If 1 $r_{j,iter} \geq AP^{i.iter}$
$x_{new}^{i,iter+1} = x_{new}^{i,iter} + |\cos(r_i)| \times (m^{j,iter} - x_{new}^{i,iter})$
Else
$x_{new}^{i,iter+1} = 0.5 \times (ub_i \times rs_i + m^{j,iter})$
End If 1
End For
Check the feasibility of new positions.
Evaluate the new position of the crows.
Update the memory of crows:
If 2 $f\,(x_{new}^{i,iter+1})$ is better than $f\,(m^{i,iter})$.
$m^{i,iter+1} = x_{new}^{i,iter+1}$
Else
$m^{i,iter+1} = m^{i,iter}$
End If 2
iter = iter + 1
End While

4. Function Experiment Results

4.1. Experiment Environments

Different functions are in Table 1. In Table 1, D is the searching dimension and $f_{\min}$ is the idea function value. Range is the searching scope. Different optimal solutions of high-dimensional testing functions in this paper are hidden in a smooth and narrow parabolic valley, with broad searching space, tall obstacles, and a large number of local minimum points. This paper uses different test functions for comparing VSCSA and standard CSA performances. This paper tests VSCSA with the cuckoo search algorithm (CS) [46], the sine cosine algorithm (SCA) [14], and the moth-flame optimization algorithm (MFO) [47]. CS was proposed by Xin-She Yang and was inspired by the cuckoo incubation mechanism in nature. The size of the cuckoo bird is similar to that of a pigeon, but it is slender and has a dark gray upper body. SCA, which was proposed by Seyedali Mirjalili in 2016, was inspired by sine and cosine mathematical terminology. MFO, which was proposed by Seyedali Mirjalili in 2015, was inspired by the moth navigation in nature called the transverse orientation. In this chapter, the CS discovery probability was set as 0.25 and the step was set as 0.25. For SCA, a = 2, r_2 = 2π, r_3, and r_4 were selected in [0, 1]. In CSA, *fl* = 2. All algorithm parameters were selected from the original algorithm literature. The population size was 20, the maximum iterations were 400, and it was ran 10 times in MATLAB (R2016b).

Table 1. Basic information of benchmark functions.

Name	Function	D	Range	$f_{\min}$
Beale	$f_1(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	2	[−50, 50]	0
Bohachevsky01	$f_2(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$	2	[−50, 50]	0
Bohachevsky02	$f_3(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1)\cos(4\pi x_2) + 0.3$	2	[−50, 50]	0
Bohachevsky03	$f_4(x) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1 + 4\pi x_2) + 0.3$	2	[−50, 50]	0
Booth	$f_5(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	2	[−50, 50]	0
Brent	$f_6(x) = (x_1 + 10)^2 + (x_2 + 10)^2 + e^{-x_1^2 - x_2^2}$	2	[−50, 50]	0
Cube	$f_7(x) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2$	2	[−50, 50]	0
Leon	$f_8(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$	2	[−50, 50]	0
Levy13	$f_9(x) = \sin^2(3\pi x_1) + (x_1 - 1)^2[1 + \sin^2(3\pi x_2)] + (x_2 - 1)^2[1 + \sin^2(2\pi x_2)]$	2	[−50, 50]	0
Matyas	$f_{10}(x) = 0.26(x_1^2 + x_2^2) - 0.48\,x_1x_2$	2	[−50, 50]	0
Ackley 01	$f_{11}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + \exp(1)$	2/30/60/200	[−20, 20]	0
Griewank	$f_{12}(x) = \sum_{i=1}^{D}\left(x_i^2/4000\right) - \prod_{i=1}^{D}\cos\left(x_i/\sqrt{i}\right) + 1$	2/30/60/200	[−20, 20]	0
Rastrigin	$f_{13}(x) = 10D + \sum_{i=1}^{D}\left(x_i^2 - 10\cos(2\pi x_i)\right)$	2/30/60/200	[−20, 20]	0
Sphere	$f_{14}(x) = \sum_{i=1}^{D} x_i^2$	2/30/60/200	[−20, 20]	0

4.2. Data Results

In Tables 2 and 3, Min, Max, Ave, and Var mean the minimum value, the maximum value, the average value, and the variance deviation. Table 2 shows two-dimension function results. Table 3 shows high-dimension functions results. For two-dimension functions, VSCSA can obtain the ideal function values in f_2 to f_5, $f_{12(D=2)}$, and $f_{13(D=2)}$. And VSCSA can obtain the ideal values of all evaluation indexes in f_2 to f_4. CSA can obtain ideal function values in $f_{12(D=2)}$, $f_{13(D=2)}$. MFO can obtain the ideal function values in f_2 to f_5, $f_{12(D=2)}$, and $f_{13(D=2)}$. MFO can obtain the ideal values of all evaluation indexes in f_2, f_3, f_5. SCA can obtain the ideal function values in f_2 to f_4, $f_{12(D=2)}$, and $f_{13(D=2)}$. SCA can obtain the ideal values of all evaluation indexes in f_2 to f_4. Min values of MFO in f_{10} and $f_{14(D=2)}$ are better than those of VSCSA. Min value of SCA in $f_{14(D=2)}$ is better than that of

VSCSA. For high dimension functions, the Min values of SCA in $f_{11(D=30)}$, $f_{12(D=60)}$, $f_{13(D=60)}$, $f_{13(D=200)}$ are better than those of VSCSA. Min value of MFO in $f_{12(D=30)}$ is better than that of VSCSA. VSCSA in other results are all less than comparative algorithms. VSCSA can ensure continuous evolution and has good convergence speed and optimization accuracy. Especially for multi-peak high dimension functions with rotational characteristics, the proposed algorithm can better overcome the interference caused by local extreme points in the solving process, can prevent premature convergence, ensure continuous population evolution, and ultimately achieve a high optimization accuracy.

Table 2. Comparison of results for two-dimension functions.

Function	Metric	VSCSA	CSA	CS	MFO	SCA
f_1	Min	1.2634×10^{-31}	1.1569×10^{-16}	0.0436	8.2737×10^{-22}	0.0001
	Max	6.4206×10^{-16}	8.2586×10^{-15}	1.8208	0.0358	0.0035
	Ave	7.9525×10^{-17}	2.6914×10^{-15}	0.4393	0.0036	0.0009
	Var	4.1382×10^{-32}	7.2893×10^{-30}	0.2594	0.0001	1.2328×10^{-6}
f_2	Min	0	2.2204×10^{-16}	0.5852	0	0
	Max	0	2.6645×10^{-14}	3.4157	0	0
	Ave	0	7.3053×10^{-15}	1.7169	0	0
	Var	0	7.5842×10^{-29}	1.2711	0	0
f_3	Min	0	1.6653×10^{-16}	0.0910	0	0
	Max	0	3.9351×10^{-12}	2.0767	0	0
	Ave	0	5.8736×10^{-13}	1.1458	0	0
	Var	0	1.6012×10^{-24}	0.4273	0	0
f_4	Min	0	5.5511×10^{-17}	0.4008	0	0
	Max	0	6.8778×10^{-14}	3.9791	3.3307×10^{-16}	0
	Ave	0	1.5071×10^{-14}	1.5279	6.6613×10^{-17}	0
	Var	0	4.8923×10^{-28}	1.4034	1.3559×10^{-32}	0
f_5	Min	0	2.2380×10^{-17}	0.1529	0	8.5105×10^{-5}
	Max	1.4374×10^{-15}	9.8969×10^{-15}	5.2062	0	0.0072
	Ave	1.4374×10^{-16}	1.3184×10^{-15}	2.3982	0	0.0015
	Var	2.0660×10^{-31}	9.2626×10^{-30}	2.5784	0	4.1821×10^{-6}
f_6	Min	1.3839×10^{-87}	1.2181×10^{-17}	0.2978	1.3839×10^{-87}	0.0001
	Max	1.2738×10^{-21}	1.3241×10^{-15}	2.2916	1.3839×10^{-87}	0.0554
	Ave	1.3301×10^{-22}	4.7342×10^{-16}	0.9435	1.3839×10^{-87}	0.0258
	Var	1.6096×10^{-43}	2.2174×10^{-31}	0.4555	5.5373×10^{-206}	0.0005
f_7	Min	1.9671×10^{-17}	6.8577×10^{-15}	0.2525	0.0002	0.0002
	Max	0.0005	1.8490×10^{-11}	15.7426	7.1992	0.0056
	Ave	9.1750×10^{-5}	2.2728×10^{-12}	5.7119	1.5894	0.0025
	Var	2.3620×10^{-8}	3.2677×10^{-23}	36.2206	7.2811	3.1310×10^{-6}
f_8	Min	3.2519×10^{-27}	7.0257×10^{-15}	0.8962	0.0064	0.0001
	Max	5.7350×10^{-6}	2.4108×10^{-12}	26.2582	39.3529	0.0368
	Ave	9.2305×10^{-7}	5.0263×10^{-13}	7.5532	5.2897	0.0104
	Var	4.0001×10^{-12}	6.7474×10^{-25}	52.2655	1.5027×10^{2}	0.0002
f_9	Min	1.3498×10^{-31}	2.5846×10^{-16}	0.2671	1.3498×10^{-31}	0.0002
	Max	1.9689×10^{-14}	7.3082×10^{-14}	4.1046	1.3498×10^{-31}	0.0099
	Ave	2.9178×10^{-15}	1.6250×10^{-14}	1.8569	1.3498×10^{-31}	0.0033
	Var	4.3618×10^{-29}	5.1105×10^{-28}	0.9865	0	1.0867×10^{-5}
f_{10}	Min	1.7336×10^{-38}	2.2204×10^{-16}	0.0014	4.8795×10^{-50}	5.0354×10^{-54}
	Max	2.4876×10^{-29}	2.0117×10^{-13}	0.2567	1.6616×10^{-10}	6.4181×10^{-41}
	Ave	2.6360×10^{-30}	2.5902×10^{-14}	0.0544	1.6789×10^{-11}	7.4082×10^{-42}
	Var	6.1178×10^{-59}	3.8412×10^{-27}	0.0057	2.7550×10^{-21}	4.0702×10^{-82}

Table 2. *Cont.*

Function	Metric	VSCSA	CSA	CS	MFO	SCA
$f_{11(D=2)}$	Min	8.8818×10^{-16}	5.5532×10^{-9}	0.4659	8.8818×10^{-16}	8.8818×10^{-16}
	Max	4.4409×10^{-15}	5.5989×10^{-8}	2.7931	2.5799	8.8818×10^{-16}
	Ave	1.2434×10^{-15}	1.8217×10^{-8}	1.7149	0.2580	8.8818×10^{-16}
	Var	1.2622×10^{-30}	3.0345×10^{-16}	0.5155	0.6656	0
$f_{12(D=2)}$	Min	0	0	0.0089	0	0
	Max	0.0074	0.0099	0.0150	0.0395	0.0085
	Ave	0.0015	0.0045	0.0115	0.0145	0.0016
	Var	9.7247×10^{-6}	1.6088×10^{-5}	4.6345×10^{-6}	1.8438×10^{-4}	1.1916×10^{-5}
$f_{13(D=2)}$	Min	0	0	2.5027	0	0
	Max	0.9950	0.9950	5.5228	1.9899	0
	Ave	0.4869	0.0995	4.4264	0.3980	0
	Var	0.2644	0.0990	1.1062	0.4840	0
$f_{14(D=2)}$	Min	9.4793×10^{-39}	3.1793×10^{-18}	0.0239	3.2958×10^{-78}	1.0019×10^{-57}
	Max	8.9804×10^{-31}	1.4926×10^{-16}	0.7788	1.8724×10^{-19}	1.0246×10^{-40}
	Ave	1.0017×10^{-31}	5.2384×10^{-17}	0.1987	1.8724×10^{-20}	1.0247×10^{-41}
	Var	7.9344×10^{-62}	3.3914×10^{-33}	0.0521	3.5060×10^{-39}	1.0497×10^{-81}

Table 3. Comparison of results for high dimension functions.

