IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 6, 2019, accepted September 23, 2019, date of current version October 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2945842

QSSA: Quantum Evolutionary Salp Swarm Algorithm for Mechanical Design

RONGZHONG CHEN[1,2], CHEN DONG[1,2,3], (Member, IEEE), YIN YE[1,2], ZHENYI CHEN[4], AND YANHUA LIU[1,3], (Member, IEEE)

[1]College of Mathematics and Computer Science, Key Laboratory of Spatial Data Mining & Information Sharing, Ministry of Education, Fuzhou University, Fuzhou 350116, China
[2]Key Lab of Information Security of Network Systems (Fuzhou University), Fuzhou 350116, China
[3]Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou 350116, China
[4]Department of Electrical Engineering, University of South Florida, Tampa, FL 33620, USA

Corresponding authors: Chen Dong (dongchen@fzu.edu.cn) and Zhenyi Chen (zhenyichen@mail.usf.edu)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672159, Grant 61872091, Grant U18042631, and Grant 61702105, in part by the Science Foundation of the Fujian Province, China, under Grant 2018J01793, and in part by the Foundation of the Education Department of Fujian Province, China, under Grant JAT170099.

ABSTRACT Salp swarm algorithm is a new meta-heuristic algorithm which has excellent advantages for solving the multidimensional optimization problem. In this paper, a hybrid model assisted evolutionary algorithm for solving engineering design optimization problems is proposed and investigated. The purpose of the optimizer is to improve the potential shortcomings of the basic salp swarm optimization, including trapping in local or deceptive optima easily. On the one hand, quantum behavior can increase an individual's searchability, which can promote the overall optimization trend. On the other hand, the elite opposition-based learning strategy is used to enhance the diversity of the population. Besides, mutation mechanism is introduced to prevent the individuals of the population from being in stagnation behavior. The proposed quantum-behaved and wavelet mutation salp swarm algorithm (QSSA) is applied on twenty-three benchmark functions and three basic constrained engineering problems. Experimental results demonstrate that the algorithm has excellent solution quality, and it can overcome the defect of the low convergence rate.

INDEX TERMS Evolutionary algorithm, salp swarm optimization, quantum behavior, wavelet mutation, structure design.

I. INTRODUCTION

In the process of biological evolution, there are various social behaviors. Different nature-inspired algorithms are creatively proposed by researchers, such as artificial bee colony algorithm (ABC) [1], particle swarm optimization (PSO) [2], [3], grasshopper optimization algorithm (GOA) [4], and salp swarm algorithm (SSA) [5]. However, most existing meta-heuristic algorithms are more sensitive to the multidimensionality of the problem, and it is customary to find the optimal solution in a flat search area. The traditional algorithm not only faces the space and time challenges, but also need to ensure the accuracy of the solution obtained under nonlinear constraints. In the practical problems [6], [7], improve efficiency is the emphasis of solution. Due to the limitations of biological algorithms and the deficiency of self-search conditions, most algorithms [8], [9] tend to reach ''premature maturity''. To overcome the algorithm's fall into the optimal local solution [10], [11], the researchers propose different ameliorative algorithms and apply their ideas in different fields.

In recent years, SSA is intensely concerned by many scholars owing to its fewer control parameters, fast convergence speed, and better flexibility, which is adapted to solve complex combinatorial optimization problems. The behavior of the salp chains has been widely used due to its excellent convergence and search accuracy. Parameters estimation of photovoltaic cells is a typical multi-objective optimization problem. Compared with other swarm intelligent algorithms in designing SPVSs, the SSA algorithm shows strong competitiveness, a high value of accuracy and practicability [12]. Tolba *et al.* [13] proposed to apply SSA for solving the problems of RDG and SCB locations and capacities simultaneously on radial distribution networks (RDNs). The article

The associate editor coordinating the review of this manuscript and approving it for publication was Malik Jahan Khan.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

has focused on proposing a type-II fuzzy PID controller for keeping both frequency and tie-line power to their nominal values in the condition of different uncertainties. It is shown that the I-SSO algorithm has greater performance advantages than other meta-heuristic algorithms according to dynamic response results are proposed [14].

According to survey statistics from the latest literature, there are also many improvements based on the basic salp swarm algorithm. Faris *et al.* [15] use the binary salp swarm algorithm to enhance the exploration, and utilize the transfer functions and crossover operator to tackle FS problems. To accomplish microarray classification, a novel hybrid salp swarm algorithm is proposed based on the improved version of the bio-inspired optimization technique. Weighted-chaotic SSA for both optimal gene selection and parameter optimization of the KELM classifier, which has higher accuracy, higher sensitivity, and better specificity [16]. Three chaotic versions of the SSA algorithm are used to automatically find the centroid of the document image dataset by adopting a k-means cluster [17]. On the best of our knowledge, the authors in [18] used enhanced theory to improve the SSA, which is called an enhanced salp swarm algorithm (ESSA). To further enhance SSA, a modified binary Salp Swarm Algorithm (BSSA) with different updating strategies rules is proposed. In the Feature Selection process, the improved algorithm is tested on 20 datasets of the UCI repository. Experiments show that BSSA has good advantages in the exploration and development of feature space [19].

A. ORGANIZATION OF THIS PAPER

The paper is organized as follows. In Section II, a brief overview of the standard SSA and an improved QSSA algorithm is contained. In Section III, on solving 23 benchmark functions, experimental results of the comparison between other methods are analyzed. In Section IV, an improved SSA is elaborated with different operations for tackling the engineering problems. Conclusions are drawn in Section V.

II. THE BASIC SSA MODEL

The salp is a transparent organism similar to jellyfish. Currently, it mainly lives in the sea near Oceania. They feed on phytoplankton. Scientists have discovered that the predation strategy of the sea squirt is a chain-like behavior that relies on the chain mechanism of the group to better forage. The salp chain is illustrated in Fig.1. Seyedali Mirjalili proposes a salp swarm algorithm inspired by the predation mechanism of the salps. SSA is one of the evolutionary computation techniques. The salp optimization algorithm is based on chain behavior to find the optimal solution. In the SSA method, the population is divided into a leader, which is at the front of the chain, and followers; the leader guides the salp chain and the followers follow one by one. In the process of predation of the salp, the leader leads the population to the optimal food, and the follower follows the recent salp and transmittings food signals to maintain the flexibility of the population, and

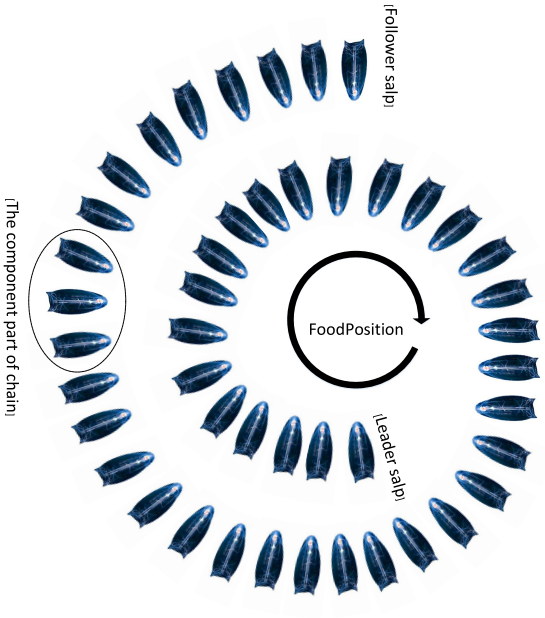


FIGURE 1. The predatory chain of the salp swarm.

avoid population falling into the local optimal solution. Next, the main steps of the salp swarm algorithm will be analyzed.

Let the position vector of each salp be an $N \times D$ dimensional, where N is number of search agents and D represents the dimension of continuous solution space. Hence, the position vector of the population can be represented by a multidimensional matrix, as shown in Eq.(1):

$$X_j^i = \begin{bmatrix} X_1^1 & X_2^1 & \cdots & X_D^1 \\ X_1^2 & X_2^2 & \cdots & X_D^2 \\ \vdots & \vdots & \cdots & \vdots \\ X_1^N & X_2^N & \cdots & X_D^N \end{bmatrix} \tag{1}$$

The following equation suggested to update the position of the leader can be represented as:

$$X_j^1 = \begin{cases} F_j + c_1((ub_j - lb_j)c_2 + lb_j) & c_3 \geq 0.5 \\ F_j - c_1((ub_j - lb_j)c_2 + lb_j) & c_3 < 0.5 \end{cases} \tag{2}$$

where X_j^1 is the position of the first generation leader of the population in the j_{th} dimension, ub_j indicates the upper bound of j_{th} dimension search space, lb_j indicates the lower bound of jth dimension search space, c_2 and c_3 are the random numbers during the interval [0, 1]. The coefficient c_1 is the most important parameter in the iterative process of the algorithm because it balances between exploration and exploitation propensities, and it can be given by:

$$c_1 = 2e^{-(\frac{4l}{l_{max}})^2} \tag{3}$$

where l is the current iteration of the algorithm and l_{max} is the maximum number of iterations. The function curve is shown in Fig.2.

