A Simplex Method-Based Salp Swarm Algorithm for Numerical and Engineering Optimization

Dengyun Wang[1], Yongquan Zhou[1,2](✉), Shengqi Jiang[1], and Xin Liu[1]

[1] College of Information Science and Engineering,
Guangxi University for Nationalities, Nanning 530006, China
yongquanzhou@126.com

[2] Guangxi High School Key Laboratory of Complex System
and Computational Intelligence, Nanning 530006, China

Abstract. Salp Swarm Algorithm (SSA) is a novel meta-inspired optimization algorithm. The main inspiration of this algorithm is the swarming behavior of salps when navigating and foraging in the ocean. This algorithm has already displayed the strong ability in solving some engineering design problems. This paper proposes an improved salp swarm algorithm based on simplex method named as simplex method-based salp swarm algorithm (SMSSA). The simplex method is a stochastic variant strategy, which increases the diversity of the population and enhances the local search ability of the algorithm. This approach helps to achieve a better trade-off between the exploration and exploitation ability of the SSA and makes SSA more robust and faster. The proposed algorithm is compared with other four meta-inspired algorithms on 4 benchmark functions. The proposed algorithm is also applied to one real-life constrained engineering design problems. The experimental results have demonstrated the MSSSA performs better than the other competitive meta-inspired algorithms.

Keywords: Simplex method · Salp swarm algorithm · Benchmark function
Global optimization

1 Introduction

The purpose of the optimization aims at finding the best possible solution(s) for given problems. In the real world, a lot of problems can be considered as optimization problems. With the scale and complexity of the problem escalating, we need new optimization techniques more than ever. Over the past few decades, many new meta-heuristic techniques have been proposed to solve these optimization problems and become very popular. These meta-heuristics like a black box just need to looking at the inputs and outputs. In the recent years, some well-known meta-heuristic algorithms are proposed in this field such as Different Evolution [1], Particle Swarm Optimization (PSO) [2], Bat algorithm (BA) [3], Moth-Flame Optimization (MFO) [4], Grey Wolf Optimization (GWO) [5], and Cuckoo Search (CS) [6]. Most of these algorithms are

Z. Shi et al. (Eds.): IIP 2018, IFIP AICT 538, pp. 150–159, 2018.
https://doi.org/10.1007/978-3-030-00828-4_16

derived from a various natural phenomenon. These algorithms are widely used in a variety of scientific and industry fields.

The salp swarm optimization algorithm is proposed by Mirjalili et al. [7]. The salps swarm optimization has been shown the powerful results, when it compared to other state-of-the-art met-heuristic optimization algorithms. The author has been applied this algorithm to engineering design problems such as welded design problem, and achieved good results. Although, the SSA has proved good performance compared with some traditional algorithms, it still has some drawbacks such as spend too long time in research phase, and need to enhance the ability of convergence speed and calculation accuracy. To overcome the above problems, a simplex method-based salp swarm algorithm is proposed. The simplex method [8] has the strong ability to avoid local optimum and enhance the ability of searching the global optimum. In this work an improved version of the SSA is based on simplex method named SMSSA which purpose aimed at enhance the precision of the convergence of basic SSA.

The rest of paper is organized as follows: In the Sect. 2 presents a briefly introduce of the original SSA algorithm and Simplex method. The detailed description of the SMSSA algorithm is introduced in the Sect. 3. In the Sect. 4, through a range of tests to demonstrate the superior performance of SMSSA and compared with other well-known five meta-heuristic algorithms (including the original algorithm SSA) via fourteen benchmark functions. In the Sect. 5, SMSSA employed to solve one engineering design problems. The analysis and discussion of the results are provided in Sect. 6. In the last section, the conclusion of the work will provided.

2 Related Works

In this part, a briefly background information about the salp swarm algorithm will be provided. The salp swarm algorithm [9] is a new meta-heuristic optimization algorithm that proposed by Seyedali Mirjalili. The SSA inspired from the behavior of the salps foraging and navigating in the ocean. Salps is one of the family of Salpidae and the body is transparent barrel-shaped. The tissues of salp are very similar to jelly fish. Salps navigate in the water by using water pumped through body to get propulsion to move forward [10].

In the SSA algorithm, the mathematically model of the salp chains are divided to two groups: leader groups and follow groups. The leader salp position updating formula as follows:

$$X_j^i = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j) & c_3 \geq 0.5 \\ F_j - c_1((ub_j - lb_j)c_2 + lb_j) & c_3 < 0.5 \end{cases} \tag{1}$$

where X_j^i indicate the position of the leader salp groups in the jth dimension, F_j indicates the position of the food source in the jth dimension, ub_j denotes the upper bound of the salps in the jth dimension, lb_j denotes the lower bound of the salps in the jth dimension, c_1, c_2, c_3 are three random coefficients.