Function	Metric	VSCSA	CSA	CS	MFO	SCA
$f_{11(D=30)}$	Min	2.2797	3.2397	16.3819	10.0741	0.6109
	Max	5.1136	6.0112	17.7649	16.5201	7.2651
	Ave	3.7778	4.6759	17.1658	13.7082	2.8626
	Var	0.6331	0.9435	0.2970	4.7736	3.6873
$f_{12(D=30)}$	Min	0.0528	0.1364	1.2198	0.0300	0.0139
	Max	0.3587	0.3208	1.4571	0.5775	0.7466
	Ave	0.1378	0.2198	1.3653	0.2246	0.3989
	Var	0.0070	0.0056	0.0058	0.0388	0.0677
$f_{13(D=30)}$	Min	1.1286×10^{2}	1.3475×10^{2}	1.4140×10^{3}	1.5609×10^{2}	1.1746×10^{2}
	Max	2.5911×10^{2}	2.0885×10^{2}	2.0780×10^{3}	9.2546×10^{2}	2.5009×10^{2}
	Ave	1.7758×10^{2}	1.6834×10^{2}	1.7499×10^{3}	3.2263×10^{2}	1.8308×10^{2}
	Var	1.7198×10^{3}	7.5248×10^{2}	4.3256×10^{4}	6.8344×10^{4}	2.4061×10^{3}
$f_{14(D=30)}$	Min	0.1507	0.8730	1.1071×10^{3}	2.5817	0.8212
	Max	0.4330	2.8546	1.8329×10^{3}	8.0147×10^{2}	59.5845
	Ave	0.2653	1.8225	1.5994×10^{3}	3.4198×10^{2}	16.5211
	Var	0.0075	0.4599	5.5006×10^{4}	9.3346×10^{4}	2.8519×10^{2}
$f_{11(D=60)}$	Min	4.1431	4.1958	16.7884	14.5856	5.0868
	Max	6.6470	5.7192	18.2277	18.1983	10.4431
	Ave	5.1825	4.8548	17.6613	16.7067	7.8758
	Var	0.6236	0.4106	0.2953	1.7958	3.1980
$f_{12(D=60)}$	Min	0.2101	0.3262	1.7140	1.0847	0.0276
	Max	0.3411	0.6327	2.1414	1.4313	1.1034
	Ave	0.2524	0.4977	1.8979	1.2450	0.8115
	Var	0.0021	0.0087	0.0125	0.0143	0.1153
$f_{13(D=60)}$	Min	3.3309×10^{2}	3.7876×10^{2}	3.6370×10^{3}	8.6975×10^{2}	2.1715×10^{2}
	Max	6.2218×10^{2}	6.1118×10^{2}	5.3243×10^{3}	3.9577×10^{3}	7.7275×10^{2}
	Ave	5.0434×10^{2}	4.7404×10^{2}	4.3379×10^{3}	2.0584×10^{3}	4.8009×10^{2}
	Var	9.9127×10^{3}	6.2803×10^{3}	4.1675×10^{5}	1.1259×10^{6}	3.6050×10^{4}

Table 3. *Cont.*

Function	Metric	VSCSA	CSA	CS	MFO	SCA
$f_{14(D=60)}$	Min	2.7331	17.9029	2.6427×10^3	2.9806×10^2	1.2890×10^2
	Max	5.5279	27.1171	4.5444×10^3	1.5615×10^3	4.8950×10^2
	Ave	4.0482	21.1575	3.7431×10^3	9.2456×10^2	2.4631×10^2
	Var	0.8931	8.1825	3.7021×10^5	1.8256×10^5	1.6048×10^4
$f_{11(D=200)}$	Min	4.9721	5.4329	16.4399	18.5124	8.4967
	Max	6.4728	6.2622	18.2605	18.9722	11.7968
	Ave	5.5735	5.7298	17.6394	18.7693	9.9552
	Var	0.2370	0.0481	0.2890	0.0161	1.2549
$f_{12(D=200)}$	Min	0.5730	0.8061	3.6256	3.9739	1.1261
	Max	0.6674	0.9795	5.1979	4.8149	2.0141
	Ave	0.6197	0.9020	4.4142	4.2953	1.5740
	Var	0.0013	0.0035	0.1957	0.0626	0.0719
$f_{13(D=200)}$	Min	1.8775×10^3	1.9453×10^3	1.5111×10^4	1.5274×10^4	1.7142×10^3
	Max	2.5561×10^3	2.3118×10^3	1.8056×10^4	1.7675×10^4	4.6564×10^3
	Ave	2.2061×10^3	2.1896×10^3	1.6534×10^4	1.6121×10^4	3.2054×10^3
	Var	3.4431×10^4	1.1518×10^4	1.1715×10^6	9.4756×10^5	1.0841×10^6
$f_{14(D=200)}$	Min	42.8433	1.5256×10^2	1.0562×10^4	1.2646×10^4	1.3509×10^3
	Max	62.5081	2.0567×10^2	1.6804×10^4	1.5405×10^4	5.0878×10^3
	Ave	52.1370	1.8448×10^2	1.3558×10^4	1.4005×10^4	3.1410×10^3
	Var	33.5172	3.3378×10^2	3.1386×10^6	6.1661×10^5	1.8390×10^6

4.3. Iteration Results

This paper gave algorithm optimal iteration curves after 10 independent operations, as shown in Figures 2 and 3. Compared with different algorithms in two-dimension iteration curves, VSCSA has the fastest iteration curve except for f_{10}, $f_{13(D=2)}$, $f_{14(D=2)}$. In f_{10}, SCA has the fastest iteration curve. In $f_{13(D=2)}$, $f_{14(D=2)}$, MFO has the fastest iteration curve, and CSA has the second fast iteration curve. Compared with different algorithms in high-dimension iteration curves, VSCSA has the fastest iteration curve except $f_{11(D=30)}$, $f_{12(D=30)}$, $f_{12(D=60)}$, $f_{13(D=60)}$. In $f_{11(D=30)}$, $f_{12(D=30)}$, $f_{12(D=60)}$, $f_{13(D=60)}$, SCA has the fastest iteration curve. The VSCSA has outstanding performances in various test functions, whereby especially the searching accuracy has been greatly improved. Therefore, the iteration curve can display that VSCSA has a strong searching performance.

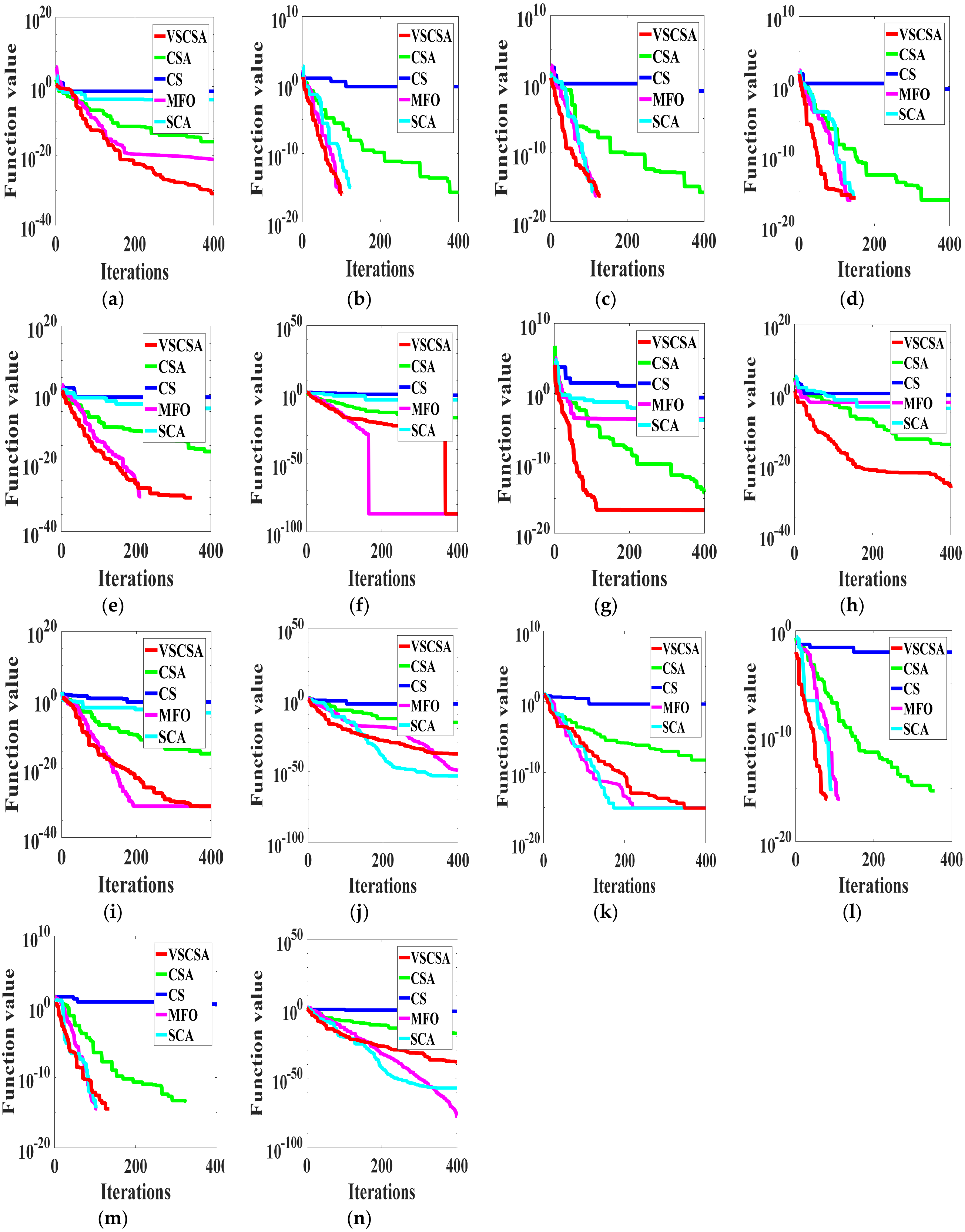

Figure 2. Iteration curves of two−dimension functions. **(a)** f_1; **(b)** f_2; **(c)** f_3; **(d)** f_4; **(e)** f_5; **(f)** f_6; **(g)** f_7; **(h)** f_8; **(i)** f_9; **(j)** f_{10}; **(k)** $f_{11(D=2)}$; **(l)** $f_{12(D=2)}$; **(m)** $f_{13(D=2)}$; **(n)** $f_{14(D=2)}$.

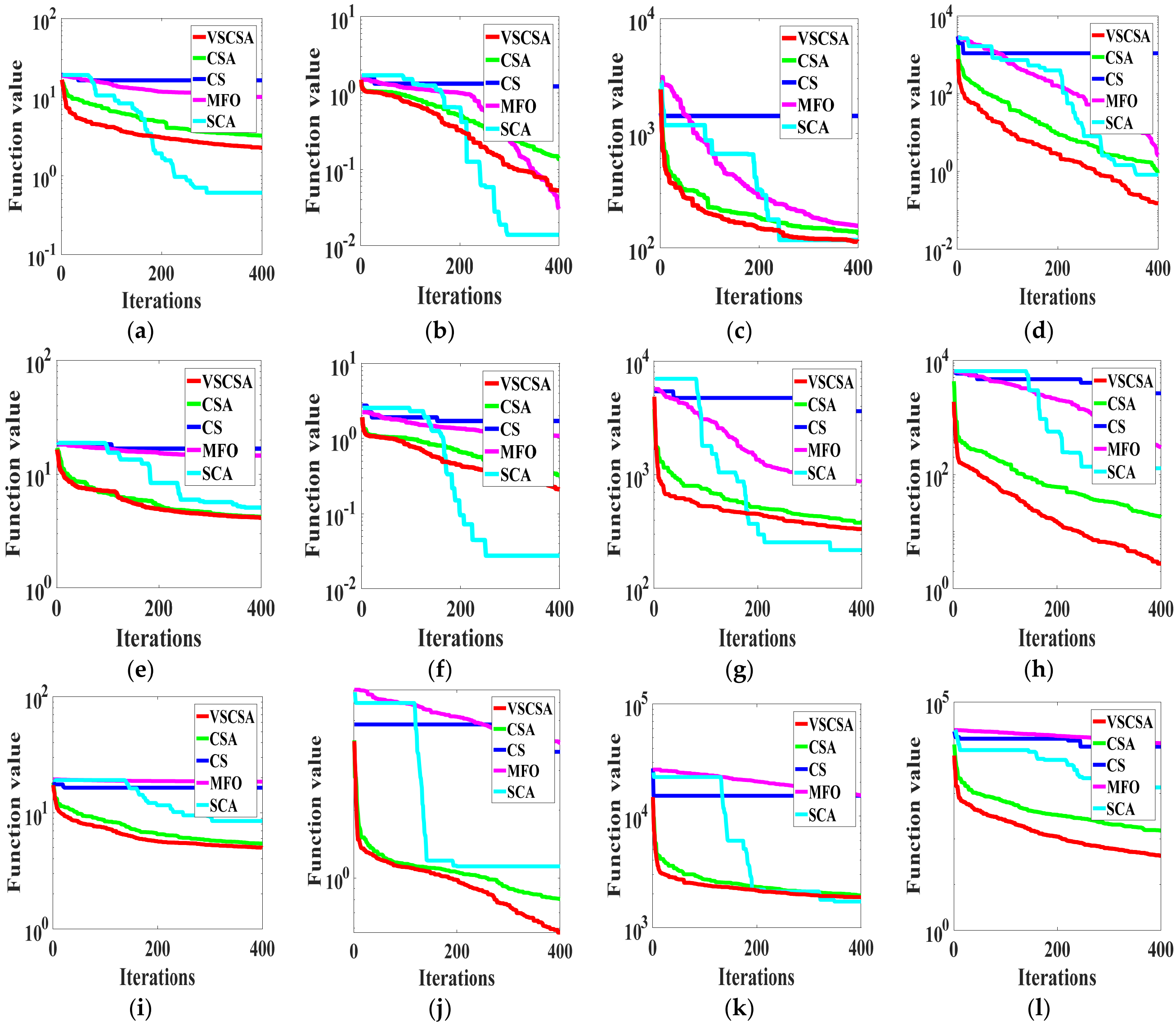

Figure 3. Iteration curves of high−dimension functions. (**a**) $f_{11(D=30)}$; (**b**) $f_{12(D=30)}$; (**c**) $f_{13(D=30)}$; (**d**) $f_{14(D=30)}$; (**e**) $f_{11(D=60)}$; (**f**) $f_{12(D=60)}$; (**g**) $f_{13(D=60)}$; (**h**) $f_{14(D=60)}$; (**i**) $f_{11(D=200)}$; (**j**) $f_{12(D=200)}$; (**k**) $f_{13(D=200)}$; (**l**) $f_{14(D=200)}$.

