Improved Salp Swarm Algorithm Based on Oscillation Inertia Weights to Solve Function Optimization Functions

Xue-Long Li, Jie-Sheng Wang *, Chun-Li Lin, Zhen-Long Zhao

Abstract—**Salp swarm algorithm (SSA) is a new meta-heuristic algorithm based on the behavior of the salps swarm. In view of the defects of SSA, such as poor exploitation and exploration ability, six kinds of inertia weights oscillated in accordance to the increase of iteration number were introduced into the optimization process of SSA algorithm so as to make it not falling into the local optimum in advance. In order to verify the effectiveness of the improved SSA, the original SSA and the improved SSA with oscillatory inertia weights are tested with 24 CEC test functions and the simulation experiments results are analyzed. The simulation results show the proposed algorithms based on six oscillating inertia weight are better than the original SSA, and improve its convergence speed and optimization accuracy.**

Index Terms—**salp swarm algorithm, oscillation inertia weight, function optimization, performance computation**

I. INTRODUCTION

AS the complexity of various problems increases, the need for low-cost, fast and more intelligent optimization algorithms have appeared over the past decades. There are many difficulties and challenges in practical problems, such as non-linearity, multi-objective, multivariability, high dimension, uncertainty and non-convex constraints. Traditional mathematical optimization methods were proposed, such as integer programming, gradient descent, and newton method, but these optimization algorithms can not solve complex practical problems well. Many meta-heuristics algorithms have been applied in engineering and scientific fields, which have successfully solved many optimization problems in recent years， making them a research hot spot for optimization [1]. The meta-heuristic methods adopt the random optimization strategy, which generates multiple random solutions through iterative generation process, and improves them in the optimization process [2]. In general, it is of great significance to research the theory, improvement and application of meta-heuristic algorithm, both in improving its optimization performance and in broadening its application fields [3].

Manuscript received March 5, 2022; revised August 8, 2022. This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. LJKZ0293), and the Project by Liaoning Provincial Natural Science Foundation of China (Grant No. 20180550700).

Xue-Long Li is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: lxl17346536920@126.com).

Jie-Sheng Wang is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (Corresponding author, phone: 86-0412-2538246; fax: 86-0412-2538244; wang_jiesheng@126.com).

Chun-Li Lin is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, P. R. China (e-mail: 356408163@qq.com).

Zhen-Long Zhao is a senior engineer of School of Electrical Engineering, Yingkou Institute of Technology, Yingkou, 115014, P. R. China (e-mail: 308056442@qq.com).

SSA was a new swarm intelligence optimization algorithm based on random population proposed by Mirjalili in 2017 [4]. It has the advantages of less computation, less control parameters and easy to implement. It shows obvious advantages in many practical engineering problems. But similar to other swarm intelligence algorithms, the standard SSA still has some defects, such as low solution accuracy and slow convergence rate. Researchers have proposed many strategies to enhance its performance. The chaotic mapping was added to adjust the random parameters of leader position in order to upgrade the classification performance of standard SSA in feature selection [5]. The refraction reverse learning mechanism and adaptive control factor were proposed to improve search performance of standard SSA [6]. The mutation operator (DE/rand/1) were adopted to mutate non-optimal individuals, which has achieved good results in the application of reactive power compensation in distribution system [7]. The combination of standard SSA and PSO algorithm to update population location can delete redundant or confused features in feature selection while maintaining high accuracy and efficiency [8]. The constant inertia weight factor was added to food source location and K-neighborhood classifier was combined to select features [9]. The time-varying inertia weight factor was proposed to update follower position and the adaptive catastrophe strategy was combined to enhance its performance [10]. The spatial transformation searching (STS) was added to SSA and the experiments results show STS-SSA has good robustness in searching process [11]. The reverse learning strategy and local search feature selection algorithm was proposed to increase the calculation accuracy of standard SSA [12]. The hybrid method obtained by combining the characteristics of the firefly algorithm with SSA effectively improved the performance of standard SSA and has been successfully applied to solve UPMSP [13]. The strategy by using the weighted distance position update mode was proposed to increase the exploration ability and calculation accuracy of standard SSA [14]. The operators of the SCA and disruption operator were added to improve the exploration efficiency of SSA [15].

To solve the problems of low accuracy and slow convergence velocity of standard SSA, an hybrid SSA based on oscillation inertia weights was proposed. The inertia weight based SSA (IWSSA) was introduced on the basis of the standard SSA to improve its randomness and increase its local searching and exploration ability. By optimizing 24 benchmark functions and comparing the performance with the standard SSA, the effectiveness and robustness of IWSSA are verified.

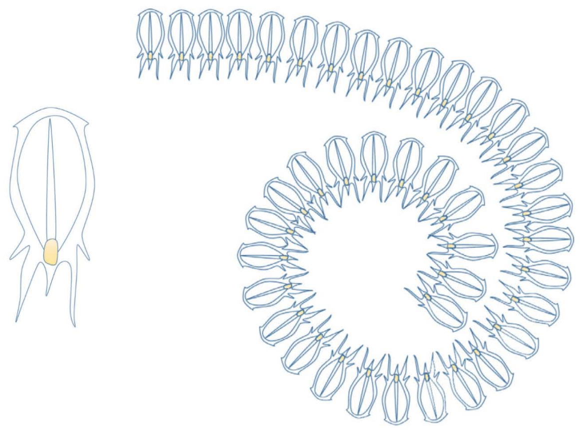
II. Basic Principles of Salp Swarm Algorithm

A. Biological Principle of Salp Swarm Algorithm

The main idea of SSA comes from the clustering behavior of salps. Salp is an almost completely transparent barrel-shaped marine organism, whose transparent property can well protect salps from predators in the ocean. Salp mainly feeds on phytoplankton, rely on inhalation and ejection of sea water to move forward. They can transport tons of carbon from the ocean surface to the deep sea every day, prevent the carbon from re-entering the atmosphere and effectively reduce the possibility of carbon entering the atmosphere to become a greenhouse gas. It has a good carbon removal effect for nature, and the special function of salp makes it play an irreplaceable role in the ocean. Most creatures in nature move and forage in groups, but salp is connected from end to end to form a chain for foraging. Salp often forms a salp chain swarm in the deep sea, which includes two parts: leaders and followers. The individual at the front of the salp chain is considered to be the leader, other individuals are considered to be followers, and the leader leads the follower to move according to the location of the food source. The biological principle of SSA is shown in Fig. 1.

B. Salp Swarm Algorithm

The whole optimization process of SSA originates from the clustering foraging behavior of salps chain, and the process of searching for the best food source of salps population is compared to the optimization of function. In order to balance the algorithm in the basic SSA, the salps in the first half of the salps chain are the leaders, and the rest are the followers. Different from other swarm intelligence optimization algorithms, leaders will not affect the movement of the whole swarm, and followers will update their positions according to the previous individual, and so on to form a salps chain.



(a) Individual salp (b) Swarm of salps (salps chain)

Fig. 1 Biological principle of salp swarm algorithm.

The leader will play a weaker and weaker role in the leadership of the followers, and the followers will not blindly move towards the leader. This behavioral characteristic maintains the diversity of the population. SSA saves the optimal solution and assigns it to the food source variable, so even if the whole population deteriorates, it will not be lost.

SSA only updates the position of the leader relative to the food source, so the leader is always exploring and developing the space around it. The progressive movement of followers reduces the situation of falling into local extreme. In salp swarm algorithm, the position vector X_j^i of each salp individual is defined for searching in D-dimensional space, where N is the number of decision variables. The position vector X_j^i in SSA will be composed of N salps individuals with dimension D. Therefore, the population vector is composed of $N \times D$-dimensional matrices, which is defined as:

$$X_j^i=\begin{pmatrix} x_1^1 & x_2^1 & \cdots & x_D^1 \\ x_1^2 & x_2^2 & \cdots & x_D^2 \\ \vdots & & \ddots & \vdots \\ x_1^N & x_2^N & \cdots & x_D^N \end{pmatrix} \tag{1}$$

SSA initializes the population by generating random numbers, and initializes the position X_j^i ($i=1,\cdots,D$, $j=1,\cdots,D$) of salps.

$$X_j^i=rand(N,D)\times(ub(j)\text{-}lb(j))+lb(j) \tag{2}$$

where, N is the population size of salps and D is the spatial dimension.

