# An Enhanced Whale Optimization Algorithm with Simplex Method

Yanbiao Niu[1,2], Zhonghua Tang[1,2], Yongquan Zhou[1,2(✉)], and Zhongmin Wang[1,2]

[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.** This paper proposes an enhanced whale optimization algorithm with the simplex method named SMWOA algorithm. SMWOA make WOA faster, more robust, and avoid premature convergence. The simplex method (SM) iteratively optimizes the current worst step size, avoids the population search at the edge, and improves the convergence accuracy and speed of the algorithm. The SMWOA algorithm is compared with other well-known meta-heuristic algorithms on 5 benchmarks and 1 classical engineering design problem. The experimental results show that the SMWOA algorithm has better performance than other meta-heuristic optimization algorithms in low and high dimensions.

**Keywords:** An enhanced whale optimization algorithm · Simplex method · Benchmark function · Meta-heuristic optimization algorithms

## 1 Introduction

In recent years, a great many nature-inspired meta-heuristic algorithms have been developed and applied to many optimization problems [1]. Swarm-based techniques have become popular, with the main inspiration coming from a collection of decentralized, self-organizing, collaborative search agent systems [2]. Compared with the traditional precise algorithm, the stochastic optimization algorithm can not only deal with a large number of decision variables but also has strong competitiveness and close to the optimal results. Among many kinds of research, some famous swarm-based optimizers are genetic algorithm (GA) [3], differential evolution (DE) [4] and particle swarm optimization (PSO) [5]. The success of these algorithms in addressing problems of large scale and complexity has encouraged other researchers to develop many new swarm algorithms such as salp swarm algorithm (SSA) [6], grasshopper optimization algorithm (GOA) [7] and grey wolf optimizer (GWO) [8].

The whale optimization algorithm is proposed by Mirjalili et al. in 2016 [9]. It is a new efficient meta-heuristic algorithm inspired by the search and hunting activities of humpback whales in the ocean. However, similar to other swarm-based meta-heuristics, this algorithm also has random operators to explore and exploit the search space. Therefore, when solving the optimization problem, it may rapidly converge to

D.-S. Huang et al. (Eds.): ICIC 2019, LNCS 11643, pp. 729–738, 2019.
https://doi.org/10.1007/978-3-030-26763-6_70

the local optimal solution rather than the global optimal solution. In order to solve the above problems, this paper proposes an enhanced WOA algorithm with simplex method named SMWOA which aimed at enhance the precision of the convergence of basic WOA algorithm. The simplex method [10] has strong ability to avoid local optimization and improve the ability of global optimal search. Experimental results show that the SMWOA algorithm is feasible and effective, and emphasize its superior approximation ability in high dimensional space.

## 2 Whale Optimization Algorithm (WOA) and Simplex Method

### 2.1 Whale Optimization Algorithm

WOA is a new meta-heuristic swarm intelligence optimization algorithm to solve optimization problems. It is derived from the feeding behavior of humpback whales in the bubble net. The hunt for humpback whales can be modeled mathematically in two stages [9].

The humpback whales searching prey is represented by the following Eqs. (1) and (2)

$$\vec{D} = \left| \vec{C} \cdot \overrightarrow{X_{rand}} - \vec{X} \right| \tag{1}$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A} \cdot \vec{D} \tag{2}$$

Where $\overrightarrow{X_{rand}}$ is the random vector (a random whale) selected from the current population. $\vec{A}$ and $\vec{C}$ are coefficient vectors, and the calculation formula is as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a} \tag{3}$$

$$\vec{C} = 2 \cdot \vec{r} \tag{4}$$

Where vector $\vec{a}$ is decreasing linearly from 2 to 0 during the iteration, and $\vec{r}$ is a random vector, within the range of 0 and 1.

$$\vec{D} = \left| \vec{C} \cdot \overrightarrow{X^*}(t) - \vec{X}(t) \right| \tag{5}$$

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A} \cdot \vec{D} \tag{6}$$

Where $t$ is the current iteration, $X^*$ is the position vector of the current optimal solution, and $\vec{X}$ is the position vector. Here, if $|A| > 1$, Eqs. (1) and (2) express the searching prey, otherwise Eqs. (5) and (6) express the encircling prey, which shows the shrinking mechanism.