The equation of the follow salps update its position can be expressed as follows:

$$X_{\mathrm{j}}^{i} = \frac{1}{2}(X_j^i + X_j^{i-1}) \tag{2}$$

Where $i \geq \frac{N}{2}$ and X_j^i denotes that the follower salp position in jth dimension. The Eq. (1) simulated the move of the salp chains. The steps of salp swarm algorithm (SSA) can be described through the pseudo code illustrated as follows (Algorithm 1):

Algorithm 1. SSA pseudo-code

```
Initialize the salp population $X_i(i = 1,2,....n)$ considering ub and lb
while(end condition is not satisfied)
calculate the fitness of each search agent(salp)
F=the best search agent
Update $C_1$ by Equation(3.2)
  for each salp ($X_i$)
    if $(i < \frac{N}{2})$
      Update the position of the leading salp by equation (3.1)
    else
      Update the position of the follower salp by equation (3.3)
    end if
  end for
 Amend the salps based on the upper and lower bounds of variables
 end while
 Return F
```

3 The Proposed SMSSA Approach

The simplex method-based on salp swarm algorithm (SMSSA) proposed in this paper is designed to improve the population diversity and enhance the speed of the convergence. The simplex method has excellent qualities that make the algorithm to jump out the local optimum and increase the diversity of the population. It is means that this approach can make a balance between exploration and exploitation ability of SSA. So, we update the location of the worst salp by using simplex method after each iterating. The modified Algorithm 2 illustrated as follows:

Algorithm 2. SMSSA pseudo-code

```
Initialize the salp population $X_i(i = 1,2,....n)$ considering ub and lb
while(end condition is not satisfied)
calculate the fitness of each search agent(salp)
F=the best search agent
Update $c_1$ by Equation(3.2)
  for each salp ( $X_i$ )
    if $(i == 1)$
      Update the position of the leading salp by equation (3.1)
    else
      Update the position of the follower salp by equation (3.3)
    end if
  end for
  Amend the salps based on the upper and lower bounds of variables
  Update the location of the worst salp by using the simplex method [Eqs. (4)-(8)]
end while
Return F
```

4 Simulation Experiments

4.1 Simulation Platform

The experimental settings for these algorithms are tested in MATLAB R2016 (a) on a windows 10 computer with an Intel Core (TM) i5-4590 Processor, 3.30 GHz, 4 GB RAM.

4.2 Benchmark Functions

Benchmark functions are widely used in this field to benchmark the performance of the algorithm by using a set of quintessential math functions to find the globally optimal. Following the same procedure, 4 standard benchmark functions are used as a comparative test bed from the literature [8, 9]. Tables 1, 2 illustrated the mathematical formulations that employed benchmark functions used respectively. In these three tables, range denotes the search space boundary of the function, and dim means the dimension of the function, and $f_{\min}$ represent the theoretical minimum (optimal value). Heuristic algorithms are stochastic optimization techniques, so they must run dozens of times to produce meaningful statistical results. The result of the last iteration is calculated as the best solution. The same method was chosen to generate and report results for over 30 independent runs.

To evaluate the performance of the proposed SMSSA algorithm, we have chosen some new and well-known algorithm for comparison: CS [6], MFO [4], PSO [2], BA [3], and SSA [7]. Every algorithm uses 30 population individuals and experiences 1000 iteration.

Table 1. Unimodal benchmark functions

Name	Function	Range	Dim	$f_{\min}$
Sphere	$f_1(x)\sum_{i=1}^{n} x_i^2$	[−100, 100]	10	0
Schewfel's 2.22	$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} x_i$	[−10, 10]	10	0
Schwefel's 1.2	$f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	[−100,100]	10	0

Table 2. Multimodal benchmark functions

Name	Function	Range	Dim	$f_{\min}$
Ackey	$f_4(x) = -20\,exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} - exp(\frac{1}{n}\sum^{n} cos2\pi x_i)\right) + 20 + e$	[−32, 32]	10	0