In SSA, the initial position of the food source is selected by the objective function. By determining the position of leaders and followers, leaders and followers is represented by a two-dimensional matrix X. The location update strategy for leaders is realized by:

$$X_j^i(l)=\begin{cases} F_j+c_1((ub_j-lb_j)c_2+lb_j), c_3\geq 0.5 \\ F_j-c_1((ub_j-lb_j)c_2+lb_j), c_3<0.5 \end{cases} \tag{3}$$

where, l represents the current number of iterations, $X_j^i(l)$ represents the follower's position at the current generation of the i-th obsidian in the j-dimensional space, F_j is the food source's position at the current generation in the j-dimensional space, ub_j and lb_j represent the upper and lower limits of the j-dimensional space, and c_2 and c_3 are random numbers evenly distributed in the scope (0, 1), which is used to adjust the changing trend of the leader's position. Coefficient c_1 is the most crucial parameter of SSA to be used in balancing the exploration and exploitation, which decreases adaptively with the increase of iterations. The value range of c_1 is shown in Eq. (4).

$$c_1=2e^{-(4l/Max\_iter)^2} \tag{4}$$

The position of followers is updated based on the Newton's law of motion shown in Eq. (5).

$$x_j^i=\frac{1}{2}at^2+v_0t \tag{5}$$

where, X_j^i represents the position of the i-th follower in the j-th dimension, $i\geq 2$, t represents the time, v_0 represents the initial velocity, acceleration $a=(v_{final}\text{-}v_0)/\Delta t$, $v_{final}=(x_j^{i-1}\text{-}x_j^i)/\Delta t$.

Because the difference between each iteration is 1 and the initial velocity v_0=0, Eq. (5) can be expressed as:

$$X_j^i(l)=\frac{1}{2}(X_j^i(l-1)+X_j^{i-1}(l-1)) \tag{6}$$

where, l represents the current number of iterations, $X_j^i(l)$ represents the position of the i -th follower at current generation and j -th dimension, $X_j^i(l-1)$ and $X_j^{i-1}(l-1)$ represent the positions of the i -th and ($i-1$)-th follower of the previous generation in the j -th dimension space, respectively.

C. *Pseudo Code of SSA*

The flowchart of SSA is shown in Fig. 2.

III. Improved SSA Based on Oscillation Inertia Weights

Inertia weight strategy introduced inertia weight factor w into PSO algorithm for the first time by Shi [16] and achieved satisfactory results. Many subsequent studies have also proved the importance of w to balance the global optimization and local optimization. It can be seen from Eq. (6) that during a certain iteration of SSA, the current position is updated according to the midpoint of the historical position of $i-1$ -th and i -th followers, which is also the mathematical model embodiment of the following characteristics of salps chain.

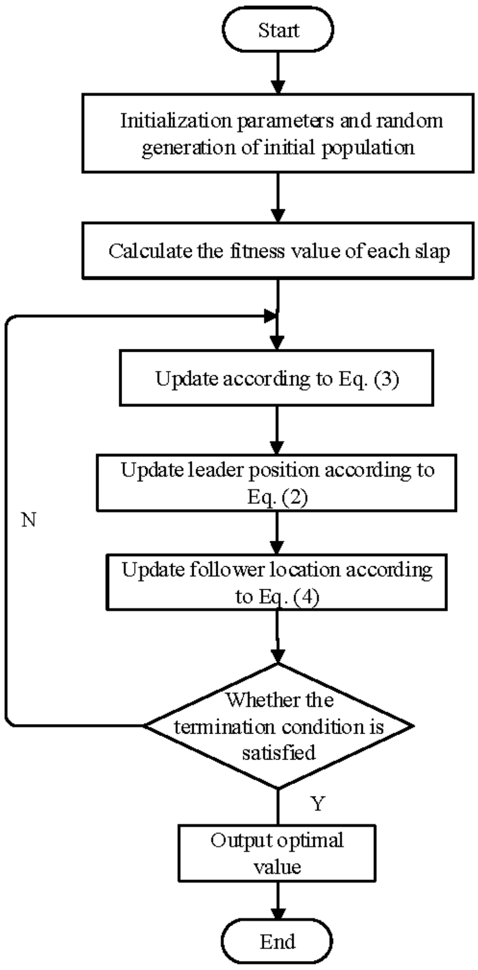


Fig. 2 Flowchart of salp swarm algorithm.

However, this is a kind of blind following behavior, which does not take into account the impact of the former on the latter, but only accepts the location information of the former to update the current location, which limits the search efficiency of the algorithm. In this paper, six convergence factors D are adopted as oscillatory inertia weights w [17]. The variation trend of six oscillatory inertia weights with the number of iterations is shown in Fig. 3, and the calculation formula is defined as:

$$w_1=2(2a\text{-}1)\exp(-(1.5l)^2/Max\_iter^2) \tag{7}$$

$$w_2=10((a-0.5)(1-\cos(l/Max\_iter-0.35\pi))) \tag{8}$$

$$w_3=5((a-0.5)(1+\sin(l/Max\_iter+\pi)) \tag{9}$$

$$w_4=(a\text{-}0.5)(2\text{-}\tan(l/Max\_iter) \tag{10}$$

$$w_5=4((a-0.5)(1-(l/Max\_iter)^3)) \tag{11}$$

$$w_6=4((a-0.5)(2-\exp(0.7l/Max\_iter))) \tag{12}$$

where, a represents a random number uniformly distributed between 0 to 1, l represents the current iteration number, and $Maxiter$ represents the maximum iteration number. So the oscillating inertia weight w can be realized by:

$$X_j^i=\frac{1}{2}(X_j^i+w\times X_j^{i-1}) \tag{13}$$

The oscillating inertia weight w decreases gradually in accordance with the increase of the iteration number. At the beginning of the iteration, the attenuation degree of w is lower and can move in a larger range. In the later process of iteration, the attenuation degree of w increases and the moving amplitude decreases, so that the optimal solution can be mined more accurately, thus the exploitation and exploration capabilities of search can be balanced.

IV. Simulation Experiments and Result Analysis

A. *Testing Functions*

In the simulation experiments, 24 benchmark functions are used to evaluate the performance of SSA, so as to show the validity of the proposed strategy. The expressions of 24 benchmark functions are listed in Table 1 [18], which includes three kinds. F_1-F_{12} are unimodal functions ,which have only one global optimal solution,and can efficiently verify the development ability of the improved algorithm. F_{13}-F_{21} are multi-peak functions, which have more than one extreme point or more than one local optimal value, so as to verify its global optimization performance. F_{22}-F_{24} are combined functions, which can verify its stability and robustness.

B. *Simulation Results Analysis of Improved SSA*

This section compares six improved SSAs based on oscillatory inertia weight (IWn-SSA, n=1, 2, ... ,6) with the original SSA and SSA with the inertia weight shown in Eq. (12) [19]. The particle number of each algorithm is set as 30. The experimental results of function optimization are illustrated in Fig. 4.

