A Multi-Strategy Integrated Optimized Salp Swarm Algorithm and Its Application for Engineering Problems.

Shiyi Huang *

School of Science, Jiangnan University, Wuxi, China, 214122

* Corresponding Author Email: hsy13813702387@163.com

Abstract. Salp Swarm Algorithm has the advantages of few adjustment parameters and easy implementation, which has been applied in many fields, such as data mining, image processing, and engineering calculation. However, this algorithm is easy to fall into local optimization and sometimes the accuracy of optimization is not high. To solve the above problems in SSA, this paper proposes an orthogonal opposition-based adaptive slap swarm algorithm OOASSA. First, an orthogonal opposition learning strategy was introduced when updating the leader position to enhance the adequacy of global search and improve the algorithm's ability to get out of the local extreme value. Then, it was introduced adaptive inertia weight when updating the follower position and introduced an adaptive adjustment strategy in the ratio of a leader-follower number to ensure that the algorithm had a good global development ability in the early stage of iteration. Good local exploration ability in late iteration. In this paper, 10 test functions and 3 engineering optimization problems in CEC2017 are used for simulation experiments, and the proposed algorithm's global exploration, local mining, and local optimization capabilities are significantly better than many of the most advanced SSA variants.

Keywords: Salp swarm algorithm, multi-strategy integrated, Engineering problems, Optimized algorithm.

1. Introduction

With the arrival of the era of big data, many complex optimization problems arise in various fields, such as communication, economics and automation. By traditional optimization methods, the optimal solution or near-optimal solution can not be obtained in a short time. With the aim of solving complex optimization problems more effectively, a series of Swarm Intelligence Algorithms(SIA) have been proposed by researchers inspired by biological intelligent behavior and physical phenomena. Salp Swarm Algorithm (SSA) is one of them. It is an optimization algorithm proposed by Professor MIRJALILI et al.[1] in 2017 inspired by aggregative predatory behavior of salps. It has the advantages of easy implementation and less adjustment parameters, and has been applied in many fields. Such as data mining, image processing and engineering calculation. However, SSA still has some problems in the process of optimization, which is easy to fall into local extreme value and sometimes cannot get a high accuracy of optimization.

In response to the above problems, Hegazy et al.[2] proposed that fixed inertia weight can accelerate the convergence speed in the search process. Sayed et al.[3] combined SSA with chaos theory to propose a chaotic SSA. Wang et al.[4] proposed an improved SSA based on simplex method, which changed the random strategy, increased the diversity of the population, and strengthened the local search ability of the algorithm. CHEN Zhong-Yun et al.[5] divided the original population into three sub-groups: leader, follower and tail of the chain, and made them implement different renewal strategies. From the above improvement measures, it can be found that researchers mainly focus on balancing the global search ability and local development ability of SSA to achieve better results, but the problems of prematurity and low accuracy still exist.

To solve the above problems of SSA, this paper proposes an orthogonal opposition-based adaptive salp swarm algorithm (OOASSA). OOASSA first introduced the orthogonal opposition learning strategy when updating the leader position to enhance the adequacy of global search and improve the ability to escape from the local extremum. Then it introduced the adaptive inertia weight when updating the follower position, and introduced the adaptive adjustment strategy in the leader-follower ratio. To ensure that the algorithm has a good global development ability in the early stage of iteration and a good local exploration ability in the later stage of iteration. In this paper, 10 functions in CEC2017 [6] are used to test performance, and the performance of OOASSA is compared with four improved SSA. The results of Friedman test based on experimental data show that OOASSA can obtain better performance.

2. Basic salp swarm algorithm

Salps are barrel-shaped Marine plankton similar to jellyfish that move by inhaling and ejecting water [7]. Professor MIRJALILI et al. proposed a model of salps chain based on the aggregative predatory behavior of salps, in which the salp swarm was divided into two groups: leaders and followers. The leaders were the first individual in the chain, and the rest were followers. In SSA, the salps chain consists of N D-dimensional individuals. In the process of iteration, the location of the moving target of the salps chain is called the source of food, and the leader gradually approaches the food by updating his position. The updating formula is as follows:

$$x_j^1 = \begin{cases} F_j + c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right), c_3 \geq 0.5 \\ F_j - c_1\left(\left(ub_j - lb_j\right)c_2 + lb_j\right), c_3 < 0.5 \end{cases} \tag{1}$$