Humpback whales prey on fish groups with a logarithmic conical spiral motion. The mathematical model is as follows:

$$\overrightarrow{X}^{'}(t+1) = \overrightarrow{D}^{'} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^{*}}(t) \tag{7}$$

Where $\overrightarrow{D}^{'} = \left|\overrightarrow{X^{*}}(t) - \overrightarrow{X}(t)\right|$ represents the distance from the ith whale to its prey (the best solution so far), $l$ is a random number in $[-1,\ 1]$, $b$ is the constant that defines the shape of the logarithmic spiral.

In the process of attacking, whales show both attack modes. Therefore, assuming that they have a 50% probability of contracting and enveloping mechanism, and their positions are updated by a spiral model with the same probability, the model can be modeled as:

$$\overrightarrow{X}(t+1) = \begin{cases} \overrightarrow{X^{*}}(t) - \overrightarrow{A} \cdot \overrightarrow{D} & \textit{if } p < 0.5 \\ \overrightarrow{D^{'}} \cdot e^{bl} \cdot \cos(2\pi l) + \overrightarrow{X^{*}}(t) & \textit{if } p > 0.5 \end{cases} \tag{8}$$

Where $p$ is a random number in [0, 1].

### 2.2 Simplex Method

The simplex method has unique advantages in local search and often gets higher search precision through optimization. The method, proposed by Spendley et al., is a simple linear search method to find local minima of functions. It is based on the idea of comparing the values of the objective function at the N + 1 vertices of a polytope (simplex) in N-dimensional space and moving the polyhedron towards the minimum point as the optimization progress. Subsequently, simplex method have been used for optimization and optimal search, and the preliminary results show it's promising capability [10].

## 3 An Enhanced Whale Optimization Algorithm with Simplex Method

In order to improve the accuracy and convergence speed of the algorithm, the SMWOA algorithm is proposed. The simplex method has good performance, makes the algorithm jump out of the local optimum, and avoids the population searching at the edge, leading it to the global optimization. This means that the method helps achieve a better balance between exploration and exploitation of WOA. Therefore, after each iteration, we use the simplex method to update the position of the worst search agent. This makes the algorithm closer to the optimal solution, improves exploit ability of the algorithm, and speed it up to find the optimal solution. The modified Algorithm 1 showed as follows:

**Algorithm 1.** **SM**WOA pseudo-code

```
1.  Initialize the whales population $X_i (i = 1,2,\ldots n)$
2.  Initialize $a$ , $A$ ,and $C$ $l$ , and $p$
3.  Calculate the fitness of each search agent
4.  $X^*$ =the best search agent
5.  while $(t < \max\_iteration)$
6.      for each search agent
7.          Update $a$ , $A$ ,and $C$ $l$ , and $p$
8.          if1 $(p < 0.5)$
9.              if2 $(|A| < 1)$
10.                 Update the position of the current search agent by Eq. (6)
11.             else if2 $(|A| \geq 1)$
12.                 Select a random search agent $(X_{rand})$
13.                 Update the position of the current search agent by Eq. (2)
14.             end if2
15.         else if1 $(p \geq 0.5)$
16.             Update the position of the current search by Eq. (7)
17.         end if1
18.     end for
19.     Update the position of the worst search agent using the simplex method [Eqs. (9)-(13)]
20.     Check if any search agent goes beyond the search space and amend it
21.     Calculate the fitness of each search agent
22.     Update $X^*$ if there is a better solution
23.     $t = t + 1$
24. end while
25. Return X*
```

## 4 Simulation Experiments

In this section, we selected 5 benchmark functions with different characteristics to verify the performance of SMWOA algorithm from different perspectives. In general, benchmark functions can be segmented into three categories: high-dimension unimodal test functions ($f_1 \sim f_2$), high-dimension multi-modal test functions ($f_3 \sim f_4$) and fixed-dimensional multi-modal test functions ($f_5$), as shown in Table 1.