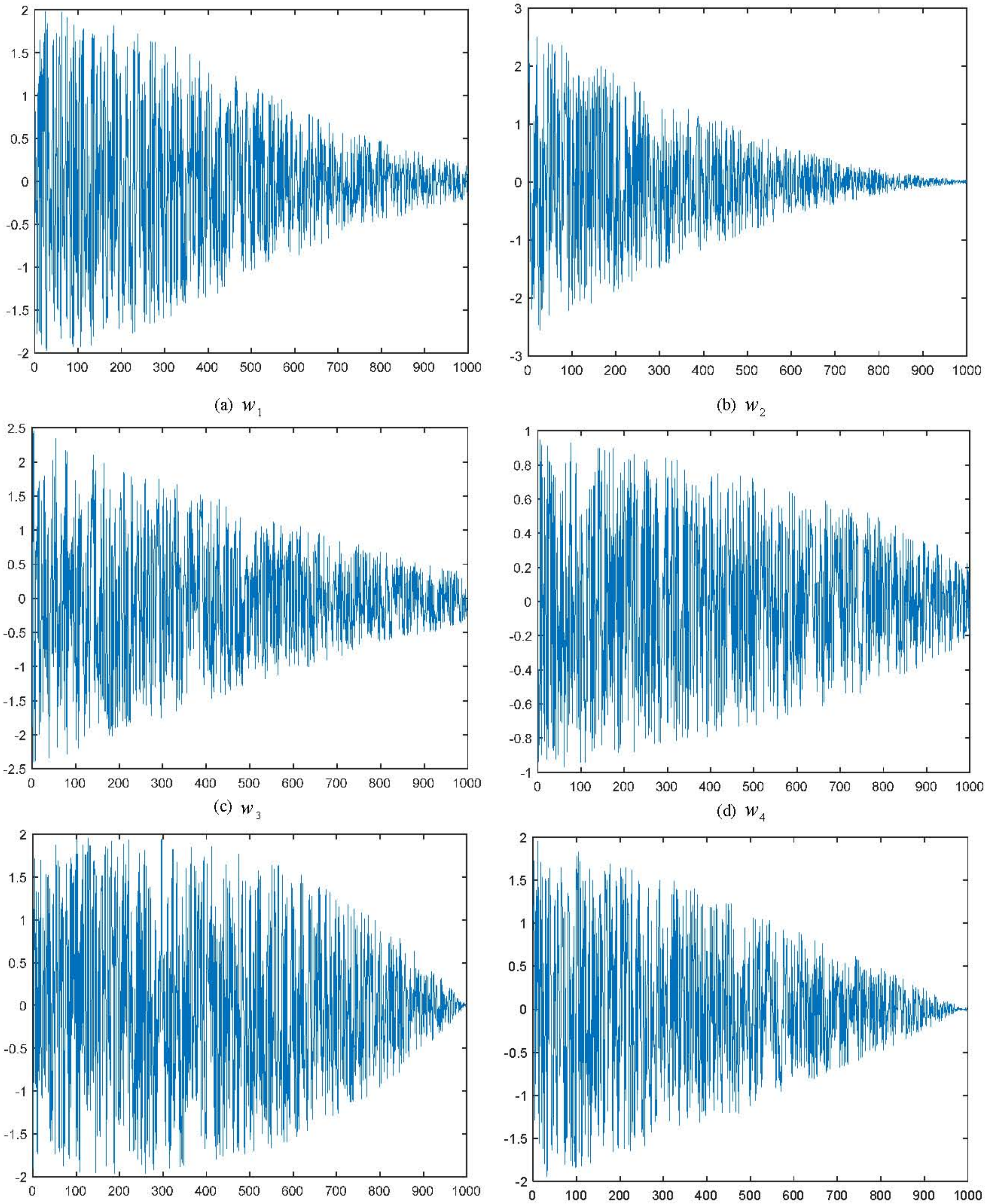


Fig. 3 Six oscillation inertia weights.

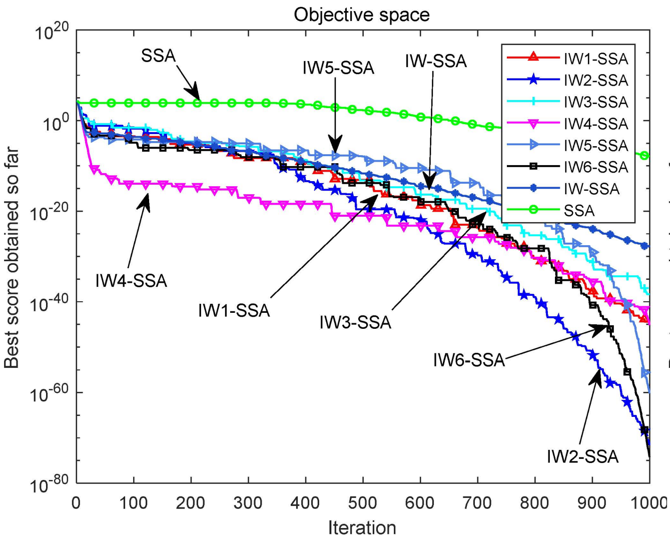
TABLE 1. BENCHMARK FUNCTIONS

Function	Dim	Range	f_{min}
$F_1(x)=\sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$F_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	[-10,10]	0
$F_3(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$	30	[-100,100]	0
$F_4(x)=\max_i\left\{\lvert x_i\rvert, 1\le i\le n\right\}$	30	[-100,100]	0

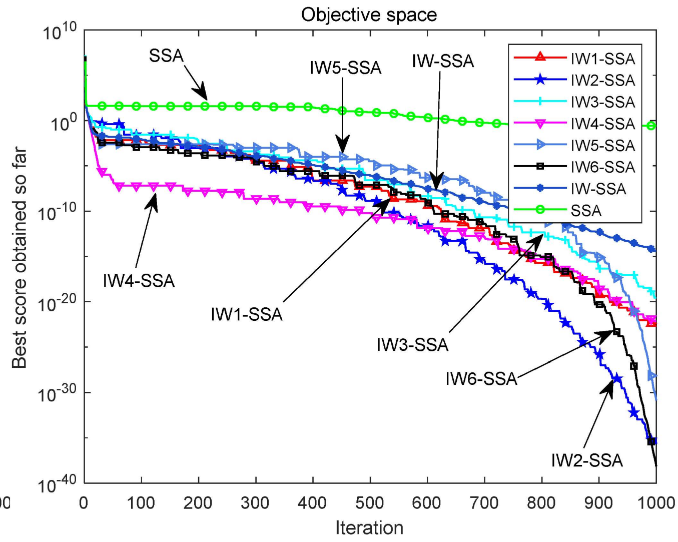
$F_5(x)=\sum_{i=1}^{n-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	30	[-30,30]	0
$F_6(x)=\sum_{i=1}^{n} ix_i^4+rand[0,1)$	30	[-100,100]	0
$F_7(x)=x_1^2+10^6\sum_{i=2}^{n} x_i^2$	30	[-10,10]	0
$F_8(x)=\sum_{i=1}^{n}\lvert x_i\rvert^{i+1}$	30	[-1,1]	0
$F_9(x)=\sum_{i=1}^{n}\lvert x_i\rvert$	30	[-100,100]	0
$F_{10}(x)=\sum_{i=1}^{n} x_i^{10}$	30	[-10,10]	0
$F_{11}(x)=\sum_{i=1}^{n-1}\left(x_i^2\right)^{\left(x_{i+1}^2+1\right)}+\left(x_{i+1}^2\right)^{\left(x_i^2+1\right)}$	30	[-1,4]	0
$F_{12}(x)=\left(x_1-1\right)^2+\sum_{i=1}^{n} i\left(2x_i^2-x_{i-1}\right)^2$	30	[-10,10]	0
$F_{13}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)\right)+20+e$	30	[-32,32]	0
$F_{14}(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[-600,600]	0
$F_{15}(x)=\frac{\pi}{n}\{10\sin\left(\pi y_1\right)+\sum_{i=1}^{n-1}\left(y_i-1\right)^2\left[1+10\sin^2\left(\pi y_{i+1}\right)\right]$ $+\left(y_n-1\right)^2\}+\sum_{i=1}^{n}u\left(x_i,10,100,4\right)$ $y_i=1+\frac{x_i+1}{4}$ $u\left(x_i,a,k,m\right)=\begin{cases}k\left(x_i-a\right)^m & x_i>a\\ 0 & -a<x_i<a\\ k\left(-x_i-a\right)^m & x_i<-a\end{cases}$	30	[-50,50]	0
$F_{16}(x)=0.1\{\sin^2\left(3\pi x_1\right)+\sum_{i=1}^{n}\left(x_i-1\right)^2\left[1+\sin^2\left(3\pi x_i+1\right)\right]$ $+\left(x_n-1\right)^2\left[1+\sin^2\left(2\pi x_n\right)\right]\}+\sum_{i=1}^{n}u\left(x_i,5,100,4\right)$	30	[-50,50]	0
$F_{17}(x)=\sum_{i=1}^{d}\lvert x_i\sin\left(x_i\right)+0.1x_i\rvert$	30	[-10,10]	0
$F_{18}(x)=\sum_{i=2}^{n}\left[\left(x_i-1\right)^2+\left(x_1-x_i^2\right)^2\right]$	30	[0,10]	0
$F_{19}(x)=\sin^2\left(\pi w_1\right)+\sum_{i=1}^{n-1}\left(w_i-1\right)^2\left[1+10\sin^2\left(\pi w_i+1\right)\right]$ $+\left(w_n-1\right)^2\left[1+\sin^2\left(2\pi w_n\right)\right]$ $w_i=\frac{x_i-1}{4}$	30	[-10,10]	0
$F_{20}(x)=\left[\frac{1}{n-1}\sum_{i=1}^{n-1}\left(\sqrt{s_i}\left(\sin\left(50.0s_i^{0.02}\right)+1\right)\right)\right]^2$ $s_i=\sqrt{x_i^2+x_{i+1}^2}$	30	[-10,10]	0
$F_{21}(x)=-\cos\left(x_1\right)\cos\left(x_2\right)e^{\left[-\left(x_1-\pi\right)^2-\left(x_2-\pi\right)^2\right]}$	30	[-100,100]	0
$F_{22}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right)^{-1}$	2	[-65,65]	1
$F_{23}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[-5,5]	0.00030
$F_{24}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10$	2	[-5,5]	-0.398

The experimental results of 10 runs are listed in Table 2, which gives the optimal values, average values and variance values. It can be concluded from the above simulation results the function optimization performance under six oscillatory inertia weights based SSAs (IWn-SSA, n=1, 2, ... ,6) is better than standard SSA.