x_j^1 is the j-th dimension of the first individual of the salps chain, F_j is the j-th dimension of the food, ub_j is the upper limit of search in the j-th dimension, and lb_j is the lower limit ofin the j-th dimension. c_2 and c_3 are random numbers subject to $N(0,1)$. c_1 is an important adaptive adjustment parameter of SSA, defined by the following formula:

$$c_1 = 2e^{-\left(\frac{4t}{max\_iter}\right)^2} \tag{2}$$

t is the current number of iterations and max_iter is the defined maximum number of iterations. With the change of iteration number, c_1 can dynamically weigh the global exploration ability and local development ability of SSA.

The formula for updating follower locations is as follows:

$$x_j^i(k+1) = \frac{1}{2}\left(x_j^i(k) + x_j^{i-1}(k)\right) \tag{3}$$

$x_j^i(k)$ is the j-th dimension of the i-th follower at the k-th iteration ($i \geq 2$).

SSA has the advantages of less adjustment parameters. However, SSA still has some problems in the process of optimization, which is easy to fall into local extremum and sometimes a high accuracy of optimization cannot be obtained.

3. Orthogonal opposition-based adaptive slap swarm algorithm

3.1. Orthogonal opposition-based learning

The leader updates its position only under the guidance of the food. The location update mechanism is relatively single, and the algorithm is easy to fall into local extreme value. With the purpose of enhancing the ability to escape from local optimum, an orthogonal opposition-based learning strategy (OOBL) [8] was proposed and applied to generate new candidates of leaders.

Orthogonal opposition-based learning strategy is a method that uses the current solution and the opposite solution to find the best combination of the levels of different factors with fewer experiments through orthogonal experimental design. This paper integrates opposition-based learning and orthogonal learning, and proposes an orthogonal opposition-based learning strategy to improve SSA. The basic idea of the strategy is that the current position and its opposite position are used to construct orthogonal candidate solutions according to the orthogonal table, and then each candidate solution is evaluated, and finally the best orthogonal combination is obtained. In this way, the information of each dimension in the individual and the opposite individual is fully utilized and the best combination is found. The formula for generating the reverse solution is:

$$x\_back^i = ub + lb - random(0,1)\, x^i \tag{4}$$

x^i is the position of the i-th leader, x_back^i is the reverse position of the i-th leader, ub and lb are respectively the upper and the lower limits of the search area, and $random(0,1)$ is a random number obeying N(0,1).

Orthogonal tables are used in the construction of orthogonal candidate solutions [9], which is defined as $L_M(Q^K)$. L is the symbol of the orthogonal matrix, Q is the number of levels and K is the number of factors. K corresponds to the number of columns of the orthogonal matrix, which is also the dimension of the problem, and M corresponds to the number of rows of the orthogonal matrix, which is also the number of experimental solutions. The following takes three factors and two levels as an example, and its orthogonal table is shown in equation (5):

$$L_4(2^3) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{bmatrix} \tag{5}$$

For the orthogonal opposition learning strategy in this paper, it is necessary to construct the orthogonal table of two levels of D factor (D is the population dimension).With the increase of the number of dimensions, the number of trial evaluations increases, but the convergence performance is better.

3.2. Adaptive inertia weight

According to the follower position update model, followers of SSA learn from the previous individual and their own characteristics to update the position. It is easy to fall into local extreme value. To optimize the follower position updating mechanism, this paper introduces a nonlinear decreasing adaptive inertia weight [10] to evaluate the influence of the previous individual on the current follower. The new follower location update formula is

$$x_j^i(k+1) = \frac{1}{2}\left[x_j^i(k) + w(t)x_j^{i-1}(k)\right] \tag{6}$$

$$w(t) = 1 - \frac{1}{1+\frac{1}{\sqrt{t}}} \tag{7}$$

From the analysis of formula (7), we can see that the value of inertia weight $w(t)$ presents a nonlinear decreasing trend with the increase of the number of algorithm iterations (from 1 gradually decreasing to approaching 0).