**Table 1.** Benchmark functions.

| Benchmark function | Dim | Range | $F_{min}$ |
|---|---|---|---|
| $f_1(x) = \sum_{i=1}^{n} x_i^2$ | 50 | $x_i \in [-100, 100]$ | 0 |
| $f_2(x) = \sum_{i=1}^{n} \|x_i\| + \prod_{i=1}^{n} \|x_i\|$ | 50 | $x_i \in [-10, 10]$ | 0 |
| $f_3(x) = \sum_{i=1}^{n} x_i \sin(x_i) + 0.1x_i$ | 50 | $x_i \in [-10, 10]$ | 0 |
| $f_4(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$ | 50 | $x_i \in [-5.12, 5.12]$ | 0 |
| $f_5(x) = -\frac{1+\cos(12\sqrt{x_1^2+x_2^2})}{0.5(x_1^2+x_2^2)+2}$ | 2 | $x_i \in [-5.12, 5.12]$ | −1 |

## 4.1 Results of the Algorithms

To better validate the performance of SMWOA, seven of the most advanced meta-heuristic algorithms are adopted: MFO [11], PSOGSA [12], BA [13], ABC [14], SBO [18], SCA [19], and WOA [9]. To make a fair comparison, each algorithm runs 30 times on the benchmark function, plus the standard deviation of the last generation of the best approximate solution. This feature presents us which algorithm behaves much more stable. The main parameter settings for all algorithms are shown in Table 2. The statistical results include Best, Worst, Mean, and Std values of the target function values. All algorithms are ranked according to the value of std.

**Table 2.** The parameters setting for six algorithms.

| Algorithm | Parameter values |
|---|---|
| MFO [11] | The population size is 30, the maximum iteration number is 1000 |
| PSOGSA [12] | The population size is 30, the maximum iteration number is 1000 |
| BA [13] | $A = 0.9,\ r = 0.5,\ f \in [-1, 1]$ the population size is 30, the maximum iteration number is 1000 |
| ABC [14] | The population size is 30, the maximum iteration number is 1000 |
| SBO [18] | Parameter values of $\alpha$ and Z are considered, 0.94 and 0.02, respectively. The mutation probability is 0.05. The population size is 30, the maximum iteration number is 1000 |
| SCA [19] | $\alpha$ Linearly decreased from 2 to 0 in [19], the population size is 30. The maximum iteration number is 1000 |
| WOA [9] | $\vec{\alpha}$ Linearly decreased from 2 to 0 in [9], the population size is 30. The maximum iteration number is 1000 |
| SMWOA | $\vec{\alpha}$ Linearly decreased from 2 to 0 in [9], the population size is 30. The maximum iteration number is 1000 |

