IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 28 February 2023, accepted 13 March 2023, date of publication 20 March 2023, date of current version 29 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3258970

RESEARCH ARTICLE

An Improved Future Search Algorithm Based on the Sine Cosine Algorithm for Function Optimization Problems

YUQI FAN, SHENG ZHANG, HUIMIN YANG, DI XU, AND YAPING WANG

Key Laboratory of Advanced Manufacturing and Intelligent Technology, School of Mechanical and Power Engineering, Ministry of Education, Harbin University of Science and Technology, Harbin 150080, China

Corresponding author: Yaping Wang (wypbl@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 52175502.

ABSTRACT Future search algorithm imitates the person living life. If one person finds that his life is not good, he will try to change his living life, and he will imitate a more successful person. To overcome insufficient performances of the basic Future search algorithm, this paper proposed an improved Future search algorithm based on the sine cosine algorithm (FSASCA). The proposed algorithm uses sine cosine algorithm to loop-progressive find the best solution. The searching method of the sine cosine algorithm can make the feasible solution to be re-positioned around another feasible solutions, which can make the proposed algorithm have a strong exploitation ability. Four coefficient factors are added in the basic FSA, and new update methods are introduced in the searching phase. To verify the searching and optimization performances of the proposed algorithm in this paper, this paper also gives data calculation results, Wilcoxon rank sum test, iteration figures, box plot figures, and searching path figures. Experimental results showed that FSASCA has a better iteration speed, the convergence precision, the solving accuracy, the strong competitive, and the high balance.

INDEX TERMS Future search algorithm, sine cosine algorithm, optimization problem, function optimization.

I. INTRODUCTION

Optimization methods has made the great progress and owned many theoretical researches and application results. Existing optimization methods can be divided into two categories, including the traditional deterministic optimization method and the intelligent optimization algorithm. Traditional deterministic optimization methods mainly includes the steepest descent method, the conjugate gradient method, the Newton method, the variable metric method, the simplex method, the penalty function method, which generally have the perfect mathematical foundation and the strict mathematical definition. There are some disadvantages for traditional deterministic optimization methods. The single point operation greatly limits the calculation efficiency. Traditional optimization methods need that the objective function is a continuous and differentiable analytic function, which even required to be high-order differentiable. So traditional optimization methods are not suitable for solving some discrete, discontinuous, derivative free and other optimization problems. The computational complexity is generally large and sensitive to the initial point selection in the traditional optimization methods [1], [2]. With the increasing complexity of the mathematical and industrial problems, traditional optimization method can not meet the optimization design requirements of practical problems in terms of the calculating convergence speed, the initial value sensitivity. Therefore, the new research goal is to find an efficient optimization algorithms with the fast calculation speed, the good convergence performance, the insensitivity to the selection of initial values, and no functional property requirements for the objective function and constraint function. With the computer field development, researchers pay

The associate editor coordinating the review of this manuscript and approving it for publication was Christos Anagnostopoulos.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

more and more attention to the development of intelligent optimization algorithms for solving nonlinear optimization problems based on the simulating natural phenomena, living habits of colony organisms, and the specie evolutionary processes. Optimization algorithms are evolutionary and stochastic, and can continuously enhance their adaptability through self-learning in a time-varying environment [3], [4].

In recent years, scholars have proposed many advanced metaheuristic algorithms, such as african vultures optimization algorithm (AVOA) [5], grey wolf optimizer (GWO) [6], crow search algorithm (CSA) [7], artificial butterfly optimization (ABO) [8], gravitational search algorithm (GSA) [9], chao game optimization (CGO) [10], wild horse optimizer (WHO) [11], whale optimization algorithm (WOA) [12], equilibrium optimizer (EO) [13], teaching learning based optimization (TLBO) [14], symbiotic organisms search(SOS) [15], Electro-search algorithm (ES) [16], water wave optimization (WWO) [17], moth name optimization algorithm (MFO) [18], spotted hyena optimizer (SHO) [19], mine blast algorithm (MBA) [20], and so on [21], [22], [23], [24], [25], [26]. The high efficiency of optimization algorithms sets the strong support in industry fields, such as global optimization problem [27], [28], [29], [30], 0-1 knapsack problem [31], [32], [33], [34], path planning problems [35], [36], [37], [38], image fields [39], [40], and so on [41], [42], [43], [44].

Future search algorithm was proposed by M. Elsisi in 2018 [45]. FSA imitates the person living life. All people in the world are pursuing the best life. If a person finds his life is not good, he will imitates the successful people to improve his life. According to this imitating behavior, FSA algorithm is proposed by mathematical equations. The feasible solutions in FSA is represented by persons. One person which can get the best life in a country is the optimal local solution. FSA renews the random initial uses the local searching between people and the global searching between the histories optimal persons. But FSA neglects neighborhood solutions in the searching process. To solve this problem, this paper uses Sine Cosine Algorithm (SCA) [46] to loop-progressive find the best solution for basic FSA, and introduces an improved Future search algorithm based on the sine cosine algorithm (FSASCA). The Sine Cosine Algorithm (SCA) is proposed by Seyedali Mirjalili in 2016. SCA can establish many initial random feasible solutions by using a mathematical model based on sine and cosine functions, which can enhance searching ability for FSA.

For experiment analyses, this paper used different testing functions and different algorithms to show FSASCA searching performances. This paper gave data calculation results, Wilcoxon rank sum test, iteration figures, box plot figures, and searching path figures. All experiment results shows that FSASCA owns a stronger exploration searching ability than FSA. The remainder of this paper is organized as follows. Section II introduces the basic FSA. Section III introduces FSASCA. In Section IV, this paper calculates functions. The conclusions are presented in Section V.

II. FUTURE SEARCH ALGORITHM

All species on the earth find the best living way. If any creatures found that their livings are not good, they will try to emulate other living life in the world. In biology, human beings are classified as hominids. All people in the world look for the best living life. The future search algorithm was inspired by finding the best living life for people. FSA formulates a mathematical equations which can update the random initial and use the local searching between people and the global searching between the histories optimal persons. Some heuristic algorithms begin random steps and build iterations based on the initial optimal solution. The feasible solution may be far from the global optimal solution, which makes the other algorithms take large iteration times to get the best solution. FSA can overcome those shortcomings update the random initial each iteration. FSA uses the local best solution and the global best solution to seek the best solutions. In FSA, the searching scope can be seen that persons find for the best life in the world. One person achieving the best performance can be seen as the optimal local solution between the other persons. One person achieving the best performance in a country over some years represents the optimal global solution between the other persons.

FSA is built based on mathematical equations and starts steps based on random solutions:

$$S(i,:) = Lb + (Ub - Lb) * rand(1, d) \tag{1}$$

where $S(i,:)$ means the i-th solution, Lb and Ub mean the lower limit bounds and the upper limit bounds, *rand* is the uniformly distributed pseudo-random number in d dimensions.

After building all solutions, each solution can be seen as the local solution (LS), and the best solution can be seen as the global solution (GS), and then the algorithm starts its iterations to find the optimal solution.

First, the searching process in FSA depends on the LS supporting the exploitation characteristic.

$$S(i,:)_L = (LS(i,:) - S(i,:)) * rand \tag{2}$$

Second, the FSA searching process in the searching scope depends on the GS supporting the exploitation characteristic.

$$S(i,:)_G = (GS - S(i,:)) * rand \tag{3}$$

After calculating the local convergence and the global convergence, each solution can be updated by:

$$S(i,:) = S(i,:) + S(i,:)_L + S(i,:)_G \tag{4}$$

The algorithm updates the GS and LS. And each solution can be updated by:

$$S(i,:) = GS + (GS - S(i,:)) * rand \tag{5}$$

TABLE 1. Basic information of benchmark function.

Name	Function	D	Range	$f_{\min}$
BOHACHEVSKY01	$f_1(x)= x_1^2+2x_2^2-0.3\cos(3\pi x_1)-0.4\cos(4\pi x_2)+0.7$	2	[-50, 50]	0
BOHACHEVSKY02	$f_2(x)= x_1^2+2x_2^2-0.3\cos(3\pi x_1) \cos(4\pi x_2)+0.3$	2	[-50, 50]	0
BOHACHEVSKY03	$f_3(x)= x_1^2+2x_2^2-0.3\cos(3\pi x_1+4\pi x_2)+0.3$	2	[-50, 50]	0
BOOTH	$f_4(x)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$	2	[-50, 50]	0
CUBE	$f_5(x)=100(x_2-x_1^3)^2+(1- x_1)^2$	2	[-50, 50]	0
LEVY13	$f_6(x)=\sin^2(3\pi x_1)+(x_1-1)^2[1+\sin^2(3\pi x_2)] +(x_2-1)^2[1+\sin^2(2\pi x_2)]$	2	[-50, 50]	0
MATYAS	$f_7(x)=0.26(x_1^2+x_2^2) -0.48x_1x_2$	2	[-50, 50]	0
THREE-HUMP	$f_8(x)=2 x_1^2-1.05 x_1^4+ x_1^6/6+x_1x_2+ x_2^2$	2	[-50, 50]	0
CHUNG REYNOLDS	$f_9(x)=\left(\sum_{i=1}^{D} x_i^2\right)^2$	2/50/100/300	[-10, 10]	0
CSENDES	$f_{10}(x)=\sum_{i=1}^{D} x_i^6\left(2+\sin 1/x_i\right)$	2/50/100/300	[-10, 10]	0
ROSENBROCK	$f_{11}(x)=\sum_{i=1}^{D-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	2/50/100/300	[-10, 10]	0
SCHUMER STEIGLITZ	$f_{12}(x)=\sum_{i=1}^{D} x_i^4$	2/50/100/300	[-10, 10]	0
SPHERE	$f_{13}(x)=\sum_{i=1}^{D} x_i^2$	2/50/100/300	[-10, 10]	0
SUM OF DIFFERENT POWERS	$f_{14}(x)=\sum_{i=1}^{D} \lvert x_i\rvert^{i+1}$	2/50/100/300	[-10, 10]	0

Finally, FSA will check the GS and LS due to the updating of initial, and will update them if there are better solutions than the GS and LS.

The FSA iterative process can be presented as follows:

Step 1. Randomly defines the initial population size, defines the objective function and its searching space. Set the maximum number of iterations *Max*. Set $t = 1$. Set the lower limit bounds Lb and the upper limit bounds Ub. Calculate the initial global solution *GS,* the initial local solution *LS.* Initialize by Equation (1).

Step 2. The search in each country depends on the LS in Equation (2), the search in the overall world depends on the GS in Equation (3). After computing the local and the global convergences, the solution of each person is defined in Equation (4).

Step 3. Compare their fitness values of all feasible solutions to determine the global solution *GS* and the local solution *LS* in the current generation. After comparing the current *GS* and *LS* with the

previous *GS* and previous *LS*, update the *GS* and *LS* if there is a better solution. Update the random initial of Equation (5)

Step 4. Calculate $t = t + 1$. Judge whether t equals to *Max*. If no, return to step 2. Otherwise, Stop and output results.

III. THE PROPOSED ALGORITHM

FSA directly updates the random initial uses the local search between people and the global search between the histories optimal persons. But it ignores peripheral feasible solutions in the local search and the global search. This paper use Sine Cosine Algorithm (SCA) [46] searching method to loop-progressive find the best solution. The Sine Cosine Algorithm is proposed by Seyedali Mirjalili in 2016. Sine and cosine are mathematical terms and a kind of trigonometric function. In the right triangle, the opposite side ratio of the acute angle ∠A to the hypotenuse is called the sine of ∠A, the ratio of the adjacent side of the acute angle ∠A to the hypotenuse is called the cosine of ∠A. The trigonometric function is a kind of function which belongs to transcendental function of elementary function in mathematics. Their essence is the mapping between the set of arbitrary angles and the variables of a set of ratios. Generally, trigonometric functions are defined in the plane rectangular coordinate system, and their definition domain is the whole real number domain.This paper adds searching steps in Equation (6) and Equation (7).

$$S(i,:) = S(i,:) + r_1 \times \sin(r_2) \times |r_3 \times GS - S(i,:)| \quad r_4 < 0.5 \tag{6}$$

$$S(i,:) = S(i,:) + r_1 \times \cos(r_2) \times \times |r_3 \times GS - S(i,:)| \quad r_4 \geq 0.5 \tag{7}$$

$$r_1 = a - a\frac{t}{Max} \tag{8}$$

where r_2 is a random number in $[0\ 2\pi]$. r_3 is the disturbance weight, this paper is to have a more accurate positioning, sets $r_3 = 1$. r_4 is a random number in [0,1].