3.3. Adaptive adjustment strategy

In SSA, leaders and followers always make up half of the population. As a result, in the early stage, the proportion of leaders performing global search is too low, the proportion of followers is too high, the leader can not effectively carry out global search. However, in the later iteration, the proportion of followers is too low, the local development is not enough, and the precision is not high [11]. In response to this problem, this paper introduces the leader-follower adaptive adjustment strategy [12]. With the increase of the number of iterations, the number of leaders adaptively decreases and the number of followers adaptively increases. In the early stage, a strong global search ability can be maintained and in the later stage, the local search ability is gradually enhanced which improves the convergence accuracy of the algorithm as a whole. The adaptive factor r is

$$r = 0.75 - \frac{0.25t}{max\_iter} \tag{8}$$

By analyzing formula (8), the value of r shows a linear decreasing trend (from gradually decreasing to 25) as the number of iterations increasees.

The algorithm flow chart is shown in Figure 1.

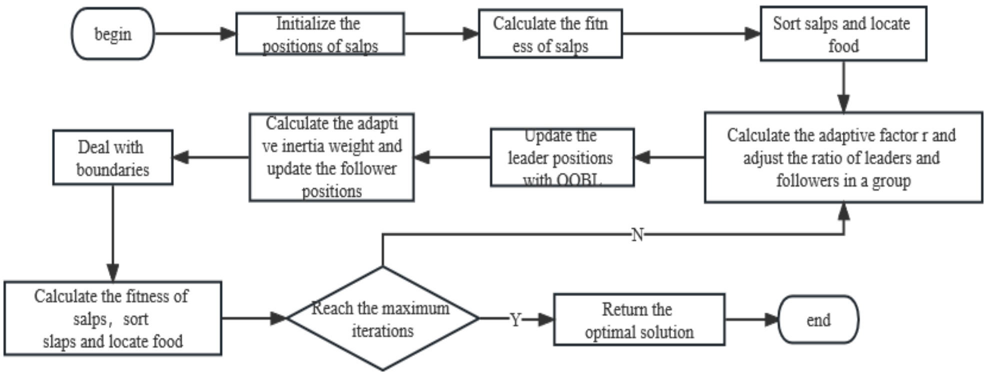

Figure 1 The flow chart of the algorithm

Combining the above improvement strategies, OOASSA pseudo-code is as follows:

```
begin
  Set algorithm parameters: population size N, maximum number of iterations max_iter,
individual dimension D.
  Initialize the positions of salps x_j^i (i=1,2,...,N, j=1,2,...,D), calculate the fitness of
salps by f(x), sort salps and locate food.
  while(t<max_iter+1)do
        Calculate the adaptive factor r by equation (8).
  for i=1 to N do
  if i≤ rN  then
              for j=1 to D do
                   Update the position of leader x_j^i by equation (1)
              end for
              Generate opposite solution of leader x^i by equation (4)
                  Orthogonal experiment.
          else
              Calculate the adaptive inertia weight w by equation (7).
              Update the position of follower x^i by equation (6)
          end if
      end for
      for i=1 to N do
          Deal with boundaries, calculate calculate the fitness of salps and sort salps.
          Locate food.
      end for
      t=t+1
   end while
  end
```

4. Experiment

4.1. Parameter setting

The experimental environment and parameter settings are shown in Table 1.

Table 1 Parameter setting

Operating system	Memory	CPU	Experimental platform
Windows 10	16GB	AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz	MATLAB R2021a

4.2. Verification of algorithm validity

In this paper, we improve SSA in three aspects, namely, orthogonal opposition learning strategy, adaptive inertia weight and adaptive adjustment strategy. With the aim of verifying the effectiveness of the three improved strategies, they are embedded in SSA respectively, and OOBL-SSA, AIW-SSA and AA-SSA algorithms are obtained. This section selects 6 basic test functions in Table 2 for simulation experiments to compare OOASSA, OOBL-SSA, AIW-SSA, AA-SSA, and SSA. The population size was set to 30, the maximum number of iterations was set to 1000, and the dimension was set to 10. Each algorithm was independently run 30 times on each function.Table 3 respectively recorded their minimum(min), maxmum(max), average value (avg) and standard deviation(std). Friedman test [13] was used to rank the algorithms, as shown in Table 4.