4.4. Box Plot Results

The box plot connects the two quartiles and connects the upper and lower edges to draw the box plot, and the median is in the middle of the box plot. If the box plot is narrower, the data is more concentrated. This paper gave algorithm box plots, as shown in Figures 4 and 5. Compared with different algorithms in low-dimension box plots, VSCSA has the narrowest box plot except for f_8, f_{13}. In f_8 and f_{13}, CSA has the narrowest box plot. Compared with different algorithms in high-dimension box plots, VSCSA has the narrowest box plot except for $f_{11(D=60)}$, $f_{13(D=60)}$, $f_{11(D=200)}$, $f_{13(D=200)}$. In $f_{11(D=60)}$, $f_{13(D=60)}$, $f_{11(D=200)}$, $f_{13(D=200)}$, CSA has the narrowest box plot. Compared to the standard CSA algorithm, the VSCSA algorithm not only has a higher solving accuracy but also runs faster in most testing functions, which fully demonstrates that the VSCSA retains outstanding local search ability and is a significant improvement in global searching performances.

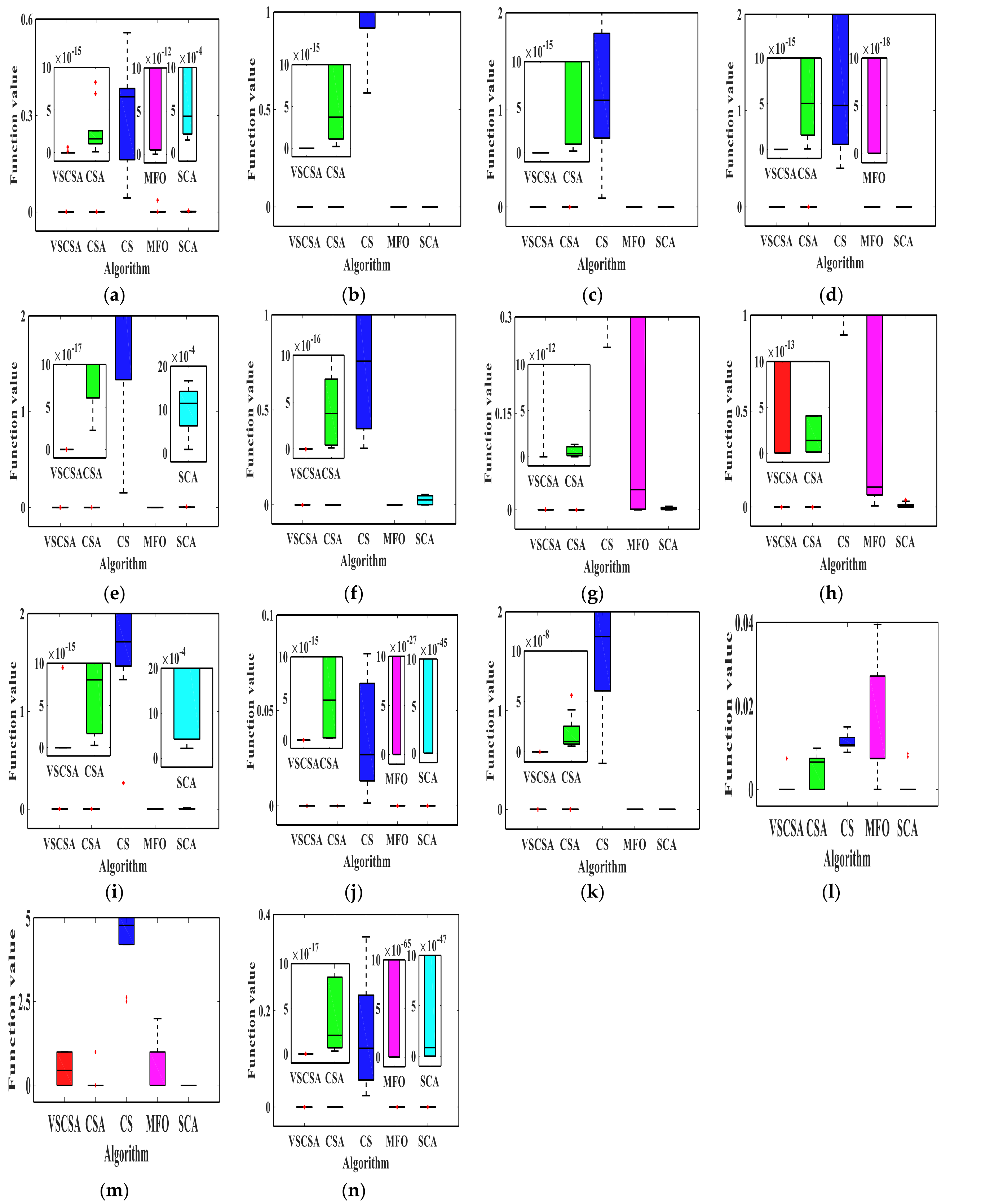

Figure 4. Box plot charts of two−dimension functions. (**a**) f_1; (**b**) f_2; (**c**) f_3; (**d**) f_4; (**e**) f_5; (**f**) f_6; (**g**) f_7; (**h**) f_8; (**i**) f_9; (**j**) f_{10}; (**k**) $f_{11(D=2)}$; (**l**) $f_{12(D=2)}$; (**m**) $f_{13(D=2)}$; (**n**) $f_{14(D=2)}$.

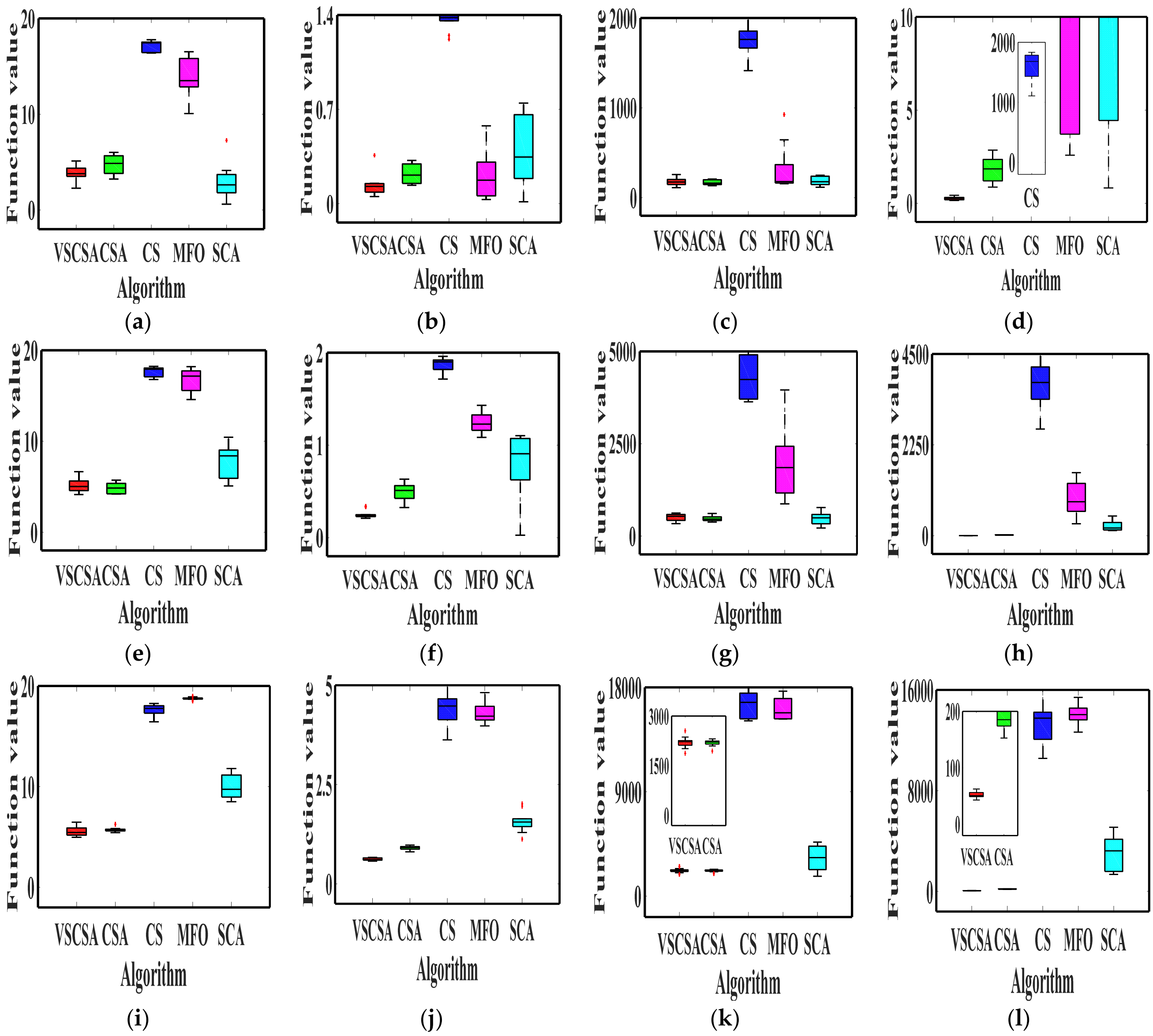

Figure 5. Box plot charts of high−dimension functions. (**a**) $f_{11(D=30)}$; (**b**) $f_{12(D=30)}$; (**c**) $f_{13(D=30)}$; (**d**) $f_{14(D=30)}$; (**e**) $f_{11(D=60)}$; (**f**) $f_{12(D=60)}$; (**g**) $f_{13(D=60)}$; (**h**) $f_{14(D=60)}$; (**i**) $f_{11(D=200)}$; (**j**) $f_{12(D=200)}$; (**k**) $f_{13(D=200)}$; (**l**) $f_{14(D=200)}$.

4.5. Sub-Sequence Runs Results

Different axes are projected at equal angular intervals from the same center point, each axis represents a quantitative variable, and points on each axis are sequentially connected into lines or geometric shapes. Each variable has its axis, with equal distances between them, and all axes have the same scale. It is equivalent to a parallel coordinate map, which is arranged radially along the axis. This paper shows the basic statistical assessment obtained in sub-sequence runs of different algorithms. Ten sub-sequence runs are shown in Figures 6 and 7. If the total length of polygon edges with different colors is longer, the lower the accuracy of the algorithm subsequence operations. For two-dimension amplification radar charts, CS subsequences have the largest radar charts except for $f_{12(D=2)}$. MFO radar charts are larger than CSA radar charts in $f_{12(D=2)}$. For high-dimension amplification radar charts, CS subsequences have the largest radar charts except for $f_{11(D=60)}$, $f_{11(D=200)}$ to $f_{14(D=200)}$.

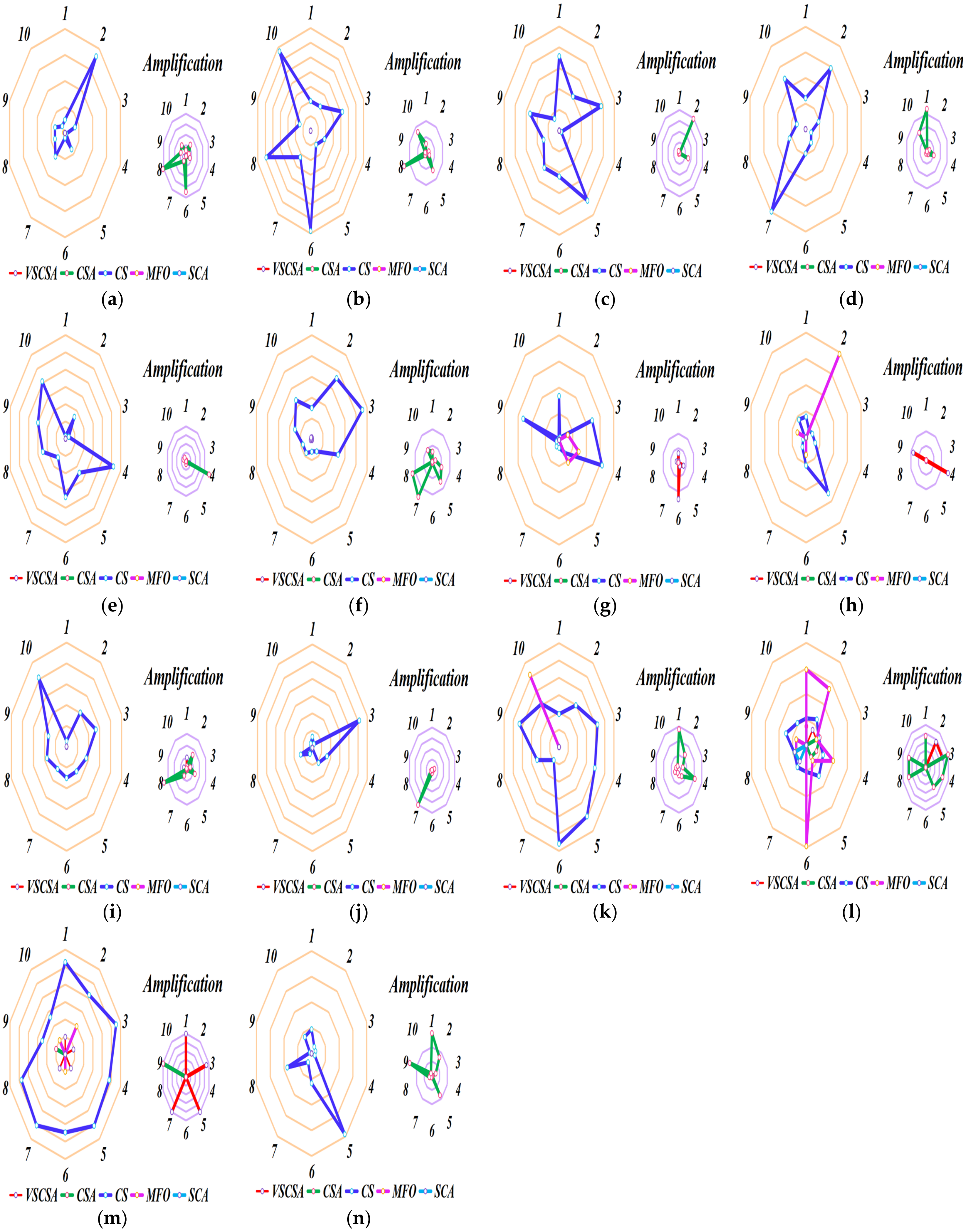

Figure 6. Sub-sequence runs radar charts of two−dimension functions. (**a**) f_1; (**b**) f_2; (**c**) f_3; (**d**) f_4; (**e**) f_5; (**f**) f_6; (**g**) f_7; (**h**) f_8; (**i**) f_9; (**j**) f_{10}; (**k**) $f_{11(D=2)}$; (**l**) $f_{12(D=2)}$; (**m**) $f_{13(D=2)}$; (**n**) $f_{14(D=2)}$.