**Table 3.** Results of benchmark functions.

| Functions result | | MFO | PSO-GSA | BA | ABC | SBO | SCA | WOA | SMW-OA |
|---|---|---|---|---|---|---|---|---|---|
| *f1* | Best | 5.457 | 0.213 | $4.1e^{+4}$ | $5.6e^{-3}$ | 0.153 | 0.845 | $1.0e^{-165}$ | 0 |
| | Worst | $3.0e^{+4}$ | $2.0e^{+4}$ | $6.5e^{+4}$ | 0.45 | 0.408 | $9.7e^{+2}$ | $7.6e^{-145}$ | $5.9e^{-323}$ |
| | Mean | $6.6e^{+3}$ | $9.3e^{+3}$ | $5.4e^{+4}$ | $8.8e^{-2}$ | 0.282 | $1.3e^{+2}$ | $2.5e^{-146}$ | $9.8e^{-323}$ |
| | Std | $8.0e^{+3}$ | $8.2e^{+3}$ | $6.1e^{+3}$ | 0.106 | 0.058 | $2.5e^{+2}$ | $1.3e^{-145}$ | 0 |
| | Rank | 7 | 8 | 6 | 4 | 3 | 5 | 2 | 1 |
| *f2* | Best | 20.21 | $7.6e^{-6}$ | $2.3e^{+11}$ | $2.9e^{-2}$ | 0.166 | $2.6e^{-5}$ | $6.3e^{-122}$ | $1.9e^{-260}$ |
| | Worst | $1.1e^{+2}$ | $1.8e^{+2}$ | $3.7e^{+20}$ | 0.136 | 0.290 | 6.9e-2 | $7.6e^{-100}$ | $8.1e^{-244}$ |
| | Mean | 70.87 | 47.01 | $1.7e^{+19}$ | $7.5e^{-2}$ | 0.229 | 9.9e-3 | $2.5e^{-101}$ | $2.9e^{-245}$ |
| | Std | 29.59 | 63.16 | $7.6e^{+29}$ | $2.5e^{-2}$ | 0.028 | 1.4e-2 | $1.3e^{-100}$ | 0 |
| | Rank | 6 | 7 | 8 | 4 | 5 | 3 | 2 | 1 |
| *f3* | Best | $7.2e^{-2}$ | 2.24 | 7.94 | $4.9e^{-2}$ | 0.101 | $1.4e^{-4}$ | $1.1e^{-113}$ | $4.6e^{-261}$ |
| | Worst | 32.65 | 14.09 | 39.6 | 0.70 | 0.341 | 16.21 | $2.1e^{-103}$ | $1.9e^{-238}$ |
| | Mean | 11.32 | 6.96 | 17.1 | 0.27 | 0.206 | 2.34 | $9.4e^{-105}$ | $8.1e^{-240}$ |
| | Std | 8.99 | 3.28 | 7.85 | 0.153 | 0.058 | 3.46 | $3.9e^{-104}$ | 0 |
| | Rank | 8 | 5 | 7 | 4 | 3 | 6 | 2 | 1 |
| *f4* | Best | $1.7e^{+2}$ | $1.4e^{+2}$ | $2.9e^{+2}$ | 11.39 | 76.74 | 0.41 | 0 | 0 |
| | Worst | $3.9e^{+2}$ | $3.8e^{+2}$ | $4.6e^{+2}$ | 24.10 | $1.3e^{+2}$ | $2.4e^{+2}$ | 0 | 0 |
| | Mean | $3.1e^{+2}$ | $2.4e^{+2}$ | $3.6e^{+2}$ | 16.8 | $1.0e^{+2}$ | 73.31 | 0 | 0 |
| | Std | 50.40 | 48.82 | 38.27 | 2.92 | 14.57 | 58.20 | 0 | 0 |
| | Rank | 6 | 5 | 4 | 2 | 3 | 7 | 1 | 1 |
| *f5* | Best | −1 | −1 | −0.936 | −0.99 | −1 | −1 | −1 | −1 |
| | Worst | −0.93 | −0.93 | −0.229 | −0.97 | −0.93 | −1 | −0.936 | −0.936 |
| | Mean | −0.96 | −0.99 | −0.657 | −0.99 | −0.97 | −1 | −0.976 | −0.997 |
| | Std | $3.2e^{-2}$ | 0.019 | 0.225 | 5.8e-3 | $3.2e^{-2}$ | 0 | 0.0297 | 0.0116 |
| | Rank | 7 | 4 | 8 | 2 | 6 | 1 | 5 | 3 |

The results of $f_1$–$f_2$ in Table 3 shows that the SMWOA algorithm can provide very competitive results on unimodal functions. From Figs. 1 and 2, we can clearly see that SMWOA has the fastest convergence rate when searching the global optimum. As can be seen from Figs. 6 and 7, the standard deviation of SMWOA is significantly less than that of the other seven algorithms.

The results of $f_3$–$f_4$ in Table 3 shows that the SMWOA outperforms all the other algorithms. Moreover, for $f_4$, WOA and SMWOA can achieve the theoretical global optimum. As can be seen from the curves in the figures, SMWOA has the fastest convergence speed and higher convergence accuracy. Figures 8 and 9, the standard deviations of the SMWOA are smaller than those of the other algorithms. It can be concluded that SMWOA can avoid local minimum and has a good convergence rate for high-dimensional multimodal test functions.