The position of follower salp is updated according to Newton's law of motion, and its formula can be briefly described

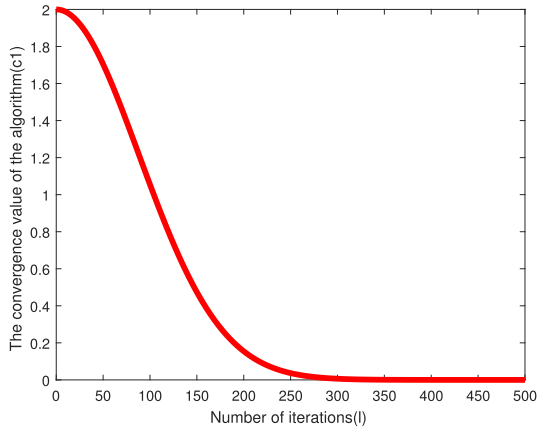


FIGURE 2. The function curve of c_1.

as follows:

$$X_j^i = \frac{1}{2}at^2 + v_0 t \tag{4}$$

where $i \geq 2$, X_j^i represents the position of i_{th} follower salp in j_{th} dimension, t is time, v_0 indicates the initial speed, and $a = \frac{v_1}{v_0}$ where $v = \frac{x - x_0}{t}$.

Combined with the classical Newton, the time interval in the algorithm is iteration, the position update time is equal to 1, and considering $v_0 = 0$, this equation can be expressed as follows:

$$X_j^i = \frac{1}{2}(X_j^i + X_j^{i-1}) \tag{5}$$

where $i \geq 2$, and X_j^i represents the position of i_{th} follower salp in j_{th} dimension.

According to Eqs.(2) and Eqs.(5), the salp chains can be formed. Based on the above explanations, the pseudocode of the SSA algorithm can be described in Algorithm 1.

Algorithm 1 Pseudocode of the SSA Algorithm.

```
Randomly initialize the positions of salps x_i(i = 1, 2, ··· , N) according to ub_j and lb_j
while termination criteria is not satisfied do
    Set F as the best search agent
    Calculate the fitness value of each salp
    Update c_1 by Eq.(3)
    if i < N/2 then
        Update the position of the leading salp by Eq.(2)
    else
        Update the position of the follower salp by Eq.(5)
    end if
    Amend the each salp that go beyond the upper and lower bounds
    Recalculate the fitness of all salps
    Update the best position of the food source(F).
end while
Next generation until stoping criterion
Return F
```

III. IMPROVED SALP SWARM ALGORITHM

In this section, the improved salp swarm algorithm is discussed from three aspects of the quantum mechanics, elite opposition-based, and wavelet variation mechanism, respectively. The relevant content will be elaborated in the following subsections.

A. QUANTUM-BASED SALP SWARM ALGORITHM

When solving the multidimensional modal problem, the salp swarm algorithm has the disadvantages of slow convergence speed and quickly falls into a local optimum. A quantum behavior particle swarm optimization is proposed to improve the overall performance of the algorithm [20]. Inspired by this idea, a quantum salp swarm algorithm is proposed.

In quantum mechanics, the general state of a free particle can be represented by a wave function.

$$\psi = \exp[ik \cdot r] = \exp[ip \cdot \boldsymbol{r}/\hbar] \tag{6}$$

Fourier expansion is made for plane wave functions, and it can be expressed as follows:

$$\varphi(p) = \frac{1}{(2\pi\hbar)^{3/2}} \int \Psi(r) e^{-ip\cdot r/\hbar} d^3 r \tag{7}$$

$$\Psi(r) = \frac{1}{(2\pi\hbar)^{3/2}} \int \varphi(p) e^{ip\cdot r/\hbar} d^3 p \tag{8}$$

The dynamic behavior of particles can be seen as a superposition of planar monochromatic waves, which can be expressed as follows:

$$\psi(\boldsymbol{r}, t) = \frac{1}{(2\pi\hbar)^{3/2}} \int \varphi(\boldsymbol{p}) \mathrm{e}^{i(p\cdot r - Et)\hbar} \mathrm{d}^3 p \tag{9}$$

According to the theory of planar monochromatic waves and De Broglie:

$$-\mathrm{i}\hbar\nabla\psi = p\psi, \quad -\hbar^2\nabla^2\psi = p^2\psi \tag{10}$$

And the relationship between energy and momentum:

$$E = p^2/2m \tag{11}$$

We can get the following simplification formula:

$$\left(\mathrm{i}\hbar\frac{\partial}{\partial t} + \frac{\hbar^2}{2\,m}\nabla^2\right)\psi = \frac{1}{(2\pi\hbar)^{3/2}} * \int \varphi(p)\left(E - \frac{p^2}{2\,m}\right)\mathrm{e}^{\mathrm{i}(pr - \overline{E})/\hbar}\mathrm{d}^3 p = 0 \tag{12}$$

Equation(12) can be written as:

$$\mathrm{i}\hbar\frac{\partial}{\partial t}\psi(\boldsymbol{r}, t) = -\frac{\hbar^2}{2m}\nabla^2\psi(\boldsymbol{r}, t) \tag{13}$$

Therefore, the particle moves in the one-dimensional potential field $V(\boldsymbol{r})$, and the motion equation can be written as

follows:

$$i\hbar\frac{\partial}{\partial t}\psi(\boldsymbol{r},t)=\left[-\frac{\hbar^2}{2m}\nabla^2+V(\boldsymbol{r})\right]\psi(\boldsymbol{r},t) \tag{14}$$

where $\hbar$ is Planck constant, and m is the masses of the particle.

The SchrÃdinger wave equation reveals the fundamental laws of material motion in the microscopic world. It is assumed that the particles have quantum behavior, which expresses different wave functions. In the one-dimensional Delta potential field, the potential energy distribution of the potential well where the particle centered on point X, as follows:

$$V(r)=-\gamma\delta(r) \tag{15}$$

where γ is the potential well depth, we can get the wave function of the particle based on Eq.(14).

$$\Psi\left(x_{j,t+1}^{i}\right)=\frac{1}{\sqrt{L_{j,t}^{i}}}\exp\left(-\frac{\left|x_{j,t+1}^{i}-s_{j,t}^{i}\right|}{L_{j,t}^{i}}\right) \tag{16}$$

$$L=\frac{\hbar^2}{m\gamma} \tag{17}$$

L is the characteristic length of the Delta well. The probability density function of particles in the jth dimension can be expressed as follows:

$$Q\left(x_{i,t+1}^{j}\right)=\frac{1}{L_{i,t}^{j}}\exp\left(-\frac{2\left|x_{i,t+1}^{j}-s_{i,t}^{j}\right|}{L_{i,t}^{j}}\right) \tag{18}$$

In the one-dimensional Delta potential field of the particle, its motion equation satisfies the SchrÃdinger formula, so the position of the particle is uncertain at any point in time. However, in the basic salp swarm algorithm, the individual position of the salp must be determined in the corresponding iteration. To obtain the precise position of individuals, the state of individuals can be collapsed from a quantum state to a classical state by using the Monte Carlo method.

$$|r_{t+1}|=\alpha\cdot|r_t|\cdot\ln(1/u) \tag{19}$$

where $r_{t+1}=x_t-P$, u is a random number.

$$x_{k+1}=P\pm\alpha\cdot|x_k-P|\cdot\ln(1/u) \tag{20}$$

Therefore, the position update strategy for the leading salp is described as follows:

$$x_{j,t+1}^{1}=\begin{cases}s_{j,t}^{1}+c_1\cdot(M-x_{j,t}^{1})\cdot\ln(1/c_2) & c_3\geq 0.5\\ s_{j,t}^{1}-c_1\cdot(M-x_{j,t}^{1})\cdot\ln(1/c_2) & c_3<0.5\end{cases} \tag{21}$$

where $s_{j,t}^{1}$ is the global optimal value, M is the average of the local optimum values and is defined in Eqs.(22).

$$M=\frac{\sum_{i=1}^{N}p_{i,t}^{j}}{N} \tag{22}$$

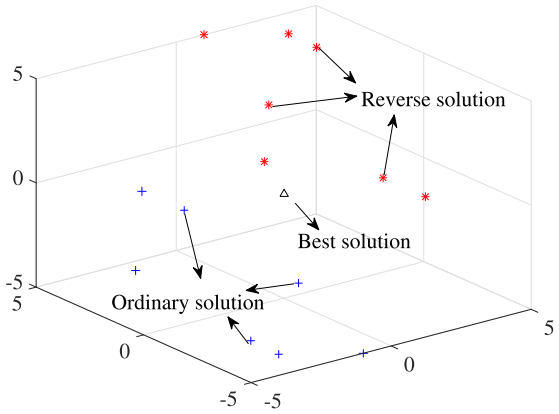


FIGURE 3. Distribution of the particle.