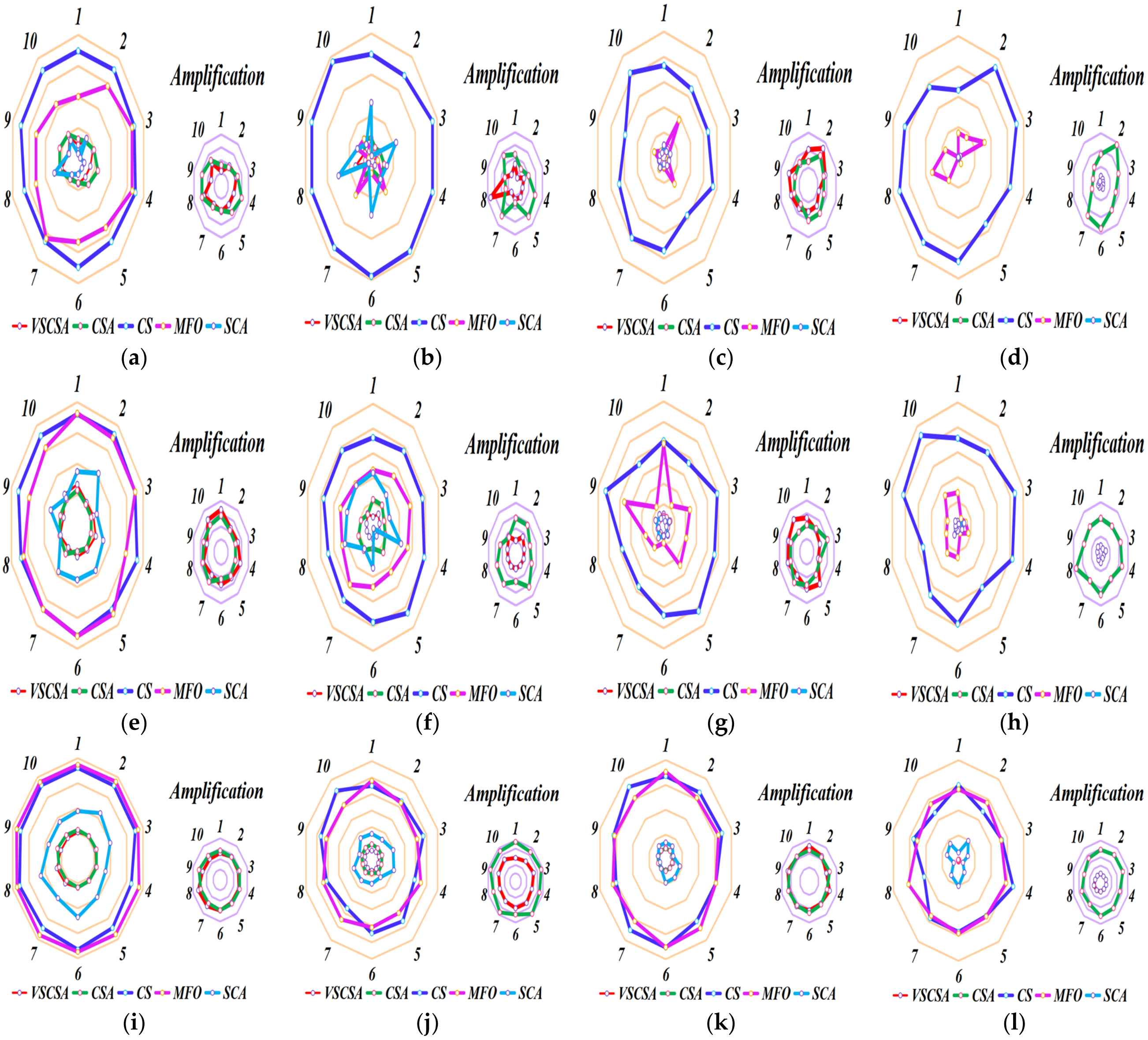

Figure 7. Sub-sequence runs radar charts of high−dimension functions. (**a**) $f_{11(D=30)}$; (**b**) $f_{12(D=30)}$; (**c**) $f_{13(D=30)}$; (**d**) $f_{14(D=30)}$; (**e**) $f_{11(D=60)}$; (**f**) $f_{12(D=60)}$; (**g**) $f_{13(D=60)}$; (**h**) $f_{14(D=60)}$; (**i**) $f_{11(D=200)}$; (**j**) $f_{12(D=200)}$; (**k**) $f_{13(D=200)}$; (**l**) $f_{14(D=200)}$.

4.6. Search Path Results

To test the structural reliability analysis, the computational efficiency, and the accuracy of the proposed algorithm, three-dimensional images of two-dimension functions are given in Figure 8, while the VSCSA path and the CSA path are refracted to a two-dimension plane in Figure 9. The red straight line is the VSCSA searching path, the green dashed line is the CSA searching path, and the pink dot is the theoretical optimal position. CSA searching paths have many short repeat searching paths and occasional long searching paths. The VSCSA algorithm has a strong performance in population diversity, representing the global optimal performance. In the early stage of the algorithm searching process, the VSCSA can quickly traverse and explore the entire solution region, lock in the approximate range of the global optimal solution, and ensure the diversity of the population. At the end stage of the algorithm searching process, the reduction of differences between individuals makes the searching process jump out of local vortices and find the ideal optimization solution, which can improve the algorithm global convergence ability.

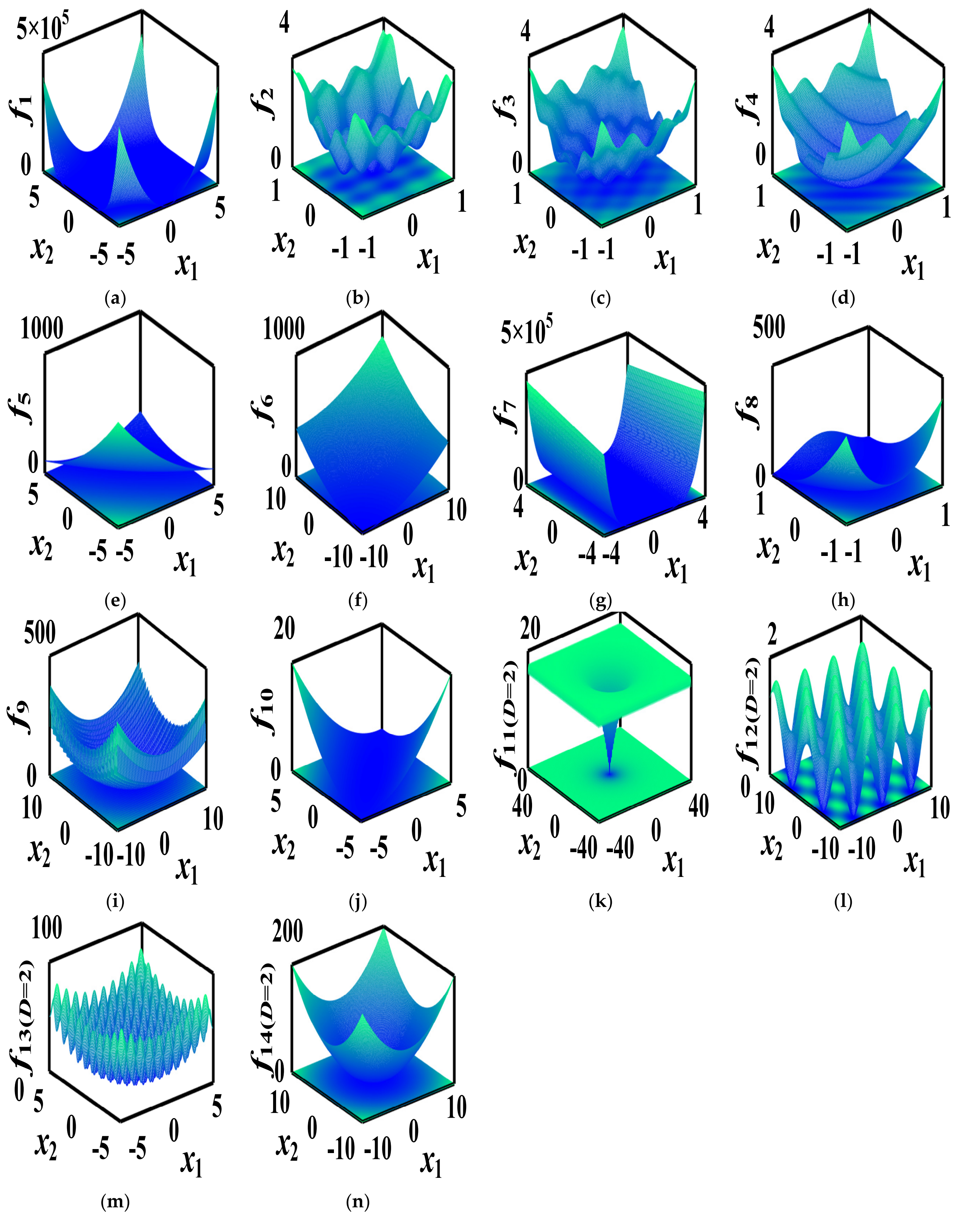

Figure 8. Three−dimension images of two-dimension functions. **(a)** f_1; **(b)** f_2; **(c)** f_3; **(d)** f_4; **(e)** f_5; **(f)** f_6; **(g)** f_7; **(h)** f_8; **(i)** f_9; **(j)** f_{10}; **(k)** $f_{11(D=2)}$; **(l)** $f_{12(D=2)}$; **(m)** $f_{13(D=2)}$; **(n)** $f_{14(D=2)}$.

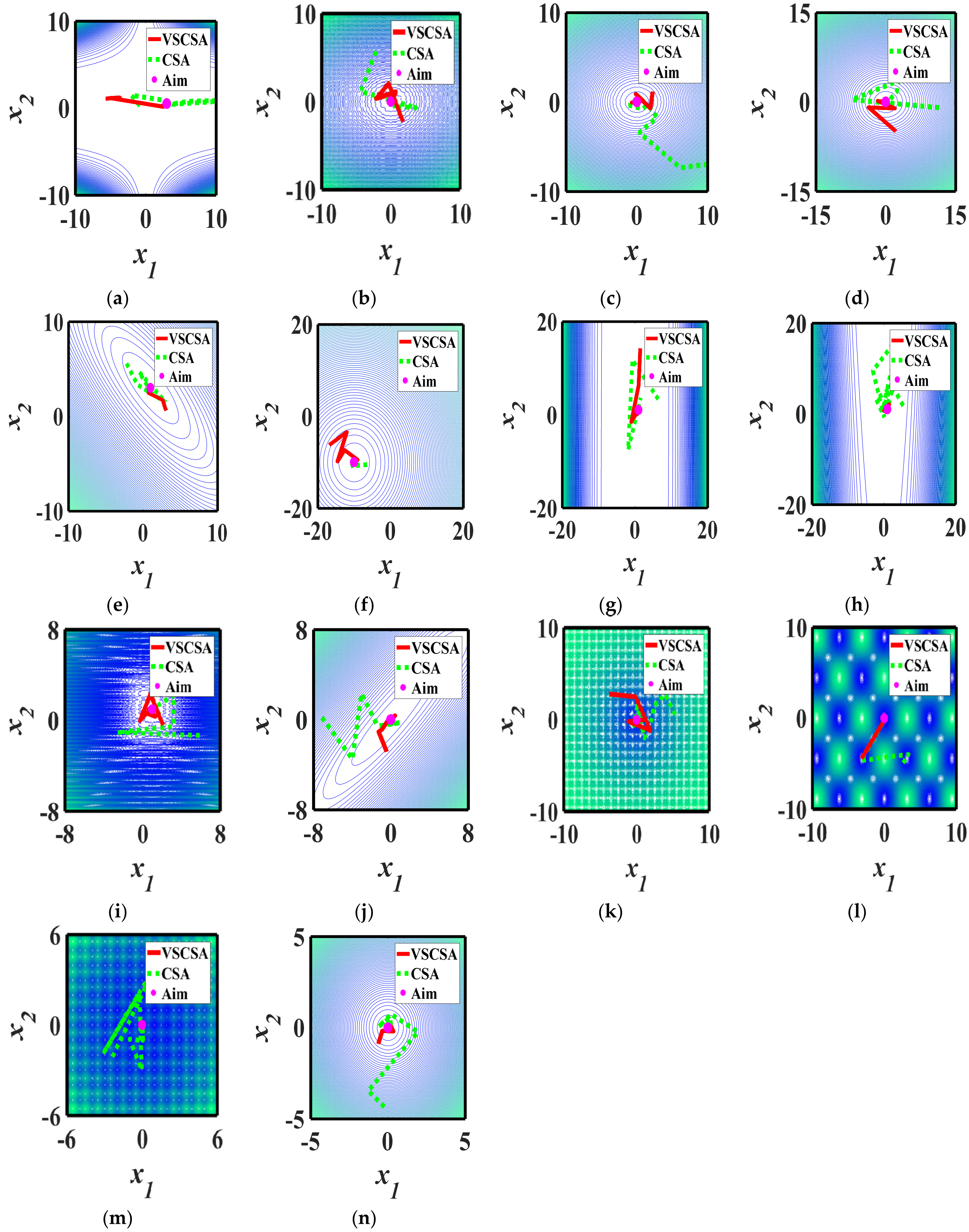

Figure 9. Search paths. (**a**) f_1; (**b**) f_2; (**c**) f_3; (**d**) f_4; (**e**) f_5; (**f**) f_6; (**g**) f_7; (**h**) f_8; (**i**) f_9; (**j**) f_{10}; (**k**) $f_{11(D=2)}$; (**l**) $f_{12(D=2)}$; (**m**) $f_{13(D=2)}$; (**n**) $f_{14(D=2)}$.

