IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 22, 2021, accepted February 4, 2021, date of publication February 9, 2021, date of current version February 18, 2021.
*Digital Object Identifier 10.1109/ACCESS.2021.3058128*

# A Modified Sine Cosine Algorithm for Solving Optimization Problems

**MENG WANG AND GUIZHEN LU**
School of Information and Communication Engineering, Communication University of China, Beijing 100024, China
Corresponding author: Meng Wang (wangmeng1993@cuc.edu.cn)

**ABSTRACT** The sine cosine algorithm (SCA) is a newly emerging optimization algorithm. It is easy for sine cosine algorithm (SCA) to sink into premature of the algorithm and obtain a slower convergence rate when solving the complicated optimization problems, especially highly ill-posed problems. A novel modified sine cosine algorithm (MSCA) is put forward for solving the optimization problems. To our limited knowledge, the linear searching path and empirical parameter have not been applied to the related improved sine cosine algorithm. The proposed MSCA improves the search path of original SCA by introducing linear searching path and empirical parameter, effectively avoiding sinking into the local optimal. In addition, the proposed algorithm changes the definition of convergence factor. Two kinds of tests, including 23 benchmark functions test and actual engineering problem tests are adopted to prove the performance of the MSCA. In addition, the performance of the proposed MSCA is compared with SCA by using benchmark functions on different dimensional (D = 30, 50,100 and 500). As expected, the result of comparisons show that the proposed MSCA can better avoid the local optima than both the SCA and other population-based algorithms. And MSCA can obtain the faster convergence than SCA on different dimensions.

**INDEX TERMS** Sine cosine algorithm, linear searching path, nature-inspired algorithm, optimization.

## I. INTRODUCTION

The theory of optimization has been widely used in the actual engineering fields. By building suitable mathematical model according to concrete problems, in order to minimize (sometimes maximize) the fitness function of the concrete problem, many actual engineering problems belong to optimization problems. Examples include structural mechanics, power systems and so on [1], [2].

According to the academic results of the researchers, stochastic optimization methods regard the optimization problem as a black box, as illustrated in Figure 1. This means that there is no need for rigorous mathematical model derivation. Only it is necessary to continuously adjust the algorithm parameters according to the optimization results to obtain the purpose needed. Another advantage of treating the problem as a black box is flexibility, which means that stochastic algorithms can be conveniently applied to solve various engineering difficulties in different fields [3], [4].

In the past few decades, the research on stochastic optimization algorithms has become a hot research, and continuously improving the performance of optimization algorithms has become the research goal of researchers [5]. Commonly used algorithms include genetic algorithm (GA), particle swarm optimization (PSO) and differential evolution (DE). Meanwhile, nature-inspired algorithms are gradually rising, such as whale optimization algorithm (WOA), grey wolf optimizer (GWO) and sine cosine algorithm (SCA) [6]–[11]. Several common algorithms are introduced in the following paragraphs.

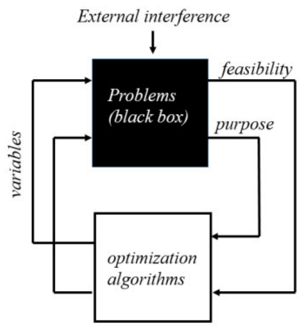


**FIGURE 1.** Stochastic optimization algorithm considers the system as black box.

Particle swarm optimization (PSO) is a global optimization algorithm, which was proposed by J. Kennedy and

The associate editor coordinating the review of this manuscript and approving it for publication was Bijoy Chand Chand Chatterjee.

R.C. Eberhaxt in 1995 [12]. It was designed to simulate the unpredictable foraging movement of a flock of birds. Imagine such a condition: all birds initially fly in search of food aimlessly, until one of them finds a little of food, then other birds have a trend to approach it and search its surrounding environment to form a specific flock of birds [13]. At the same time, because each bird tries to stop in the flock without colliding with each other, they determine their own flight direction and speed according to their own previous knowledge. When a bird finds the most satisfactory food at present, it will attract the other flock fly to the neighborhood, and the birds are continuously searching for places where more satisfactory food may exist as they approach them. Finally, the birds find the most plentiful foraging locations.