The SMWOA failed to show the best standard deviation on $f_5$, while at the same time, the SMWOA can find the best fitness value. Figure 5 show the convergence curves of all the algorithms on fixed-dimension multimodal benchmark functions. In

Fig. 5, SMWOA is the third best and obtains the same final optimum. SMWOA is the third best standard deviations in all algorithms.

In summary, the experimental results show that SMWOA algorithm is highly competitive and superior to other well-known algorithms (Figs. 3, 4 and 10).

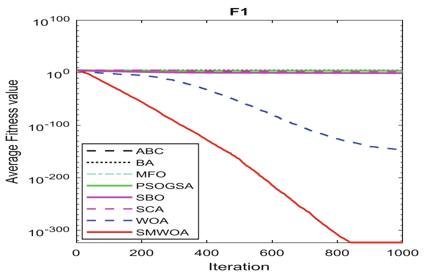

**Fig. 1.** Evolution curves of fitness value for *$f_{01}$*

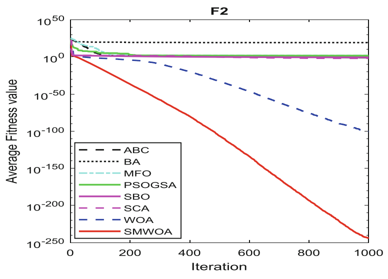

**Fig. 2.** Evolution curves of fitness value for *$f_{02}$*

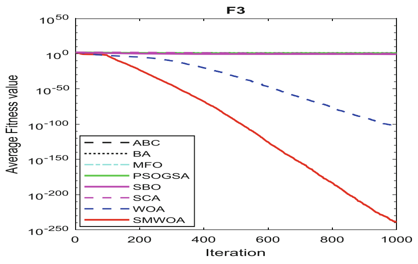

**Fig. 3.** Evolution curves of fitness value for *$f_{03}$*

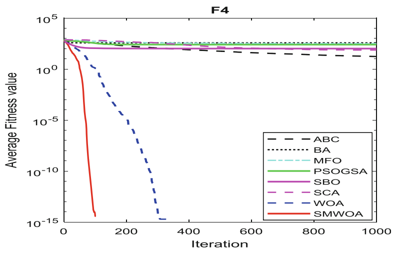

**Fig. 4.** Evolution curves of fitness value for *$f_{04}$*

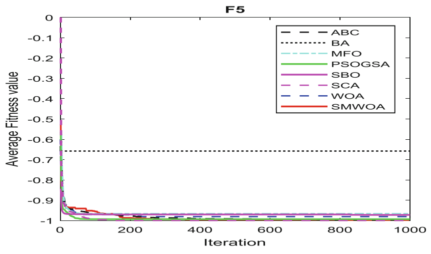

**Fig. 5.** Evolution curves of fitness value for *$f_{05}$*

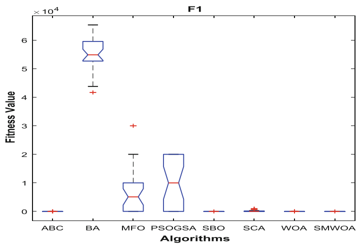

**Fig. 6.** Anova test of global minimum for *$f_{01}$*

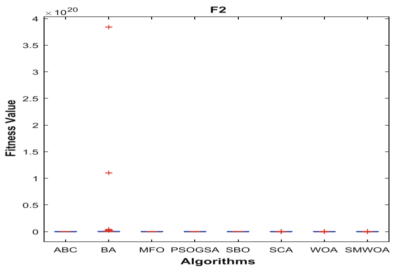

**Fig. 7.** Anova test of global minimum for $f_{02}$

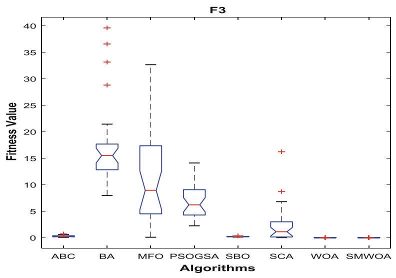

**Fig. 8.** Anova test of global minimum for $f_{03}$

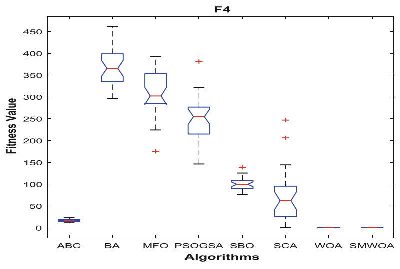

**Fig. 9.** Anova test of global minimum for $f_{04}$

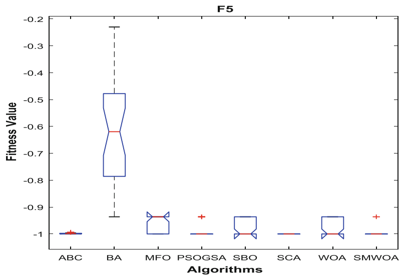

**Fig. 10.** Anova test of global minimum for $f_{05}$

## 5 Constrained Optimization Using the SMWOA Algorithm

Reducer design problem is one of the basic structural engineering problems. The objective function of the problem is to minimize the total weight of the reducer. The problem involves 7 variables, including width $b(x_1)$, number of teeth $m(x_2)$, number of pinion teeth $z(x_3)$, length of the first shaft between bearings $l_1(x_4)$, length of the second shaft between bearings $l_2(x_5)$, a diameter of the first shaft $d_1(x_6)$ and diameter of the second shaft $d_2(x_7)$.