4.7. Wilcoxon Rank Sum Test Results

In the process of detecting algorithms, more different experimental results often appear. When comparing and analyzing algorithms, conclusions cannot be drawn solely based on differences in a few results, so statistical analysis should be conducted to test the significance of differences in the data. The Wilcoxon rank sum test result is the *p*-value. If the *p*-value is greater than 0.05, there is no significant change for two sets of data. If the *p*-value is less than 0.05, two algorithm performances are significant. In Table 4, N means that the computer cannot give the *p*-value because of the too-large or too-small *p*-value. In function $f_8, f_{13(D=2)}, f_{11(D=30)}), f_{13(D=30)}, f_{11(D=60)}, f_{13(D=60)}, f_{11(D=200)}, f_{13(D=200)}$, the *p*-value in CSA is larger than 0.05. In function $f_4, f_{10}, f_{11(D=2)}, f_{13(D=2)}, f_{12(D=30)}, f_{13(D=30)}$, the *p*-value in MFO is larger than 0.05. In function $f_{11(D=2)}, f_{12(D=2)}, f_{11(D=30)}, f_{13(D=30)}, f_{13(D=60)}$, the *p*-value in SCA is larger than 0.05. For other algorithms, the Wilcoxon rank sum test results are all less than 0.05. From the results of the Wilcoxon rank sum test by VSCSA, the searching accuracy of the algorithm has been significantly improved, and the improved algorithm is significantly better than the standard CSA in terms of searching accuracy and speed.

Table 4. Comparison of the Wilcoxon rank sum test results.

Function	CSA	CS	MFO	SCA
f_1	0.00033	0.00018	0.00131	0.00018
f_2	6.39×10^{-5}	6.39×10^{-5}	N	N
f_3	6.39×10^{-5}	6.39×10^{-5}	N	N
f_4	6.39×10^{-5}	6.39×10^{-5}	0.07758	N
f_5	0.00219	0.00018	0.00023	0.00018
f_6	0.00018	0.00018	0.00221	0.00018
f_7	0.00283	0.00018	0.00033	0.00033
f_8	0.27304	0.00018	0.00018	0.00018
f_9	0.00443	0.00017	0.00597	0.00017
f_{10}	0.00018	0.00018	0.79134	0.00018
$f_{11(D=2)}$	0.00009	0.00009	1.00000	0.36812
$f_{12(D=2)}$	0.01903	0.00013	0.01914	0.88154
$f_{13(D=2)}$	0.87766	0.00015	0.60255	0.01429
$f_{14(D=2)}$	0.00018	0.00018	0.00283	0.00018
$f_{11(D=30)}$	0.07566	0.00018	0.00018	0.06402
$f_{12(D=30)}$	0.01133	0.00018	0.67758	0.03121
$f_{13(D=30)}$	0.73373	0.00018	0.18588	0.79134
$f_{14(D=30)}$	0.00018	0.00018	0.00018	0.00018
$f_{11(D=60)}$	0.57075	0.00018	0.00018	0.00283
$f_{12(D=60)}$	0.00033	0.00018	0.00018	0.00283
$f_{13(D=60)}$	0.42736	0.00018	0.00018	0.62318
$f_{14(D=60)}$	0.00018	0.00018	0.00018	0.00018
$f_{11(D=200)}$	0.14047	0.00018	0.00018	0.00018
$f_{12(\mathrm{D}=200)}$	0.00018	0.00018	0.00018	0.00018
$f_{13(\mathrm{D}=200)}$	0.85011	0.00018	0.00018	0.02113
$f_{14(\mathrm{D}=200)}$	0.00018	0.00018	0.00018	0.00018

4.8. Algorithm Ranking Results

Algorithm ranking radar charts are shown in Figure 10. The positions of different colored dots in the radar image represent the algorithm searching accuracy. If the algorithm point is close to the center point, the algorithm has a high ranking. It can be seen that the VSCSA surrounds the center point. From the radar graph, it can be seen that VSCSA has the best results among multiple test functions and has the highest searching accuracy among comparison algorithms. Although VSCSA did not achieve comprehensive advantages in some test functions, it achieved optimal searching results in more than half of the test functions, indicating that VSCSA has strong competitiveness. It can be seen that the proposed VSCSA in this paper greatly enhances the CSA searching performance.

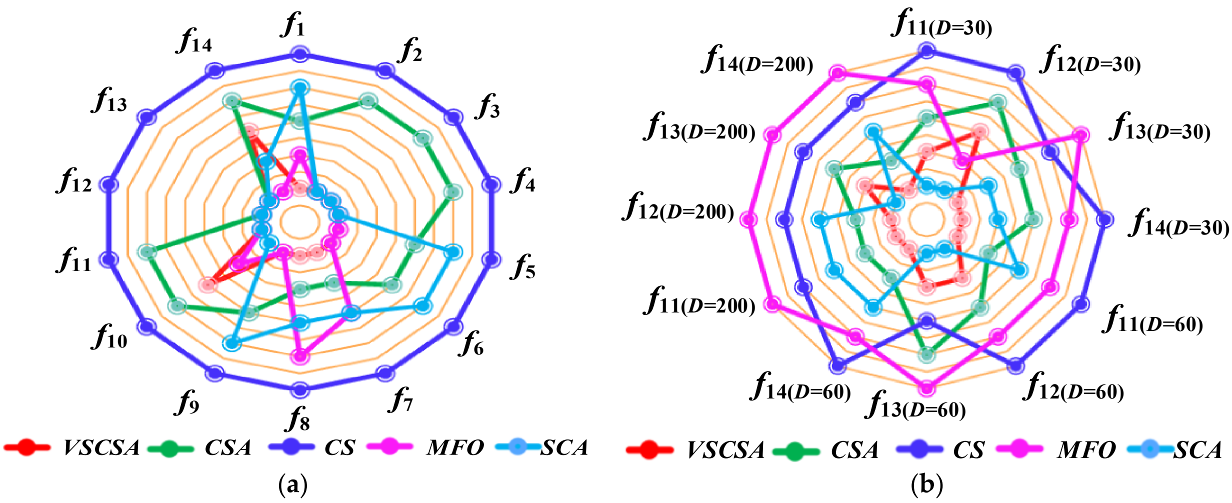

Figure 10. Algorithm ranking figures. (**a**) Two-dimension functions. (**b**) High-dimension functions.

5. CEC2017 Test Function Experiment Results

5.1. Experiment Environments

The IEEE Congress on Evolutionary Computation (CEC) is one of the largest and most significant conferences within Evolutionary Computation (EC). CEC test functions under the CEC conference series are among the widely used benchmarks to test different algorithms. CEC2017 is the test function in the 2017 CEC conference. CEC2017 consists of different problems, including Unimodal, Multimodal, Hybrid, and Composition functions. To further show the proposed algorithm, this paper selected CEC2017 in F_1 to F_{20}. F_1 to F_{20} of CEC2017 are given in Table 5. In Table 5, D is the searching dimension, $F_{\min}$ is the idea function value, and range is the searching scope. F_1 and F_2 are Unimodal Functions, F_3 to F_9 are Simple Multimodal Functions, F_{10} to F_{19} are Hybrid Functions, and F_{20} is the Composition. In this paper, the proposed method compares with state-of-the-art algorithms (SOTA) in recent years. SOTA includes the bald eagle search algorithm (BES) [48], COOT bird algorithm (COOT) [49], wild horse optimizer (WHO) [18], and whale optimization algorithm (WOA) [10]. All algorithm parameters were selected according to original literature. The population size and the maximum number of iterations were 20 and 5000, respectively. To obtain a fair comparison result, all algorithms independently ran 10 times in MATLAB(R2016b). The experimental environment was the Windows 7 operating system, Intel (R) Core (TM) i3-7100CPU, 8GBRAM.

5.2. Experiment Results

The statistical results of algorithms on CEC2017 benchmark functions are shown in Table 6. In Table 6, Min, Max, and Var mean the minimum value, the maximum value, and the variance deviation. For Unimodal Functions, VSCSA, CSA, BES, COOT, and WHO can obtain the ideal value in F_1. All six algorithms can obtain the ideal value in F_2. For Simple Multimodal Functions, all six algorithms can obtain the ideal value in F_3 to F_9. For Hybrid Functions, all six algorithms can obtain the ideal value in F_{10} and F_{11}. CSA can obtain the minimum value in F_{12}. BES can obtain the minimum value in F_{17}. COOT can obtain the minimum value in F_{14} F_{16}. WHO can obtain the minimum value in F_{13} F_{15} F_{18} F_{19}. For the Composition, COOT can obtain the minimum value in F_{20}.

Figure 11 gives the best iteration curves of different algorithms in 10 independent runs. From Figure 11 we can see that VSCSA has the fastest initial iteration speed in function F_1, F_3, and F_9. And VSCSA has the fastest iteration speed in the later stage for function F_2, F_4 to F_8, and F_{10}. VSCSA has the slowest iteration speed in F_{12}, F_{15}, and F_{16}.

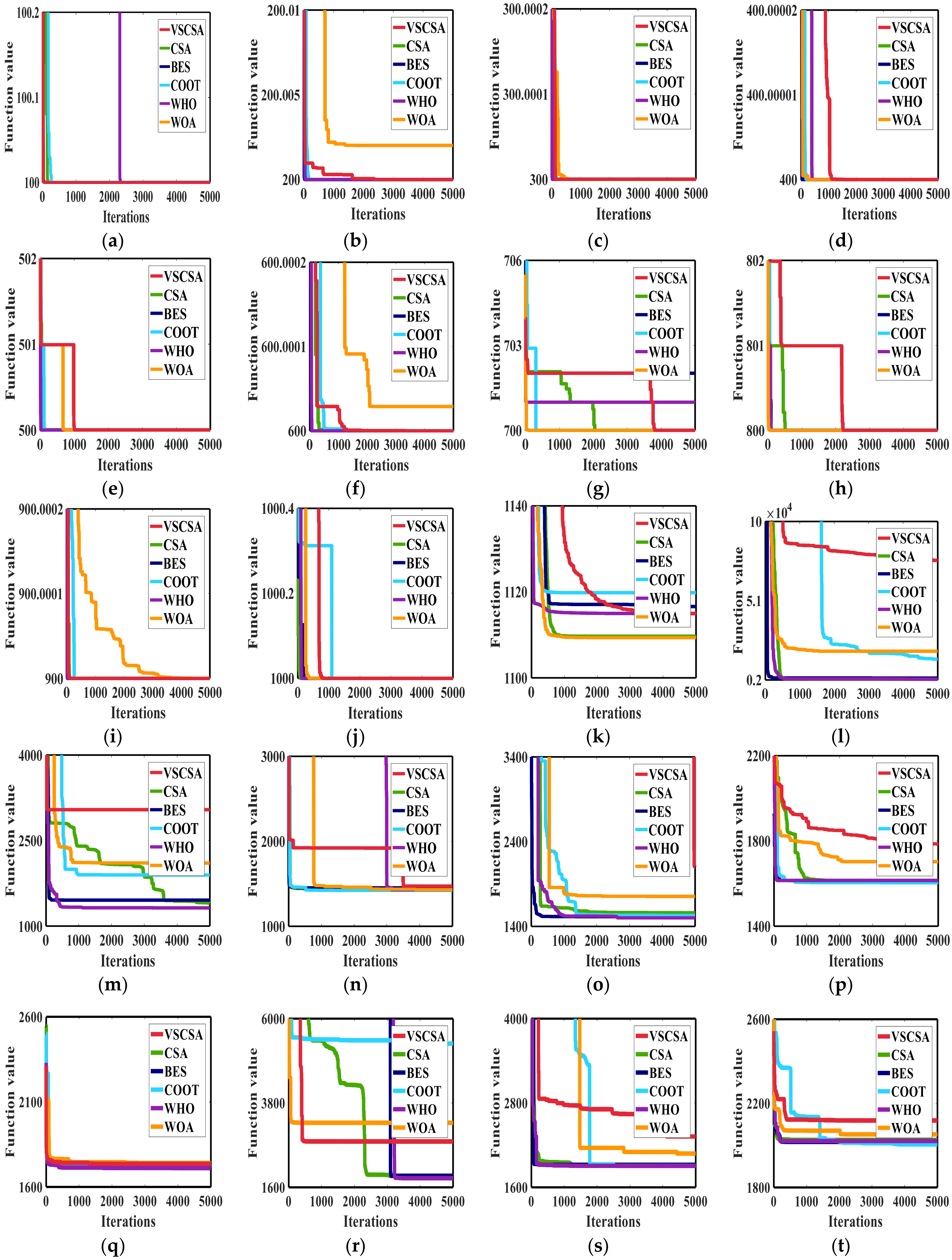

Figure 11. Iteration curves of CEC2017 functions. **(a)** F_1; **(b)** F_2; **(c)** F_3; **(d)** F_4; **(e)** F_5; **(f)** F_6; **(g)** F_7; **(h)** F_8; **(i)** F_9; **(j)** F_{10}; **(k)** F_{11}; **(l)** F_{12}; **(m)** F_{13}; **(n)** F_{14}; **(o)** F_{15}; **(p)** F_{16}; **(q)** F_{17}; **(r)** F_{18}; **(s)** F_{19}; **(t)** F_{20}.