Inspired by this model, people use information society sharing to learn from the surrounding successful people, while maintaining a certain degree of autonomy based on individual experience and cognition, and ultimately the group finds the best [14], [15]. The above process is abstracted to obtain particles in a multidimensional space without volume and mass at a certain speed. The basic PSO algorithm traverses the search space based on its own and other particle history information [16]. Researchers have been working on the improvement of the PSO algorithm, such as GPSO, CPSO and CLPSO [17], [18].

Genetic algorithm (GA) belongs to a type of random search and optimization method which simulates biological evolution theory and its underlying genetic mechanism to solve optimization problems [19]. It was originally inspired by Michigan University professor Holland who proposed it in the 1970s and was enriched and applied by Goldberg and others [20].

Together with evolutionary strategies and evolutionary rules, it constitutes the foremost framework of genetic algorithms. GA has its solid biological foundation, namely, group random search strategy and the exchange mechanism of group individual information, which is in line with the cognitive view of the biological intelligence generation process; it does not rely on additional gradient information, and is available for any functions, and can be easily used in parallel computing [21], [22]. These advantages make GA become one of the best methods for solving complex optimization problems, widely used in path planning, engineering design, nonlinear problem, multi-objective problem, pattern recognition and other fields [23].

Differential evolution (DE) was proposed by Storn and Price in 1995. Differential evolution (DE) is a familiar optimization algorithm [24]. The algorithm principle of DE is very similar to GA. They all mainly include three processes: mutation, hybridization and selection, as shown in the Figure 2. But the specific process definitions of the algorithm are different. The method of DE algorithm to generate the initial population is to encode by using floating point vectors. [25]–[28].

Whale optimization algorithm (WOA) is a novel optimization algorithm, which was proposed by Mirjalili in 2016 [29].

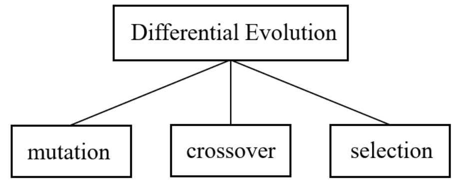


**FIGURE 2.** Three main processes of DE algorithm.

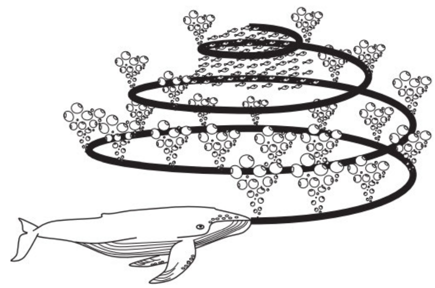

**FIGURE 3.** Bubble-net feeding behavior of humpback whale.

**FIGURE 4.** Foraging behavior of wolves.

The algorithm simulates the hunting behavior of humpback whales. It is divided into three stages: encircling, hunting and attacking prey. A corresponding theoretical model is established to achieve the best solution to an actual problem. According to the characteristics of humpback whale prey-predation behavior, the WOA mainly includes three stages: surrounding prey, bubble net hunting and prey search [30]. The bubble-net feeding behavior is as shown in Figure 3. (Figure 3 is from [29, Fig. 2])

Compared to other traditional optimization methods, WOA has the superiorities of simple structure, simple principle, fewer parameters, and strong optimization ability [31], [32]. The algorithm has been widely applied in water resources optimization configuration and truss design, fault diagnosis, photovoltaic cell parameter estimation and many other engineering issues [33].

The gray wolf optimization is a nature inspired optimization algorithm, which was proposed by Mirjalili *et al.* in 2014 [34]. The algorithm is inspired by the predation action and method of gray wolves, as the Figure 4 shows: (The Figure 4 is from [34, Fig.2]).

The gray wolf optimization algorithm is roughly divided into five steps: social caste system, surrounding prey, hunting, attacking and searching [35]. Gray wolves strictly obey the social dominance hierarchy. When the gray wolves are searching for the prey they need, they will slowly approach the prey and then surround the prey [9]. Gray wolves generally identify the prey first and then surround it. During each iteration, the three best gray wolves are kept, and the positions of the remaining search agents are updated based on their location information. When the prey is no longer moving, the gray wolf will attack to capture the prey [36], [37].