The parameter r_1 sets the next searching position which could be either in the searching scope between the solution

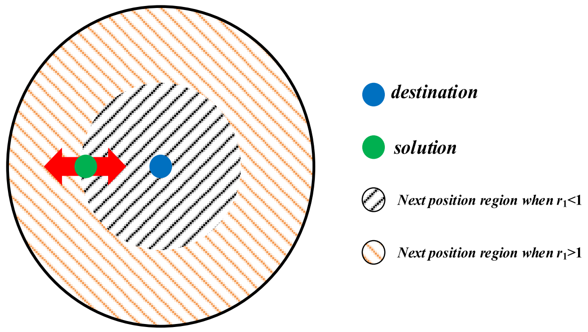


FIGURE 1. The effects of Sine and Cosine.s.

in destination or outside. The parameter r_2 sets the moving distance which should be towards or outwards in the searching scope. The parameter r_4 set the weight between the sine and cosine components in Equation (6) and Equation (7). The effects of Sine and Cosine in Equation (6) and Equation (7) are illustrated in Figure 1 [46].

Figure 1 shows the moving trend of the two dimension feasible solution in the searching space. If r_1 is bigger than 1, the feasible solution shows the outward expansion trend. If r_1 is less than 1, the feasible solution shows the inward expansion trend. The sine and cosine calculation pattern allows a feasible solution to be re-positioned around another feasible solution, which can ensure algorithm exploitation ability of the space defined between two solutions.

The FSASCA main step can be summarized in the pseudocode shown in Algorithm1.

Algorithm 1 FSASCA

Input: *Function f.Searching range. Set Max,a, Lb,Ub. Set t = 1. Population size N. Initial global solution GS. Initial local solution LS. . Initial global function GF.*

Output: *GS.*

For 1 *i=1:N*
$S(i,:)=Lb+(Ub\text{-}Lb)*rand$
Record initial global solution GS, initial local solution LS, initial global function GF.
End For 1
While *(t<Max)*
  For 2 *in =1:N*
    $S(i,:)_L=(LS(i,:)\text{-}S(i,:))*rand$
    $S(i,:)_G=(GS\text{-}S(i,:))*rand$
    $S(i,:)=S(i,:)+S(i,:)_L+S(i,:)_G$
    If 1 $f\ (S(i,:))$. is better than $f\ (GS)$.
      $GS=S(i,:)$
      $GF=f\ (S(i,:))$.
    End If 1
    Update LS.
    $S(i,:)=GS+(GS\text{-}LS(i,:))*rand$
    $r_1=a\text{-}at/Max$, $r_2\in[0\ 2\pi]$, $r_3=1$, $r_4\in[0\ 1]$
    If 2 $r_4<0.5$.
      $S(i,:)=S(i,:)+r_1\times\sin(r_2)\times|r_3\times GS\text{-}S(i,:)|$
    Else
      $S(i,:)=S(i,:)+r_1\times\cos(r_2)\times|r_3\times GS\text{-}S(i,:)|$
    End If 2
    Check the GS and LS due to the updating of initial
  End For 2
$t=t+1$
End While

IV. FUNCTION EXPERIMENTS

A. TESTING ENVIRONMENTS

The benchmark function is a set of functions to test different algorithm performances. To display different abilities of the proposed algorithm more exactly and completely, this paper chooses 14 benchmark functions which are widely appiled in algorithm fields to analysis the proposed algorithm. Therefore, algorithm performances can be got by 14 testing functions which can objectively reflect the algorithm optimization ability. These functions can be divided into low dimension functions and variable dimension functions.

Low dimension functions usually do not own local optimal values, so it is easy to find the feasible solution for algorithms. Variable dimension functions have many local extreme points which have strong oscillation and non-convexity, so the problem solving in variable dimension functions is deceptive, which makes the algorithm more difficult to the feasible solution. Benchmark functions are showed in Table 1. In Table 1, D is the searching dimension, Range is the searching scope of the algorithm independent variable, is the searching scope of the function solution, so it starts from negative. $f_{\min}$ is the ideal function value. In the original FSA iterature, the author compare FSA with CS, FPA, and SA, so this paper test different algorithms including earlier algorithms and latest algorithms, earlier algorithms include the cuckoo search algorithm (CS) [47], the flower pollination algorithm (FPA) [48], and the simulated annealing algorithm (SA) [49]. Latest algorithms include the crow search algorithm (CSA) [7], the giza pyramids construction (GPC) [50], the jellyfish search (JS) optimizer [25], the sine cosine algorithm (SCA) [46], and the spotted hyena optimizer (SHO) [19]. All algorithm initial parameters in this paper were selected by original algorithm literatures. All algorithm searching process can be found in original algorithm literatures. CS and FPA were proposed by Xin-She Yang. Cuckoos reproduces their eggs in other nests, and remove host eggs to enhance incubation probability. CS was encouraged by the cuckoos hatch mechanism. In CS, the discovery probability and the step was equal to 0.25. FPA has the biotic searching step and the cross-pollination searching step. Two searching steps can be seen as the pollination action with pollen-carrying pollinators having the lévy flight trajectory. FPA has two parameters p and β, $p=0.8$, $\beta=1.5$. SA proposed in the early 1980s was the physical annealing process algorithm. The SA principle is to make the solid metal get a large temperature, and make metal atoms have some random states. SA has two parameters T and k, $T=100$, $k=0.95$. CSA,which was proposed by Alireza Askarzadeh in 2016, is a Swarm intelligence algorithm which is based on crows storing their excess food in hiding places

TABLE 2. Comparison of results for 2 dimension functions.

Function	Index	FSASCA	FSA	CS	FPA	SA	CSA	GPC	JS	SCA	SHO
f_1	Min	0	0	0.3829	0.0958	0.0001	0	0	0	0	0
	Max	0.4129	19.6317	2.3038	0.4686	0.0070	9.4369E-15	0	0	0	0
	Std	0.1306	7.9121	0.6138	0.1446	0.0027	2.9645E-15	0	0	0	0
f_2	Min	0	0	0.2634	0.0545	0.0010	0	0	0	0	0
	Max	0	0.2475	7.4197	0.3042	0.0048	6.0507E-15	0	0	0	0
	Std	0	0.1183	2.1241	0.0864	0.0015	1.8699E-15	0	0	0	0
f_3	Min	2.4498E-12	1.0051E-05	4.7309E-01	1.0091E-01	3.6190E-05	5.5511E-17	0	0	0	0
	Max	0.1330	0.5783	3.2922	0.2323	0.0033	3.3862E-15	0	0	0	0
	Std	0.0431	0.2121	0.9917	0.0505	0.0011	1.0236E-15	0	0	0	0
f_4	Min	1.0687E-05	2.5374E-05	0.0044	0.0595	0.0001	8.5729E-20	3.9101E-05	0	0.0004	0.1428
	Max	0.0492	1.5942	6.2584	0.2096	0.0018	5.9533E-16	0.0092	0	0.0113	11.4735
	Std	0.0183	0.5019	1.8997	0.0534	0.0006	1.9328E-16	0.0028	0	0.0040	3.2276
f_5	Min	0.0007	0.0152	0.0356	0.0282	0.0009	0.0040	0.0040	1.46923E-06	0.0058	0.0002
	Max	18.0055	20.5089	27.4343	1.6595	0.0841	23.8004	0.0888	4.8150	1.0012	0.0030
	Std	7.4558	7.1953	8.1087	0.5460	0.0357	8.2427	0.0303	1.5550	0.3264	0.0010
f_6	Min	1.3498E-31	1.3498E-31	0.4582	0.0981	0.0021	3.4088E-18	0.0005	1.3498E-31	3.9254E-05	2.7105E-06
	Max	0.1101	4.2581	3.1541	0.4495	0.0496	2.2857E-14	0.0553	1.3498E-31	0.0055	0.3347
	Std	0.0348	1.3347	0.8291	0.1079	0.0146	7.9687E-15	0.0168	0	0.0017	0.1058
f_7	Min	0	0	0.0082	0.0004	3.6272E-08	2.7500E-20	2.5099E-46	8.9761E-88	3.0398E-68	0
	Max	0	5.0265	0.2331	0.0105	8.6600E-05	6.7527E-17	1.0338E-30	8.5452E-81	1.1069E-46	0
	Std	0	1.5942	0.0654	0.0034	2.3816E-05	2.5182E-17	3.2581E-31	2.6770E-81	3.4839E-47	0
f_8	Min	0	6.4697E-37	0.0194	0.0039	0.0002	8.2575E-19	9.1467E-34	6.9102E-122	5.6357E-71	0
	Max	0.2987	0.3229	0.7211	0.0928	0.0023	2.1556E-15	4.2518E-28	5.0028E-110	1.0350E-58	0
	Std	0.1574	0.1492	0.2158	0.0358	0.0007	6.6708E-16	1.3341E-28	1.9581E-110	3.2711E-59	0
$f_{9(D=2)}$	Min	0	0	8.4040E-07	8.7785E-08	2.2711E-10	4.5699E-39	6.0794E-97	9.2844E-246	4.6121E-134	0
	Max	6.6247E-119	5.2397E-11	5.1167E-03	3.7534E-06	9.8869E-08	4.4328E-36	4.6207E-60	3.6904E-217	2.0648E-111	0
	Std	2.0949E-119	1.6569E-11	1.5895E-03	1.1704E-06	3.9318E-08	1.6587E-36	1.4592E-60	0	6.5290E-112	0
$f_{10(D=2)}$	Min	0	5.0588E-215	8.0631E-11	4.9148E-11	1.6975E-14	9.0704E-61	3.2600E-124	0	9.8277E-217	3.6477E-313
	Max	0	13.4481	1.1733E-05	3.5425E-08	1.2992E-11	2.8262E-54	2.2123E-95	5.5635E-302	1.2633E-181	1.6543E-196
	Std	0	4.2527	4.8245E-06	1.0844E-08	4.1403E-12	8.8543E-55	6.9826E-96	0	0	0
$f_{11(D=2)}$	Min	2.6801E-09	1.0563E-05	0.0756	0.0035	0.0002	7.7008E-18	0.0135	1.1440E-19	2.0190E-05	1.9854E-06
	Max	8.1360	7.4652	1.0588	0.1160	0.0038	1.3663E-15	0.3090	7.4083E-13	0.0134	0.0001
	Std	2.6799	2.7422	0.3070	0.0377	0.0011	4.5425E-16	0.0944	2.3426E-13	0.0040	3.4114E-05
$f_{12(D=2)}$	Min	0	3.7236E-181	5.4247E-06	3.9894E-08	5.7156E-11	1.1608E-41	8.3169E-80	2.0847E-268	1.5500E-151	0
	Max	0	9.9512E-05	2.8490E-03	2.5887E-05	1.8179E-07	7.5012E-34	1.7163E-57	1.8145E-232	1.1931E-125	0
	Std	0	3.1468E-05	9.4749E-04	8.3433E-06	5.5013E-08	2.3701E-34	5.4274E-58	0	3.7656E-126	0
$f_{13(D=2)}$	Min	0	1.7009E-72	8.0163E-03	1.7703E-06	9.4267E-06	2.7327E-21	5.9128E-41	2.8545E-126	2.7583E-74	0
	Max	0	0.0262	0.0551	0.0035	0.0005	1.4773E-17	1.8834E-30	6.0623E-118	1.2552E-62	0
	Std	0	0.0082	0.0152	0.0014	0.0002	4.7718E-18	5.8215E-31	1.9076E-118	3.9694E-63	0
$f_{14(D=2)}$	Min	0	0	8.9575E-06	8.9012E-05	1.4915E-06	3.5334E-22	2.9366E-50	1.0964E-156	5.3567E-86	0
	Max	0	0.9752	0.0219	0.0004	7.2093E-05	1.2576E-19	1.9760E-31	6.9302E-148	2.2669E-72	0
	Std	0	0.3078	0.0071	0.0001	2.7648E-05	3.7900E-20	6.2482E-32	2.1915E-148	7.1664E-73	0

and retrieving it.In CSA, fl leads to the local searching and large values results in global searching, $fl = 2$. GPC, which was mainly proposed in 2020, is inspired by the ancient past having good metaheuristic algorithm characteristics to deal with many issues. JS, which was mainly proposed by Jui-Sheng Chou and Dinh-Nhat Truong in 2020.is inspired by the behavior of jellyfish involving their following the ocean current, $\gamma = 0.1$ SCA, which was mainly proposed by Seyedali Mirjalil, is inspired by searching the best solution using a mathematical model based on sine and cosine function. SHO, which was mainly proposed by Gaurav Dhiman and Vijay Kumar in 2017, is the social relationship

TABLE 3. Comparison of results for 50/100/300 dimension functions.