Figure 12 gives box plots for different algorithms after 10 independent runs. VSCSA has the narrowest box plot in function F_2 to F_9. CSA has the narrowest box plot in function F_{12}, $F_{13,}$ and F_{18}. BES has the narrowest box plot in function F_{15}. COOT has the narrowest box plot in function F_{10}, F_{11}, F_{14}, F_{16}, F_{17}, F_{19}, and F_{20}. VSCSA has the worst box plot in function F_1 and F_{15}. BES has the worst box plot in function F_5, F_{16}, F_{17}, and F_{20}. COOT has the worst box plot in function F_7. WHO has the worst box plot in function F_8. WOA has the worst box plot in function F_6, F_{11}, F_{12}, F_{13}, F_{14}, F_{18}, and F_{19}. For F_{10}, VSCSA, BES, WHO, and WOA have large box plots.

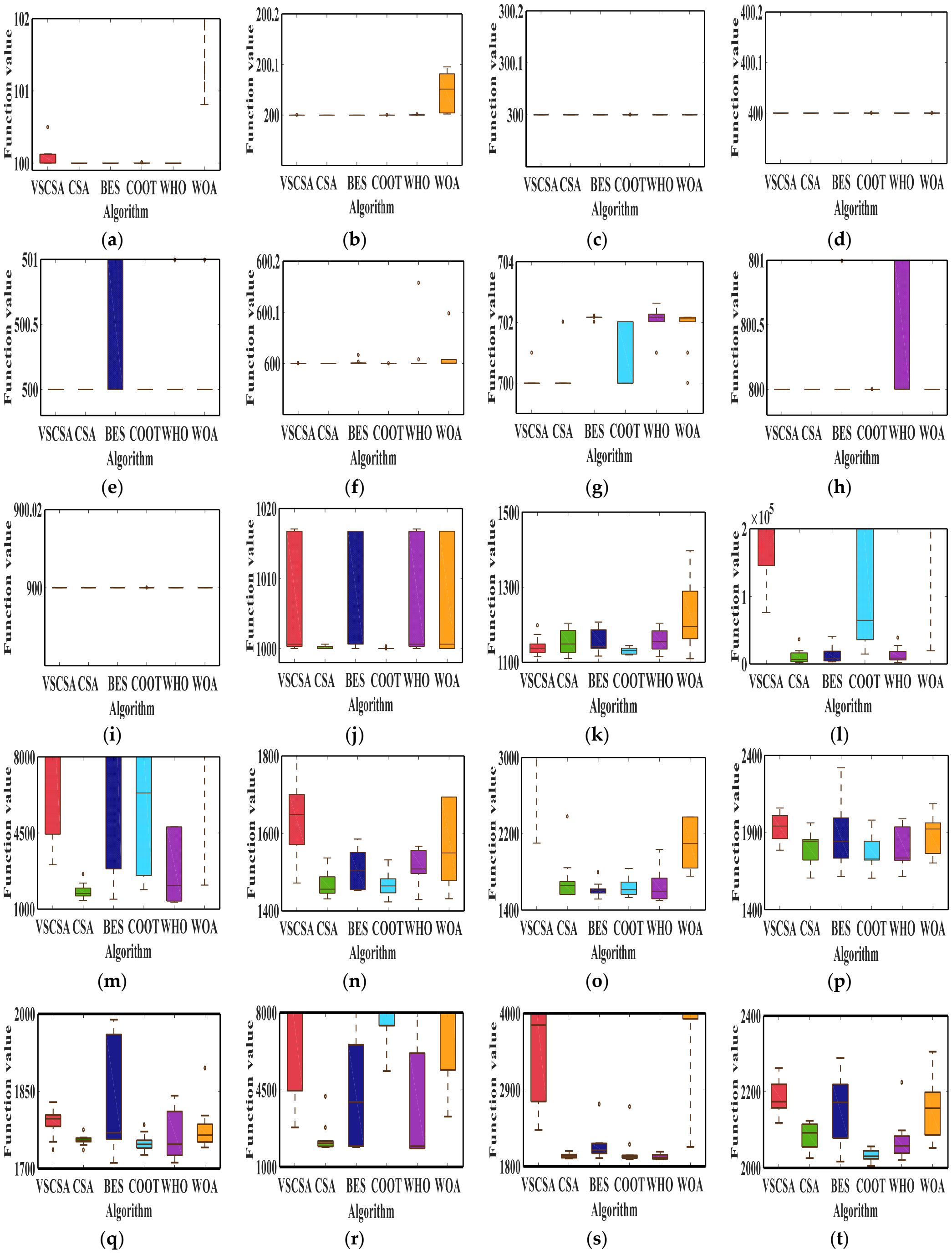

Figure 12. Box plot charts of CEC2017 functions. (**a**) F_1; (**b**) F_2; (**c**) F_3; (**d**) F_4; (**e**) F_5; (**f**) F_6; (**g**) F_7; (**h**) F_8; (**i**) F_9; (**j**) F_{10}; (**k**) F_{11}; (**l**) F_{12}; (**m**) F_{13}; (**n**) F_{14}; (**o**) F_{15}; (**p**) F_{16}; (**q**) F_{17}; (**r**) F_{18}; (**s**) F_{19}; (**t**) F_{20}.

Figure 13 gives radar charts for different algorithms after 10 independent runs. For Figure 13, VSCSA subsequences have the largest radar charts for function F_{15}. BES has the largest radar charts for function F_5 and F_8. WHO has the largest radar charts for function F_6. WOA has the largest radar charts for function F_1 to F_4, F_9, F_{12}, F_{14}, $F_{18,}$ and F_{19}. For function F_7, F_{10}, F_{11}, F_{13}, F_{16}, $F_{17,}$ and F_{20}, many algorithms have large radar charts.

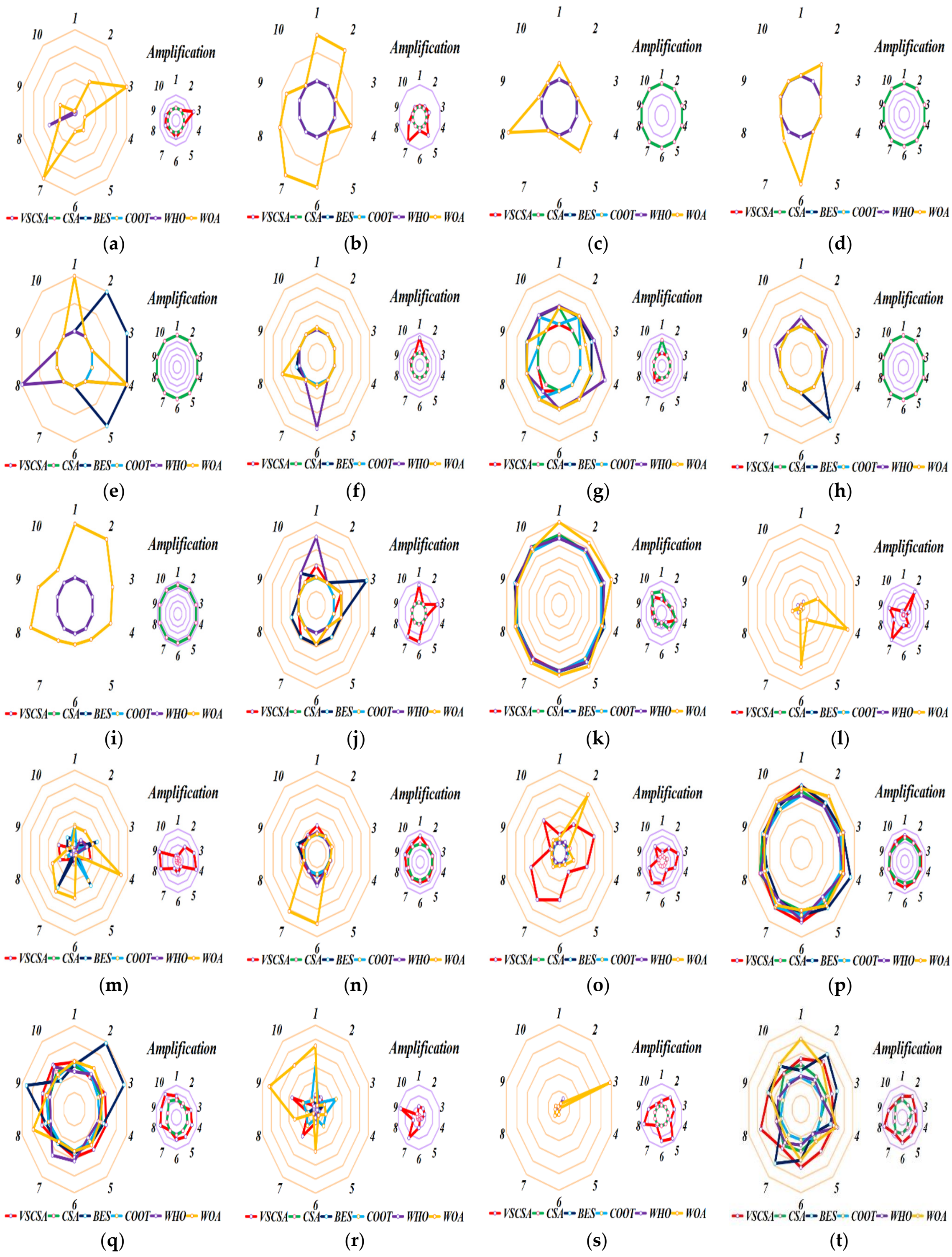

Figure 13. Sub-sequence runs radar charts of CEC2017 functions. **(a)** F_1; **(b)** F_2; **(c)** F_3; **(d)** F_4; **(e)** F_5; **(f)** F_6; **(g)** F_7; **(h)** F_8; **(i)** F_9; **(j)** F_{10}; **(k)** F_{11}; **(l)** F_{12}; **(m)** F_{13}; **(n)** F_{14}; **(o)** F_{15}; **(p)** F_{16}; **(q)** F_{17}; **(r)** F_{18}; **(s)** F_{19}; **(t)** F_{20}.

Table 7 shows the Wilcoxon rank sum test results. In Table 7, N means that the computer cannot give the p-value because of the too-large or too-small p-value. In function F_7, F_{11}, the p-value in CSA is larger than 0.05. In functions F_6, F_8, F_{10}, F_{11}, F_{13}, F_{16}, F_{17}, F_{20}, the p-value in BES is larger than 0.05. In function F_1, F_3, F_6, F_8, F_9, F_{11}, F_{13}, F_{18}, the p-value in COOT is larger than 0.05. In function F_2, F_5, F_6, F_8, F_{10}, F_{11}, F_{14}, F_{17}, the p-value in WHO is larger than 0.05. In function F_{10}, F_{12}, F_{10}, F_{14}, F_{16} to F_{20}, the p-value in WOA is larger than 0.05. For other algorithms, the Wilcoxon rank sum test results are all less than 0.05.

Table 5. Basic information of CEC2017 benchmark functions.

No.	Function	D	Range	F_{min}
F_1	Shifted and Rotated Bent Cigar Function	2	$[-100, 100]$	100
F_2	Shifted and Rotated Zakharov Function	2	$[-100, 100]$	200
F_3	Shifted and Rotated Rosenbrock's Function	2	$[-100, 100]$	300
F_4	Shifted and Rotated Rastrigin's Function	2	$[-100, 100]$	400
F_5	Shifted and Rotated Expanded Scaffer's F6 Function	2	$[-100, 100]$	500
F_6	Shifted and Rotated Lunacek Bi-Rastrigin Function	2	$[-100, 100]$	600
F_7	Shifted and Rotated Non-Continuous Rastrigin's Function	2	$[-100, 100]$	700
F_8	Shifted and Rotated Levy Function	2	$[-100, 100]$	800
F_9	Shifted and Rotated Schwefel's Function	2	$[-100, 100]$	900
F_{10}	Hybrid Function 1 (N = 3)	2	$[-100, 100]$	1000
F_{11}	Hybrid Function 2 (N = 3)	10	$[-100, 100]$	1100
F_{12}	Hybrid Function 3 (N = 3)	10	$[-100, 100]$	1200
F_{13}	Hybrid Function 4 (N = 4)	10	$[-100, 100]$	1300
F_{14}	Hybrid Function 5 (N = 4)	10	$[-100, 100]$	1400
F_{15}	Hybrid Function 6 (N = 4)	10	$[-100, 100]$	1500
F_{16}	Hybrid Function 6 (N = 5)	10	$[-100, 100]$	1600
F_{17}	Hybrid Function 6 (N = 5)	10	$[-100, 100]$	1700
F_{18}	Hybrid Function 6 (N = 5)	10	$[-100, 100]$	1800
F_{19}	Hybrid Function 6 (N = 6)	10	$[-100, 100]$	1900
F_{20}	Composition Function 1 (N = 3)	10	$[-100, 100]$	2000

Table 6. Comparison of results for CEC2017 benchmark functions.