The sine cosine algorithm (SCA) is a novel optimization algorithm, which was proposed by Mirjalili in 2016 [38]. It summarizes and absorbs the iterative strategy of part of the swarm intelligent optimization algorithm. It takes a set containing a specific number of random solutions as the initial solution set of the algorithm, repeatedly evaluates the fitness of the solution through the objective function and randomly follows a specific update strategy to iterate the solution set, and finally obtains the optimal solution or a satisfactory solution that meets the fitness requirements [39]. Like most swarm intelligence optimization algorithms, SCA relies on an iterative strategy to achieve a random search in the solution space, and cannot guarantee that the optimal solution is found in one operation. But when the initial value setting and the number of iterations setting are large enough, the ability of obtaining the optimal solution is greatly improved.

SCA summarizes and deconstructs the iteration strategies of the previous intelligent optimization algorithms into two threads. First, in the global search thread, it applies large random fluctuations to the solution in the current solution set to search for unknown regions in the solution space. Second, in the local development thread, it applies a weak random disturbance to the solution set to fully search the neighborhood of the current solution. The periodic volatility of the trigonometric functions is used to construct an iterative equation that realizes the two thread functions of global search and local development. The succinct update iterative equation is used to impose disturbance and update the solution set [40]–[42].

In addition, other evolutionary computation algorithms, such as monarch butterfly optimization (MBO) [43], earthworm optimization algorithm (EWA) [44], elephant herding optimization (EHO) [45], moth search algorithm (MSA) [46] and rhino herd (RH), were all proposed for global optimization, inspired from the biological evolution and behaviors. And they have been also widely used in the engineering problems [47], [48]. The good performances have been shown in the actual engineering applications.

To summarize, every random optimization algorithm has advantages and disadvantages. According to the theory of "*No Free Lunch*", there is no optimization algorithm that is available for the whole optimization problems effectively. The capacity of the random optimization algorithm algorithms is decided by their ability of global search and local search to search solutions.

Since the original SCA was proposed, many researchers have begun to work on the improvement of SCA, in order to further improve the performance of the algorithm. Nenavath and Jatoth proposed a hybrid algorithm, called SCA-DE [49], which introduced the differential evolution (DE) into SCA. A combination algorithm, called GWO-SCA, was proposed by N Singh and SB Singh [50]. It hybridized the Grey Wolf Optimizer (GWO) and Sine Cosine Algorithm (SCA). Numerical results proved that the proposed algorithm can solve benchmark functions and engineering applications effectively. The Improved SCA (ISCA) [51] was proposed, which changed the position searching equation by introducing an inertia weight, in order to avoid falling into local optima. Researchers proposed a novel improved symmetric SCA (SSCA-APS) to improve the algorithm performance [42]. In the proposed algorithm, adaptive probability selection was proposed.

However, due to the existence of the absolute value item in the traditional SCA position updating path, the traditional SCA in engineering applications, especially the highly nonlinear and highly ill-posed problems, is easy to fall into the premature of algorithm and cannot find the global optimum well. The searching path cannot constrain the search direction of the algorithm well. At the same time, as the dimensionality increases, the convergence speed of traditional SCA will decrease.

As it is all known, in the existing SCA improvement strategies, linear search paths and empirical factor haven't been used to improve performance. In this article, the linear search path and empirical parameter are used to improve the performance of traditional SCA, which has better global search capability and can obtain the more stable solution, effectively avoiding falling into the local optimal. In the testing of 23 benchmark functions and engineering design testing, it is proved that the proposed MSCA obtains the better performance. At the same time, compared with traditional SCA, MSCA has a better convergence rate. In addition, the highly ill-posed inverse scattering problem in physics is used as an optimization verification for the first time. It is proved that the proposed algorithm can solve highly ill-conditioned problems compared to traditional algorithms well.

The chapters of the article are arranged as follows. Part II describes the traditional sine cosine algorithm. Part III explains the modified algorithm, including the search path and convergence factor in detail. Part IV discusses the numerical results of the comparison with other algorithm, including benchmark function tests and actual engineering problem. Part V concludes with contributions and the outlook of research.

## II. TRADITIONAL SINE COSINE ALGORITHM

For the traditional sine cosine algorithm, it was the first time that adaptive sine and cosine mathematics parameters were introduced into the position updating formula. The formula of position updating is shown as follows (more details of the SCA are shown in the reference [38]):

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times \left| r_3 P_i^t - X_i^t \right|, & r_4 < 0.5 \\ X_i^t + r_1 \times \cos(r_2) \times \left| r_3 P_i^t - X_i^t \right|, & r_4 \geq 0.5 \end{cases} \tag{1}$$

Here, $r_2 \in [0, 2\pi]$, is a random variable. $r_3$ is a random variable. $r_4$ is used to choose different search paths, sine or cosine, according to different random values in Eq. (1). $P_i$ is the objective solution.