Table 2 Function expression

Function expression	Dimension	Value range	Global minimum
$f_1(x)=\sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$f_2(x)=\sum_{j=1}^{n}\left(\sum_{i=1}^{j} x_i\right)^2$	10	[-100,100]	0
$f_3(x)=\max_i\left\{\lvert x_i\rvert,1\le i\le n\right\}$	10	[-100,100]	0
$f_4(x)=\sum_{i=1}^{n}(x_i+0.5)^2.$	10	[-100,100]	0
$f_5(x)=\sum_{i=1}^{n} ix_i^4+random[0,1)$	10	[-1.28,1.28]	0
$f_6(x)=\sum_{i=1}^{n}\left(x_i^2-10cos(2\pi x_i)+10\right)$	10	[-5.12,5.12]	0

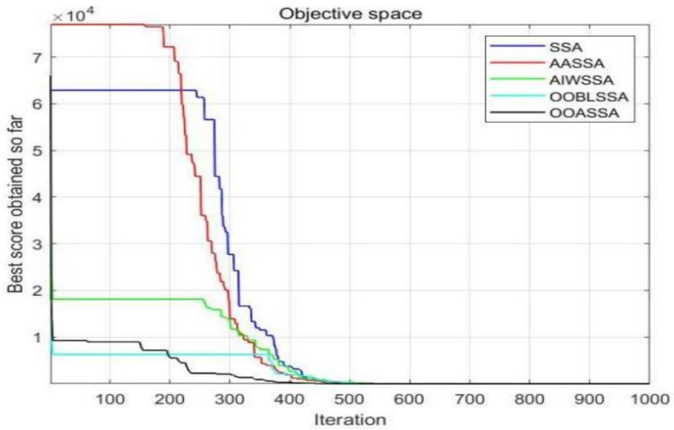

Figure 2 Convergence diagram of function 1

Table 3 Statistics data

Function	Statistics	SSA	AA-SSA	AIW-SSA	OOBL-SSA	OOASSA
f_1	avg	2.72E-08	1.40E-08	1.29E-08	5.24E-09	1.07E-09
	std	9.44E-09	3.35E-09	3.30E-09	4.04E-10	1.14E-09
	min	1.62E-08	8.52E-09	7.91E-09	3.38E-09	4.01E-10
	max	5.89E-08	2.09E-08	2.03E-08	8.09E-09	2.25E-09
f_2	avg	7.61E-09	2.71E-09	2.29E-09	1.70E-09	1.35E-09
	std	4.65E-09	2.25E-09	1.53E-09	9.29E-10	7.87E-10
	min	2.76E-09	5.57E-10	4.94E-10	4.21E-10	3.11E-10
	max	2.23E-08	1.04E-08	7.15E-09	4.98E-09	3.27E-09
f_3	avg	2.08E-05	2.07E-05	1.76E-05	1.72E-05	1.26E-05
	std	8.50E-06	5.71E-06	4.21E-06	4.23E-06	4.22E-06
	min	1.28E-05	1.16E-05	1.11E-05	9.22E-06	6.23E-06
	max	4.84E-05	3.92E-05	3.32E-05	2.44E-05	3.16E-05
f_4	avg	1.28E-09	8.07E-10	7.69E-10	6.71E-10	6.27E-10
	std	4.59E-10	3.16E-10	2.36E-10	2.28E-10	1.51E-10
	min	5.16E-10	3.65E-10	3.24E-10	2.53E-10	1.76E-10
	max	2.52E-09	1.59E-09	1.08E-09	1.07E-09	1.02E-09
f_5	avg	0.010753	0.007018747	0.006760328	0.001933758	0.001173734
	std	0.007787	0.005831102	0.004912189	0.002229347	0.001036112
	min	0.001637	0.001236233	0.000873576	0.000109604	2.15E-05
	max	0.035214	0.025956242	0.018900007	0.008519782	0.004176989
f_6	avg	30.41247	21.49107055	18.30721318	7.296343497	6.798873267
	std	11.67542	9.236449471	8.064451062	7.792019209	5.689141039
	min	9.949585	3.979836229	1.989918114	5.00E-10	1.73E-10
	max	56.71235	44.77300099	41.78792578	36.81334364	28.85367622