Function	Metric	VSCSA	CSA	BES	COOT	WHO	WOA
F_1	Min	100.0000	100.0000	100.0000	100.0000	100.0000	100.8089
	Max	100.4967	100.0000	100.0000	100.0079	2476.9326	4991.5872
	Var	0.0239	0	0	5.8721×10^{-6}	5.6498×10^{5}	3.0578×10^{6}
F_2	Min	200.0000	200.0000	200.0000	200.0000	200.0000	200.0020
	Max	200.0000	200.0000	200.0000	200.0000	200.0012	200.0951
	Var	3.2226×10^{-13}	0	0	7.7463×10^{-11}	1.6905×10^{-7}	1.3605×10^{-3}
F_3	Min	300.0000	300.0000	300.0000	300.0000	300.0000	300.0000
	Max	300.0000	300.0000	300.0000	300.0000	300.0000	300.0000
	Var	0	0	0	3.5902×10^{-28}	0	9.2862×10^{-22}
F_4	Min	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
	Max	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
	Var	0	0	0	1.3381×10^{-21}	6.9650×10^{-26}	2.1903×10^{-14}
F_5	Min	500.0000	500.0000	500.0000	500.0000	500.0000	500.0000
	Max	500.0000	500.0000	500.9950	500.0000	500.9950	500.9950
	Var	0	0	0.2640	0	0.1760	0.1760
F_6	Min	600.0000	600.0000	600.0000	600.0000	600.0000	600.0000
	Max	600.0002	600.0000	600.0164	600.0000	600.1573	600.0976
	Var	5.1834×10^{-9}	0	2.6128×10^{-5}	8.0042×10^{-11}	0.0025	0.0009

Table 6. *Cont.*

Function	Metric	VSCSA	CSA	BES	COOT	WHO	WOA
F_7	Min	700.0000	700.0000	702.0163	700.0000	700.9950	700.0000
	Max	700.9950	702.0163	702.2136	702.0163	704.7119	702.1708
	Var	0.0990	0.4066	0.0027	1.0842	0.8721	0.5292
F_8	Min	800.0000	800.0000	800.0000	800.0000	800.0000	800.0000
	Max	800.0000	800.0000	804.9748	800.0000	800.9950	800.0000
	Var	0	0	2.4639	5.7443×10^{-27}	0.2310	4.8755×10^{-24}
F_9	Min	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
	Max	900.0000	900.0000	900.0000	900.0000	900.0000	900.0000
	Var	0	0	0	5.7443×10^{-27}	0	8.2264×10^{-14}
F_{10}	Min	1000.0000	1000.0000	1000.0000	1000.0000	1000.0000	1000.0000
	Max	1017.0694	1000.6243	1074.9496	1000.3122	1058.5045	1016.7572
	Var	71.9334	0.0444	465.5578	0.0097	337.6272	63.1675
F_{11}	Min	1114.9624	1109.6715	1116.5732	1119.8084	1114.9368	1109.3504
	Max	1198.3403	1204.5114	1207.1855	1144.6353	1204.4832	1396.9177
	Var	663.0311	1057.8398	958.1988	70.7817	889.1115	1.0239×10^{4}
F_{12}	Min	7.5856×10^{4}	2604.2573	3128.5674	1.4793×10^{4}	2.5114×10^{3}	1.9748×10^{4}
	Max	9.7061×10^{5}	3.6182×10^{4}	4.0349×10^{4}	5.7144×10^{5}	3.8765×10^{4}	1.0693×10^{7}
	Var	$1.0431 \times 10^{+11}$	1.1384×10^{8}	1.4756×10^{8}	$4.7970 \times 10^{+10}$	1.3494×10^{8}	$1.48761 \times 10^{+13}$
F_{13}	Min	3041.5027	1403.7263	1455.2229	1895.3341	1318.7813	2105.0934
	Max	1.8156×10^{4}	2602.3091	3.1311×10^{4}	2.5062×10^{4}	1.3722×10^{4}	5.2478×10^{4}
	Var	3.8814×10^{7}	1.2894×10^{5}	1.3522×10^{8}	6.6879×10^{7}	2.0050×10^{7}	2.3650×10^{8}
F_{14}	Min	1472.2338	1431.3035	1453.1541	1423.2821	1429.3626	1431.4869
	Max	2069.1500	1536.7192	2282.9765	1531.6019	2297.9909	5143.6059
	Var	3.1788×10^{4}	1113.3125	6.3312×10^{4}	1214.3278	6.3780×10^{4}	2.2832×10^{6}
F_{15}	Min	2102.0812	1561.6616	1514.5328	1530.4193	1501.4497	1755.3194
	Max	8248.5418	2380.6647	1.7944×10^{3}	1835.0032	2035.3400	1.0501×10^{4}
	Var	4.5125×10^{6}	6.0137×10^{4}	6405.4267	9095.8887	2.7671×10^{4}	7.1257×10^{6}
F_{16}	Min	1785.9923	1605.1444	1614.9232	1604.0291	1614.1054	1703.0662
	Max	2058.8581	1962.8200	2319.6410	1980.6734	1989.2964	2086.8136
	Var	9501.8979	1.0527×10^{4}	3.8956×10^{4}	1.0911×10^{4}	1.9357×10^{4}	1.4474×10^{4}
F_{17}	Min	1736.2506	1735.7399	1711.0671	1726.6221	1711.3399	1741.0307
	Max	1828.8309	1774.8088	1988.8544	1784.6624	1841.2146	1894.6790
	Var	756.9156	102.0301	1.1332×10^{4}	297.7841	2068.8956	2046.5244
F_{18}	Min	2795.7978	1899.0415	1900.8064	5348.1482	1837.5630	3284.2186
	Max	2.6628×10^{4}	4194.3397	9071.9493	3.2147×10^{4}	8313.3384	5.2491×10^{4}
	Var	6.6663×10^{7}	4.9868×10^{5}	7.3939×10^{6}	6.8506×10^{7}	6.0211×10^{6}	3.3418×10^{8}
F_{19}	Min	2323.2878	1914.0998	1919.9589	1906.5796	1905.3476	2080.4660
	Max	5351.9820	2022.0524	2.6936×10^{3}	2656.9216	3.2933×10^{4}	2.4271×10^{5}
	Var	1.2781×10^{6}	1145.4605	4.8170×10^{4}	5.2873×10^{4}	9.6078×10^{7}	5.3813×10^{9}
F_{20}	Min	2118.6468	2025.8065	2016.9142	2004.9954	2020.8626	2052.4900
	Max	2262.7814	2123.8647	2289.2135	2056.6773	2224.7862	2305.7172
	Var	1758.7689	1230.6049	9242.5172	231.5231	3534.4072	5965.3790

Table 7. Comparison of the Wilcoxon rank sum test results in CEC2017 functions.

Function	CSA	BES	COOT	WHO	WOA
F_1	0.00075	0.00075	0.51989	0.02384	0.00018
F_2	0.00006	0.00006	0.03764	0.47100	0.00018
F_3	N	N	0.36812	N	0.00006
F_4	N	N	0.03498	0.01493	0.00006
F_5	N	0.03359	N	0.16749	0.00023
F_6	0.01493	0.57148	0.10957	0.39943	0.00069
F_7	1.00000	0.00007	0.04981	0.00010	0.00012
F_8	N	0.16808	0.36812	0.07672	0.00023
F_9	N	N	0.36812	N	0.00006
F_{10}	0.00978	0.35909	0.00445	1.00000	0.96975
F_{11}	0.52052	0.42736	0.34470	0.27304	0.01726
F_{12}	0.00018	0.00018	0.04515	0.00018	0.42736
F_{13}	0.00018	0.96985	0.42736	0.01402	0.03764
F_{14}	0.00101	0.03121	0.00077	0.06402	0.30749
F_{15}	0.00025	0.00018	0.00018	0.00018	0.00911
F_{16}	0.00911	0.27304	0.00361	0.01726	0.47268
F_{17}	0.01402	0.57075	0.00459	0.18588	0.14047
F_{18}	0.00033	0.02113	0.27304	0.00283	0.12122
F_{19}	0.00018	0.00033	0.00033	0.00283	0.05390
F_{20}	0.00033	0.73373	0.00018	0.00220	0.34470

6. Engineering Applications

6.1. Three Bar Truss Design Problem

The three bar truss design problem is a civil engineering problem, and the weight of the bar structure is the key problem in the Gear Train Problem which owns a problematic and constrained space. The constraints of this problem are based on the stress constraints of each bar. Figure 14 is the structural diagram of the three bar truss design problem. A_1 A_2 A_3 respectively represents the length of the bar, P means the force value, L means the space length.

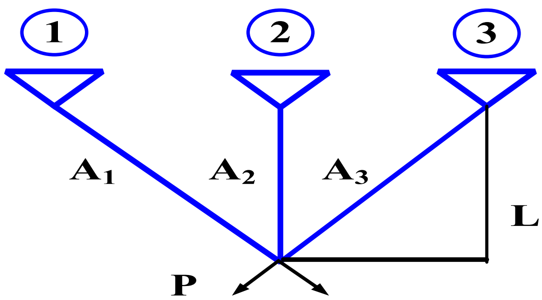

Figure 14. Three bar truss design problem.

This problem can be described mathematically as follows:

$$\begin{aligned}
&\textit{Consider} && \vec{X} = [x_1 x_2] = [A_1 A_1] \\
&\textit{Minimize} && f\left(\vec{X}\right) = \left(2\sqrt{2}x_1 + x_2\right) \times L \\
&\textit{Subject to} && g_1\left(\vec{X}\right) = \frac{2\sqrt{2}x_1 + x_2}{\sqrt{2x_1^2 + 2x_1x_2}} \times P - \sigma \leq 0 \\
& && g_2\left(\vec{X}\right) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}} \times P - \sigma \leq 0 \\
& && g_3\left(\vec{X}\right) = \frac{1}{\sqrt{2}x_2 + x_1} \times P - \sigma \leq 0 \\
& 0 \leq x_1, x_2 \leq 1 \quad P = 2\mathrm{KN/cm}^2 \quad L = 100\mathrm{cm} \quad \sigma = 2\mathrm{KN/cm}^2
\end{aligned}$$

In this paper, basic CSA were selected for the CSA literature. Comparison algorithms and parameters selected the algorithm literature [50], with each method tested 30 times with 1000 iterations and a maximum of 60,000 number function evaluations (NFEs). The results of best, mean, minimum values, maximum values, and the standard deviation value are given in Table 8.

Table 8. Results of three bar truss problem.

Algorithm	Min	Max	Std	Avg
WHO	263.8958433765	263.8958433765	$1.2710574865 \times 10^{-13}$	263.8958433765
PSO	263.8958433827	263.8960409745	$5.3917161119 \times 10^{-5}$	263.8959010895
GA	263.8958919373	263.9970875475	0.0252055577	263.9095296976
AEFA	265.1001279647	280.9534461900	4.0558625686	271.8733092380
FA	263.8958477145	263.8989975836	$8.8455344984 \times 10^{-4}$	263.8964634153
GSA	263.8968857660	264.1972851298	0.0948941056	264.0059193538
HHO	263.8959528570	264.0672685182	0.0467621287	263.9419743129
MVO	263.8958747019	263.9000377233	$9.8601397499 \times 10^{-4}$	263.8967256362
WOA	263.8959383525	265.6916186134	0.5029074306	264.3105859277
SSA	263.8958435096	263.8998220362	$7.2678747873 \times 10^{-4}$	263.8962415757
GWO	263.8959818300	263.9028435626	0.0014371714	263.8975822284
CSA	263.8958433765	263.8958433765	$6.4741204424 \times 10^{-12}$	263.8958433765
VSCSA	263.8958433765	263.9145156687	0.0037434952	263.8981466437

The VSCSA Min value is the same as the CSA Min value and the WHO Min value. The VSCSA Max value is larger than the CSA Max value. WHO and CSA obtain the less Max value and the Avg value. WHO obtains the less Std value. AEFA obtains the worst Min value, the Max value, the Std value, and the Avg value.

6.2. The Gear Train Problem

The cost of the gear ratio of the gear train is the key problem in the Gear Train Problem which owns only four parameters in boundary constraints. Four parameters are discrete because each gear should have an integral number of teeth in this problem. Discrete variables add different complexities for this problem. Figure 15 is the structural diagram of the Gear Train Problem. Four parameters are the numbers of teeth on the gears: nA, nB, nC, and nD. A, B, C, and D mean centre points.

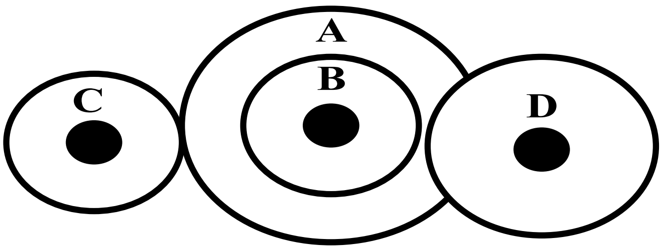

Figure 15. Gear train problem.