Function	Index	FSASCA	FSA	CS	FPA	SA	CSA	GPC	JS	SCA	SHO
$f_{9(D=50)}$	Min	0	0	3.5203E+05	2.4964E+05	1.8100E+03	3.3255	2.3198E-48	5.7152E-63	0.6307	0
	Max	24.4598	1.0396E+05	8.1431E+05	6.3220E+05	3.2678E+03	19.5395	1.5241E-39	5.4834E-05	3.5257E+03	0
	Std	7.7334	3.2757E+04	1.4041E+05	1.4288E+05	4.3937E+02	4.7199	4.7182E-40	1.7340E-05	1.0529E+03	0
$f_{10(D=50)}$	Min	0	2.3094E-47	6.1301E+05	1.2191E+06	2.4229E+03	0.1740	5.4315E-67	8.9671E-99	3.1505E+03	1.4007E+04
	Max	3.1810E-30	7.3542E+03	4.8920E+06	3.4868E+06	9.4482E+03	4.0815	2.0910E-58	1.6333E-22	4.0998E+05	1.0018E+07
	Std	1.0059E-30	2.5081E+03	1.3690E+06	7.8390E+05	2.1415E+03	1.3876	6.6053E-59	6.0804E-23	1.5448E+05	3.1534E+06
$f_{11(D=50)}$	Min	0	0	2.0266E+06	1.2926E+06	2.1042E+04	2.4291E+02	48.2832	0.0033	8.0153E+03	48.5806
	Max	48.5042	53.9814	3.8683E+06	2.4732E+06	4.0253E+04	5.7119E+02	48.8489	0.4089	2.2384E+05	48.9822
	Std	25.5366	25.5733	6.6144E+05	4.5013E+05	6.3355E+03	96.7163	0.1752	0.1209	7.8265E+04	0.1572
$f_{12(D=50)}$	Min	0	1.6348E-106	1.4258E+04	1.1189E+04	1.3714E+02	0.1600	2.6784E-50	5.9966E-66	2.7008E+02	0
	Max	5.0616E-18	6.3465E+02	4.1227E+04	4.1981E+04	3.5052E+02	1.4364	4.1339E-40	1.6778E-42	4.8480E+03	0
	Std	1.6006E-18	1.9831E+02	8.2693E+03	9.3691E+03	85.6151	0.4305	1.3035E-40	5.3056E-43	1.4359E+03	0
$f_{13(D=50)}$	Min	0	0	6.3737E+02	4.6483E+02	44.5984	1.3886	1.3283E-22	7.1325E-34	0.0381	0
	Max	2.0552E-28	1.3336E+02	9.2006E+02	7.3904E+02	56.8124	3.7975	7.7425E-20	1.4991E-13	59.9973	0
	Std	6.4993E-29	41.6650	88.8625	89.9085	3.8454	0.7056	2.5633E-20	4.7403E-14	20.6883	0
$f_{14(D=50)}$	Min	0	0	1.3426E+28	1.4927E+31	1.7235E+16	6.1100	4.8965E-39	1.4794E-92	3.3622E+07	0
	Max	5.5345E+11	5.4430E+24	6.2408E+36	9.9388E+33	4.8586E+20	3.9686E+02	3.9847E-29	6.0258E-77	2.4532E+17	0
	Std	1.7502E+11	1.7212E+24	2.0025E+36	3.0489E+33	1.5714E+20	1.2322E+02	1.2582E-29	1.9050E-77	7.7222E+16	0
$f_{9(D=100)}$	Min	0	2.5173E-171	1.8584E+06	1.0338E+06	1.2221E+05	1.3884E+02	6.4978E-44	1.5105E-46	2.4518E+02	0
	Max	19.5416	2.2231E+03	4.1006E+06	2.3383E+06	1.8539E+05	3.0204E+02	5.8181E-38	9.3351E-06	6.4441E+04	0
	Std	6.1796	6.8873E+02	7.0668E+05	4.9284E+05	2.1611E+04	60.6630	1.8414E-38	2.9448E-06	1.9847E+04	0
$f_{10(D=100)}$	Min	0	4.0125E-110	7.4376E+06	1.2493E+06	3.7194E+05	5.7495	2.8087E-60	1.9240E-98	1.4821E+06	3.5922E+03
	Max	0	2.2566E+05	1.2749E+07	5.6442E+06	7.9566E+05	16.3899	1.1634E-52	2.4294E-27	7.3410E+06	2.1480E+07
	Std	0	7.3009E+04	1.3241E+06	1.6615E+06	1.3627E+05	4.2938	3.6418E-53	7.6826E-28	1.8279E+06	7.9789E+06
$f_{11(D=100)}$	Min	0	0	5.1736E+06	3.8229E+06	5.5700E+05	1.8671E+03	98.7615	0.0801	7.9600E+05	98.0038
	Max	97.9996	1.3790E+03	1.0455E+07	7.0102E+06	7.9545E+05	3.6533E+03	98.8319	5.4257	2.6172E+06	98.9816
	Std	50.5865	4.2512E+02	1.6916E+06	1.2302E+06	8.1298E+04	5.6581E+02	0.0197	1.6215	6.6706E+05	0.3045
$f_{12(D=100)}$	Min	0	5.1764E-174	3.6663E+04	3.1151E+04	2.8173E+03	5.0706	4.8440E-39	2.8671E-68	1.0134E+04	0
	Max	1.5165E-61	2.5121E+02	9.6735E+04	6.7479E+04	8.2860E+03	10.9955	3.6404E-35	1.1830E-40	3.1454E+04	0
	Std	4.7955E-62	78.9992	1.9326E+04	1.3267E+04	1.6560E+03	2.0069	1.1480E-35	3.7411E-41	6.8979E+03	0
$f_{13(D=100)}$	Min	0	1.4351E-190	1.3963E+03	1.0327E+03	3.2918E+02	11.0415	1.0675E-21	7.7031E-26	31.2038	0
	Max	0.3388	78.1688	1.9094E+03	1.6662E+03	4.6432E+02	17.3363	1.6668E-19	0.0069	2.3479E+02	0
	Std	0.1071	25.6024	1.5119E+02	2.1231E+02	45.0782	1.7987	5.1211E-20	0.0022	61.8621	0
$f_{14(D=100)}$	Min	0	0	1.6526E+74	7.3763E+69	2.2945E+55	4.4343E+05	4.3146E-53	1.7467E-95	4.5390E+42	0
	Max	5.3948E+25	7.2339E+42	1.1779E+80	5.1152E+74	7.6622E+58	1.1086E+14	6.9598E-30	1.2221E-76	4.9131E+59	0
	Std	1.7060E+25	2.2734E+42	3.7277E+79	2.1200E+74	2.7934E+58	3.4965E+13	2.1872E-30	3.8648E-77	1.5782E+59	0
$f_{9(D=300)}$	Min	0	9.8795E-146	2.2664E+07	1.2370E+07	1.5027E+07	3.7624E+03	1.2825E-42	1.0417E-54	1.4621E+05	0
	Max	0.0071	5.5483E+05	4.3227E+07	2.5257E+07	1.9860E+07	6.0717E+03	8.5687E-35	2.5568E-04	1.7549E+06	0
	Std	0.0022	1.7323E+05	5.7470E+06	4.6594E+06	1.5756E+06	7.2448E+02	2.6672E-35	8.0853E-05	5.1811E+05	0
$f_{10(D=300)}$	Min	0	8.0482E-122	1.6682E+07	1.1147E+07	7.4394E+07	1.0535E+02	3.3872E-57	5.9044E-101	1.6903E+07	5.3115E+03
	Max	7.3237E+02	2.2492E+05	3.9218E+07	2.4382E+07	9.5810E+07	1.7268E+02	1.0240E-51	5.3399E-07	2.9852E+07	7.5853E+07
	Std	2.3154E+02	8.5201E+04	6.3799E+06	4.5910E+06	6.3344E+06	21.0109	3.2485E-52	1.6886E-07	4.4162E+06	3.1189E+07
$f_{11(D=300)}$	Min	0	0	3.4654E+07	1.5705E+07	1.9385E+07	1.1398E+04	2.9860E+02	0.2533	1.0434E+07	2.9772E+02
	Max	2.9598E+02	7.9014E+05	4.8430E+07	1.9068E+07	2.3245E+07	1.5065E+04	2.9882E+02	4.1017	1.8662E+07	2.9898E+02
	Std	1.4297E+02	2.4952E+05	3.9304E+06	1.3014E+06	1.2461E+06	1.2063E+03	6.3664E-02	1.3492	2.9968E+06	0.4384
$f_{12(D=300)}$	Min	0	1.1138E-304	2.3404E+05	9.5850E+04	1.9801E+05	52.2306	8.9428E-40	4.1104E-64	7.5796E+04	0
	Max	1.4786E-18	3.2772E+03	2.9514E+05	2.3827E+05	2.2548E+05	85.0609	1.8575E-35	1.7014E-13	2.0527E+05	0
	Std	4.6757E-19	1.3409E+03	1.8298E+04	4.8173E+04	1.1414E+04	9.1422	5.9828E-36	5.3784E-14	4.3362E+04	0
$f_{13(D=300)}$	Min	2.7579E-289	0	4.7696E+03	2.9787E+03	3.8186E+03	75.8091	2.1340E-19	4.7925E-25	2.0687E+02	0
	Max	6.2163E-36	4.9370E+02	6.2478E+03	5.1687E+03	4.2906E+03	89.7034	8.3921E-18	0.3849	2.3808E+03	0
	Std	1.9753E-36	1.9996E+02	5.1484E+02	7.2238E+02	1.4706E+02	4.2401	2.3843E-18	0.1217	6.1847E+02	0
$f_{14(D=300)}$	Min	0	0	6.6308E+248	3.5746E+226	7.0862E+220	5.5170E+43	7.5688E-41	1.4847E-93	1.1930E+218	0
	Max	2.7867E+153	4.8137E+178	4.2983E+263	1.3961E+261	6.4542E+237	1.1215E+70	1.8446E-30	6.6344E-82	2.4667E+239	0
	Std	8.8122E+152	NO	NO	NO	NO	3.5464E+69	5.7599E-31	2.0932E-82	NO	0

between spotted hyenas and collaborative behaviors, mainly steps include the prey searching, the encircling, and the attacking prey. In SHO, $rd1$ and $rd2$ are random vectors in the range of [0, 1]. The population size and the maximum number of iterations were 20 and 500 in this paper, and each algorithm was ran in 10 times. All algorithm experiment environment was in the Windows 7 operating system, Intel (R) Core (TM) i3-7100 CPU, RAM equals to 8GB.