$$\text{Minimize} f\left(\overrightarrow{x}\right) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) \\ - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right)$$

$$\text{Subject to} \quad g_1(\vec{x}) = -x_1 + 0.0193x_3 \le 0, \quad g_2(\vec{x}) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0,$$

$$g_3(\vec{x}) = \frac{1.93x_4^3}{x_2x_6^4x_3} - 1 \le 0, \quad g_4(\vec{x}) = \frac{1.93x_5^3}{x_2x_7^5x_3} - 1 \le 0,$$

$$g_5(\vec{x}) = \frac{\left[(745x_4/x_2x_3)^2 + 16.9 \times 10^6\right]^{1/2}}{110x_6^3} - 1 \le 0,$$

$$g_6(\vec{x}) = \frac{\left[(745x_5/x_2x_3)^2 + 157.5 \times 10^6\right]^{1/2}}{85x_7^3} - 1 \le 0,$$

$$g_7(\vec{x}) = \frac{x_2x_3}{40} - 1 \le 0, g_8(\vec{x}) = \frac{5x_2}{x_1} - 1 \le 0, g_9(\vec{x}) = \frac{x_1}{12x_2} - 1 \le 0,$$

$$g_{10}(\vec{x}) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0, g_{11}(\vec{x}) = \frac{1.1x_6 + 1.7}{x_5} - 1 \le 0,$$

$$\text{Where} \quad 2.6 \le x_1 \le 3.6, 0.7 \le x_2 \le 0.8, 17 \le x_3 \le 28, 7.3 \le x_4 \le 8.3,$$
$$7.3 \le x_5 \le 8.3, 2.9 \le x_6 \le 3.9, 5.0 \le x_7 \le 5.5.$$

**Table 4.** Comparison results of the speed reducer design problem.

| Algorithms | Optimal values for variables | | | | | | | Optimal cost |
|---|---|---|---|---|---|---|---|---|
| | *b* | *m* | *z* | $l_1$ | $l_2$ | $d_1$ | $d_2$ | |
| CS [15] | 3.501 | 0.7 | 17 | 7.605 | 7.818 | 3.352 | 5.287 | 3000.98 |
| HCPS [16] | 3.5 | 0.7 | 17 | 7.3 | 7.715 | 3.350 | 5.286 | 2994.47 |
| SCA [17] | 3.5 | 0.7 | 17 | 7.327 | 7.715 | 3.350 | 5.286 | 2994.74 |
| ABC [14] | 3.499 | 0.7 | 17 | 7.3 | 7.8 | 3.350 | 5.287 | 2997.05 |
| SMWOA | 3.5 | 0.7 | 17 | 7.3 | 7.3 | 3.350 | 5.286 | **2985.13** |

Table 4 shows that SMWOA can obtain the optimal cost value and is superior to all algorithms.