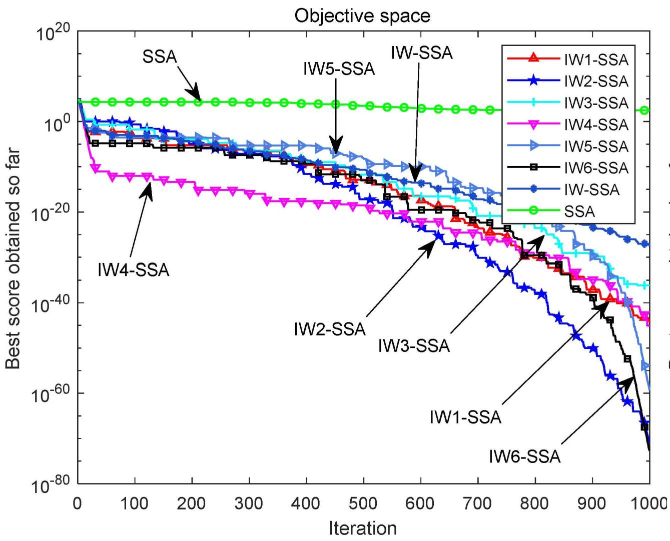
Seen from the results of the convergence curves on the testing functions, most of the results show that the oscillatory inertia weights based SSAs improve the convergence speed and accuracy than other algorithms to a great extent. In the mean time, it can be seen from Table 2 that the effectiveness of six oscillatory inertia weights based SSA is very stable in the process during 10 times optimization. From all the test results, they greatly improve the exploration ability and convergence speed and accuracy in most cases. In particular, in the fixed-dimensional test function, the robustness and reliability are improved under ensuring its optimization accuracy.