In this work, the best, the average, the worst, and the standard represent the best fitness value, the worst fitness value, and the standard deviation, respectively. The experimental results are shown in Tables 3, 4. The best results are shown in bold type. In addition, for the randomness of the algorithm, statistical tests should be conducted to confirm the significance of the results [10]. To determine whether the SMSSA results differed statistically from the best results for CS, MFO, PSO, BA, and SSA, a non-parametric test called the Wilcoxon rank-sum test [11] is performed at 5% significance level. Tables 6, 7 illustrated the pairwise comparisons of the best values for the six groups generated by the Wilcoxon test. Such groups are formed by CS versus SMSSA, MFO versus SMSSA, PSO versus SMSSA, BA versus SMSSA, SSA versus SMSSA. Generally, p values < 0.05 can harbor the idea that it is strong evidence against the null

Table 3. The results of unimodal benchmark functions

Benchmark functions	Result	Algorithm						Rank
		MFO	PSO	CS	BA	SSA	SMSSA	
f1 (D = 10)	Best	1.33E−32	2.51E−11	0.001754	7.85E−05	9.71E−11	**0**	1
	Worst	5.68E−28	1.88E−08	0.007574	977.8178	4.56E−10	0	
	Mean	2.23E−29	2.27E−09	0.004869	142.7748	2.78E−10	0	
	Std	1.03E−28	4.17E−09	0.001515	233.6054	1.13E−10	0	
f2 (D = 10)	Best	7.82E−21	1.82E−06	0.086866	0.020074	2.46E−06	**1.06E−63**	1
	Worst	10	0.000193	0.379926	46.0267	6.35E−06	4.97E−18	
	Mean	1	2.41E−05	0.181721	18.95383	3.89E−06	1.67E−19	
	Std	3.051286	3.46E−05	0.046013	18.41805	8.49E−07	9.07E−19	
f3 (D = 10)	Best	1.59E−10	0.00029	0.002953	0.000241	9.47E−11	**5.93E−95**	1
	Worst	6666.667	0.019537	0.025104	3570.337	4.75E−10	3.72E−57	
	Mean	888.8889	0.002678	0.011328	1100.737	2.65E−10	1.24E−58	
	Std	2040.459	0.003616	0.005053	901.8556	1.09E−10	6.79E−58	

Table 4. The results of multimodal benchmark functions.

Benchmark functions	Result	Algorithm						Rank
		MFO	PSO	CS	BA	SSA	SMSSA	
f4(D = 10)	Best	4.44E−15	1.2E−05	1.176417	17.25099	3.38E−06	**8.88E−16**	1
	Worst	1.155149	0.000157	3.590626	19.94035	8.69E−06	2.316849	
	Mean	0.038505	5.23E−05	2.64839	18.62497	6.09E−06	0.485117	
	Std	0.2109	3.9E−05	0.53206	0.606456	1.35E−06	0.73783	

hypothesis. Through the statistical test, we can confirm that the results are not produced by chance.

In addition, through nonparametric Wilcoxon statistical tests and calculate the *p* values are reported as the standards of significance as well. Tables 5, 6 show the experimental results of the rank-sum test.

4.3 Unimodal Benchmark Functions

The unimodal benchmark functions have only one global optimum and have no local optimum. So, this type of functions is very suitable for benchmarking the convergence of the algorithm. The results shown in Table 4 show that SMSSA algorithm is more competitive in researching the global optimum. According to the Table 4, the results of SMSSA are superior to some of other algorithms in $f_1 \sim f_3$. Therefore, the SMSSA has higher performance in finding the global minimum of unimodal benchmark functions. As shown in Table 5, the *p* values of $f_1 \sim f_3$ illustrated that SMSSA achieves paramount improvement in some unimodal benchmark functions against other algorithm. Therefore, it is proved by unimodal benchmark functions that SMSSA has better performance in searching global optimal value. Figures 1, 2, 3 shows that the average convergence curve for all algorithms tested with the unimodal benchmark functions are obtained from 30 times independent run.

Table 5. *p*-values rank-sum test on unimodal benchmark functions

Functions	CS vs SMSSA	MFO vs SMSSA	PSO vs SMSSA	BAvs SMSSA	SSA vs SMSSA
f_1	3.16E−12	3.16E−12	3.16E−12	3.16E−12	3.16E−12
f_2	3.02E−11	5.06E−10	3.02E−11	3.02E−11	3.02E−11
f_3	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11

4.4 Multimodal Benchmark Functions

Compared with the unimodal benchmark functions, the multimodal benchmark functions have many local optimal solutions (minima) which increases exponentially with the dimension. This feature makes them good at benchmarking the exploration ability of the algorithm. The result obtained from the benchmark function test reflects the ability of an algorithm to avoid the local minimum and finally can reach the global