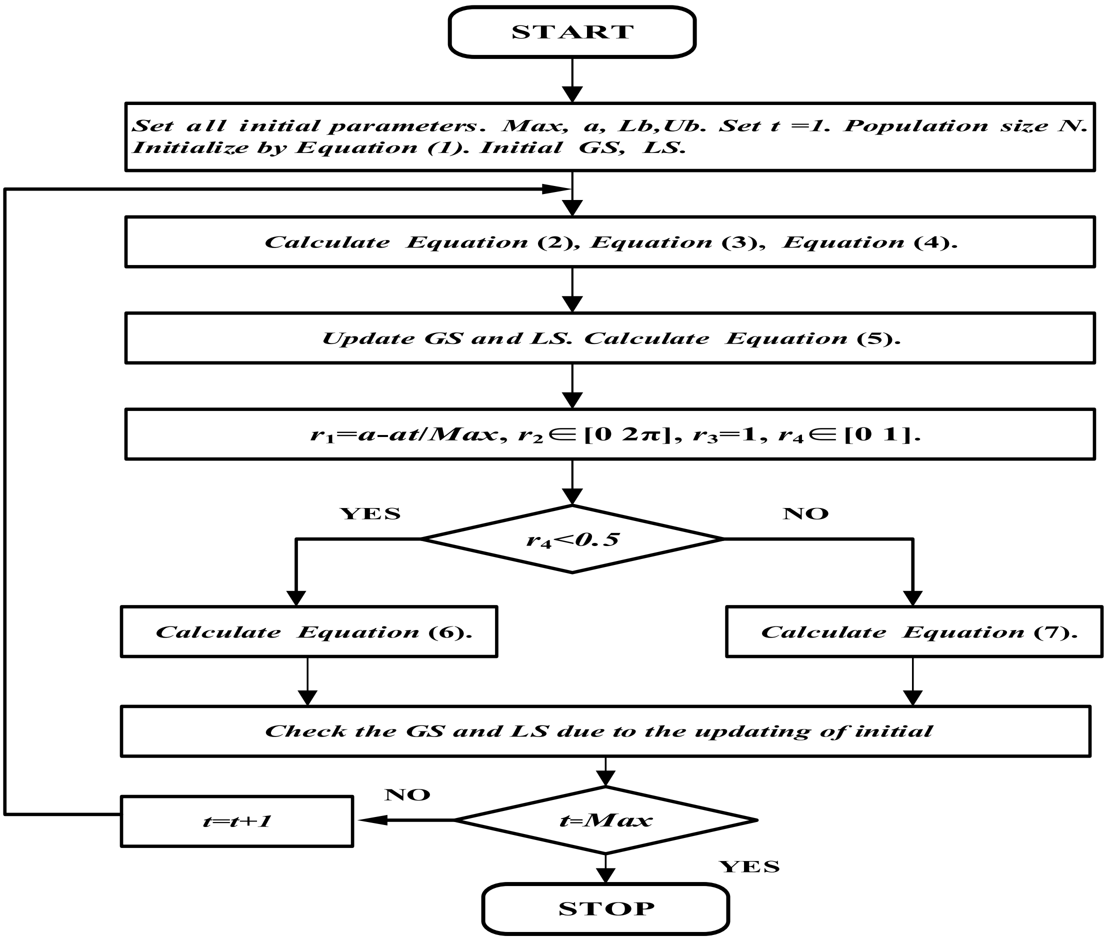


FIGURE 2. The flowchart for FSASCA.

All algorithm programs, data, and figures were completed in MATLA.

B. NUMERICAL CALCULATION DISCUSSIONS

To display different algorithm performances, this paper selects three calculation evaluation indexes. In Tables 2 and Tables 3, *Min*, *Max* and *Std* represent the minimum function value, the maximum function value, and standard deviation. Table 2 shows the two dimension functions calculation results. Tables 3 show 50/100/300 dimension functions calculation results. ***NO*** means that the computer cannot give the calculated value because of the large data. The standard deviation is most commonly used in the probability statistics as the measurement basis for the degree of statistical distribution, it can reflect the degree of dispersion among individuals in the group. For two dimension functions, FSASCA can get the ideal function values in f_1, f_2, f_7 to $f_{10(D=2)}$, $f_{12(D=2)}$ to $f_{14(D=2)}$. FSASCA can get the ideal values of three calculation evaluation indexes in f_2, f_7, $f_{10(D=2)}$, $f_{12(D=2)}$, $f_{13(D=2)}$, $f_{14(D=2)}$. FSA can get the ideal function values in f_1, f_2, f_7, $f_{9(D=2)}$, $f_{14(D=2)}$. CSA can get the ideal function values in f_1, f_2. GPC can get the ideal function values in f_1 to f_3. GPC can get the ideal values of three calculation evaluation indexes in f_1 to f_3. JS can get the ideal function values in f_1 to f_4, $f_{10(D=2)}$. JS can get the ideal values of three calculation evaluation indexes in f_1 to f_4. SCA can get the ideal function values in f_1 to f_3. SCA can get the ideal values of three calculation evaluation indexes in f_1 to f_3. SHO can get the ideal function values of three calculation evaluation in f_1 to f_3, f_7 to f_9, $f_{12(D=2)}$ to $f_{14(D=2)}$. For variable dimension functions (D=50/100/D=300), FSASCA can get all ideal function values except $f_{13(D=300)}$. FSA can get the ideal function values in $f_{9(D=50)}$, $f_{11(D=50)}$ to $f_{14(D=50)}$, $f_{11(D=100)}$, $f_{14(D=100)}$, $f_{11(D=300)}$, $f_{13(D=300)}$, $f_{14(D=300)}$. SHO can get the ideal function values in $f_{9(D=50)}$, $f_{12(D=50)}$, $f_{14(D=50)}$, $f_{9(D=100)}$, $f_{12(D=100)}$ to $f_{14(D=100)}$, $f_{9(D=300)}$, $f_{12(D=300)}$ to $f_{14(D=300)}$. Numerical calculation discussions show the FSASCA has a strong searching ability in solving different dimension.

C. ITERATION DISCUSSIONS

Iteration is a mathematic activity that repeats the feedback process, usually to approximate the desired goal or result. Each mathematic activity is called an iteration, and each iteration result will be applied as the initial value of the next iteration. In mathematics, iteration function is the object of in-depth study in algorithm. Figure 3 and

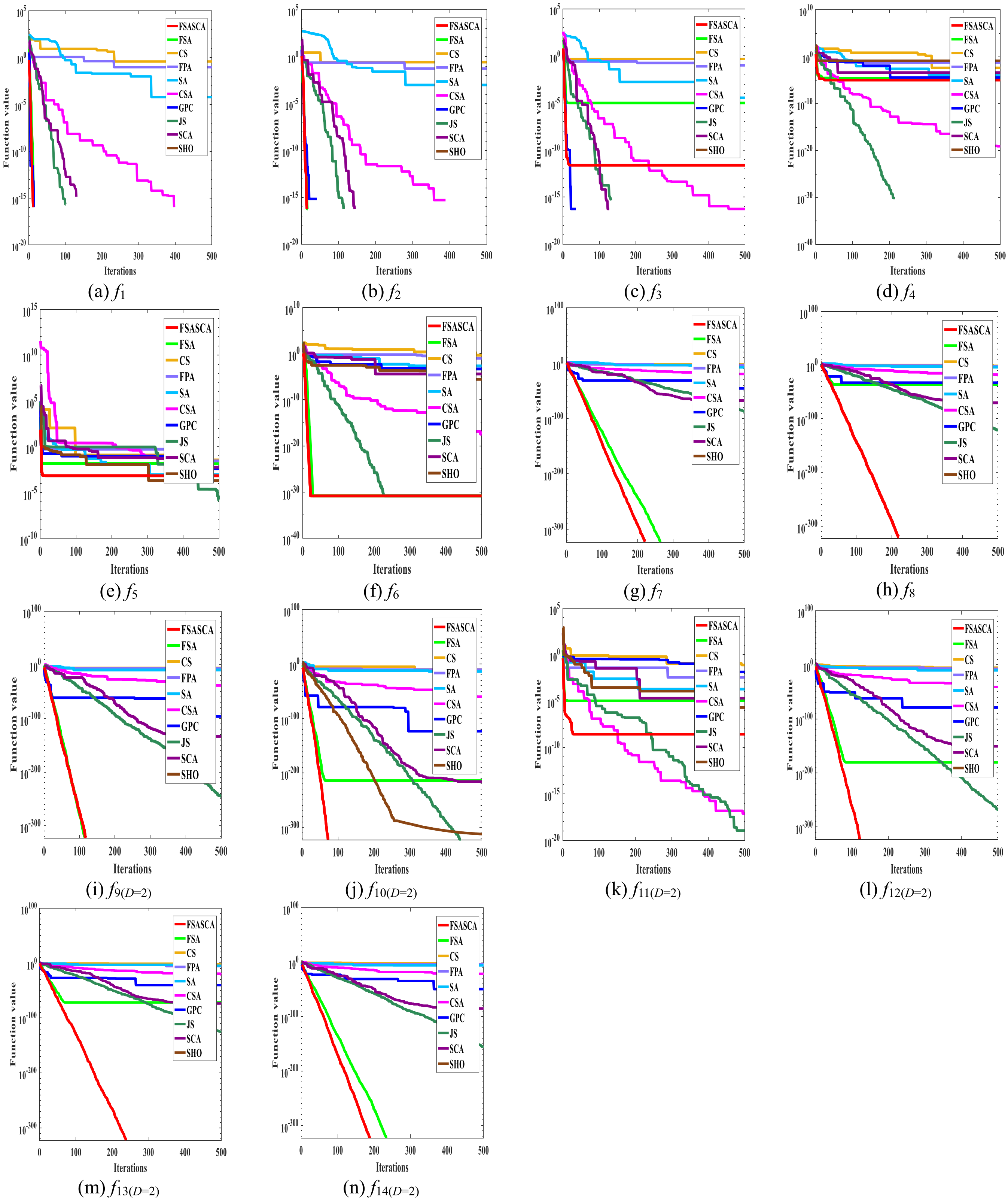


FIGURE 3. Convergence curves for functions (D=2).

Figure 4 show the optimal iteration curve of all algorithms handling different functions under 10 independent runs. To see the algorithm performance more intuitively, this paper uses the semilogarithmic coordinate system to show iteration figures. Based on the plane rectangular coordinate system, if there is only one logarithmic coordinate axis in