In the basic salp swarm algorithm, the individual of the population lacks communication as leading salps have comprehensive guidance for the subsequent update the position of the follower salps. The introduction of quantum mechanics can effectively enhance the mobility of salps. The results show that quantum mechanics has better convergence speed and robustness, and the solution results are excellent.

B. ELITE OPPOSITION-BASED LEARNING

Reverse learning [21] is a new strategy in the field of intelligent computing, and it has the advantage of low cost and high effect of solving problems by comparing with other algorithms. In multi-dimensional complex issues, the quality of solutions is often affected by many factors. Most group intelligence algorithms use the random optimization method to search space and find globally optimal solutions until some termination criterion is reached. Search strategy based on reverse learning can significantly enhance the searchability of the population.

- **Definition 1. Opposite Point**
  Let $X_i=\left(x_{i,1},x_{i,2},\ldots,x_{i,D}\right)$ be a solution in D-dimensional space. Its reverse solution $Y_i=y_{i,1},y_{i,2},\ldots,y_{i,D}$ is defined by

$$Y_i=ub_j+lb_j-X_i \tag{23}$$

  where $[lb_j,ub_j]$ is the dynamic boundary of the j_{th} search space and its defined as follows:

$$\begin{aligned}lb_j&=\min\left(x_{i,j}\right)\\ ub_j&=\max\left(x_{i,j}\right)\end{aligned} \tag{24}$$

- **Definition 2. Elite Opposite Solution**
  It is observed from Fig.3 that the convergence trend of the particles. For a solution in population, its fitness function is expressed as $f(X)$. For a problem to be minimized, if $f(X)<f(Y)$ then Y can be replaced by X in the algorithm iteration. Therefore, random and reverse population are simultaneously evaluated, and a better solution among them is regarded as the next-generation solution.

In this paper, the position of the follower salp will only be affected by the leader salp. There is no doubt that the chain predation behavior of salps leads to a single solution. Non-dominated solutions are often viewed as elite individuals in multi-objective optimization problems, and these individuals are considered to enhance the population to find the global optimal Pareto frontier. Follower salps in the algorithm are also essential parts of the population, so analyzing the elite inverse solution of the follower salps can enhance the algorithm diversity. Formally, elite reverse solution [22] can be expressed as follows:

$$X_j^i(elite) = k(ub_j + lb_j) - X_j^i(normal) \tag{25}$$

where k is a random number generated by a uniform distribution in [0, 1]. $X_j^i(normal)$ and $X_j^i(elite)$ are the positions of follower salp and their elite reverse solutions, respectively. Calculate the fitness for them separately, and choose the best individual among them as the next generation while preserving reverse elitism property up to some predefined stopping criteria. Therefore, elite reverse strategy is introduced at the position of the follower salps can provide more appropriate and robust candidate solution.

C. WAVELET MUTATION

It is well known that mutational manipulations are often used in mutated individuals to improve convergence. In the salp swarm optimization, we introduce wavelet mutation [23] into the later stage of evolution to maintain population diversity. The specific details of the wavelet variation can be briefly described as follows. Each follower salp of the population has an opportunity for variation, which is controlled by a probability of mutation $p_m \in [0, 1]$. For each individual, a random number between 0 and 1 is generated for assessing the variability condition that if it is less than or equal to $\overline{x}_j^i(t) = \left[x_j^1(t), x_j^2(t), \ldots, x_j^i(t)\right]$, the mutation operation will take place on that individual. Next, we suppose $\overline{X}_j^i(t)$ is the selected individual of the path generation, and its range is inside the boundaries $[lb_j, ub_j]$. $\overline{X}_j^i(t)$ will be mutated according to the following formula.

$$\overline{x}_j^i(t) = \begin{cases} x_j^i(t) + \sigma \times \left(ub_j - x_j^i(t)\right), & \text{if } \sigma > 0 \\ x_j^i(t) + \sigma \times \left(x_j^i(t) - lb_j\right), & \text{if } \sigma \le 0 \end{cases} \tag{26}$$

with

$$\sigma = \frac{1}{\sqrt{a}} e^{\frac{-\left(\frac{x}{a}\right)^2}{2}} \cos\left(5\left(\frac{x}{a}\right)\right) \tag{27}$$

where a is dilation parameter that usually changes with the individual's iteration to achieve the goal of fine-tuning.

Therefore, the curve trend of different dilated Morlet Wavelets can be seen in Fig.4. It can be seen from the figure that the amplitude of $\psi_{a,0}(x)$ will be scaled down as the parameter a increases. This variation character reflects the slight adjustment of the algorithm at a later stage. Since more than 99% of the total energy of the morlet wavelet function is concentrated in the interval $[-2.5, 2.5]$, x can be randomly obtained within the interval $[-2.5 \times a, 2.5 \times a]$. The value of the dilation parameter a increases with the change of l/L, where l represents the current number of iterations and L represents the maximum number of iterations. This linear change can increase the search performance of the algorithm. Therefore, the expression for the dilation parameter can be obtained as follows.

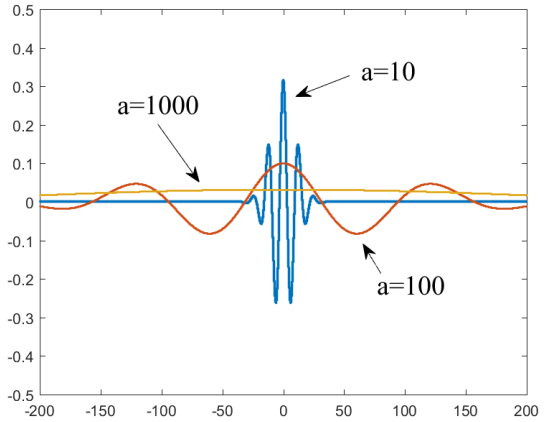


FIGURE 4. Morlet wavelet dilated by different values of parameter *a*.

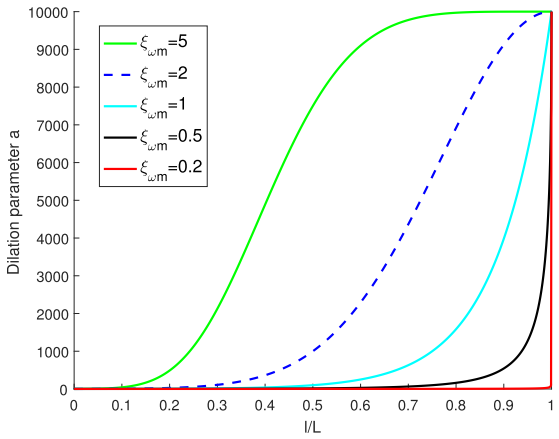


FIGURE 5. Effect of the shape parameter $\xi_{\omega m}$ to a with respect to l/L.

$$a = e^{-\ln(g_1)\times\left(1-\frac{l}{L}\right)^{\xi_{\omega m}}+\ln(g_1)} \tag{28}$$

where $\xi_{\omega m}$ is the shape parameter of the monotonic increasing function, and g_1 is the upper limit of the parameter a. The domain of parameter a varies from 0 to 1, and it follows the increase in the number of iterations l and the tendency to change nonlinearly is shown in Fig.5. It is suggested that the value of a is between 1 and $g_1(= 10000)$ through the scientific analysis of specific experiments. The characteristics of the morlet wavelet function can help the algorithm jump out of the local optimum during the search phase to achieve fine tuning.

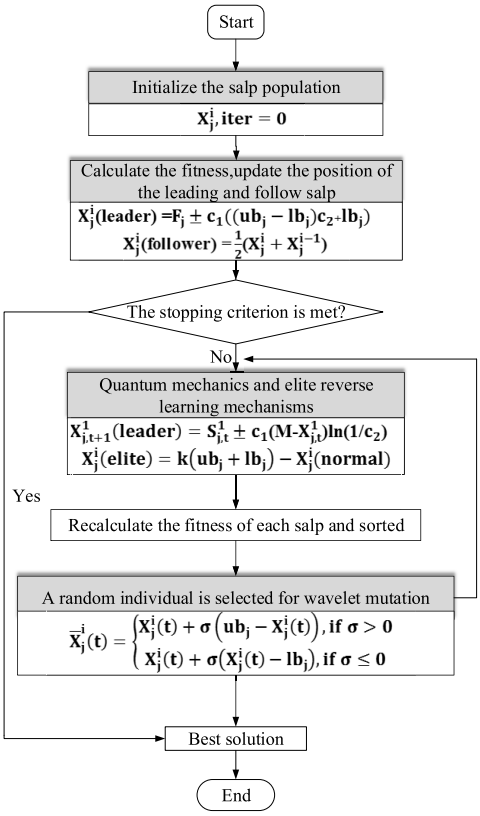


FIGURE 6. Flowchart of the QSSA algorithm.