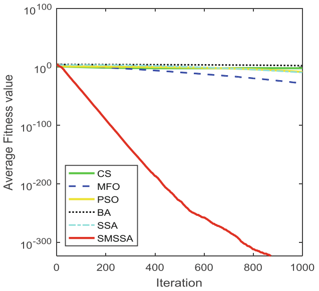

Fig. 1. The convergence curves for f_1

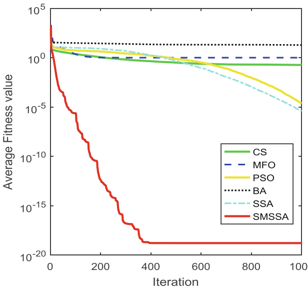

Fig. 2. The convergence curves for f_2

minimum. Table 4 illustrated the results of the algorithm on multimodal benchmark functions. All the results of the Best, Worst, Mean, and Std values illustrated in the table, SMSSA can provide more competitive results on the multimodal benchmark functions. All the results in the table show that the SMSSA has advantage in exploration. As the p values of f_4 shown in the Table 6 are less than 0.05 mostly, which demonstrated it is not the null hypothesis. Therefore, these evidence shows that the results of SMSSA not occurring by accident in the statistic's sense.

As the Table 4 and Fig. 5 shown, the speed of convergence of SMSSA on the multimodal benchmark functions is faster than other algorithms In summary, these evidence shows that this algorithm is more stable and robust than other algorithms (Figs. 4 and 6).

Table 6. p-values rank-sum test on multimodal benchmark functions

Functions	CS vs SMSSA	MFO vs SMSSA	PSO vs SMSSA	BAvs SMSSA	SSA vs SMSSA
f_4	5.32E-11	0.292688	0.026289	2.38E-11	0.026289

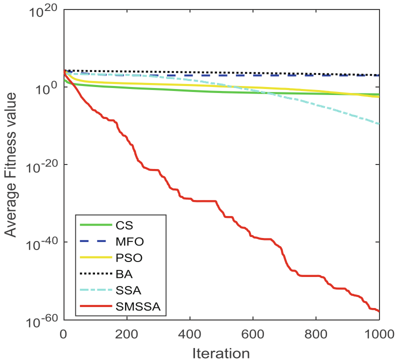

Fig. 3. The convergence curves for f_3

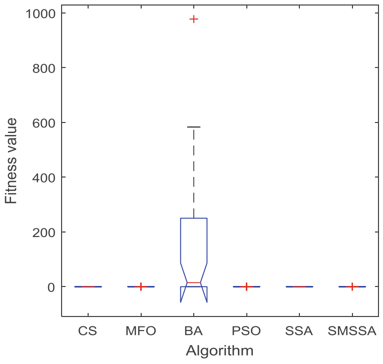

Fig. 4. The standard deviation for f_1

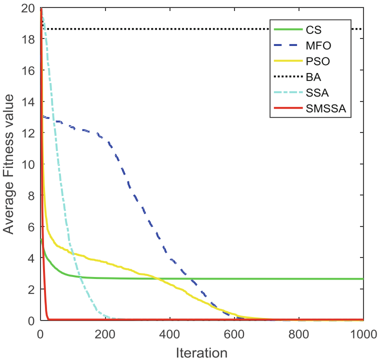

Fig. 5. The convergence curves for f_4

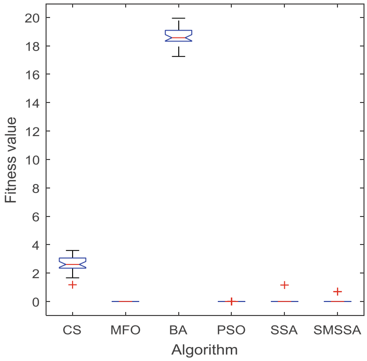

Fig. 6. The standard deviation for f_4

5 SMSSA for Engineering Optimization Problems

In this part, solve an engineering problem (spring design problem) by applying SMSSA algorithm to prove the good performance of SMSSA. Assuming the use of the SMSSA algorithm, some inequality constraints of real problems will be solved. Some methods have been employed to deal with constraints in the paper: special operators, penalty function, repaired algorithms, and hybrid methods [12]. In this work, penalty method is applied to solve the constraints of spring design problem. The spring design problem, [13] is a classic engineering design problem. The main purpose of this problem is to minimize the weight of the spring illustrated in Fig. 7. The model of this problem described as follows:

$$\begin{array}{ll}
\text{Consider} & \vec{x} = [x_1\, x_2\, x_3] = [RDN], \\
\text{Minimize} & f(\vec{x}) = (x_3 + 2)x_2x_1^2, \\
\text{Subject to} & g_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0,\ g_2(\vec{x}) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} \leq 0, \\
 & g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2 x_3} \leq 0,\ g_4(\vec{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0, \\
\text{Variable range} & 0.05 \leq x_1 \leq 2.00,\ 0.25 \leq x_2 \leq 1.30,\ 2.00 \leq x_3 \leq 15.0
\end{array}$$