Table 4 Friedman test results

P-Value		SSA	AA-SSA	AIW-SSA	OOBL-SSA	OOASSA
1.8983E-30		4.93	4.00	2.93	1.94	1.21

As can be seen from convergence Figure 2, compared with SSA, although the effect of AA-SSA is not obvious in the early stage, the convergence speed is faster in the later stage, and the final convergence accuracy is also higher. In addition, the convergence speed of OOBL-SSA, AIW-SSA and OOASSA is always higher than that of SSA during the whole iteration, and the final convergence accuracy is also higher than that of SSA. Among them, although the convergence speed of OOASSA is not as fast as that of OOBLSSA in the early stage, it will not easily fall into local extreme value in the later stage, so the final optimal solution has the highest precision. Table 3 shows that the average value of OOASSA on the 6 functions is always optimal, which is consistent with the above analysis, but the standard deviation is not always optimal. Based on the above analysis, the three improvement strategies all have certain effects, and the simultaneous action of the three improvement strategies can further improve the optimization performance of the algorithm, but the effect is weakened in terms of stability. In addition, Table 4 presents Friedman test results, which show that P-Value is much less than 0.01, indicating that there are significant differences between SSA, AA-SSA, AIW-SSA, OOBL-SSA and OOASSA. OOASSA ranked first, followed by OOBL-SSA, AIW-SSA, AA-SSA and SSA, which further verified the effectiveness of the improved strategy and the obvious advantages of OOASSA over the single-strategy improved algorithm.

4.3. Verification of algorithm superiority

OOSSA was used to solve 10 test functions in CEC2017[6] (including 2 Unimodal Functions F_1 and F_3,3 Simple Multimodal Functions F_5, F_8 and F_9 ,3 Hybrid Functions F_{11}, F_{12} and F_{16},2 Compostion Functions F_{21} and F_{22}), and compared with SSA[1], CASSA[14], LECUSSA[15], MSNSSA[5], and RCSSA [16]. The population size is set to 30, the maximum number of iterations is set to 5000, and the dimension is set to 100. Each algorithm is independently run on each function 30 times. Table 5 respectively recorded their minimum(min), maxmum(max), average value (avg) and standard deviation(std). Friedman test was used to rank the algorithms, as shown in Table 6.