In this section, a wavelet mutation optimization algorithm is proposed to find the optimal solution in multi-dimensional problems. Morlet wavelet mutation can coordinate performance and expand the contradiction of the search area.

The salp-chain predation behavior is a dynamic structure in which elements are closely linked to maintaining a dominant relationship between individuals. This rigorous structure has a good advantage in biological predation. Multidimensional problems are often limited by the individual's distribution and convergence speed, so it's hard to find the optimal solution. This paper introduces quantum mechanism to improve the position of the followers and looks at the quantum changes of individuals from physics. Next, elite reverse learning is used to allow follower salp changes their current position, and overcome the deficiency which absolute obedience. In the final stage of the iteration, some individuals are randomly selected for wavelet mutations to help the algorithm jump out of the locally optimal solution. The SSA and QSSA are the two-layer loops, so the time complexity is $O(n^2)$. Flowchart of the QSSA algorithm is proposed in this paper, as shown in Fig.6. The QSSA optimizer is proved its superiority with high performance and accuracy for solving many complex problems.

D. BENCHMARKS AND EXPERIMENTAL SETTINGS

The proposed QSSA discussed in the preceding sections will be tested on some uni-modal, multi-modal, and fixed-dimension multi-modal benchmark functions under the same conditions in this section.

TABLE 1. Uni-modal benchmark functions.

Function	Dim	Range	f_{min}
$F_1(x) = \sum_{i=1}^{n} x_i^2$	n	[-100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	n	[-10,10]	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j\right)^2$	n	[-100,100]	0
$F_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	n	[-100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} [100 (x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	n	[-30,30]	0
$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	n	[-100,100]	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0, 1]$	n	[-128,128]	0

As we all know, unimodal functions has only globally optimal but no locally optimal, so they are more capable of detecting the development capabilities of the algorithm. Multimodal functions are often unable to find global optimal solutions because of many disturbing factors. Therefore, multi-modal functions can detect the exploration ability of the algorithm.

To validate the effectiveness of the QSSA proposed in this paper, we firstly conduct experiments on twenty-three well-known benchmark functions in this section to determine the related optimal parameters, and the maximum iteration is set to 500 for test functions. Table 1, Table 2, and Table 3 list the characteristics of twenty-three benchmark functions.

In the experiment, the population size is set to 30. The parameters c_2 and c_3 are the random number of [0, 1], and the value of c_1 decreases from 2 to 0 in a linear way through the iterations. The experiments are executed in computer intel i3-4150 CPU 3.50 GHz and 4G memory with system Windows 7 Ultimate. The programs are written in MATLAB R2016a and perform 30 times for every function. Without loss of generality, the performance is also estimated in terms of the fitness mean (Mean) and standard deviation (Std) of the fitness errors.

E. COMPARISON RESULTS ON THE BENCHMARK PROBLEMS

In this section, tables and figures are shown in a very intuitive way to improve the overall visual effect. It is mainly compared with the recent meta-heuristic algorithm (ESSA [18], SSA, CSSA [30], GWO [25], WOA [35], PSO, GSA [27], FEP [28] and CS [29]). Different key parameters of intelligence algorithms can be seen in table 4. In the uni-modal benchmark function, as shown in table 5, the improved QSSA algorithm achieves the best ranking in most of the test functions except for the f_5 and f_6 functions. For the multi-modal benchmark function(f_9, f_{10}, f_{11}, f_{12} and f_{13}), the improved algorithm can find the optimal solution than other algorithms. In the f_8 function, although the QSSA algorithm fail to achieve a good ranking, it obtain the smaller standard deviation. Fixed-dimension multimodal benchmark

TABLE 2. Multi-modal benchmark functions.

Function	Dim	Range	f_{min}
$F_8(x)=\sum_{i=1}^{n}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	n	[-500,500]	-418.9829×5
$F_9(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos\left(2\pi x_i\right)+10\right]$	n	[-5.12,5.12]	0
$F_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)\right)+20+e$	n	[-32,32]	0
$F_{11}(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	n	[-600,600]	0
$F_{12}(x)=\frac{\pi}{n}\{10\sin\left(\pi y_1\right)+\sum_{i=1}^{n-1}\left(y_i-1\right)^2\left[1+10\sin^2\left(\pi y_{i+1}\right)\right]+\left(y_n-1\right)^2\}$ $+\sum_{i=1}^{n}u\left(x_i,10,100,4\right)$ $y_l=1+\frac{x_i+1}{4}$ $u\left(x_i,a,k,m\right)=\begin{cases}k\left(x_i-a\right)^m & x_i>a\\ 0 & -a<x_i<a\\ k\left(-x_i-a\right)^m & x_i<-a\end{cases}$	n	[-50,50]	0
$F_{13}(x)=0.1\{\sin^2\left(3\pi x_1\right)+\sum_{i=1}^{n}\left(x_i-1\right)^2\left[1+\sin^2\left(3\pi x_i+1\right)\right]$ $+\left(x_n-1\right)^2\left[1+\sin^2\left(2\pi x_n\right)\right]\}+\sum_{i=1}^{n}u\left(x_i,5,100,4\right)$	n	[-50,50]	0

TABLE 3. Fixed-dimension multi-modal benchmark functions.

Function	Dim	Range	f_{min}
$F_{14}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right)^{-1}$	2	[-65,65]	1
$F_{15}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1\left(b_i^2-b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[-5,5]	0.003
$F_{16}(x)=4x_1^2-2.1x_1^2+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5,5]	-1.0316
$F_{17}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10(1-\frac{1}{8\pi})\cos x_1+10$	2	[-5,5]	0.398
$F_{18}(x)=\left[1+\left(x_1+x_2+1\right)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]$ $\times\left[30+\left(2x_1-3x_2\right)^2\times\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right]$	2	[-2,2]	3
$f_{19}=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}\left(x_j-p_{ij}\right)^2\right)$	3	[1,3]	-3.86
$f_{19}=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{6}a_{ij}\left(x_j-p_{ij}\right)^2\right)$	6	[0,1]	-3.32
$F_{21}(x)=-\sum_{i=1}^{5}\left[\left(X-a_i\right)\left(X-a_i\right)^T+c_i\right]^{-1}$	4	[0,10]	-10.1532
$F_{22}(x)=-\sum_{i=1}^{7}\left[\left(X-a_i\right)\left(X-a_i\right)^T+c_i\right]^{-1}$	4	[0,10]	-10.4028
$F_{23}(x)=-\sum_{i=1}^{10}\left[\left(X-a_i\right)\left(X-a_i\right)^T+c_i\right]^{-1}$	4	[0,10]	-10.5363

functions can detect the ability of the algorithm to jump out of the local optimal solution. For the multimodal benchmark problems in Table 6, one can see that QSSA can obtain better mean values than other algorithms. Table 7 shows the convergence accuracy between the improved algorithm and other algorithms. In this table, NA indicates "not available" value. Specifically, the QSSA algorithm can find the optimal solution in the f_{15}, f_{16}, f_{18}, f_{19}, f_{21}, f_{22} and f_{23} benchmark functions. Although the QSSA algorithm fails to find the best solution in the comparison with other benchmark functions, the results are not far from the algorithm with the highest ranking. For example, the gap result of the improved algorithm and PSO are only 0.0994 in the f_{14}. The results prove the stability and robustness of the QSSA in the benchmark function.

In this paper, the benchmark functions $f_1, f_3, f_7, f_9, f_{15}$ and f_{18} are selected for the trajectory, the average fitness value and the convergence curve are shown in Figure 7. Individual convergence trajectory is used to evaluate the performance of the algorithm, we only record the first generation trajectory of the salps to observe the search area in this problem. Observing the trajectory of the benchmark function, it is found that most individuals will change their positions randomly and tend to be stable in the iteration. Another indicator is the change in the average fitness, especially the initial period of the algorithm. The average fitness reflects the global optimization

TABLE 4. Updating strategies.

Algorithm	Key parameter	Updating formula
ESSA	c_1	$c_1=2e^{-\left(\frac{4r_1l}{L}\right)^2}$
SSA	c_1	$c_1=2e^{-\left(\frac{4l}{L}\right)^2}$
CSSA	$c=4$	$o_{s+1}=c\cdot o_s\left(1-o_s\right)$
GWO	a	$a=2-\frac{2l}{L}$
WOA	a	$a=2-\frac{2l}{L}$
PSO	w,c_1,c_2	$w=1$,$c_1=2$,$c_2=2$
GSA	G_0	$G(t)=G_0\times\mathrm{e}^{-\alpha t/T}$
FEP	τ',τ	$\eta_i'(j)=\eta_i(j)\exp(\tau'N(0,1)$ $+\tau N_j(0,1))$
CS	β	$\sigma=\left\{\frac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\beta\Gamma\left(\frac{1+\beta}{2}\right)2^{\frac{\beta-1}{2}}}\right\}^{\frac{1}{\beta}}$

TABLE 5. Results of uni-modal benchmark functions.