## 6 Conclusions

An enhanced WOA algorithm, SMWOA algorithm, is proposed to solve the optimization problem. To evaluate the performance of SMWOA, we used 5 benchmark functions and compared them with the other seven most advanced meta-heuristic algorithms. As shown in Sect. 4, it proves that SMWOA is superior to all algorithms in most test cases and that SMWOA is a feasible and quite effective method in global optimization problems. What's more, We solved one classical engineering problems with our proposed SMWOA method, as shown in Sect. 5. The results show that SMWOA performs well in unknown and challenging search space.

**Acknowledgment.** This work is supported by National Science Foundation of China under Grant No. 61563008. Project of Guangxi University for Nationalities Science Foundation under Grant No. 2018GXNSFAA138146.

## References

1. Abdel-Basset, M., El-Shahat, D., El-Henawy, I., et al.: A modified flower pollination algorithm for the multidimensional knapsack problem: human-centric decision making. Soft. Comput. **22**(13), 4221–4239 (2018)
2. Heidari, A.A., Abbaspour, R.A., Jordehi, A.R.: Gaussian bare-bones water cycle algorithm for optimal reactive power dispatch in electrical power systems. Appl. Soft Comput. **57**, 657–671 (2017)
3. Hossam, F., Al-Zoubi, A.M., Asghar, H.A., et al.: An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks. Inf. Fusion **48**, 67–83 (2018). S1566253518303968
4. Wang, L., Zeng, Y., Chen, T.: Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. Expert Syst. Appl. **42**(2), 855–863 (2015)
5. Jordehi, A.R.: A review on constraint handling strategies in particle swarm optimisation. Neural Comput. Appl. **26**(6), 1265–1275 (2015)
6. Faris, H., Mafarja, M.M., Heidari, A.A., et al.: An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. Knowl.-Based Syst. **154**, 43–67 (2018)
7. Asghar, H.A., Hossam, F., Ibrahim, A., et al.: An efficient hybrid multilayer perceptron neural network with grasshopper optimization. Soft Comput. (2018). https://doi.org/10.1007/s00500-018-3424-2
8. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
9. Mirjalili, S., Lewis, A.: The whale optimization algorithm. Adv. Eng. Softw. **95**, 51–67 (2016)
10. Davoodi, E., Hagh, M.T., Zadeh, S.G.: A hybrid improved quantum-behaved particle swarm optimization-simplex method (IQPSOS) to solve power system load flow problems. Appl. Soft Comput. **21**, 171–179 (2014)
11. Mirjalili, S.: Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl.-Based Syst. **89**, 228–249 (2015)
12. Mirjalili, S., Hashim, S.Z.M.: A new hybrid PSOGSA algorithm for function optimization. In: International Conference on Computer and Information Application, pp. 374–377 (2010)
13. Yang, X.S.: A new metaheuristic bat-inspired algorithm. Comput. Knowl. Technol **284**, 65–74 (2010)
14. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim **39**(3), 459–471 (2007)
15. Gandomi, A.H., Yang, X.-S., Alavi, A.H.: Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng. Comput. **29**, 17–35 (2013)
16. Long, W., et al.: A hybrid cuckoo search algorithm with feasibility-based rule for constrained structural optimization. J. Central South Univ. **21**(8), 3197–3204 (2014)
17. Ray, T., Liew, K.M.: Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans. Evol. Comput. **7**(4), 386–396 (2003)
18. Moosavi, S.H.S., Bardsiri, V.K.V.: Satin bowerbird optimizer: a new optimization algorithm to optimize ANFIS for software development effort estimation. Eng. Appl. Artif. Intell. **60**, 1–15 (2017)
19. Mirjalili, S.: SCA: a sine cosine algorithm for solving optimization problems. Knowl. Based. Syst. **96**, 120–133 (2016)