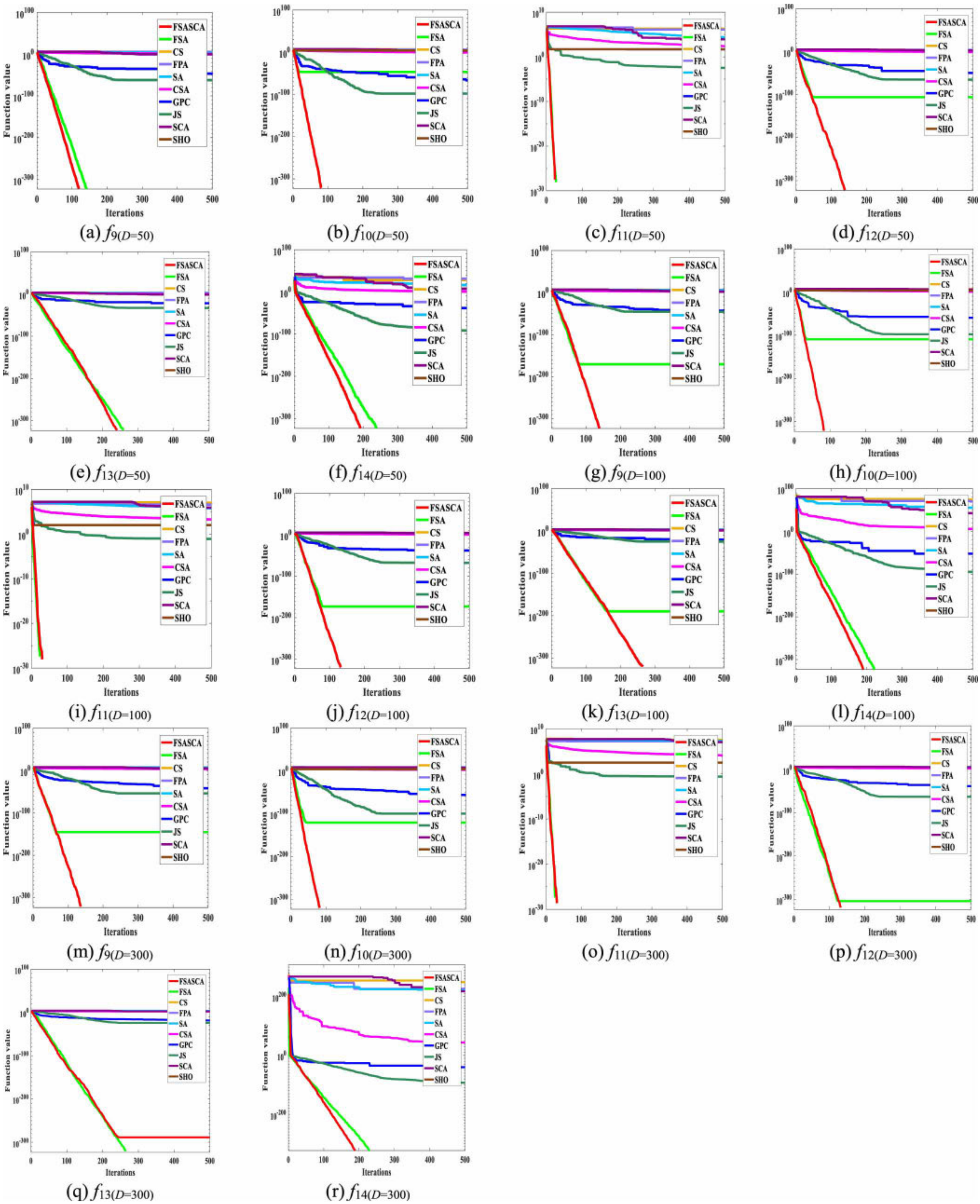


FIGURE 4. Convergence curves for functions (D=50/100/300).

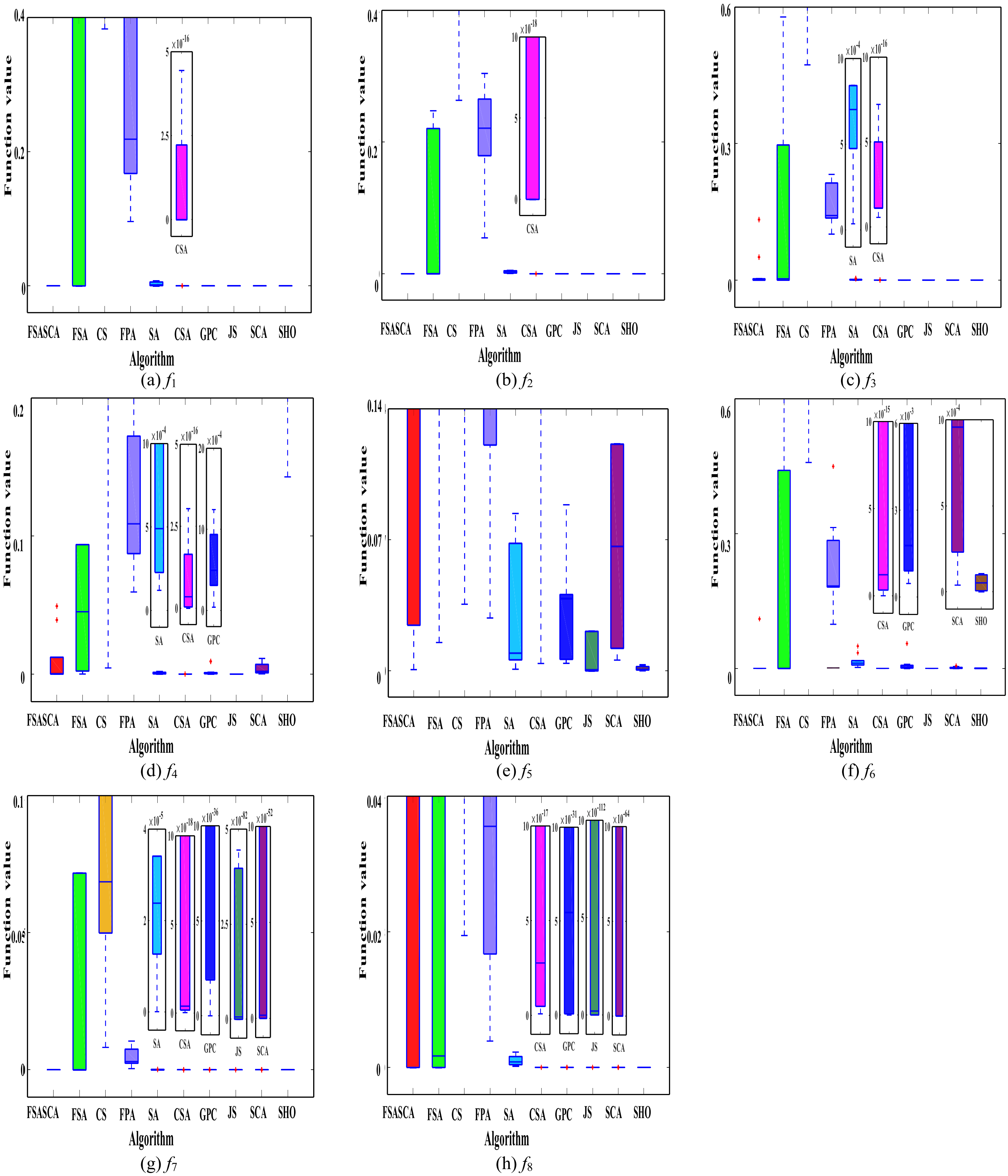
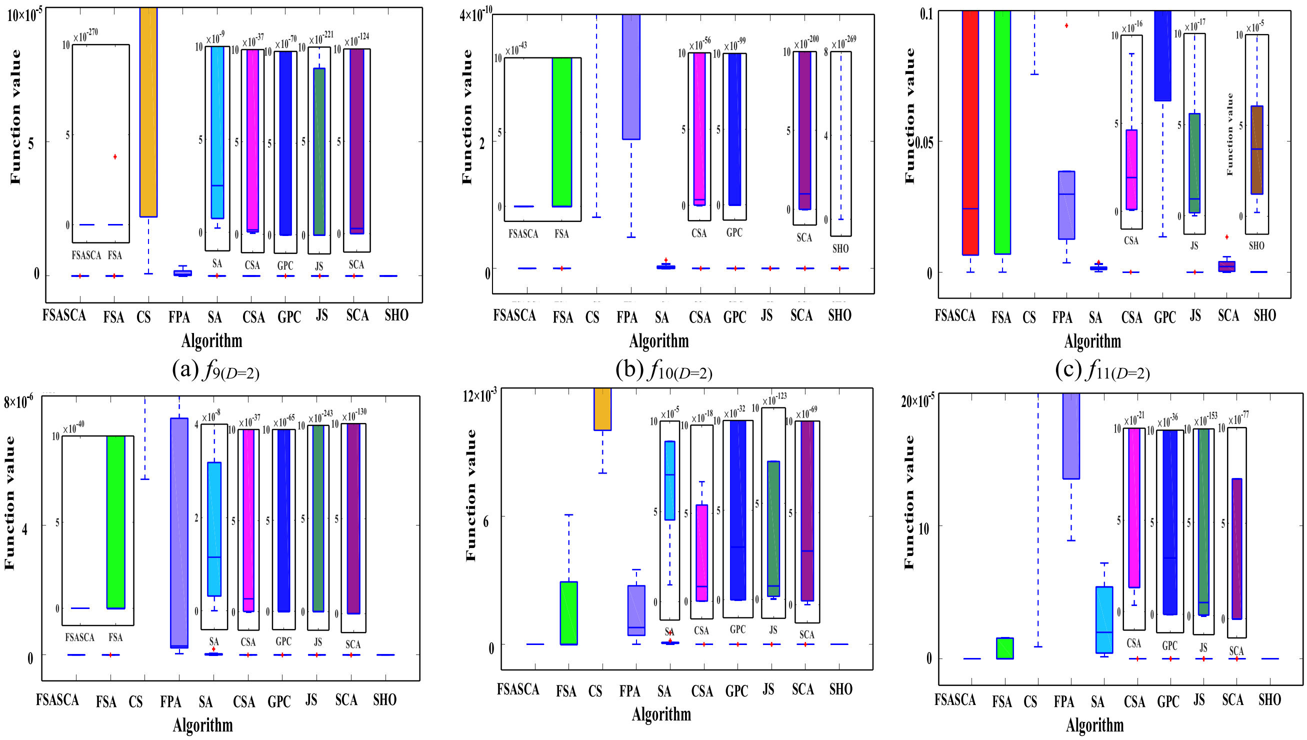


FIGURE 5. Box-plot charts of low dimension functions.

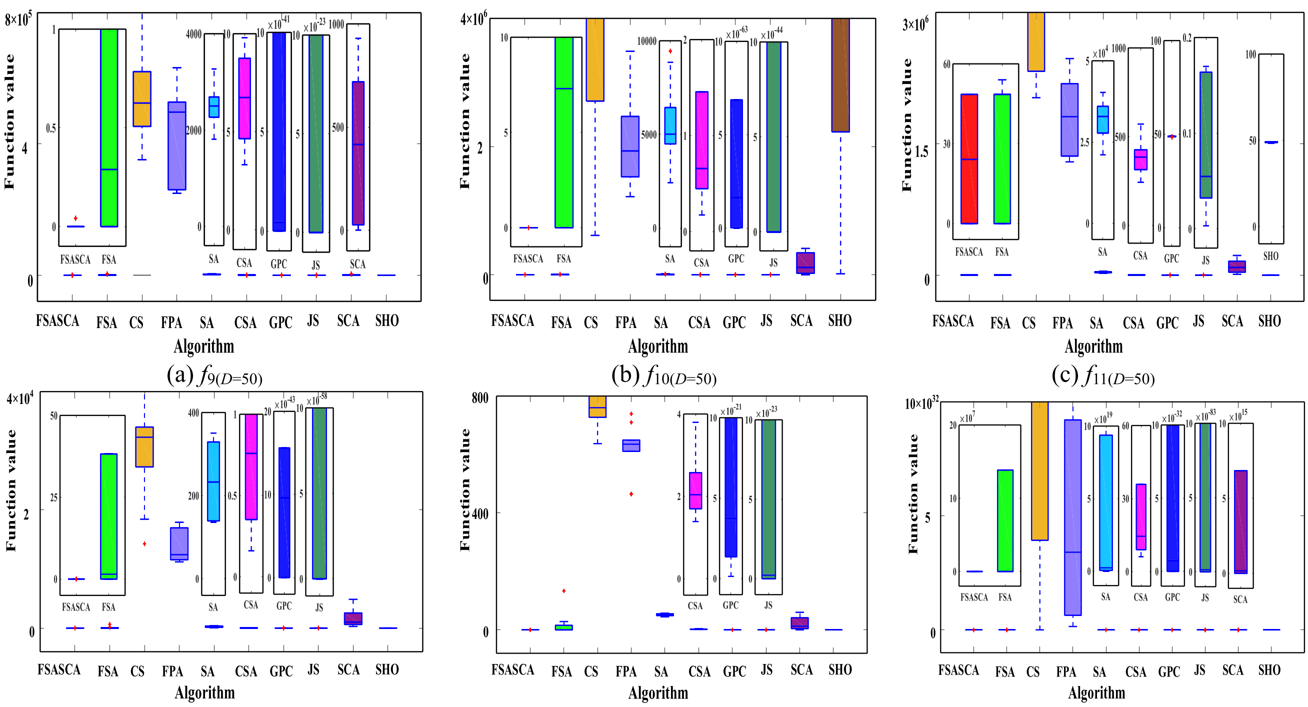
horizontal and vertical coordinates axes, the plane rectangular coordinate system means a semi logarithmic coordinate system. The logarithmic coordinate system refers to the position of each point corresponding to the logarithmic image in the two dimension rectangular coordinate system Figure 3 is two dimension function iteration curve ($D = 2$).