Function	Index	QSSA	ESSA	SSA	CSSA	GWO	WOA	PSO	GSA	FEP	CS
f_1	Mean	0	5.12E-38	3.29E-07	1.40E-09	1.27E-27	1.41E-30	1.31E-05	2.53E-16	0.00057	5.78E-03
	Std	0	2.44E-37	5.92E-07	3.31E-10	3.11E-27	4.91E-30	2.39E-05	9.67E-17	0.00013	2.41E-03
	Rank	1	2	7	6	4	3	8	5	10	9
f_2	Mean	2.47E-323	2.01E-25	1.9111	4.14E-06	8.52E-17	1.06E-21	0.0076	5.57E-02	0.0081	2.08E-01
	Std	0	7.74E-25	1.6142	1.68E-06	6.62E-17	2.39E-21	0.0262	0.19407	0.00077	3.17E-02
	Rank	1	2	10	5	4	3	8	6	9	7
f_3	Mean	0	3.36E-35	1.50E+03	5.40E-10	2.43E-05	5.39E-07	2.13E+03	8.97E+02	0.016	2.63E-01
	Std	0	1.79E-34	707.0529	1.94E-10	8.14E-05	2.93E-06	878.7605	318.96	0.014	2.97E-02
	Rank	1	2	9	3	5	4	10	8	7	6
f_4	Mean	4.94E-324	4.25E-18	2.44E-05	1.08E-05	7.69E-07	0.072581	16.4104	7.35E+00	0.3	1.43E-05
	Std	0	2.1E-17	1.89E-05	1.55E-06	6.51E-07	0.39747	4.3603	1.74145	0.5	4.83E-06
	Rank	1	2	6	4	3	7	10	9	8	5
f_5	Mean	3.4174	28.8803	136.5676	8.9354	27.1786	27.86558	106.6465	67.543	5.06	0.008
	Std	3.7151	0.0298	1.54E+02	0.0173	0.814	0.763626	104.3333	62.225	5.87	0.054
	Rank	2	7	10	4	5	6	9	8	3	1
f_6	Mean	5.71E-11	3.3387	1.72E-07	1.77E-10	0.70757	3.116266	5.59E-05	2.50E-16	0	6.17E-04
	Std	2.35E-10	0.8243	2.44E-07	3.89E-11	0.3632	0.532429	1.42E-04	1.74E-16	0	2.8E-05
	Rank	3	10	5	4	8	9	6	2	1	7
f_7	Mean	6.92E-05	8.82E-05	0.169	9.06E-05	1.72E-03	0.001425	6.05E-02	8.94E-02	0.1415	0.02855
	Std	6.33E-05	6.94E-05	0.0686	9.27E-05	1.10E-03	0.001149	0.0206	0.04339	0.3522	0.001277
	Rank	1	2	10	3	5	4	7	8	9	6

TABLE 6. Results of multi-modal benchmark functions.

Function	Index	QSSA	ESSA	SSA	CSSA	GWO	WOA	PSO	GSA	FEP	CS
f_8	Mean	-7014.46	-4830	-7460	-2710	-5776.12	-5080.7	-9347.8	-2821.07	-12554	-2128.91
	Std	117.69	709.30	634.6745	639.7074	682.0101	695.796	754.483	493.037	52.6	0.0084
	Rank	4	7	3	9	5	6	2	8	1	10
f_9	Mean	0	0	55.4523	1.30E-10	3.1496	0	58.0837	25.968	0.046	0.246
	Std	0	0	18.2751	2.26E-10	4.0294	0	13.1635	7.47	0.012	0.0018
	Rank	1	1	10	4	7	1	9	8	6	5
f_{10}	Mean	8.88E-16	8.88E-16	2.8401	7.03E-06	1.027E-13	7. 4043	0.9353	0.0621	0.018	4.1E-10
	Std	2.01E-31	0	0.6581	2.41E06	1.60E-14	9.89757	0.9909	0.2363	0.0021	5.2E-09
	Rank	1	1	9	5	3	10	8	7	6	4
f_{11}	Mean	0	0	0.2291	1.64E-09	6.08E-03	0.00028	1.9E-02	27.702	0.016	0.1852
	Std	0	0	0.1295	8.19E-10	0.0111	0.00158	0.0232	5.0403	0.022	0.039
	Rank	1	1	9	3	5	4	7	10	6	8
f_{12}	Mean	1.72E-09	0.2724	6.8274	0.1462	0.03669	0.33967	0.9099	1.79962	9.0E-06	0.01258
	Std	6.51E-09	0.1555	2.7192	0.0627	0.0175	0.21486	1.0678	0.9511	3.0E-06	4.1E-09
	Rank	1	6	10	5	4	7	8	9	2	3
f_{13}	Mean	8.15E-09	2.8525	21.3116	0.0015	0.62239	1.88901	0.119	8.8991	0.0002	0.485
	Std	3.44E-08	0.3519	16.9894	3.49E-04	0.2837	0.26608	0.2396	7.1262	0.00007	6.8E-08
	Rank	1	8	10	3	6	7	4	9	2	5

ability of the population. It can be seen from the figure that average fitness decrease in proportion to the iteration number. The convergence curve comparison is used to evaluate the advantages and disadvantages of this algorithm. In the convergence curves of various basic intelligent algorithms, QSSA can find the optimal solution faster and more accurate than other algorithms. The comparison with traditional methods suggests that the QSSA can achieve better results than other algorithms. Therefore, we have great significance for the improvement of the classic SSA.

Quantum behavior mechanisms can make leading salps generate quantize variation, which can better prevent the population from falling into local optimization. At the same time, the elite reverse learning strategy increases the ''survival of the fittest'' of the follower salps, which can make the algorithm evolve in a better direction. In the later stage of iteration, taking into consideration the wavelet mutation strategy, and we mainly mutate the individuals whose fitness value is less than a certain value. The correctness of theoretical analysis is validated by experiments, so we conclude that QSSA can find the optimal solution of the model within a very short time.

IV. ENGINEERING OPTIMIZATION OF IMPROVED ALGORITHM

In this section, the propose algorithm is used to optimize engineering structure problems. It is difficult to solve in

TABLE 7. Results of fixed-dimension multimodal benchmark functions.

Function	Index	QSSA	ESSA	SSA	CSSA	GWO	WOA	PSO	GSA	FEP	CS
f_{14}	Mean	1.0974	3.9752	1.1301	NA	4.917	2.111973	0.998	5.8598	1.22	1. 4236
	Std	0.3033	3.6295	0.5659	NA	4.125	2.498594	0	3.83	0.56	1.3E-02
	Rank	2	7	3	10	8	6	1	9	4	5
f_{15}	Mean	3.32E-04	0.0024	0.0027	0.0015	5.08E-03	0.000572	3.20E-03	0.003673	0.0005	5.0E-04
	Std	6.34E-05	0.0053	0.00543	3.5E-04	0.0086	0.000324	0.0071	1.65E-03	0.00032	1.11E-04
	Rank	1	6	7	5	10	4	8	9	2	2
f_{16}	Mean	-1.0316	-1.0316	-1.0316	NA	-1.0316	-1.03163	-1.0316	-1.0316	-1.03	-1.0316
	Std	0	6.5189E-06	5.78E-14	NA	2.64E-08	4.2E-07	6.71E-16	4.88E-16	4.9E-07	1.49E-08
	Rank	1	1	1	3	1	1	1	1	2	1
f_{17}	Mean	0.3979	0.3979	0.3979	NA	0.3979	0.397914	0.3979	0.397887	0.398	0.3979
	Std	5.48E-07	0.000011	1.26E-14	NA	8.86E-07	2.7E-05	0	0	1.5E-07	3.24E-06
	Rank	2	2	2	5	1	4	2	3	1	2
f_{18}	Mean	3	3.0002	3	NA	3	3	3	3	3.02	3.0014
	Std	0	0.00036902	2.66E-13	NA	4.27E-05	4.22E-15	1.26E-15	4.17E-15	0.11	0.00258
	Rank	1	2	1	5	1	1	1	1	4	3
f_{19}	Mean	-3.8628	-3.8261	-3.8628	NA	-3.861	-3.85616	-3.8625	-3.86278	-3.86	-3.268
	Std	4.52E-16	0.0533	1.1304E-11	NA	0.0023	0.002706	0.0014	2.29E-15	0.000014	1.85E-05
	Rank	1	8	1	10	5	7	4	3	6	9
f_{20}	Mean	-3.2621	-3.1703	-3.22526	NA	-3.2568	-2.98105	-3.2518	-3.31778	-3.27	-3.32185
	Std	0.0664	0.1191	0.05772	NA	0.0885	0.376653	0.1086	0.02308	0.059	7.21E-03
	Rank	4	8	7	10	5	9	6	2	3	1
f_{21}	Mean	-10.1532	-10.1478	-7.33	-9.8918	-9.3955	-7.04918	-6.2948	-5.95512	-5.52	-9.728
	Std	3.6134E-15	0.0143	3.3158	0.187	2	3.629551	2.9249	3.73708	1.59	0.2881
	Rank	1	2	6	3	5	7	8	9	10	4
f_{22}	Mean	-10.4029	-10.4004	-8.480225	NA	-10.2241	-8.18178	-7.0398	-9.68447	-5.53	-9.873
	Std	7.23E-15	0.005	3.094025	NA	0.9701	3.829202	3.5751	2.014	2.12	0.32034
	Rank	1	2	6	10	3	7	8	5	9	4
f_{23}	Mean	-10.5364	-10.5331	-8.64195	NA	-10.0838	-9.34238	-7.7011	-10.5364	-6.57	-9.7822
	Std	9.03E-15	0.0051	3.1316	NA	1.7514	2.414737	3.6189	2.6E-15	3.14	0.5002
	Rank	1	3	7	10	4	6	8	1	9	5

precision because engineering optimization is a typical NP-hard problem in combinatorial optimization. So the swarm intelligence algorithm is widely considered to the best way to get the optimal solution compared with other methods. Through comparing the latest known optimizer, the experimental results show the validity and availability of the improve salp swarm algorithm.