This problem can be described mathematically as follows:

$$\begin{array}{ll} \textit{Consider} & \vec{X} = [x_1 x_2 x_3 x_4] \\ \textit{Minimize} & f\left(\vec{X}\right) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4}\right)^2 \\ \textit{Subject to} & 12 \leq x_1, x_2, x_3, x_4 \leq 60 \end{array}$$

In this paper, basic CSA was selected for the CSA literature. Comparison algorithms and parameters selected the algorithm literature, the number of population sizes is set to 50, and the maximum number of iterations is set to 1000. All algorithms are executed for

30 independent runs. The results of the best, mean, minimum values, maximum values, and the standard deviation value are given in Table 9. Comparison algorithms include CS [46], FPA [50], FSA [51], SA [52], and SCA [14]. The VSCSA Min value is the same as the CSA Min value. SCA obtains the worst Min value. SCA obtains the worst Max value, Std value, and Avg value. The VSCSA Min value, Max value, Std value, and Avg value are larger than those of CSA. There is no specific algorithm that can perfectly solve all engineering problems. Different algorithms can be selected for different engineering problems.

Table 9. Results of the gear train design problem.

Algorithm	Min	Max	Std	Avg
CS	$2.7008571489 \times 10^{-12}$	$8.7008339998 \times 10^{-9}$	$2.5469034697 \times 10^{-9}$	$2.5277681200 \times 10^{-9}$
FPA	$2.3078157333 \times 10^{-11}$	$1.3616491391 \times 10^{-9}$	$5.1819924289 \times 10^{-10}$	$5.5155436237 \times 10^{-10}$
FSA	$1.0935663792 \times 10^{-9}$	$4.4677248806 \times 10^{-7}$	$8.5620977463 \times 10^{-8}$	$4.7845971457 \times 10^{-8}$
SA	$2.3078157333 \times 10^{-11}$	$1.3616491391 \times 10^{-9}$	$4.8777877665 \times 10^{-10}$	$6.1683323242 \times 10^{-10}$
SCA	$3.6358329757 \times 10^{-9}$	$2.0768133383 \times 10^{-1}$	$4.9002989331 \times 10^{-2}$	$1.6613443644 \times 10^{-2}$
CSA	$2.7008571489 \times 10^{-12}$	$2.3576406580 \times 10^{-9}$	$5.5363138249 \times 10^{-10}$	$2.7032649321 \times 10^{-10}$
VSCSA	$2.7008571489 \times 10^{-12}$	$2.7264505977 \times 10^{-8}$	$7.3324954585 \times 10^{-9}$	$4.4138792095 \times 10^{-9}$

7. Conclusions

In this paper, VSCSA is introduced to solve function problems. The proposed algorithm uses the cosine function to enhance the CSA searching ability. VSCSA has strong problem applicability and can effectively find the global optimum in a short iteration period, greatly improving the solution accuracy. In conclusion, the proposed algorithm VSCSA has significant advantages over CSA in CEC-2017 fitness values, iteration curves, box plots, and search paths. In addition, the Wilcoxon test results statistically indicate differences between VSCSA and other comparative algorithms. Engineering applications show that the proposed algorithm has strong competitiveness. The above data and conclusions indicate that the improvement strategy proposed in this paper has achieved good results, greatly improving the performances of the original CSA. Many algorithms have been applied to specific fields such as medicine, aerospace, and industry and have achieved good results. Therefore, combining VSCSA with practical problems in specific fields is a direction for future research.

Author Contributions: Conceptualization, Y.F.; methodology, Y.F.; formal analysis, H.Y.; investigation, Z.X.; resources, Y.W. and Y.F.; writing—original draft preparation, Y.F.; writing—review and editing, H.Y. and Z.X.; visualization, D.L.; supervision, Y.W.; project administration, Y.W.; funding acquisition, Y.W and Y.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 52175502, as well as the Natural Science Foundation of Heilongjiang Province, grant number LH2023E082, in addition to the basic research business fee projects of provincial undergraduate universities in Heilongjiang Province, grant number 2022-KYYWF-0144.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are included in this article.

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

References

1. Ray, T.; Liew, K.M. Society and civilization: An optimization algorithm based on the simulation of social behavior. *IEEE Trans. Evol. Comput.* **2003**, *7*, 386–396. [CrossRef]
2. Iba, K. Reactive power optimization by genetic algorithm. *IEEE Trans. Power Syst.* **1994**, *9*, 685–692. [CrossRef]
3. Mafarja, M.; Aljarah, I.; Heidari, A.A.; Hammouri, A.I.; Faris, H.; Al-Zoubi, A.M.; Mirjalili, S. Evolutionary Population Dynamics and Grasshopper Optimization approaches for feature selection problems. *Knowl. Based Syst.* **2018**, *145*, 25–45. [CrossRef]

4. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]
5. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
6. Borchers, A.; Pieler, T. Programming pluripotent precursor cells derived from Xenopus embryos to generate specific tissues and organs. *Genes* **2010**, *1*, 413–426. [CrossRef]
7. Storn, R.; Price, K. Differential Evolution-A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
8. Abdollahzadeh, B.; Gharehchopogh, F.S.; Mirjalili, S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Comput. Ind. Eng.* **2021**, *158*, 107408. [CrossRef]
9. Zhong, C.; Li, G.; Meng, Z. Beluga whale optimization: A novel nature-inspired metaheuristic algorithm. *Knowl. Based Syst.* **2022**, *251*, 109215. [CrossRef]
10. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
11. Karami, H.; Anaraki, M.V.; Farzin, S.; Mirjalili, S. Flow Direction Algorithm (FDA): A Novel Optimization Approach for Solving Optimization Problems. *Comput. Ind. Eng.* **2021**, *156*, 107224. [CrossRef]
12. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
13. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
14. Mirjalili, S. SCA: A Sine Cosine Algorithm for Solving Optimization Problems. *Knowl. Based Syst.* **2016**, *96*, 120–133. [CrossRef]
15. Dhiman, G.; Kumar, V. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Adv. Eng. Softw.* **2017**, *114*, 48–70. [CrossRef]
16. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [CrossRef]
17. Cheng, M.-Y.; Prayogo, D. Symbiotic Organisms Search: A new metaheuristic optimization algorithm. *Comput. Struct.* **2014**, *139*, 98–112. [CrossRef]
18. Naruei, I.; Keynia, F. Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems. *Eng. Comput.* **2021**, *38*, 3025–3056. [CrossRef]
19. Rezaei, F.; Safavi, H.R.; Abd Elaziz, M.; Mirjalili, S. GMO: Geometric mean optimizer for solving engineering prob lems. *Soft Comput.* **2023**, *27*, 10571–10606. [CrossRef]
20. Chopra, N.; Ansari, M.M. Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Syst. Appl.* **2022**, *198*, 116924. [CrossRef]
21. Dehghani, M.; Trojovska, E.; Trojovsky, P.; Montazeri, Z. Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowl.-Based Syst.* **2023**, *259*, 110011. [CrossRef]
22. Zhao, S.; Zhang, T.; Ma, S.; Chen, M. Dandelion Optimizer: A nature-inspired metaheuristic algorithm for engineering applications. *Eng. Appl. Artif. Intell. Int. J. Intell. Real-Time Autom.* **2022**, *114*, 105075. [CrossRef]
23. Jia, H.; Peng, X.; Lang, C. Remora Optimization Algorithm. *Expert Syst. Appl.* **2021**, *185*, 115665. [CrossRef]
24. Guan, Z.; Ren, C.; Niu, J.; Wang, P.; Shang, Y. Great Wall Construction Algorithm: A novel meta-heuristic algorithm for engineer problems. *Expert Syst. Appl.* **2023**, *233*, 120905. [CrossRef]
25. Zhang, Y.; Jin, Z.; Mirjalili, S. Generalized normal distribution optimization and its applications in parameter extraction of photovoltaic models. *Energy Convers. Manag.* **2020**, *224*, 113301. [CrossRef]
26. Trojovsky, P.; Dehghani, M. Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications. *Sensors* **2022**, *22*, 855. [CrossRef] [PubMed]
27. Moosavi, S.H.S.; Bardsiri, V.K. Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation. *Eng. Appl. Artif. Intell.* **2017**, *60*, 1–15. [CrossRef]
28. Rao, R.V.; Savsani, V.J.; Vakharia, D.P. Teaching–Learning-Based Optimization: An optimization method for continuous non-linear large scale problems. *Inf. Sci.* **2012**, *183*, 1–15. [CrossRef]
29. Seyyedabbasi, A.; Kiani, F. Sand Cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. *Eng. Comput.* **2022**, *39*, 2627–2651. [CrossRef]
30. Jiang, X.; Lin, Z.; He, T.; Ma, X.; Ma, S.; Li, S. Optimal Path Finding With Beetle Antennae Search Algorithm by Using Ant Colony Optimization Initialization and Different Searching Strategies. *IEEE Access* **2020**, *8*, 15459–15471. [CrossRef]
31. Pan, H.; Gong, J. Application of Particle Swarm Optimization (PSO) Algorithm in Determining Thermodynamics of Solid Combustibles. *Energies* **2023**, *16*, 5302. [CrossRef]
32. Zandavi, S.M.; Chung, V.Y.Y.; Anaissi, A. Stochastic Dual Simplex Algorithm: A Novel Heuristic Optimization Algorithm. *IEEE Trans. Cybern.* **2021**, *51*, 2725–2734. [CrossRef] [PubMed]
33. Liang, X.; Cai, Z.; Wang, M.; Zhao, X.; Chen, H.; Li, C. Chaotic oppositional sine–cosine method for solving global optimization problems. *Eng. Comput.* **2020**, *38*, 1223–1239. [CrossRef]
34. Pazhaniraja, N.; Basheer, S.; Thirugnanasambandam, K.; Ramalingam, R.; Rashid, M.; Kalaivani, J. Multi-objective Boolean grey wolf optimization based decomposition algorithm for high-frequency and high-utility itemset mining. *AIMS Math.* **2023**, *8*, 18111–18140. [CrossRef]

35. Huang, Z.; Li, F.; Zhu, L.; Ye, G.; Zhao, T. Phase Mask Design Based on an Improved Particle Swarm Optimization Algorithm for Depth of Field Extension. *Appl. Sci.* **2023**, *13*, 7899. [CrossRef]
36. Akın, P. A new hybrid approach based on genetic algorithm and support vector machine methods for hyperparameter optimization in synthetic minority over-sampling technique (SMOTE). *AIMS Math.* **2023**, *8*, 9400–9415. [CrossRef]
37. Askarzadeh, A. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. *Comput. Struct.* **2016**, *169*, 1–12. [CrossRef]
38. Shekhawat, S.; Saxena, A. Development and applications of an intelligent crow search algorithm based on opposition based learning. *ISA Trans.* **2020**, *99*, 210–230. [CrossRef] [PubMed]
39. Chen, Y.; Ye, Z.; Gao, B.; Wu, Y.; Yan, X.; Liao, X. A Robust Adaptive Hierarchical Learning Crow Search Algorithm for Feature Selection. *Electronics* **2023**, *12*, 3123. [CrossRef]
40. Díaz, P.; Pérez-Cisneros, M.; Cuevas, E.; Avalos, O.; Gálvez, J.; Hinojosa, S.; Zaldivar, D. An Improved Crow Search Algorithm Applied to Energy Problems. *Energies* **2018**, *11*, 571. [CrossRef]
41. Bhullar, A.K.; Kaur, R.; Sondhi, S. Enhanced crow search algorithm for AVR optimization. *Soft Comput.* **2020**, *24*, 11957–11987. [CrossRef]
42. Gadekallu, T.R.; Alazab, M.; Kaluri, R.; Maddikunta, P.K.R.; Bhattacharya, S.; Lakshmanna, K.; M, P. Hand gesture classification using a novel CNN-crow search algorithm. *Complex Intell. Syst.* **2021**, *7*, 1855–1868. [CrossRef]
43. Braik, M.; Al-Zoubi, H.; Ryalat, M.; Sheta, A.; Alzubi, O. Memory based hybrid crow search algorithm for solving numerical and constrained global optimization problems. *Artif. Intell. Rev.* **2022**, *56*, 27–99. [CrossRef]
44. Samieiyan, B.; MohammadiNasab, P.; Mollaei, M.A.; Hajizadeh, F.; Kangavari, M. Solving dimension reduction problems for classification using Promoted Crow Search Algorithm (PCSA). *Computing* **2022**, *104*, 1255–1284. [CrossRef]
45. Guo, Q.; Chen, H.; Luo, J.; Wang, X.; Wang, L.; Lv, X.; Wang, L. Parameter inversion of probability integral method based on improved crow search algorithm. *Arab. J. Geosci.* **2022**, *15*, 180. [CrossRef]
46. Gandomi, A.H.; Yang, X.-S.; Alavi, A.H. Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Eng. Comput.* **2013**, *29*, 17–35. [CrossRef]
47. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* **2015**, *89*, 228–249. [CrossRef]
48. Alsattar, H.A.; Zaidan, A.A.; Zaidan, B.B. Novel meta-heuristic bald eagle search optimisation algorithm. *Artif. Intell. Rev.* **2020**, *53*, 2237–2264. [CrossRef]
49. Naruei, I.; Keynia, F. A New Optimization Method Based on Coot Bird Natural Life Model. *Expert Syst. Appl.* **2021**, *183*, 115352. [CrossRef]
50. Yang, X.-S.; Karamanoglu, M.; He, X. Flower pollination algorithm: A novel approach for multiobjective optimiza tion. *Eng. Optim.* **2014**, *46*, 1222–1237. [CrossRef]
51. Elsisi, M. Future search algorithm for optimization. *Evol. Intell.* **2018**, *12*, 21–31. [CrossRef]
52. Osman, I.H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Ann. Oper. Res.* **1993**, *41*, 421–451. [CrossRef]