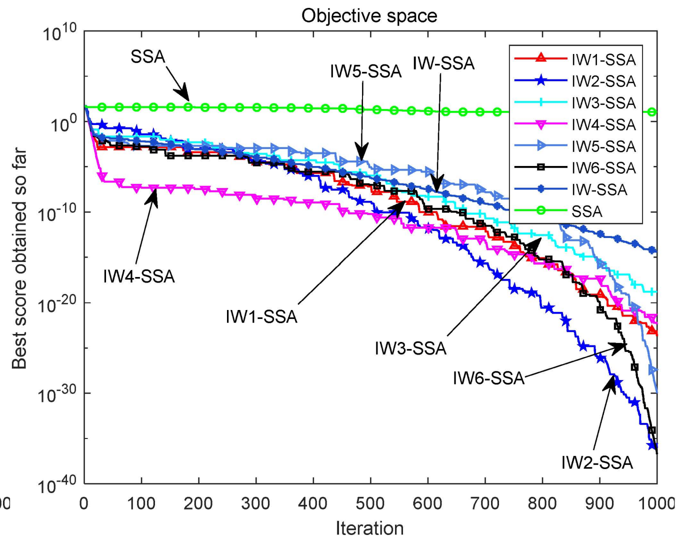
(1) F_1



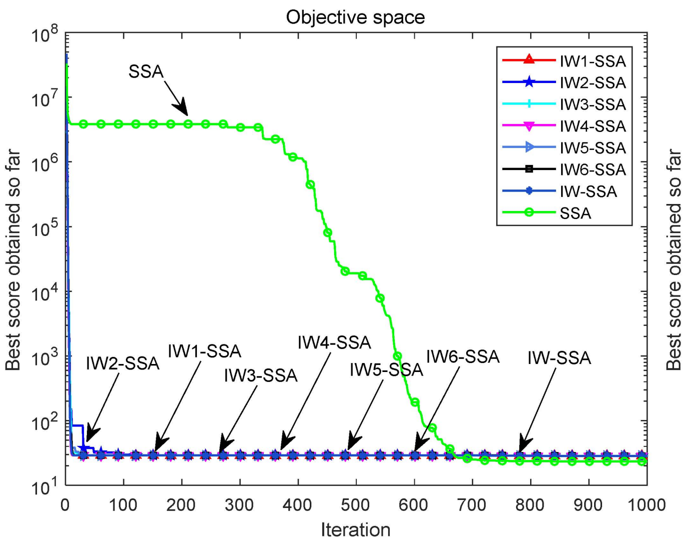
(2) F_2



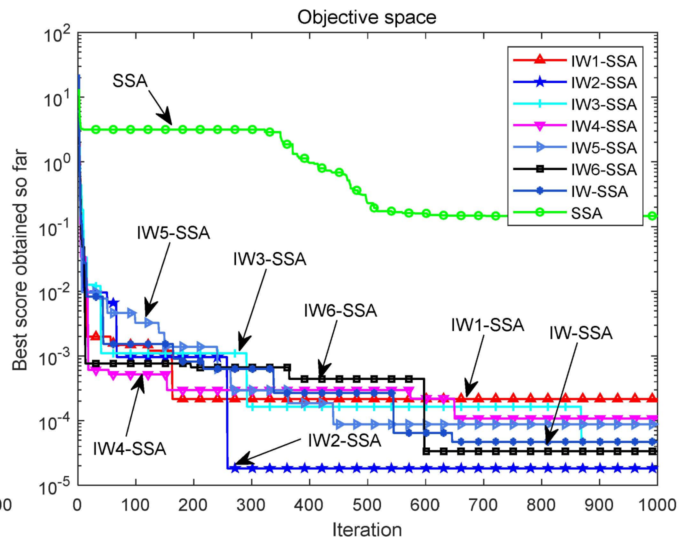
(3) F_3



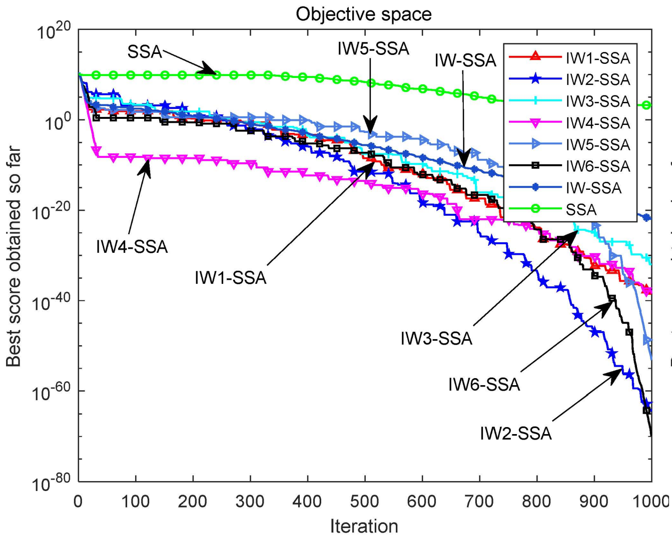
(4) F_4



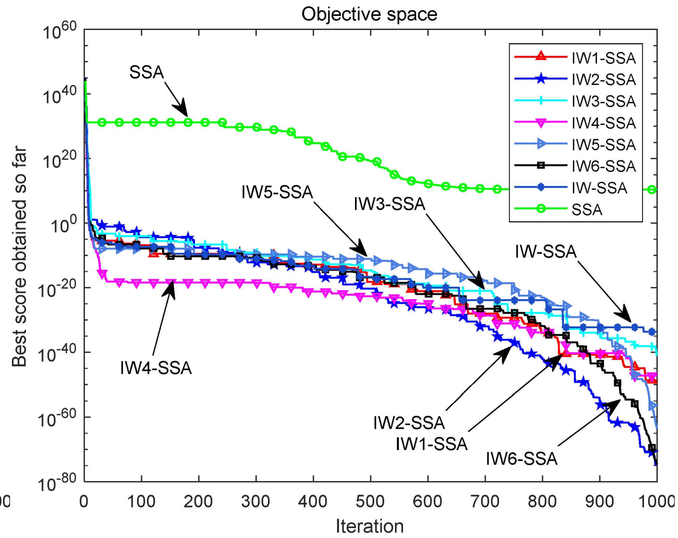
(5) F_5



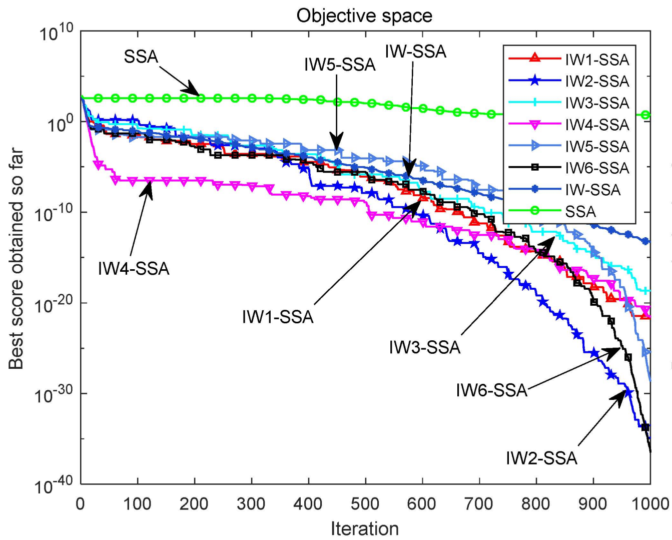
(6) F_6



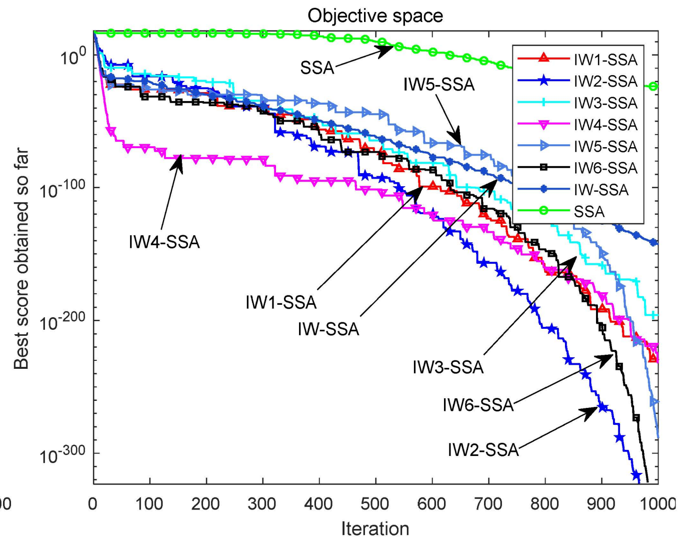
(7) F_7



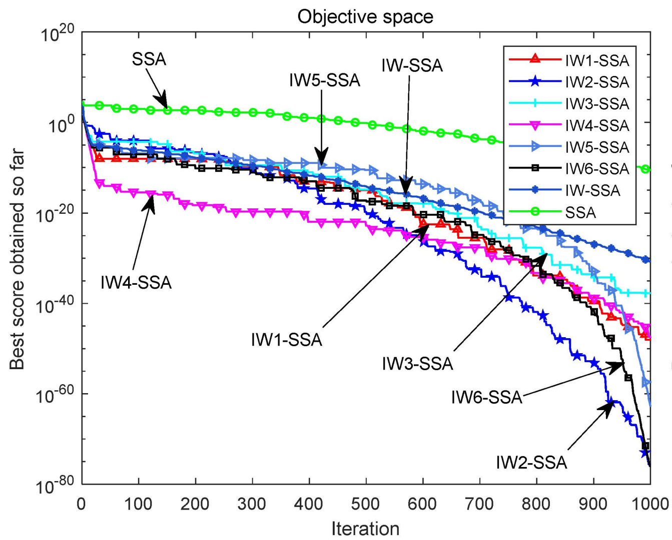
(8) F_8



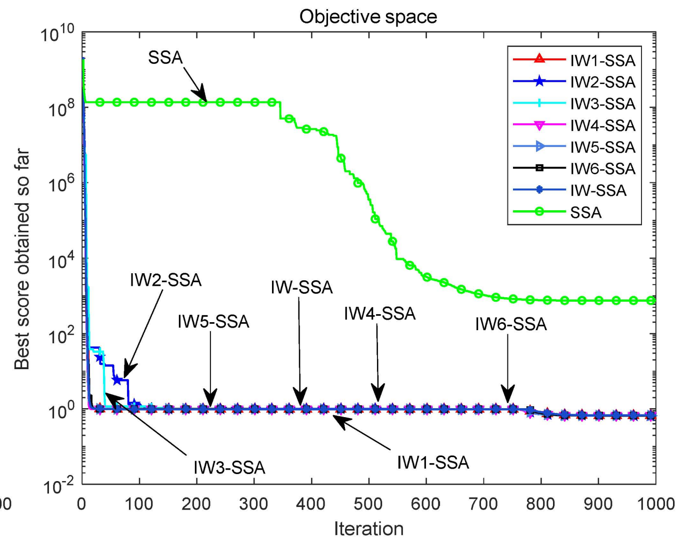
(9) F_9



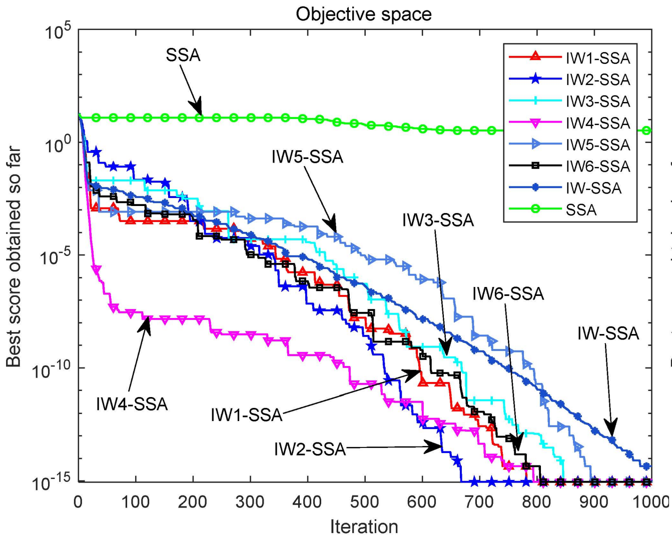
(10) F_{10}



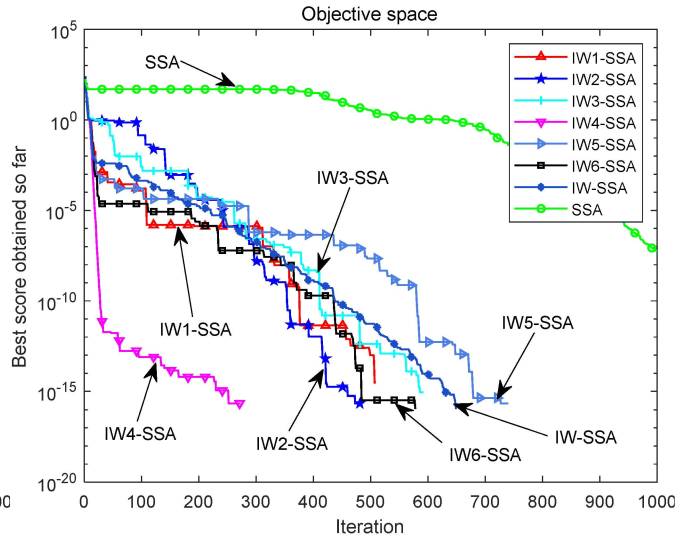
(11) F_{11}



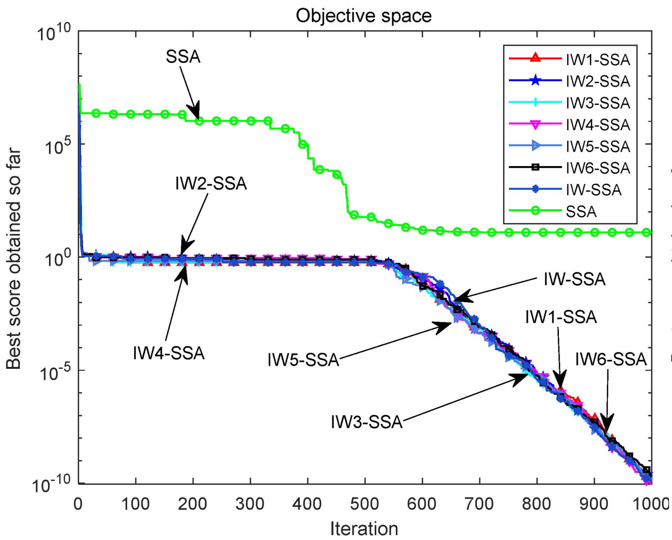
(12) F_{12}



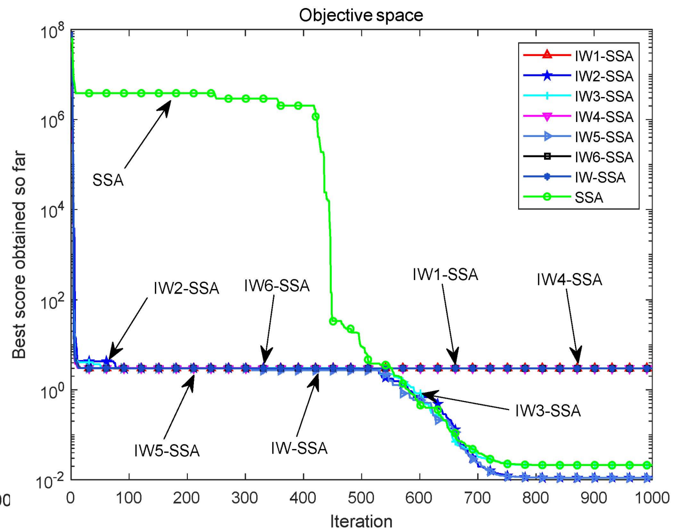
(13) F_{13}



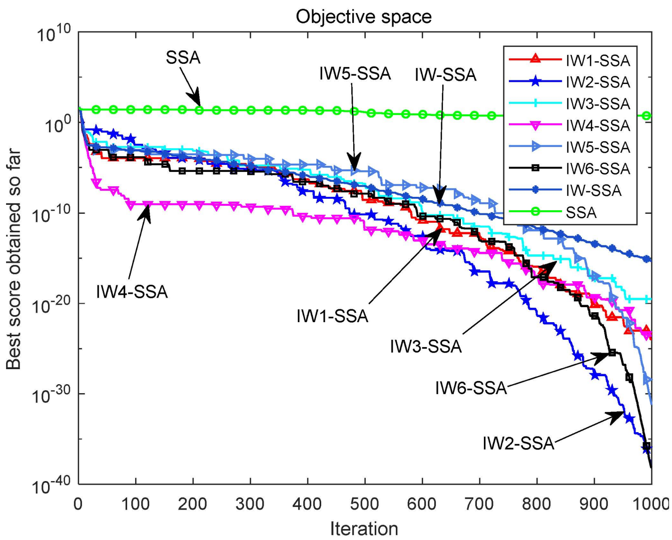
(14) F_{14}



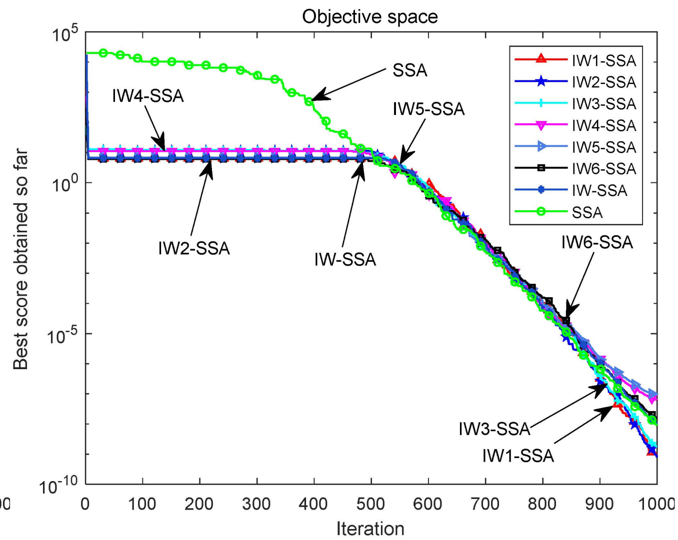
(15) F_{15}



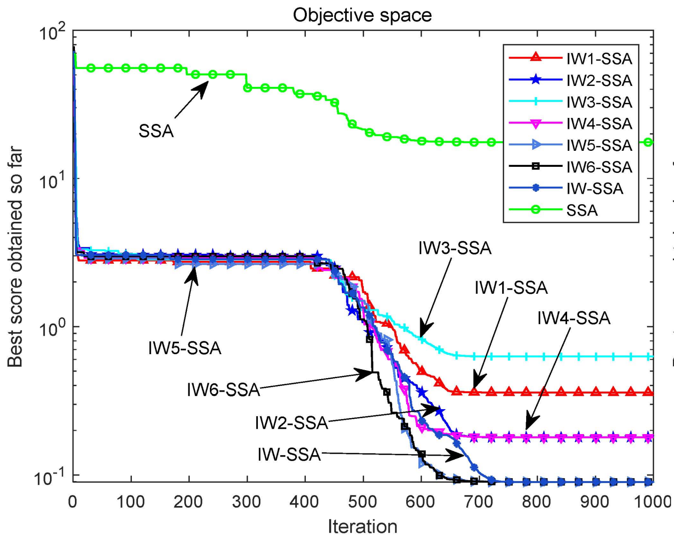
(16) F_{16}



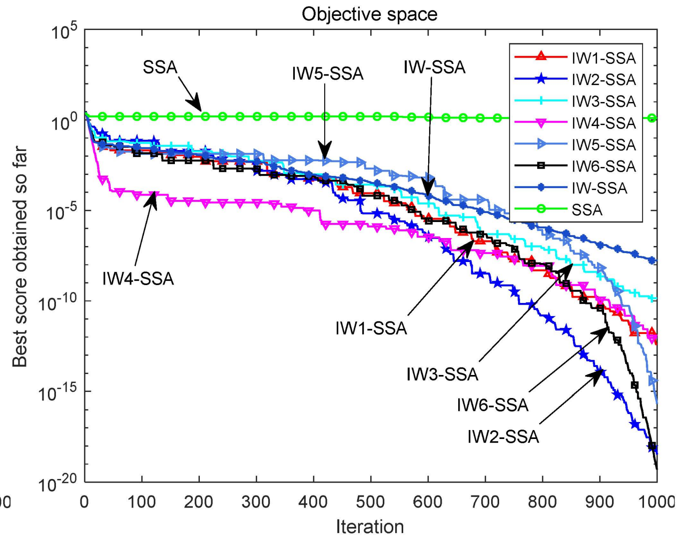
(17) F_{17}



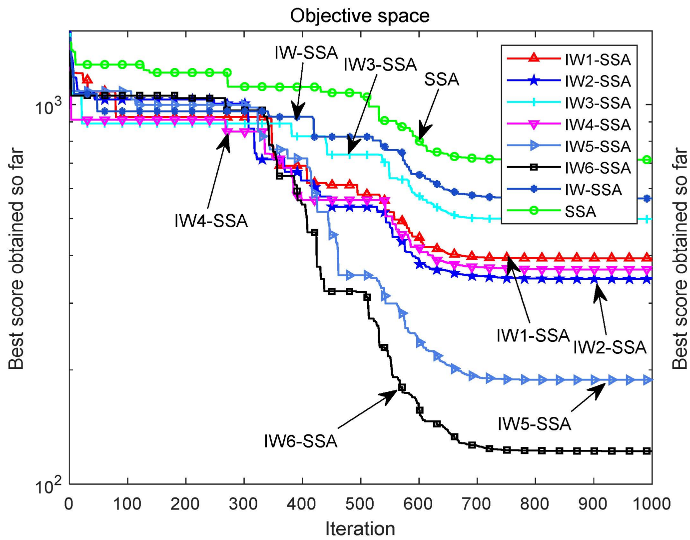
(18) F_{18}



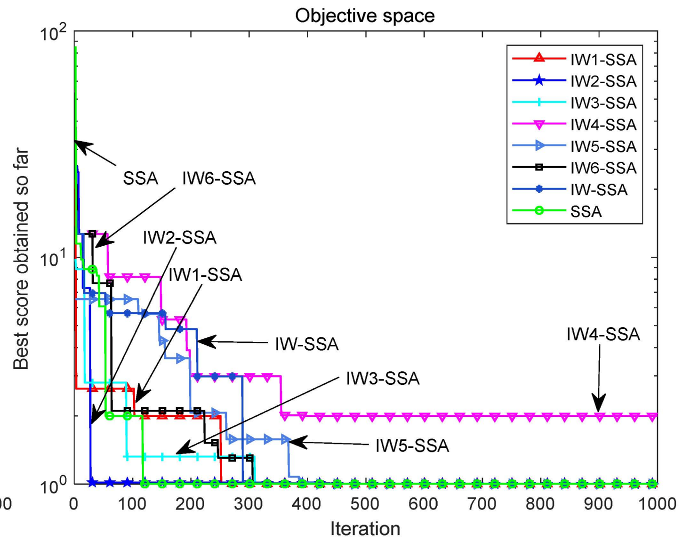
(19) F_{19}



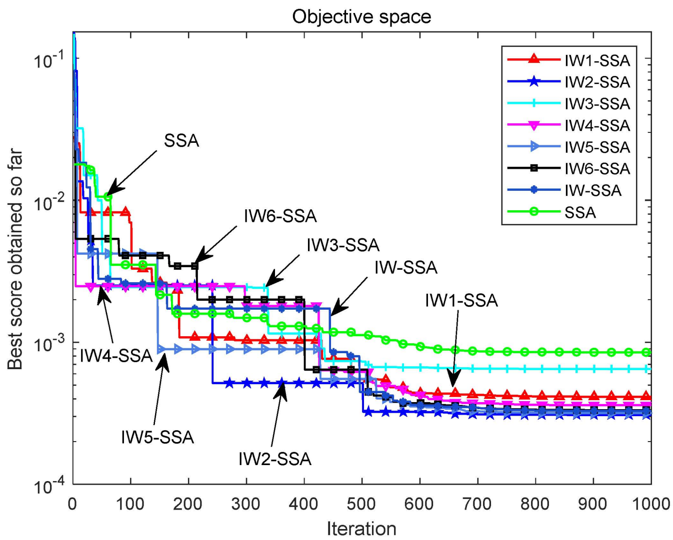
(20) F_{20}



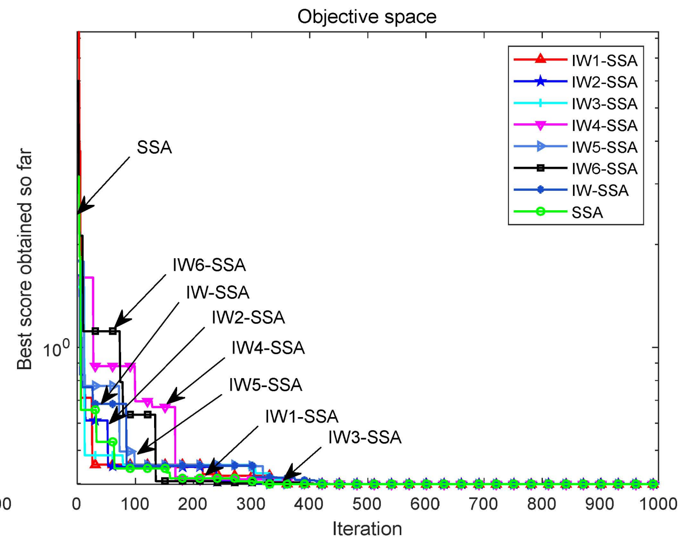
(21) F_{21}



(22) F_{22}



(23) F_{23}



(24) F_{24}

Fig. 4 Convergence curves of function optimization.

TABLE 2. QUANTITATIVE INDICATORS OF ALGORITHM OPTIMIZING FUNCTIONS

Function		IW1-SSA	IW2-SSA	IW3-SSA	IW4-SSA	IW5-SSA	IW6-SSA	IW-SSA	SSA
	Best	5.25E-24	5.52E-38	3.1E-20	3.44E-24	1.82E-31	8.77E-39	3.99E-15	0.028657
F_1	Ave	1.65E-23	1.11E-36	1.12E-19	2.04E-23	1.28E-30	1.1E-37	4.86E-15	0.701742
	Std	1.6E-23	9.1E-37	5.71E-20	2.38E-23	1.54E-30	2.08E-37	4.52E-16	0.548558