A. WELDED BEAM STRUCTURE DESIGN PROBLEM

Welded beam design is a typical optimization problem in the civil engineering field, which can be described simply as finding the minimum cost under a particular constraint. Four main variables affect the optimization results, including the length of the bar attached to the weld (l), the weld thickness (h), the height of the bar (t), and the thickness of the bar(b)(as shown in the Fig.8). It can be expressed as: $X^T = \{x_1, x_2, x_3, x_4\} = \{l, h, t, b\}$. The optimization model is expressed as follows:

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \quad (29)$$

we can get the relevant constraints according to the engineering conditions.

$$\begin{cases} g_1(x) = \tau(x) - 13000 \le 0 \\ g_2(x) = \sigma(x) - 30000 \le 0 \\ g_3(x) = x_1 - x_4 \le 0 \\ g_4(x) = 0.1047x_1^2 \\ +0.04811x_3x_4\,(14.0 + x_2) - 5.0 \le 0 \\ g_5(x) = 0.125 - x_1 \le 0 \\ g_6(x) = \delta(x) - 0.25 \le 0 \\ g_7(x) = 6000 - P_c(x) \le 0 \end{cases} \quad (30)$$

where :

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},$$
$$\tau' = \frac{6000}{\sqrt{2}x_1x_2}\tau'' = \frac{MR}{J},$$
$$M = 6000\left(14 + \frac{x_2}{2}\right),$$

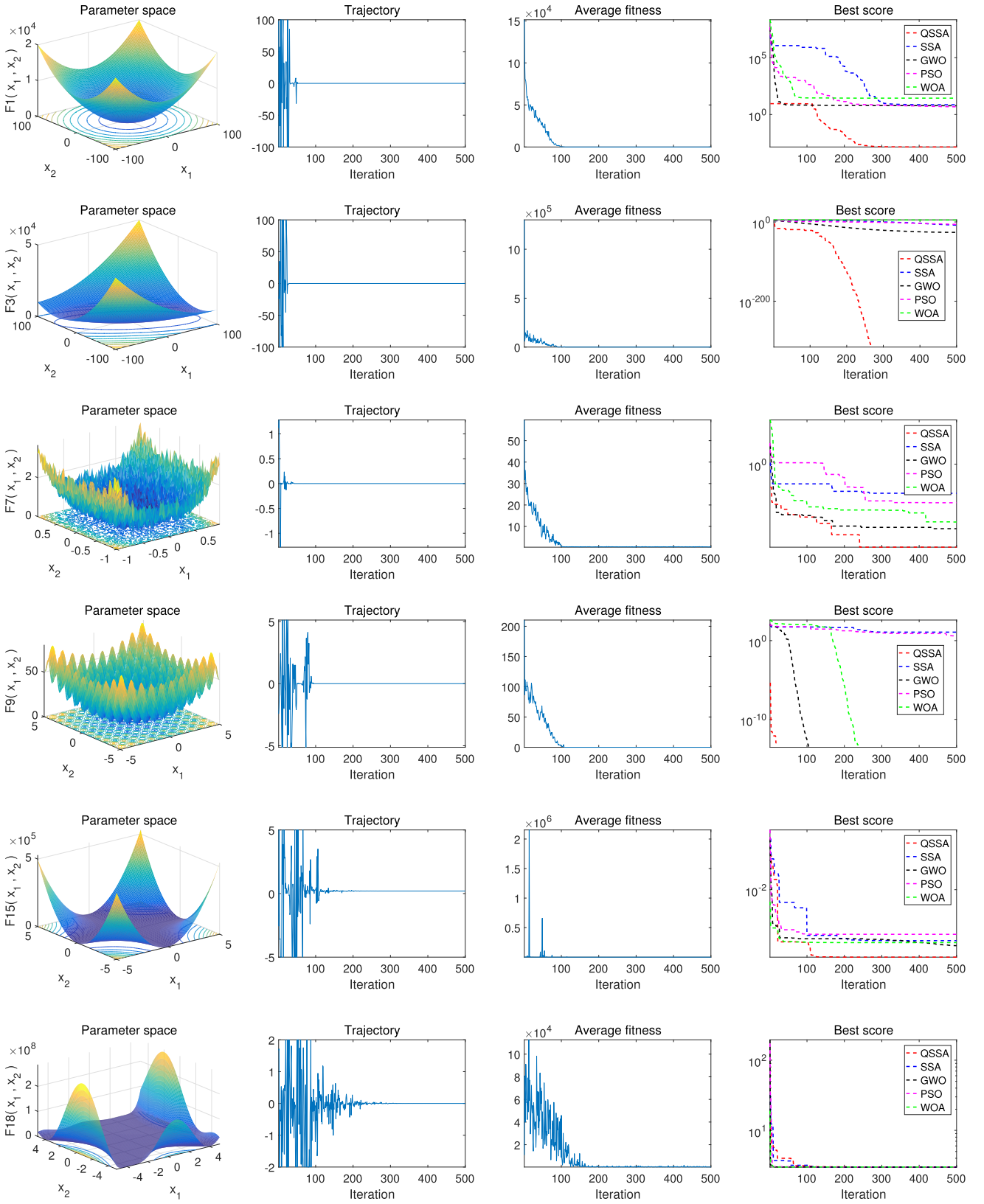


FIGURE 7. Welded beam design problem.

TABLE 8. Comparison of best obtained results for welded beam problem.

algorithm	x_1	x_2	x_3	x_4	$f(x)$
QSSA	0.2028	3.3757	9.0368	0.2057	1.7076
SSA	0.2057	3.4714	9.0366	0.2057	1.7249
SA-MFO	0.2057	3.4724	9.0367	0.2057	1.7245
MFO	0.2051	3.4847	9.0365	0.2057	1.7249
MBA	0.2057	3.4705	9.0366	0.2057	1.7248
MHS-PCLS	0.2057	3.4705	9.0366	0.2057	1.7249
NM-PSO	0.2058	3.4683	9.0366	0.2057	1.7247

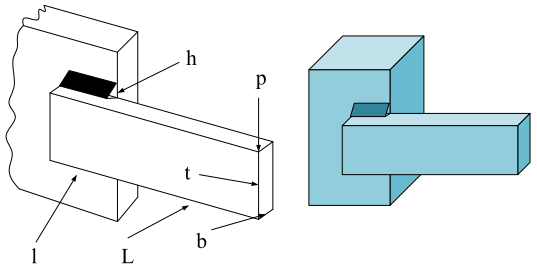


FIGURE 8. 2K-H planetary transmission design problem.

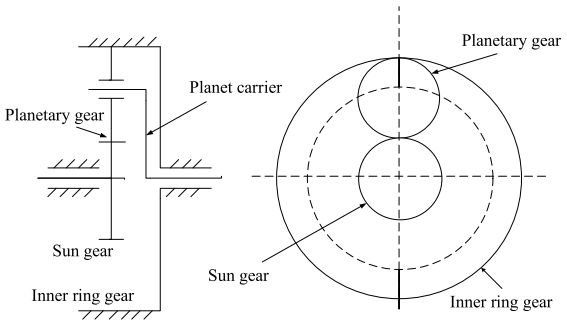


FIGURE 9. Trapezoidal leaf spring design problem.

$$
\begin{aligned}
&R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \\
&J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}, \\
&\sigma(\boldsymbol{x}) = \frac{504000}{x_4x_3^2}, \delta(\boldsymbol{x}) = \frac{2.1952}{x_3^2x_4}, \\
&P_c(\boldsymbol{x}) = 64746.022\,(1 - 0.0282346x_3)\,x_3x_4^3 \\
&0.1 \le x_1 \le 2, 0.1 \le x_2 \le 10, \\
&0.1 \le x_3 \le 10, 0.1 \le x_4 \le 2.
\end{aligned} \tag{31}
$$

The results of the welded beam design problem are shown in table 8 in terms of SSA, SA-MFO [30], MFO [31], MBA [32], MHS-PCLS [33], and NM-PSO [34]. From the experimental results, QSSA obtaine better solutions than other algorithms after 500 iterations.