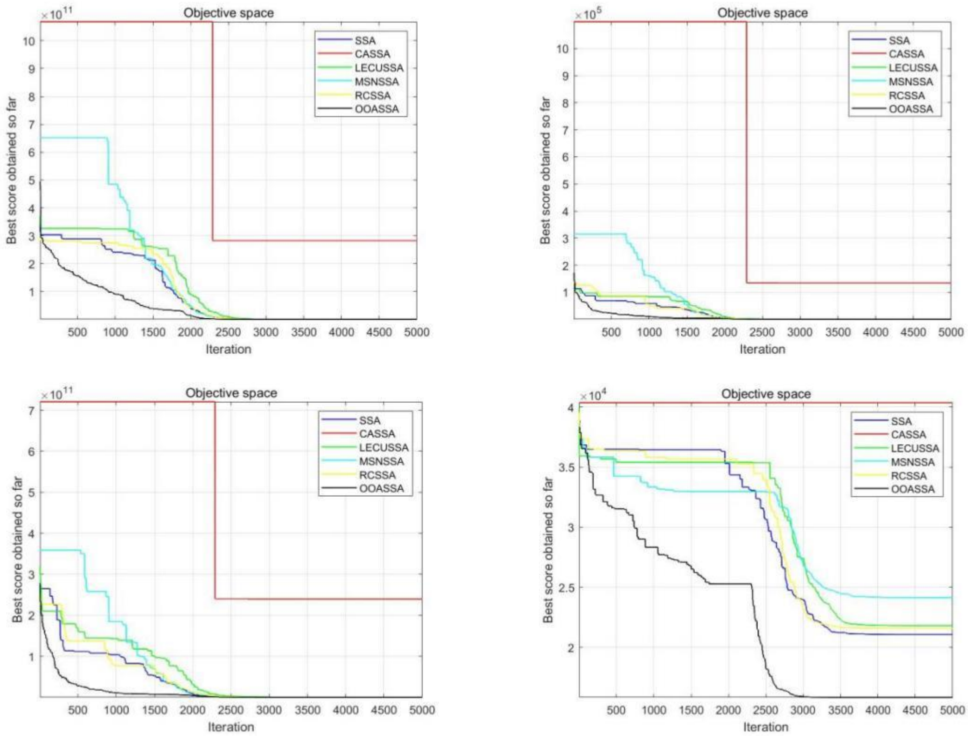

Figure 3 Convergence diagram of different functions

Table 5 Ststistics data

Function	Min	Statistics	SSA	CASSA	LECUSSA	MSNSSA	RCSSA	OOASSA
F_1	100	avg	12997.0701	2.926E+11	22443.15832	19486.05404	13980.84749	9849.571699
		std	17272.5489	3676197527	17915.87827	20557.7957	14667.71257	10801.14599
		min	277.444662	2.81647E+11	368.3624925	136.1442705	117.7761763	100.0074384
		max	61467.8556	2.97615E+11	61414.58117	60883.43269	46730.41018	35467.09554
F_3	300	avg	162228.527	2.08762E+12	166373.7019	215256.6541	161219.0206	151298.6279
		std	28611.0977	6.77266E+12	29456.59671	21415.51254	22390.33717	25412.55925
		min	106403.252	1680832.06	105243.1457	176944.3782	117167.9812	97279.13278
		max	228480.917	3.17739E+13	217388.3494	260259.3148	199801.2285	214428.6616
F_5	500	avg	1285.18202	2372.78293	1291.953434	1448.353517	1203.846	1153.699604
		std	96.5968599	28.7197466	120.4531267	84.13456735	109.0664774	104.8216931
		min	1060.31976	2253.44941	1057.172307	1239.247915	993.497267	985.6744244
		max	1496.93802	2412.93056	1596.427343	1603.391563	1406.397493	1400.230578
F_8	800	avg	1595.76547	2820.58045	1562.003467	1870.536519	1490.880647	1433.127387
		std	165.490397	27.4581220	109.5387697	96.7899073	96.24339499	112.8921992
		min	1275.58766	2746.85520	1346.229015	1691.472752	1276.582382	1249.719149
		max	2043.67888	2849.1543	1818.136953	2048.648478	1681.522763	1716.346328
F_9	900	avg	25250.3819	108767.338	25949.84131	25317.54441	23057.17256	21644.42893
		std	2695.54104	9087.20943	3211.425582	3294.955479	916.9058215	2473.890835
		min	18786.3019	92894.8137	17519.16684	20125.06453	19723.71059	16565.24269
		max	30118.5989	137735.933	31654.20521	31949.71382	24092.34787	26739.4979
F_{11}	1100	avg	2820.30673	3714818777	2967.557632	3452.058465	2816.955145	2813.00051
		std	272.887697	1904539600	370.5783981	501.3748555	256.0163767	212.706290
		min	2332.27268	1339375.08	2373.697039	2502.672776	2239.967682	2400.567873
		max	3401.32582	1.04383E+11	4242.284158	4803.09588	3316.387643	3249.23040
F_{12}	1200	avg	165330825	2.50561E+11	172190167	259263949.3	155122438.2	104811562.2
		std	73235822.8	5096869295	68925171.5	102154094.2	62766111.19	55980724.02
		min	34340497.8	2.39456E+11	85777581.41	70950260.72	29645734.54	14175845.31
		max	401591522	2.57697E+11	311087587.3	507960569.6	273647704.4	241747260.1
F_{16}	1600	avg	6008.41568	32965.82518	6362.337948	7391.052594	5734.560422	5599.47599
		std	761.7549	1209.817904	760.9742098	879.7717017	843.2375013	699.839193
		min	4414.67083	30372.69646	5015.011465	6032.451882	4258.790914	3927.60972
		max	7583.42602	35921.96745	8330.720805	9541.983747	7317.643631	6639.98282
F_{21}	2100	avg	3034.76861	7034.73839	3090.827558	3525.427736	2960.881684	2925.880409
		std	115.660646	132.8338414	142.1316716	233.9817977	112.3849823	98.60270476
		min	2819.01231	6794.691228	2900.04491	2975.198637	2768.453485	2727.32798
		max	3458.37711	7311.012882	3399.577456	4045.144585	3242.632881	3104.145615
F_{22}	2200	avg	17878.6269	38691.18344	18276.40403	19625.91492	17731.06855	17674.07957
		std	1484.97566	1017.989215	1852.412475	2295.84033	1778.071787	2126.673964
		min	14692.1844	36292.78108	15258.93222	15832.00367	15146.90895	13970.76454
		max	21062.3941	40358.82353	21808.73586	23571.22582	21583.32125	24138.45998