(a) $f_{9(D=2)}$ (b) $f_{10(D=2)}$ (c) $f_{11(D=2)}$

(d) $f_{12(D=2)}$ (e) $f_{13(D=2)}$ (f) $f_{14(D=2)}$

FIGURE 6. Box-plot charts of variable dimension functions (D=2).



(a) $f_{9(D=50)}$ (b) $f_{10(D=50)}$ (c) $f_{11(D=50)}$

(d) $f_{12(D=50)}$ (e) $f_{13(D=50)}$ (f) $f_{14(D=50)}$

FIGURE 7. Box-plot charts of variable dimension functions (D=50).

Figure 4 is high dimension function iteration curve ($D = 50/100/300$). Compared with different algorithms, FSASCA has the fastest iteration curve except $f_3, f_4, f_{13(D=300)}$, which displays that FSASCA can arrive the feasible solution more quickly. In f_3, GPC has the fastest iteration curve. In f_4, JS has the fastest iteration curve. In $f_{13(D=300)}$, FSA has the fastest iteration curve.Therefore, iteration curve results show that the proposed algorithm owns an eminent capability

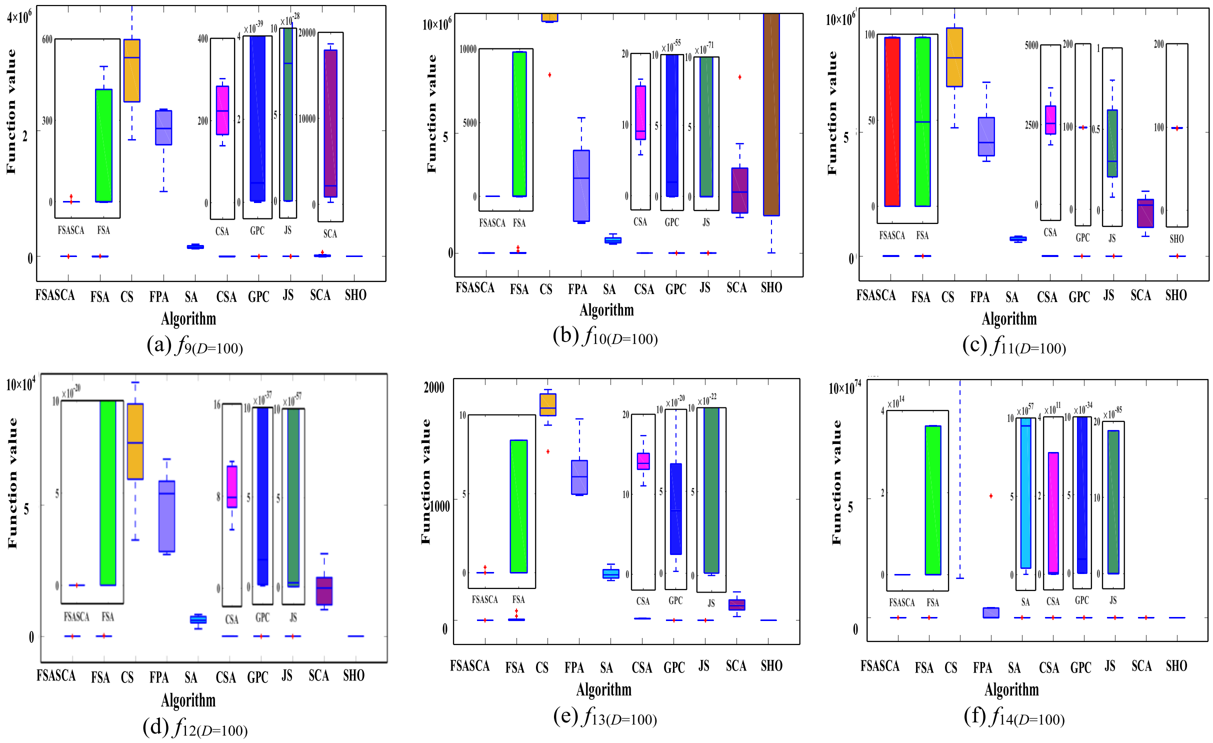


FIGURE 8. Box-plot charts of Variable dimension functions (D=100).

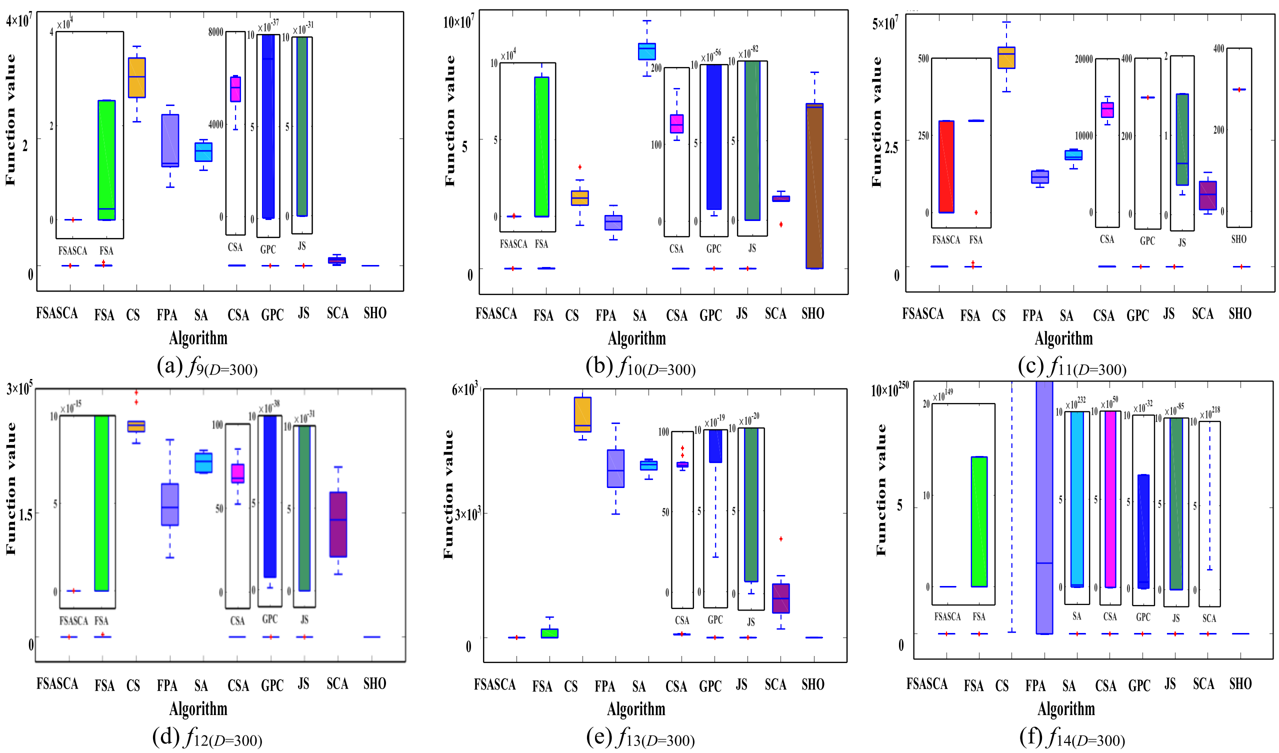


FIGURE 9. Box-plot charts of variable dimension functions (D=300).

to find feasible function solution areas and enhance the searching large drifting capability. FSASCA can adequately escape unstable searching areas and function subsidence point.

D. BOX PLOT DISCUSSIONS

The box plot is mainly used to show distribution characteristics of original data and multiple groups data. Box plot can find out the upper edge, lower edge, median and two quartiles

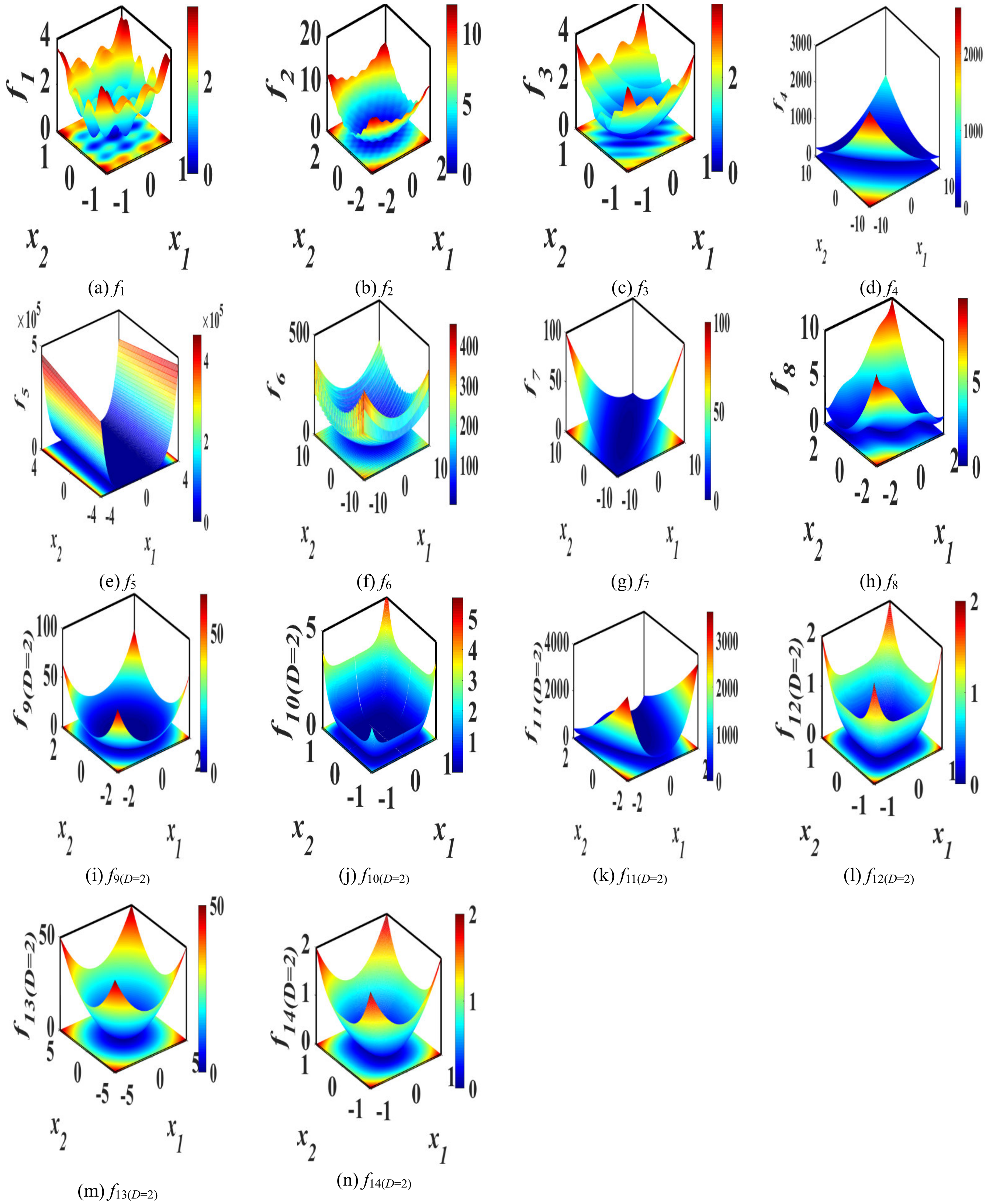


FIGURE 10. Three dimension graphical of benchmark functions.

of a group of data, the median is in the middle of the box plot, and two quartile points are connected in the box plot. The box plot was invented by the American statistician John Tukey in 1977, and it can show the maximum, minimum, median, and