B. PLANETARY GEAR TRANSMISSION DESIGN PROBLEM

Planetary gear transmission design is a multivariable, multi-constrained, nonlinear optimization design problem. The volume of the 2K-H planetary transmission mechanism is optimized based on the acquired background knowledge in this paper, and we can see the basic structure model diagram from Fig.9.

Our optimization model is mainly affected by three variables including the number of sun gear(Z), the fixed width of the inner gear ring(B) and the gear modulus(M). It can be expressed as: $X^T = \{x_1, x_2, x_3\} = \{Z, B, M\}$. The optimization model is expressed as follows:

$$f(x) = \tfrac{\pi}{4}x_3^2\,(x_1 + 2.5)^2\,x_2 \tag{32}$$

we can get the relevant constraints according to the engineering conditions.

$$
\begin{cases}
Y_1 = x_1 - 17 \ge 0 \\
Y_2 = x_2 - 10 \ge 0 \\
Y_3 = x_3 - 2 \ge 0 \\
Y_4 = x_2 - 5x_3 \ge 0 \\
Y_5 = 17x_3 - x_2 \ge 0 \\
Y_6 = x_1^2x_3^2x_2k \\
-A_H \cdot T_a \cdot u/(u-2) \ge 0 \\
Y_7 = [i_{aH}\sin(\pi/k) \\
-i_{aH} + 1]x_1 - 2 \ge 0 \\
Y_8 = x_1x_3^2x_2k \\
-A_F T_{\mathrm{a}}\,(4.69 - 0.63\ln x_1) \ge 0
\end{cases} \tag{33}
$$

where :

$$
\begin{aligned}
&A_H = 800^3K_AK_{\beta\mathrm{H}}/\left[\sigma_{\mathrm{H}}\right]^2, A_{\mathrm{F}} = 13^3K_{\mathrm{A}}K_{\beta\mathrm{F}}/\left[\sigma_{\mathrm{F}}\right] \\
&k = 3, K_A = 1.4, K_{\beta\mathrm{H}} = 1.1, [\sigma_{\mathrm{H}}] = 790, T_a = 1.8, \\
&u = 1.57, i_{aH} = 4.14, K_{\beta\mathrm{F}} = 1.15, [\sigma_{\mathrm{F}}] = 158
\end{aligned} \tag{34}
$$

To assess the precision of the QSSA compared to other optimization algorithms. Simulation results of the QSSA compared to SSA, GOA, and different state-of-the-art intelligence algorithms like WOA, SCA [35], MFO, and MVO [36] for optimizing the volume of 2K-H planetary transmission mechanism are recorded in Table 9. Simulation results indicate that the result of Planetary reducer volume optimization is satisfactory, and application of the improved algorithm to the engineering is feasible.

TABLE 9. The volume result of the planetary gears.

function	x_1	x_2	x_3	$f(x)$
QSSA	17.0000	85.7125	5.17496	9575582.48
SSA	17.0000	89.8279	5.28399	10462655.01
GOA	17.0007	88.7268	5.21927	10083622.46
WOA	17.0000	87.6907	5.39555	10649564.48
SCA	17.0015	87.0031	5.70173	11801315.78
MFO	17.0123	86.1056	5.34533	10277624.19
MVO	17.0570	90.3776	5.39662	11051503.93

C. TAPER-LEAF SPRING DESIGN PROBLEM

A small piece of variable-section leaf spring is an important part of the vehicle, and the purpose of the design is to reduce

the weight of the objective function. The objective function is constrained by four variables(h_1), length(h_2), thickness(l_1), blade width and the number of blades(l)(as shown in the Fig.10). It can be expressed as: $X^T = \{x_1, x_2, x_3, x_4\} = \{h_1, h_2, l_1, l\}$

The mathematical expression for taper-leaf spring is as follows:

$$f(x) = 2\rho bn[x_1x_3 + x_2l_3 + 0.5\,(x_4 - x_3 - l_3)\,(x_1 + x_2)] \quad (35)$$

Taking into account the arrangement of steel spring, stiffness, strength, material, size and manufacturing process of the steel spring, the following constraint equations can be established.

$$\begin{cases} g_1(X) = x_1 - H_1 \geq 0 \\ g_2(X) = H_2 - x_2 \geq 0 \\ g_3(X) = x_2 - x_1 - 1 \geq 0 \\ g_4(X) = x_3 \geq 0 \\ g_5(X) = L/2 - x_4 \geq 0 \\ g_6(X) = K_e - \left|\dfrac{K - K_c}{K_c}\right| \geq 0 \\ g_7(X) = [\sigma]_1 - \dfrac{6Px_3}{nbx_1^2} \geq 0 \\ while: \\ x_3 > (x_4 - l_3)\left(2\dfrac{x_1}{x_2} - 1\right), \\ g_8(X) = [\sigma]_2 - \dfrac{1.5P\,(x_2 - x_1)}{nb\,[x_1\,(x_4 - l_3) - x_2x_3]} \\ \left[\dfrac{(x_4 - l_3) - x_3}{x_2 - x_1}\right]^2 \geq 0 \\ else: \\ x_3 \leq (x_4 - l_3)\left(2\dfrac{x_1}{x_2} - 1\right), \\ g_8(X) = [\sigma]_2 - \dfrac{6P\,(x_4 - l_3)}{nbx_2} \geq 0 \end{cases} \quad (36)$$

where:

$$K = \frac{6EJ_2\zeta}{x_4^3} \cdot \frac{1}{\eta}, J_2 = \frac{nbx_2^3}{12}$$

$$\eta = 1 + \left(\frac{x_4 - l_3}{x_4}\right)^3 \cdot \left[1 - 2\left(\frac{x_3}{x_4 - l_3}\right)^{\frac{3}{2}}\right] + \left(\frac{x_3}{x_4}\right)^3 \cdot \left(\frac{x_2}{x_1}\right)$$

$$\rho = 7.8 \times 10^{-6}, b = 90, n = 3$$

$$H_1 = 8, H_2 = 15, L = 1720$$

$$K_e = 0.008, K_c = 176, \zeta = 0.92$$

$$[\sigma]_1 = 350, [\sigma]_2 = 500, l_3 = 40$$

$$P = 16170, E = 2.1 \times 10^5 \quad (37)$$

From Table 10, we observe that QSSA has the lowest optimizing accuracy for solving the taper-leaf spring. The design of this article can make the structure of the taper-leaf spring cross section more reasonable and ensure its stability and smoothness.

TABLE 10. The results of taper-leaf spring.

algorithm	x_1	x_2	x_3	x_4	$f(x)$
QSSA	8.0000	12.6497	54.8482	557.3966	24.0948
SSA	9.5718	13.2140	89.2402	586.6462	27.7736
GOA	8.0095	13.4735	60.1924	592.7502	26.5856
WOA	8.0000	13.6283	50.0000	596.6981	27.0605
SCA	9.8358	12.3987	67.9163	547.8393	25.5024
MFO	8.8835	13.1039	75.8601	576.9112	26.3953
MVO	8.0000	14.0163	61.0710	615.3451	28.2643

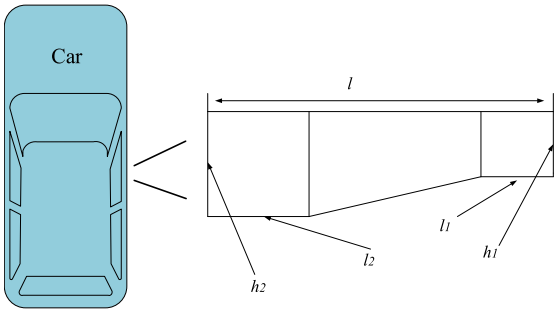


FIGURE 10. Comparison of algorithm results: function, trajectory, average fitness, and convergence.

V. CONCLUSION AND FUTURE DIRECTIONS

Due to the effect on salp swarm optimization, we have proposed a improved SSA algorithm in this paper. The main contributions of this work can be summarized as follows. Firstly, quantum mechanics are used to update the position of leading salps and enhances the global search ability of the population. Secondly, we applyÂ the elite reverse learning strategy toÂ guide the follower salps, which can keep the balance between exploration and exploitation. In the later stage of iteration, taking into consideration the wavelet mutation strategy, and we mainly mutate the individuals whose fitness value is less than a certain value. So that, the probability that the population escapes from the locally optimal region can be improved. A large quantity of simulation data indicates that the improved algorithm can obtain better solutions than other algorithms in both uni-mode and multi-modal benchmark functions. It is ideal for solving complex engineering optimization problems in the real world, including welded beam structure optimization, planetary gear transmission optimization, and taper-leaf spring optimization.