	Best	5.25E-24	5.52E-38	3.1E-20	3.44E-24	1.82E-31	8.77E-39	3.99E-15	0.028657
F_2	Ave	1.65E-23	1.11E-36	1.12E-19	2.04E-23	1.28E-30	1.1E-37	4.86E-15	0.701742
	Std	1.6E-23	9.1E-37	5.71E-20	2.38E-23	1.54E-30	2.08E-37	4.52E-16	0.548558
	Best	7.29E-47	2.56E-73	3.98E-39	4.96E-45	5.02E-60	4.85E-75	2.39E-28	41.12348
F_3	Ave	9.78E-45	2.4E-70	4.29E-37	8.58E-44	1.23E-58	1.31E-73	4.97E-28	397.7453
	Std	1.6E-44	6.45E-70	4.67E-37	1.02E-43	1.54E-58	1.89E-73	2.31E-28	259.1304
	Best	3.01E-24	2.13E-37	2.42E-20	2.52E-24	1.04E-31	5.11E-39	3.29E-15	0.957931
F_4	Ave	1.16E-23	1.14E-36	1.2E-19	3.14E-23	6.01E-31	6.13E-38	4.18E-15	5.824102
	Std	6.76E-24	1.37E-36	7.71E-20	3.47E-23	5.43E-31	7.72E-38	4.47E-16	3.014837
	Best	28.32494	28.2824	28.37659	28.31332	28.31379	28.35955	28.26724	23.72297
F_5	Ave	28.38628	28.36828	28.40951	28.38415	28.38772	28.41726	28.38462	149.7875
	Std	0.040314	0.04761	0.027578	0.045633	0.045969	0.040153	0.070064	258.8257
	Best	5.92E-06	2.66E-06	3.65E-08	1.68E-06	1.49E-06	8.31E-06	2.38E-07	0.026192
F_6	Ave	6.62E-05	7.06E-05	3.4E-05	7.94E-05	6.6E-05	4.88E-05	4.07E-05	0.076214
	Std	6.15E-05	6.09E-05	3.94E-05	6.59E-05	7.78E-05	3.63E-05	4.99E-05	0.027131
	Best	5.34E-41	5.43E-68	1.13E-33	7.46E-40	1.63E-55	1.69E-70	5.94E-23	0.514017
F_7	Ave	1.29E-39	2.34E-65	2.4E-31	1.09E-38	1.76E-53	1.37E-68	9.7E-23	960.127
	Std	2.12E-39	3.53E-65	4.61E-31	9.67E-39	4.11E-53	1.42E-68	2.2E-23	1108.555
	Best	8.07E-52	9.41E-77	5.47E-43	1.07E-49	8.19E-64	9.75E-79	9.55E-35	2.88E+08
F_8	Ave	2.51E-47	1.09E-73	2.68E-40	4.35E-47	9.65E-61	5.66E-76	1.48E-32	8.91E+13
	Std	5.88E-47	1.94E-73	4.27E-40	7.58E-47	1.22E-60	7.7E-76	3.8E-32	2.47E+14