As shown in Table 7, the spring design problem has been solved by some different approaches. Some meta-heuristic algorithms such as GSA [14], PSO [15] Evolutionary

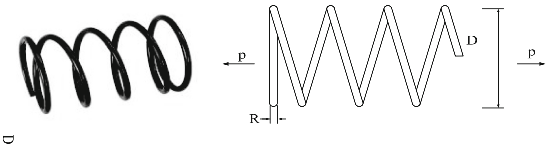

Fig. 7. The spring design problem.

Table 7. Comparison results for spring design problem

Algorithms	Optimal values for variables			Optimal cost
	R	*D*	*N*	
GSA [14]	0.050276	0.323680	13.525410	0.0127022
PSO (Ha and Wang) [15]	0.051728	0.357644	11.244543	0.0126747
GA (Coello) [16]	0.051480	0.351661	11.632201	0.0127048
SSA	0.051207	0.345215	12.004032	0.0126763
SMSSA	0.051783	0.358988	11.15708	0.0126757

genetic algorithms GA [16] has been employed to solve this problem. The statistics lead us to the conclusion that the SMSSA is better than other algorithms.

6 Conclusion

This work proposed an improved algorithm named SMSSA based on simplex method aims at increases the performance of the original SSA algorithm. It can be seen from the Sect. 4, experimental results show that SMSSA achieves not only faster convergence speed and better solutions compared with other algorithms. The conclusion is derived from the comparison between SMSSA and other algorithms. This proposed algorithm demonstrated its outstanding performance by 4 benchmark functions. In addition, this algorithm (SMSSA) also applied to solve engineering problems. The results in the Sect. 5 illustrated that SMSSA algorithm has good performance in solving engineering constraint problems. By combining the advantages of both techniques, SMSSA can get a balance between exploitation and exploration to deal with classical engineering problems.

Acknowledgment. This work is supported by National Science Foundation of China under Grants No. 61463007; 61563008. Project of Guangxi University for Nationalities Science Foundation under Grant No. 2016GXNSFAA380264.

References

1. Storn, R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. **11**(4), 341–359 (1997)
2. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: International Symposium on MICRO Machine and Human Science, pp. 39–43. IEEE (2002)
3. Yang, X.S.: A new metaheuristic bat-inspired algorithm. Comput. Knowl. Technol. **284**, 65–74 (2010)
4. Mirjalili, S.: Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl. Syst. **89**, 228–249 (2015)
5. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**(3), 46–61 (2014)
6. Yang, X.S., Deb, S.: Cuckoo search via levy flights. Mathematics, pp. 210–214 (2010)

7. Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, M.S.: Salp swarm algorithm: a bio-inspired optimizer for engineering design problems. Adv. Eng. Softw. **114**(11), 163–191 (2017)
8. Yang, X.S.: Appendix A: Test problems in optimization. In: Yang, X.-S. (ed.) Engineering Optimization, pp. 261–266. Wiley, Hoboken (2010)
9. Tang, K., Yao, X., Suganthan, P.N., et al.: Benchmark functions for the CEC’ 2008 special session and competition on large scale global optimization. University of Science and Technology of China, Hefei, China (2007)
10. Derrac, J., García, S., Molina, D., et al.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol. Comput. **1**(1), 3–18 (2011)
11. Wolfe, D.A., Hollander, M.: Nonparametric Statistical Methods Robust Nonparametric Statistical Methods. Arnold, London (1998)
12. Coello, C.A.C.: theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput. Methods Appl. Mech. Eng. **191**(11), 1245–1287 (2002)
13. Belegundu, A.D., Arora, J.S.: A study of mathematical programming methods for structural optimization. Int. J. Numer. Methods Eng. **21**(9), 1601–1623 (1985)
14. Rashedi, Esmat, Nezamabadi-pour, Hossein, Saryazdi, Saeid: GSA: a gravitational search algorithm. Intell. Inf. Manag. **4**(6), 390–395 (2012)
15. He, Q., Wang, L.: An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng. Appl. Artif. Intell. **20**(1), 89–99 (2007)
16. Coello, C.A.C.: Use of a self-adaptive penalty approach for engineering optimization problems. Comput. Indus. **41**, pp. 113–127 (2000)