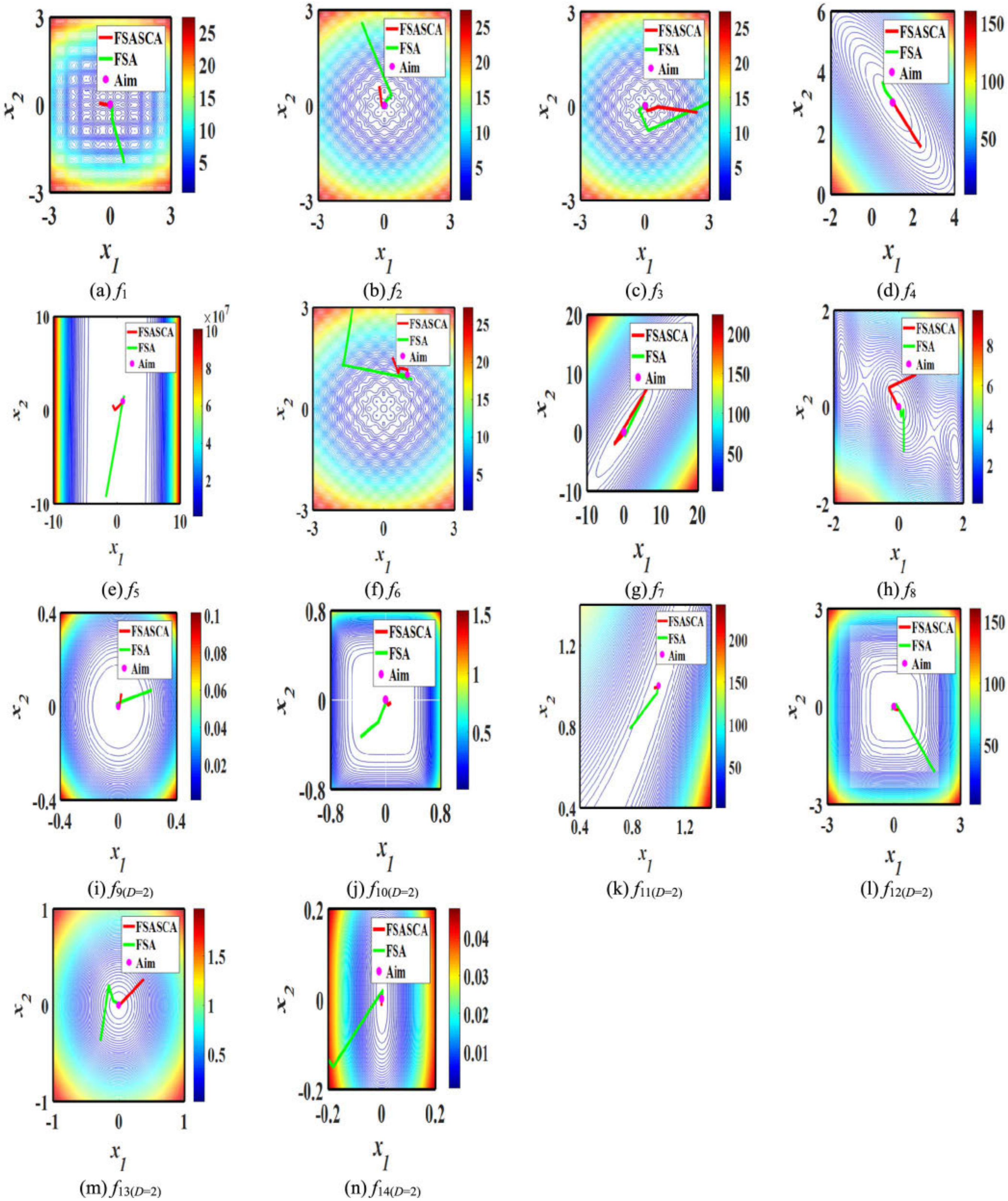


FIGURE 11. Algorithm searching paths.

upper and lower quartiles of a group of data. The narrower the boxplot, the more aggregated the data, the better the stability and accuracy of the algorithm If the boxplot is narrower, the data is more aggregated, the stability and accuracy of the

TABLE 4. Comparison of the Wilcoxon rank sum test.

Function	FSA	CS	FPA	SA	CSA	GPC	JS	SCA	SHO
f_1	0.006	1.212E-04	7.413E-04	1.745E-03	0.387	0.368	0.368	0.368	0.368
f_2	7.512E-04	6.386E-05	6.249E-05	6.386E-05	0.006	No	No	No	No
f_3	0.054	1.827E-04	3.231E-04	0.385	1.630E-04	6.386E-05	6.386E-05	6.386E-05	6.386E-05
f_4	0.045	4.396E-04	1.786E-04	0.791	1.827E-04	0.791	6.386E-05	0.345	1.827E-04
f_5	0.910	0.623	0.427	0.017	0.385	0.045	0.009	0.104	0.001
f_6	0.030	8.745E-05	1.628E-04	0.002	0.002	0.002	0.368	0.002	0.001
f_7	7.512E-04	6.386E-05	6.249E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05
f_8	0.269	0.137	1	1	1	1	1	1	0.015
$f_{9(D=2)}$	0.584	8.745E-05	8.512E-05	8.745E-05	8.745E-05	8.745E-05	0.002	0.001	0.368
$f_{10(D=2)}$	6.386E-05	6.386E-05	6.249E-05	6.386E-05	6.386E-05	6.386E-05	3.498E-02	6.386E-05	6.386E-05
$f_{11(D=2)}$	0.850	0.427	0.733	0.021	1.827E-04	0.571	1.827E-04	0.021	0.003
$f_{12(D=2)}$	6.386E-05	6.386E-05	6.249E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	No
$f_{13(D=2)}$	6.386E-05	6.386E-05	6.249E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	No
$f_{14(D=2)}$	0.015	6.386E-05	6.203E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	No
$f_{9(D=50)}$	0.025	1.727E-04	1.697E-04	1.727E-04	0.003	0.053	0.044	3.131E-04	0.006
$f_{10(D=50)}$	2.028E-04	1.494E-04	1.468E-04	1.494E-04	1.494E-04	0.002	0.001	1.494E-04	1.494E-04
$f_{11(D=50)}$	0.165	1.630E-04	1.602E-04	1.630E-04	1.630E-04	5.290E-04	1	1.630E-04	1.630E-04
$f_{12(D=50)}$	3.288E-04	1.317E-04	1.286E-04	1.317E-04	1.317E-04	0.002	0.012	1.317E-04	0.078
$f_{13(D=50)}$	0.012	1.786E-04	1.756E-04	1.786E-04	1.786E-04	1.786E-04	5.718E-04	1.786E-04	0.002
$f_{14(D=50)}$	0.003	8.745E-05	8.570E-05	8.745E-05	0.002	0.002	0.002	3.110E-04	0.368
$f_{9(D=100)}$	0.031	1.817E-04	1.776E-04	1.817E-04	1.817E-04	0.473	0.384	1.817E-04	7.512E-04
$f_{10(D=100)}$	6.386E-05	6.386E-05	6.249E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05	6.386E-05
$f_{11(D=100)}$	0.671	1.786E-04	1.756E-04	1.786E-04	1.786E-04	1.786E-04	0.472	1.786E-04	1.786E-04
$f_{12(D=100)}$	5.555E-04	1.727E-04	1.697E-04	1.727E-04	1.727E-04	1.727E-04	3.131E-04	1.727E-04	5.972E-03
$f_{13(D=100)}$	0.121	1.827E-04	1.796E-04	1.827E-04	1.827E-04	0.140	0.104	1.827E-04	2.312E-04
$f_{14(D=100)}$	0.037	8.745E-05	8.512E-05	8.745E-05	0.002	0.002	0.002	8.745E-05	0.368
$f_{9(D=300)}$	0.001	1.727E-04	1.688E-04	1.727E-04	1.727E-04	1	0.426	1.727E-04	0.006
$f_{10(D=300)}$	0.005	1.494E-04	1.468E-04	1.494E-04	0.002	0.024	0.100	1.494E-04	1.494E-04
$f_{11(D=300)}$	0.012	1.494E-04	1.459E-04	1.494E-04	1.494E-04	1.494E-04	0.135	1.494E-04	1.494E-04
$f_{12(D=300)}$	0.005	1.630E-04	1.602E-04	1.630E-04	1.630E-04	0.003	0.002	1.630E-04	0.015
$f_{13(D=300)}$	0.186	1.827E-04	1.796E-04	1.827E-04	1.827E-04	1.827E-04	1.827E-04	1.827E-04	6.386E-05
$f_{14(D=300)}$	0.003	8.745E-05	8.512E-05	8.745E-05	0.002	0.002	0.002	8.745E-05	0.368

algorithm is better. Figure 5 to Figure 9 show all box plots of different algorithms. Figure 5 is low dimension function box plots. Figure 6 to Figure 9 is variable dimension function box plots ($D = 2/50/100/300$). Most FSASCA box plots are narrower than those of other algorithms. The box plot can provide the key information about the data location and dispersion, when comparing different algorithm searching data. The box plot results show that the FSASCA not only has a wonderful robustness accuracy and a good balance, but also can avoid falling into the region of local optimal solution.

E. SEARCHING PATH DISCUSSIONS

In algorithms, the optimization path can reflect the distribution and dynamic changing of the algorithm population during iteration optimization in a real time. Therefore, the algorithm population diversity can be monitored by analyzing the optimization path, and the related algorithm performances can be analyzed. Figure 10 shows the three dimension and colorbar, and is the function three-dimensional graph. Figure 11 shows the optimization path between FSASCA and FSA. The comparison diagram of optimization path is refracted to the two dimension plane and the contour map on the two dimension plane. The red line in the figure is the FSASCA optimization path, the green line is the FSA optimization path, and the pink origin in the figure is the ideal search position. It can be seen from. Figure 11 shows that in the two dimension functions f_4, f_7 and f_8, the FSASCA optimization path is close to the FSA optimization path, and FSASCA optimization paths in other low dimension functions are smaller than FSA optimization path. There are some short distances, repeated, invalid and occasional long distance paths in FSA optimization path. FSASCA can save a lot of computing time to obtain a better searching path in only a few iterations, which shows that FSASCA has a high positioning accuracy. At the end of iteration, FSASCA can prevent the premature phenomenon of excessive attenuation of the algorithm.

F. WILCOXON RANK SUM TEST DISCUSSION

The rank sum test is a nonparametric test, it does not rely on the specific form of the overall distribution. In the case of arbitrary population distribution, the rank sum test is often used to check whether there is a significant difference in the distribution of paired test data. The Wilcoxon rank sum areapplied to test some differences in the distribution data. The Wilcoxon rank sum test result is $\boldsymbol{p}$ value, The dividing line of the $\boldsymbol{p}$ value is 0.05, it means that two sets of data are statistically analyzed when the significance level is 0.05. If $\boldsymbol{p}$ value exceeds 0.05, there is no significant changing for two sets of data. If $\boldsymbol{p}$ value is less than 0.05 and is close to 0, there is a significant changing for two sets of data. The Wilcoxon rank sum test results are shows in Table 4. ***NO*** means that the computer cannot give the calculated value because of the too large or too small data. From Table 4 we can find that the Wilcoxon rank sum test result of FSA in function f_3, f_5, f_8, $f_{9(D=2)}$, $f_{11(D=2)}$, $f_{11(D=50)}$, $f_{11(D=100)}$, $f_{13(D=100)}$, $f_{13(D=300)}$ is larger than 0.05. The result of CS and FPA in function f_5, f_8, $f_{11(D=2)}$ is larger than 0.05. The result of SA in function f_3, f_4, f_8 is larger than 0.05. The result of CSA in function f_1, f_5, f_8 is larger than 0.05. The result of GPC in function f_1, f_4, f_8, $f_{11(D=2)}$, $f_{9(D=50)}$, $f_{9(D=100)}$, $f_{9(D=300)}$ is larger than 0.05. The result of JS in function f_1, f_6, f_8, $f_{11(D=50)}$, $f_{9(D=100)}$, $f_{11(D=100)}$, $f_{9(D=300)}$ to $f_{11(D=300)}$ is larger than 0.05. The result of SCA function f_1, f_4, f_5, f_8, is larger than 0.05. The result of SHO in function f_1, $f_{9(D=2)}$, $f_{12(D=50)}$, $f_{14(D=50)}$, $f_{14(D=100)}$, $f_{14(D=300)}$ is larger than 0.05. The Wilcoxon rank sum test result of other algorithms are all less than 0.05. The Wilcoxon rank sum test shows that the algorithm proposed in this paper can quickly get effective results in the searching space and can use the exploration trend of the searching space to accelerate the iteration speed and improve the convergence accuracy.