	Best	8.96E-24	4.05E-36	1.76E-19	4.86E-23	1.89E-30	8.79E-38	3.79E-14	1.444358
F_9	Ave	1.53E-22	1.45E-35	9.97E-19	6.72E-22	2.21E-29	5.63E-37	4.71E-14	12.26086
	Std	1.15E-22	1.88E-35	7.02E-19	6.87E-22	2.34E-29	4.88E-37	6.66E-15	9.452518
	Best	1.3E-240	0	7.6E-198	1.3E-231	1.7E-305	0	3.6E-145	6.54E-36
F_{10}	Ave	9.7E-224	0	7.5E-187	6.7E-222	1.3E-298	0	2.2E-143	1.39E-24
	Std	0	0	0	0	0	0	2.9E-143	4.16E-24
	Best	5.6E-50	3.08E-77	5.75E-42	3.04E-49	3.89E-64	1.35E-78	2.37E-31	1.73E-11
F_{11}	Ave	1.94E-48	6.95E-75	2.72E-40	9.15E-48	6.9E-61	1.7E-76	3.29E-31	3.14E-11
	Std	2.72E-48	7.95E-75	3.38E-40	7.31E-48	1.75E-60	3.45E-76	7.05E-32	9.04E-12
	Best	0.66667	0.666669	0.666718	0.666725	0.666672	0.666699	0.666674	0.666668
F_{12}	Ave	0.667059	0.666867	0.667077	0.667033	0.667155	0.667047	0.66687	489.9772
	Std	0.000341	0.000159	0.000544	0.000294	0.000602	0.000407	0.000274	515.1308
	Best	-8496	-8786.64	-9347.1	-8511.84	-9055.16	-9022.17	-8740.42	-8869.5
F_{13}	Ave	-7419.46	-7826.89	-7790.37	-7675.4	-7849.52	-7758.54	-7585.91	-7957.54
	Std	605.3683	576.2334	914.975	659.9136	933.3411	763.7965	654.9152	383.5722
	Best	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	4.44E-15	0.931305
F_{14}	Ave	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	8.88E-16	4.44E-15	2.2348
	Std	0	0	0	0	0	0	0.22321	0.63445
	Best	6.78E-11	1.06E-10	9.82E-11	7.88E-11	1.05E-10	9.43E-11	9.25E-11	1.513352
F_{15}	Ave	0.001519	1.55E-10	1.7E-10	0.000168	1.86E-10	1.68E-10	1.33E-10	4.949989
	Std	0.004556	2.69E-11	7.11E-11	0.000504	6.24E-11	5.94E-11	2.11E-11	2.929871
	Best	0.010987	0.010987	1.1E-09	1.89E-09	1.04E-09	1.21E-09	1.79E-09	1.79E-09