Although the QSSA algorithm has good performance in solving single-objective problems, but the study had its limitations. We are not sure whether the algorithm has a good solution for multi-objective problems and dynamic optimization. These problems will be the focus of future research.

The future work includes studies on how to extend QSSA to solve more problems and explore the efficiency of it on engineer problems. It's a new way to improve the SSA combines with other optimizers, and more tests should be

conducted in this way. Besides, the optimal values of the parameters in the proposed methods will be investigated in the future.

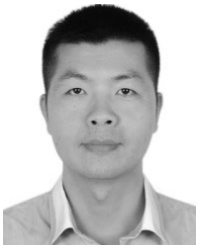
REFERENCES

[1] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm," *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Apr. 2007.
[2] J. Kennedy, "Particle swarm optimization," *Encyclopedia Mach. Learn.*, pp. 760–766, 2010.
[3] W. Guo, J. Li, G. Chen, Y. Niu, and C. Chen, "A PSO-optimized real-time fault-tolerant task allocation algorithm in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 12, pp. 3236–3249, Dec. 2015.
[4] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017.
[5] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.
[6] G. L. Zhang, B. Li, B. Yu, D. Z. Pan, and U. Schlichtmann, "TimingCamouflage: Improving circuit security against counterfeiting by unconventional timing," in *Proc. Design, Automat. Test Eur. Conf. Exhib. (DATE)*, Mar. 2018, pp. 91–96.
[7] G. L. Zhang, B. Li, J. Liu, Y. Shi, and U. Schlichtmann, "Design-phase buffer allocation for post-silicon clock binning by iterative learning," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 37, no. 2, pp. 392–405, Feb. 2018.
[8] C. Dong, Z. Xiong, X. Liu, Y. Ye, Y. Yang, and W. Guo, "Dual-search artificial bee colony algorithm for engineering optimization," *IEEE Access*, vol. 7, pp. 24571–24584, 2019.
[9] D. Tian and Z. Shi, "MPSO: Modified particle swarm optimization and its applications," *Swarm Evol. Comput.*, vol. 41, pp. 49–68, 2018.
[10] C. Dong, Y. Ye, X. Liu, Y. Yang, and W. Guo, "The sensitivity design of piezoresistive acceleration sensor in industrial IoT," *IEEE Access*, vol. 7, pp. 16952–16963, 2019.
[11] G. Liu, W. Guo, R. Li, Y. Niu, and G. Chen, "XGRouter: high-quality global router in X-architecture with particle swarm optimization," *Front. Comput. Sci.*, vol. 9, no. 4, pp. 576–594, Jul. 2015.
[12] R. Abbassi, A. Abbassi, A. A. Heidari, and S. Mirjalili, "An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models," *Energy Convers. Manage.*, vol. 179, pp. 362–372, Jan. 2019.
[13] M. Tolba, H. Rezk, A. Diab, and M. Al-Dhaifallah, "A novel robust methodology based salp swarm algorithm for allocation and capacity of renewable distributed generators on distribution grids," *Energies*, vol. 11, no. 10, pp. 362–372, Oct. 2018.
[14] P. C. Sahu, S. Mishra, R. C. Prusty, and S. Panda, "Improved-salp swarm optimized type-II fuzzy controller in load frequency control of multi area islanded AC microgrid," *Sustain. Energy, Grids Netw.*, vol. 16, pp. 380–392, Dec. 2018.
[15] H. Faris, M. M. Mafarja, A. A. Heidari, I. Aljarah, A. M. Al-Zoubi, S. Mirjalili, and H. Fujita, "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems," *Knowl.-Based Syst.*, vol. 154, pp. 43–67, Aug. 2018.
[16] S. K. Baliarsingh, S. Vipsita, K. Muhammad, B. Dash, and S. Bakshi, "Analysis of high-dimensional genomic data employing a novel bio-inspired algorithm," *Appl. Soft Comput.*, vol. 77, pp. 520–532, Apr. 2019.
[17] M. A. Elfattah, A. E. Hassanien, and S. Abuelenin, "A hybrid swarm optimization approach for document binarization," *Stud. Informat. Control*, vol. 28, no. 1, pp. 65–76, Mar. 2019.
[18] M. H. Qais, H. M. Hasanien, and S. Alghuwainem, "Enhanced salp swarm algorithm: Application to variable speed wind generators," *Eng. Appl. Artif. Intell.*, vol. 80, pp. 82–96, Apr. 2019.
[19] I. Aljarah, M. Mafarja, A. A. Heidari, H. Faris, Y. Zhang, and S. Mirjalili, "Asynchronous accelerating multi-leader salp chains for feature selection," *Appl. Soft Comput.*, vol. 71, pp. 964–979, Oct. 2018.
[20] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proc. Congr. Evol. Comput.*, vol. 1. Jun. 2004, pp. 325–331.
[21] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," in *Proc. Int. Conf. Comput. Intell. Modelling, Control Automat. Int. Conf. Intell. Agents, Web Technol. Internet Commerce (CIMCA-IAWTIC)*, vol. 1, Nov. 2005, pp. 695–701.
[22] Z. Guo, J. Shi, X. Xiong, X. Xia, and X. Liu, "Chaotic artificial bee colony with elite opposition-based learning," *Int. J. Eng. Sci.*, vol. 18, no. 4, pp. 383–390, 2019.
[23] S. K. Saha, R. Kar, D. Mandal, and S. P. Ghoshal, "Optimal IIR filter design using gravitational search algorithm with wavelet mutation," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 27, no. 1, pp. 25–39, Jan. 2015.
[24] G. I. Sayed, G. Khoriba, and M. H. Haggag, "A novel chaotic salp swarm algorithm for global optimization and feature selection," *Appl. Intell.*, vol. 48, no. 10, pp. 3462–3481, Oct. 2018.
[25] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.
[26] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.
[27] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.
[28] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.
[29] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. IEEE World Congr. Nature Biologically Inspired Comput. (NaBIC)*, Aug. 2009, pp. 210–214.
[30] G. I. Sayed and A. E. Hassanien, "A hybrid SA-MFO algorithm for function optimization and engineering design problems," *Complex Intell. Syst.*, vol. 4, pp. 195–212, Oct. 2018.
[31] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.
[32] A. Sadollah, A. Bahreininejad, H. Eskandar, and M. Hamdi, "Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2592–2612, May 2013.
[33] J. Yi, X. Li, C.-H. Chu, and L. Gao, "Parallel chaotic local search enhanced harmony search algorithm for engineering design optimization," *J. Intell. Manuf.*, vol. 30, no. 1, pp. 405–428, Jan. 2019.
[34] E. Zahara and Y.-T. Kao, "Hybrid Nelder–Mead simplex search and particle swarm optimization for constrained engineering design problems," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3880–3886, Mar. 2009.
[35] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.
[36] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Feb. 2016.

RONGZHONG CHEN received the B.S. degree in computer science and technology from Shandong Jiaotong University, China, in 2017. He is currently pursuing the master's degree with the School of Mathematics and Computer Science, Fuzhou University, and the Fujian Provincial Key Laboratory of Information Security of Network Systems. His current research interests include intelligent computing and integrated circuit physical design, and their applications to engineering optimization and voltage island.

CHEN DONG received the B.S. and M.S. degrees from the College of Mathematics and Computer Science, Fuzhou University, China, in 2002 and 2005, respectively, and the Ph.D. degree in computer science from the Computer School, Wuhan University, China, in 2011. She was a Visiting Researcher with the University of California, Los Angeles, from 2015 to 2016. She is currently an Assistant Professor with the College of Mathematics and Computer Science, Fuzhou University. Her research interests include artificial intelligence, intelligent computing, and integrated circuit physical design.

YIN YE received the B.S. degree in civil engineering from the Jiangxi University of Science and Technology, China, in 2017. He is currently pursuing the master's degree with the Fujian Provincial Key Laboratory of Information Security of Network Systems, College of Mathematics and Computer Science, Fuzhou University. His research interests include meta-heuristic algorithm, intelligent computing, and engineering application.



ZHENYI CHEN received the B.S. and M.S. degrees from the College of Mathematics and Computer Science, Fuzhou University, China, in 2000 and 2005, respectively, and the Ph.D. degree in computer science from the Computer School, Wuhan University, China, in 2012. He is currently pursuing the second Ph.D. degree with the University of South Florida. His research interests include swarm intelligence optimization and machine learning.

YANHUA LIU received the B.S. and M.S. degrees from the College of Mathematics and Computer Science, Fuzhou University, China, in 1996 and 2003, respectively, and the Ph.D. degree from the College of Physics and Information Engineering, Fuzhou University, in 2016, where he is currently an Associate Professor and the Core Researcher of the Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing. His research interests include artificial intelligence and intelligent computing. His research work received several government awards.