Table 6 Friedman test results

P-Value	SSA	CASSA	LECUSSA	MSNSSA	RCSSA	OOASSA
4.0793E-33	3.18	5.61	3.59	4.55	2.32	1.45

Figure 3 shows that OOASSA has the fastest convergence speed and the highest convergence precision, and the optimization effect is better than SSA and other improved algorithms. Table 5 shows that OOASSA has the best optimization effect on most functions, which is consistent with the conclusion of convergence graph. However, OOASSA has the smallest standard deviation on almost none of the 10 functions, indicating that OOASSA's optimization stability is not as good as other algorithms. According to the Friedman test results in Table 6, first of all, P-Value is much less than 0.01, indicating that there are significant differences between the six algorithms compared in the experiment. The second result shows that OOASSA ranks first, followed by RCSSA, SSA, LECUSSA, MSNSSA and CASSA. It shows that LECUSSA, MSNSSA and CASSA is not as good as SSA in solving high-dimensional function optimization problems. The improvement strategy of RCSSA has a certain effect, and the solution performance is better than SSA, but inferior to OOASSA.

4.4. Engineering application

The design problems of pressure vessel, three-bar truss and tension spring are solved by OOASSA and compared with other improved SSA. The population size is set to 30, the maximum number of function evaluations is set to 5000, and the optimal solution is obtained after 30 independent runs.

4.4.1 Pressure vessel design issue

Pressure vessel design issue[17] is aim to obtain the minimum production cost of the pressure vessel (pairing, forming and welding), with a lid on both ends of the pressure vessel and a hemispherical cap on one end of the head. Table 7 presents the results.

Table 7 Result of pressure vessel design issue

Algorithm	x_1	x_2	x_3	x_4	Cost
SSA	1.348773084	0.666704884	69.88451844	10.98871358	9019.924584
CASSA	0.199661789	0.337009866	10	10	100081.522
LECUSSA	1.3005503	0.642862663	67.38602182	11.46515	8138.679103
MSNSSA	1.300889345	0.643030277	67.40359298	10.24909546	8071.986161
RCSSA	1.308106157	0.646595816	67.77716245	15.53754836	8524.019803
OOASSA	1.300564758	0.642867764	67.3865559	10.23274481	8005.961

4.4.2 Three bar truss design issue

Three-bar truss design issue[17] is aim to obtain the minimum volume of the three-bar truss by adjusting the cross-sectional area. Table 8 presents the results.

Table 8 Result of three bar truss design issue

Algorithm	x_1	x_2	Cost
SSA	0.788663481	0.408281252	263.8958435
CASSA	0.791236919	0.401710797	263.966676
LECUSSA	0.788725634	0.408105475	263.8958453
MSNSSA	0.788687632	0.408212944	263.8958435
RCSSA	0.78855854	0.408578171	263.8958535
OOASSA	0.788671963	0.408257262	263.8958434

4.4.3 Tension spring design issue

Tension spring design issue [17] is aimed to obtain the minimum weight of the tension spring under the constraints of minimum deflection, vibration frequency and shear stress. Table 9 presents the results.

Table 9 Result of tension spring design issue

Algorithm	x_1	x_2	x_3	Cost
SSA	0.052608189	0.379234898	10.08206776	0.012681081
CASSA	0.05	0.310521163	15	0.013197149
LECUSSA	0.052272766	0.370923257	10.50227965	0.012671389
MSNSSA	0.054317908	0.423319687	8.238099196	0.012787153
RCSSA	0.05	0.315469731	14.29028268	0.012847728
OOASSA	0.052171307	0.368430552	10.63392688	0.01266944

It can be seen from Table 7, 8 and 9 that OOASSA obtains the lowest design cost in these three problems, which is better than the other five improved algorithms, proving that OOASSA can be effectively applied to engineering optimization problems.