F_{16}	Ave	2.079548	2.375057	1.484136	1.192927	0.894122	2.377254	1.782846	0.017491
	Std	1.354191	1.182035	1.481942	1.447776	1.356426	1.177681	1.449164	0.017804
	Best	6.97E-26	1.34E-38	2.13E-21	2.25E-25	2.97E-32	2.27E-39	4.14E-16	1.042736
F_{17}	Ave	8.53E-25	9.66E-38	8.12E-21	2.78E-24	9.07E-32	6.2E-39	4.93E-16	3.289686
	Std	7.8E-25	6.29E-38	7.89E-21	1.45E-24	5.36E-32	3.74E-39	4.86E-17	1.656366
	Best	5.42E-10	7.91E-10	1.3E-09	9.82E-10	7.06E-10	8.99E-10	1.08E-09	9.79E-10
F_{18}	Ave	2.41E-07	9.16E-09	2.61E-08	4.31E-07	3.06E-08	2.9E-08	6.69E-08	0.188543
	Std	6.19E-07	9.93E-09	3.23E-08	1.04E-06	5.21E-08	6.38E-08	8.69E-08	0.56563
	Best	1.94E-08	2.16E-07	0.089529	5.06E-07	6.64E-09	0.089528	8.39E-07	2.516747
F_{19}	Ave	0.188009	0.197656	0.303969	0.204927	0.232524	0.295451	0.177008	9.584603
	Std	0.109283	0.08901	0.151235	0.177468	0.154247	0.126932	0.140147	5.042292
	Best	3.68E-13	1.12E-19	3.54E-11	4.02E-13	1.4E-16	2.11E-20	1.29E-08	1.030016
F_{20}	Ave	6.32E-13	2.1E-19	6.69E-11	1.4E-12	3.24E-16	4.2E-20	1.5E-08	1.314713
	Std	1.81E-13	8.91E-20	2.65E-11	7.16E-13	2.4E-16	1.61E-20	1.34E-09	0.13531
	Best	214.5671	107.7467	223.1278	233.1786	242.4359	192.9462	335.6792	349.6402
F_{21}	Ave	345.6051	292.7521	339.6085	344.9242	348.8447	311.1972	435.5458	583.6005
	Std	107.0131	99.35754	96.22215	86.28318	80.95211	117.1058	66.50761	136.0176
	Best	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004
F_{22}	Ave	0.998004	1.196809	1.097407	0.998004	0.998004	0.998004	1.493834	0.998004
	Std	2.33E-16	0.397611	0.298208	1.79E-16	2.05E-16	2.98E-16	0.91324	1.72E-16
	Best	0.000307	0.000308	0.000307	0.000308	0.000307	0.000307	0.000307	0.00062
F_{23}	Ave	0.000341	0.000319	0.000355	0.000397	0.000369	0.000335	0.000463	0.000832
	Std	3.76E-05	1.66E-05	0.0001	0.000118	0.000102	3.76E-05	0.000286	0.000205
	Best	0.397887	0.397887	0.397887	0.397887	0.397887	0.397887	0.397887	0.397887
F_{24}	Ave	0.397887	0.397887	0.397887	0.397887	0.397887	0.397887	0.397887	0.397887
	Std	8.85E-15	1.60E-14	1.55E-14	1.4E-14	8.38E-15	2.46E-14	1.31E-14	2.02E-15

V. CONCLUSION

As a new biological heuristic algorithm, SSA can solve some optimization problems to some extent. However, in some specific cases, it can not show a good search ability and high optimization accuracy ability. In this paper, the adaptive inertia weights are firstly introduced. To represent the performance of the proposed strategy, 24 benchmark functions are adopted to be optimized and compared the performance of SSA. The optimization results show IW-SSA has a good optimization effect on most of the functions, and it reveal that proposed IW-SSA presented competitive and greater results compared to SSA. While increasing its convergence speed and accuracy, it ensures the balance of exploration and exploitation capabilities of SSA, and achieves best results.

REFERENCES

[1] I. Boussad, J. Lepagnot, and P. Siarry, "A Survey on Optimization Metaheuristics," *Information Sciences,* vol. 237, pp. 82-117, 2013.

[2] Ali Rıza Yıldız, "An Effective Hybrid Immune-hill Climbing Optimization Approach for Solving Design and Manufacturing Optimization Problems in Industry," *Journal of Materials Processing Technology*, vol. 209, no. 6, pp. 2773-2780, 2009.

[3] S. Arora, and P. Anand, "Chaotic Grasshopper Optimization Algorithm for Global Optimization," *Neural Computing and Applications*, vol. 31, no. 8, pp. 4385-4405, 2018.

[4] S. Mirjalili, A. H. Gandomi, and S. Z. Mirjalili, "Salp Swarm Algorithm: a Bio-inspired Optimizer for Engineering Design Problems," *Advances in Engineering Software*, vol. 114, pp. 163-191, 2017.

[5] G. I. Sayed, G. Khoriba, and M. H. Haggag, "A Novel Chaotic Salp Swarm Algorithm for Global Optimization and Feature Selection," *Applied Intelligence*, vol. 48, no. 10, pp. 3462-3481, 2018.

[6] Q. Fan, Z. J. Chen, and Z. H. Xia, "A Modified Salp Swarm Algorithm Based on Refracted Opposition-based Learning Mechanism and Adaptive Control Factor," *Journal of Harbin Institute of Technology*, vol. 52, no. 10, pp. 183-191, 2020.

[7] K. Gholami, and M. H. Parvaneh, "A Mutated Salp Swarm Algorithm for Optimum Allocation of Active and Reactive Power Sources in Radial Distribution Systems," *Applied Soft Computing*, vol. 85, pp. 105833, 2019.

[8] R. A. Ibrahim, A. A. Ewees, and D. Oliva, "Improved Salp Swarm Algorithm Based on Particle Swarm Optimization for Feature Selection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 8, pp. 3155-3169, 2019.

[9] A. E. Hegazy, M. A. Makhlouf, and G. S. El-Tawel, "Improved Salp Swarm Algorithm for Feature Selection," *Journal of King Saud University-Computer and Information Sciences,* vol. 32, no. 3, pp. 335-344, 2020.

[10] J. Wu, R. J. Nan, and L. Chen, "Improved Salp Swarm Algorithm Based on Weight Factor and Adaptive Mutation," *Journal of Experimental & Theoretical Artificial Intelligence,* vol. 31, no. 3, pp. 493-515, 2019.

[11] N. Panda, and S. K. Majhi, "Improved Salp Swarm Algorithm with Space Transformation Search for Training Neural Network," *Arabian Journal for Science and Engineering*, vol. 45, no. 3, pp. 2743-2761, 2020.

[12] M. Tubishat, N. Idris, L. Shuib, M. Abushariah, and S. Mirjalili, "Improved Salp Swarm Algorithm Based on Opposition Based Learning and Novel Local Search Algorithm for Feature Selection," *Expert Systems with Applications,* vol. 145, pp. 113122, 2020.

[13] Aaea B, Aq C, Mae D. "Enhanced salp swarm algorithm based on firefly algorithm for unrelated parallel machine scheduling with setup times," *Applied Mathematical Modelling,* vol. 94, pp. 285-305, 2021.

[14] Syed M A, Syed R. "Weighted salp swarm algorithm and its applications towards optimal sensor deployment," *Journal of King Saud University-Computer and Information Sciences,* vol.2, pp.1327-1337, 2019.
[15] Neggaz N , Ewees A A , Elaziz M A , et al. "Boosting Salp Swarm Algorithm by Sine Cosine algorithm and Disrupt Operator for Feature Selection," *Expert Systems with Applications,* vol.145, pp.113103, 2019.
[16] Y. Shi, and R. Eberhart, "Modified Particle Swarm Optimizer," IEEE World Congress on Computational Intelligence, vol. 112, no. 6, pp. 67-72, 1998.
[17] L. F. Zhu, J. S. Wang, and H. Y. Wang, "Data Clustering Method Based on Improved Bat Algorithm with Six Convergence Factors and Local Search Operators," *IEEE Access*, vol. 8, pp. 80536-80560, 2020.
[18] F. A. Hashim, E. H. Houssein, S. M. Mai, W. Al-Atabany, and S. Mirjalili, "Henry Gas Solubility Optimization: a Novel Physics-based Algorithm," *Future Generation Computer Systems*, vol. 101, pp. 645-667, 2019.
[19] L. X. Chen, and Y. M. Mu, "An Improved Salp Swarm Algorithm," *Application Research of Computers*, vol. 38, no. 6, pp. 1648-1652, 2021.