V. CONCLUSION

In this paper, FSASCA is proposed to solve different optimization problems. FSASCA is a mixed algorithm based the basic FSA and basic SCA, which can overcome that FSA neglects neighborhood solutions in the searching process. SCA allow a feasible solution to be re-positioned around another feasible solutions, which can enhance the real-time stability and the precision in basic FSA. FSASCA can greatly avoid excessive precocity and missing feasible solutions around the global optimal solution in all searching phase, and can get a better accuracy. In comparison experiments, this paper used different functions and different algorithms to show FSASCA searching performances. FSASCA can get the best function values. This paper also gives data calculation results, Wilcoxon rank sum test, iteration figures, box plot figures, and searching path figures, which are used to show that FSASCA can offer a large competitive ability compared with basic FSA. And testing results show that FSASCA is outperforms the basic FSA for all testing functions. In the future, FSASCA will be applied in industry problems to find different parameters. Then, the research will discuss on the discrete FSASCA, and used the discrete FSASCA to solve the actual industry problem.

REFERENCES

[1] M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm Evol. Comput.*, vol. 33, pp. 1–17, Apr. 2017.

[2] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments—A survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.

[3] L. T. Biegler and I. E. Grossmann, "Retrospective on optimization," *Comput. Chem. Eng.*, vol. 28, no. 8, pp. 1169–1192, Jul. 2004.

[4] R. Storn and K. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[5] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, "African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems," *Comput. Ind. Eng.*, vol. 158, Aug. 2021, Art. no. 107408.

[6] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[7] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," *Comput. Struct.*, vol. 169, pp. 1–12, Jun. 2016.

[8] X. Qi, Y. Zhu, and H. Zhang, "A new meta-heuristic butterfly-inspired algorithm," *J. Comput. Sci.*, vol. 23, pp. 226–239, Nov. 2017.

[9] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009.

[10] S. Talatahari and M. Azizi, "Chaos game optimization: A novel metaheuristic algorithm," *Artif. Intell. Rev.*, vol. 54, pp. 917–1004, Feb. 2021.

[11] I. Naruei and F. Keynia, "Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems," *Eng. Comput.*, vol. 38, no. S4, pp. 3025–3056, Oct. 2021.

[12] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, Feb. 2016.

[13] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowledge-Based Syst.*, vol. 191, Mar. 2020, Art. no. 105190.

[14] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: An optimization method for continuous non-linear large scale problems," *Inf. Sci.*, vol. 183, no. 1, pp. 1–15, Jan. 2012.

[15] M.-Y. Cheng and D. Prayogo, "Symbiotic organisms search: A new metaheuristic optimization algorithm," *Comput. Struct.*, vol. 139, pp. 98–112, Jul. 2014.

[16] A. Tabari and A. Ahmad, "A new optimization method: Electro-search algorithm," *Comput. Chem. Eng.*, vol. 103, pp. 1–11, Aug. 2017.

[17] Y.-J. Zheng, "Water wave optimization: A new nature-inspired metaheuristic," *Comput. Oper. Res.*, vol. 55, no. 1, pp. 1–11, Mar. 2015.

[18] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

[19] G. Dhiman and V. Kumar, "Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications," *Adv. Eng. Softw.*, vol. 114, pp. 48–70, Dec. 2017.

[20] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2592–2612, May 2013.

[21] I. Naruei and F. Keynia, "A new optimization method based on COOT bird natural life model," *Exp. Syst. Appl.*, vol. 183, Nov. 2021, Art. no. 115352.

[22] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: Charged system search," *Acta Mechanica*, vol. 213, nos. 3–4, pp. 267–289, 2010.

[23] H. Karami, M. V. Anaraki, S. Farzin, and S. Mirjalili, "Flow direction algorithm (FDA): A novel optimization approach for solving optimization problems," *Comput. Ind. Eng.*, vol. 156, Jun. 2021, Art. no. 107224.

[24] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.

[25] J.-S. Chou and D.-N. Truong, "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean," *Appl. Math. Comput.*, vol. 389, Jan. 2021, Art. no. 125535.

[26] A. Mohammadi-Balani, M. D. Nayeri, A. Azar, and M. Taghizadeh-Yazdi, "Golden eagle optimizer: A nature-inspired metaheuristic algorithm," *Comput. Ind. Eng.*, vol. 152, Feb. 2021, Art. no. 107050.

[27] Y. Wang, H. Liu, G. Ding, and L. Tu, "Adaptive chimp optimization algorithm with chaotic map for global numerical optimization problems," *J. Supercomputing*, 6507–6537, 2022.

[28] B. M. Nguyen, T. Tran, T. Nguyen, and G. Nguyen, "Hybridization of galactic swarm and evolution whale optimization for global search problem," *IEEE Access*, vol. 8, pp. 74991–75010, 2020.

[29] B. Chen, H. Lei, H. Shen, Y. Liu, and Y. Lu, "A hybrid quantum-based PIO algorithm for global numerical optimization," *Sci. China Inf. Sci.*, vol. 62, no. 7, Jul. 2019.

[30] D. Wu, H. Rao, C. Wen, H. Jia, Q. Liu, and L. Abualigah, ''Modified sand cat swarm optimization algorithm for solving constrained engineering optimization problems,'' *Mathematics*, vol. 10, no. 22, p. 4350, Nov. 2022.

[31] A. Sulaiman, M. Sadiq, Y. Mehmood, M. Akram, and G. A. Ali, ''Fitness-based acceleration coefficients binary particle swarm optimization (FACBPSO) to solve the discounted knapsack problem,'' *Symmetry*, vol. 14, no. 6, p. 1208, Jun. 2022.

[32] Y.-H. Feng and G.-G. Wang, ''Binary moth search algorithm for discounted 0–1 knapsack problem,'' *IEEE Access*, vol. 6, pp. 10708–10719, 2018.

[33] M. Abdel-Basset, D. El-Shahat, and I. El-Henawy, ''Solving 0–1 knapsack problem by binary flower pollination algorithm,'' *Neural Comput. Appl.*, vol. 31, no. 9, pp. 5477–5495, Sep. 2019.

[34] Y. Feng, X. Yu, and G. G. Wang, ''A novel monarch butterfly optimization with global position updating operator for large-scale 0–1 knapsack problems,'' *Mathematics*, vol. 7, no. 11, p. 1056, Nov. 2019.

[35] D. Chen, Z. Wang, G. Zhou, and S. Li, ''Path planning and energy efficiency of heterogeneous mobile robots using Cuckoo–Beetle swarm search algorithms with applications in UGV obstacle avoidance,'' *Sustainability*, vol. 14, no. 22, p. 15137, Nov. 2022.

[36] Q. Wu, Z. Chen, L. Wang, H. Lin, Z. Jiang, S. Li, and D. Chen, ''Real-time dynamic path planning of mobile robots: A novel hybrid heuristic optimization algorithm,'' *Sensors*, vol. 20, no. 1, p. 188, Dec. 2019.

[37] C. Mou, W. Qing-xian, and J. Chang-sheng, ''A modified ant optimization algorithm for path planning of UCAV,'' *Appl Soft Comput.*, vol. 8, no. 4, pp. 1712–1718, 2008.

[38] Q. Jin, C. Tang, and W. Cai, ''Research on dynamic path planning based on the fusion algorithm of improved ant colony optimization and rolling window method,'' *IEEE Access*, vol. 10, pp. 28322–28332, 2022.

[39] M. Ma, J. Liang, M. Guo, Y. Fan, and Y. Yin, ''SAR image segmentation based on artificial bee colony algorithm,'' *Appl. Soft Comput.*, vol. 11, pp. 5205–5214, Dec. 2011.

[40] Z. Wang, H. Pen, T. Yang, and Q. Wang, ''Structure-priority image restoration through genetic algorithm optimization,'' *IEEE Access*, vol. 8, pp. 90698–90708, 2020.

[41] Z. Daoqing and J. Mingyan, ''Parallel discrete lion swarm optimization algorithm for solving traveling salesman problem,'' *J. Syst. Eng. Electron.*, vol. 31, no. 4, pp. 751–760, Aug. 2020.

[42] T. Milan and N. Bacanin, ''Improved seeker optimization algorithm hybridized with firefly algorithm for constrained optimization problems,'' *Neurocomputing*, vol. 143, pp. 197–207, Nov. 2014.

[43] H. C. Ong, S. L. Tilahun, W. S. Lee, and J. M. T. Ngnotchouye, ''Comparative study of prey predator algorithm and firefly algorithm,'' *Intell. Autom. Soft Comput.*, vol. 24, no. 2, pp. 359–366, 2018.

[44] X. Jiang, Z. Lin, T. He, X. Ma, S. Ma, and S. Li, ''Optimal path finding with beetle antennae search algorithm by using ant colony optimization initialization and different searching strategies,'' *IEEE Access*, vol. 8, pp. 15459–15471, 2020.

[45] M. Elsisi, ''Future search algorithm for optimization,'' *Evol. Intell.*, vol. 12, no. 1, pp. 21–31, Mar. 2019.

[46] S. Mirjalili, ''SCA: A sine cosine algorithm for solving optimization problems,'' *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.

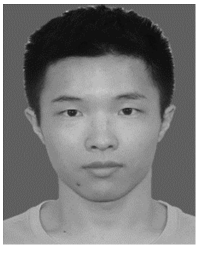
[47] Y. G. Xue and H. W. Deng, ''The cuckoo search algorithm based on dynamic grouping to adjust fight scale,'' *Appl. Mech. Mater.*, vols. 543–547, pp. 1822–1826, Mar. 2014.

[48] X. S. Yang, M. Karamanoglu, and X. He, ''Flower pollination algorithm: A novel approach for multiobjective optimization,'' *Eng. Optim.*, vol. 46, no. 9, pp. 1222–1237, 2014.

[49] I. H. Osman, ''Metastrategy simulated annealing and Tabu search algorithms for the vehicle routing problem,'' *Ann. Oper. Res.*, vol. 41, no. 4, pp. 421–451, 1993.

[50] S. Harifi, J. Mohammadzadeh, M. Khalilian, and S. Ebrahimnejad, ''Giza pyramids construction: An ancient-inspired metaheuristic algorithm for optimization,'' *Evol. Intell.*, vol. 14, no. 4, pp. 1743–1761, Dec. 2021.

YUQI FAN received the B.S. degree in mechanical design manufacture and automation and the M.S. and the Ph.D. degrees in mechanical and electronic engineering from the Harbin University of Science and Technology, in 2014, 2017, and 2021, respectively. Her current research interest includes intelligent algorithm.



SHENG ZHANG received the B.S. degree in mechanical and electronic engineering from Jiangsu University, China, in 2020. He is currently pursuing the M.S. degree in mechanical design and theory with the Harbin University of Science and Technology. His current research interests include rolling bearing fault diagnosis and life prediction.

HUIMIN YANG received the B.S. degree in mechanical and electronic engineering from Tonghua Normal University, China, in 2016. She is currently pursuing the M.S. degree in mechanical engineering with the Harbin University of Science and Technology. Her research interest includes rotating machinery fault diagnosis.

DI XU received the B.S. degree in mechanical engineering and automation from the Harbin Far East Institute of Technology, in 2014, and the M.S. degree in mechanical design and theory from the Harbin University of Science and Technology, in 2017, where she is currently pursuing the Ph.D. degree in mechanical and electronic engineering. Her current research interests include rotating machinery fault diagnosis and equipment health management.

YAPING WANG received the B.S. degree in computer science and technology from the Harbin Institute of Technology, in 2000, and the M.S. degree in mechanical and electronic engineering and the Ph.D. degree in mechanical manufacturing and automation from the Harbin University of Science and Technology, in 2006 and 2010, respectively. She is a Professor with the Harbin University of Science and Technology. Her current research interests include rotating machinery fault diagnosis and equipment health management.