5. Conclusions

Salp Swarm Algorithm has the advantages of easy implementation and less adjustment parameters, which has been applied in many fields. However, this algorithm is easy to fall into local extreme value and sometimes cannot get a high accuracy of optimization. In response to the above problems in SSA, this paper presents an orthogonal opposition-based adaptive slap swarm algorithm. Firstly, the orthogonal opposition learning strategy is introduced when the leader position is updated, which enhances the global search ability and the ability to escape from the local optimal solution. Then, the adaptive inertia weight is introduced when the follower position is updated, and the adaptive adjustment strategy is introduced in the ratio of the number of leaders and followers, so that the algorithm has a good global development ability in the early iteration stage and a good local exploration ability in the late iteration stage. 10 CEC2017 benchmark functions and 3 engineering optimization problems were used to conduct simulation experiments. And the proposed algorithm's ability to explore globally, exploit locally and jump out of local optimality is analyzed in detail. The results show that the proposed variant can effectively improve the performance of salp swarm algorithm, and its overall performance is significantly better than many of the most advanced variants.

References

[1] Mirjalili S, Gandomi H A, Mirjalili Z S, et al. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems [J]. Advances in Engineering Software, 2017,114.

[2] Hegazy A E, Makhlouf M A, Eltawel G S. Improved salp swarm algorithm for feature selection. Journal of King Saud University–Computer and Information Sciences, 2018, 6(3): 1-10.

[3] Sayed G I, Khoriba G, Haggag M H. A novel chaotic salp swarm algorithm for global optimization and feature selection. Applied Intelligence, 2018, 48(3): 1−20.

[4] WANG D Y, ZHOU Y Q, JIANG S Q, et al. A simple method-based salp swarm algorithm for numerical and engineering optimization. Intelligent Information Processing. Berlin, German: Springer, 2018: 189-211.

[5] Chen Z, Zhang D, Xin Z. A symbiotic nonuniform Gaussian variational bottle sea squirt group algorithm for multisubgroups [J]. Journal of Automation, 2022, 48(05).

[6] N. H. Awad, M. Z. Ali, P. N. Suganthan, J. J. Liang and B. Y. Qu. Definitions of CEC2017 benchmark suite final version updated.

[7] Everton T G, Barbara S B, Xiomara D G F, et al. Thaliacean community responses to distinct thermohaline and circulation patterns in the Western Tropical South Atlantic Ocean [J]. Hydrobiologia, 2022, 849(21).

[8] Houssein, H. E, Helmy, et al. An efficient orthogonal opposition-based learning slime mould algorithm for maximum power point tracking [J]. Neural Computing and Applications, 2022, 34(5).

[9] Xiangyu W, Yongtao W, Peng J, et al. Orthogonal Experimental Design Based Nozzle Geometry Optimization for the Underwater Abrasive Water Jet [J]. Machines, 2022, 10(12).

[10] Maodong L, Guanghui X, Yuanwang F, et al. Improved whale optimization algorithm based on variable spiral position update strategy and adaptive inertia weight [J]. Journal of Intelligent & Fuzzy Systems, 2022,42(3).

[11] QIN J, HUANG C, LUO Y. Adaptive Multi-swarm in Dynamic Environments [J]. Swarm and Evolutionary Computation, 2021, 63(4): 100870.

[12] Hongwu Q, Lizheng W, Muxuan S, et al. Research on Grey Wolf Optimization Algorithm Based on Adaptive Adjustment Strategy [J]. Journal of Physics: Conference Series, 2022, 2395(1).

[13] Tomasz N, Agnieszka S, Ahmad H, et al. Evaluation of the Effects of Coal Jigging by Means of Kruskal–Wallis and Friedman Tests [J]. Energies, 2023, 16(4).

[14] Tong, He Q, Chen J. Adaptive Bottle Sea Sheath Swarm Algorithm Based on Chaotic Mapping [J]. Journal of Sensing Technology, 2021, 34(01): 41-48.

[15] Zhang Y, Qin L. Improved Bottle Sea Sheath Swarm Algorithm Based on Levy Flight Strategy [J]. Computer Science, 2020, 47(07): 154-160.

[16] Fan Q, Chen Z, Xia Z. An improved bottled sea squirt swarm algorithm based on refractive inverse learning mechanism with adaptive control factors [J]. Journal of Harbin Institute of Technology, 2020, 52(10): 183-191.

[17] Farouq Z, Saad H, Ramdane M. The Solar System Algorithm: A Novel Metaheuristic Method for Global Optimization [J]. IEEE ACCESS, 2021,9.