$r_1$ is a number which is decreased from m to 0 during the process of iterations to make the algorithmic search process balanced. It is shown as follows:

$$r_1(i) = \mathrm{m} \times (1 - \frac{i}{\mathrm{i_{max}}}) \tag{2}$$

where $i$ is the present iteration, $\mathrm{i_{max}}$ is the total number of iteration, and $\mathrm{m} > 0$.

The pseudo code of SCA is shown as follows:

**Algorithm** Since Cosine Algorithm

1. Initialize a set of search agents (solutions) (X)
2. ***repeat***
3. Evaluate each of the search agents by the objective function
4. Update the best solution obtained so far ($P = X^*$)
5. Update the parameters r1, r2, r3, and r4
6. Update the poison of search agents using Eq. (1)
7. ***until*** ($t$ < maximum number of iterations)
8. Return the best solution obtained so far as the global optimum

In the reference [38], the excellent performance of SCA has been proved and applied on the airfoil design problem successfully.

## III. PROPOSED ALGORITHM

### A. IMPROVED POSITION-UPDATING EQUATION

Although it has been proved that traditional SCA possesses obvious superiority in solving optimization problems, which means it has better convergence rate and higher accuracy. Because of the existence of the absolute value term and the trigonometric function term in the position updating equations, the original SCA search path is nonlinear, it is difficult to limit the algorithm search direction when solving complex optimization problem. The search path does not search towards the global best. Sometimes the result obtained is only a local optimum when solving multi-parameter optimization and highly ill-conditioned problems. which can be found in the part IV. In addition, in the reference [38], the benchmark function test on SCA only uses 30 search agents, and does not analyze the problems of multi-dimensional situations. In fact, in the benchmark function test, the convergence rate of traditional SCA will decrease as the number of dimensions increases.

As it is all known, the search path of the PSO algorithm is a classical linear search path. Considering this problem, this article improves the equation (1) by introducing the linear path into SCA, inspired from particle swarm optimization (PSO).

Compared with the search path of traditional SCA, the new search path retains the random selection of sine and cosine parameters and the dual path characteristics, cancels the absolute value and changes the definition of the convergence factor.

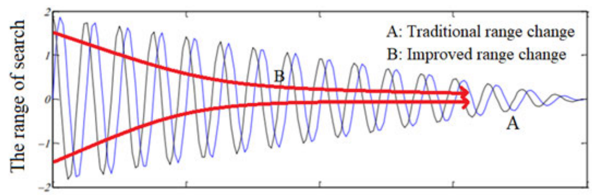


**FIGURE 5.** Traditional and improved range change.

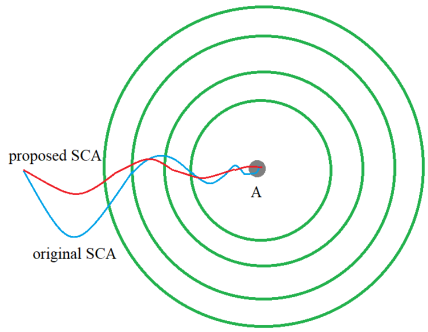


**FIGURE 6.** Comparison of different search paths.

The path search expression is as follows:

$$X_i^{k+1} = \begin{cases} r_1 X_i^k + \mathrm{c}_1 \times \sin(r_2) \times (P_{best-i}^k - X_i^k), & r_3 < 0.5 \\ r_1 X_i^k + \mathrm{c}_1 \times \cos(r_2) \times (P_{best-i}^k - X_i^k), & r_3 > 0.5 \end{cases} \tag{3}$$

where $X_i$ is the location of the present optimal solution, $k$ is the number of present iteration, $P_i$ is the optimal solution.

The parameter $r_1$ is called convergence factor. It is a number which makes the prospection and development balanced.

Through the association of the linear path and the original path, the search will be more rigorous and suitable for the complex optimization problem. The search range will find the global optimum in a more precise range. as illustrated in Figure 5. (The curve A is from [38])

As shown in the Figure 6, the point A is the global optimum which needs to be found. The search path of original SCA requires constant oscillations to find the global optimum. Due to the introduction of the linear path, the degree of oscillation of the proposed improved path is significantly lower than that of the traditional path, and the search speed is improved. Therefore, the global optimum can be found efficiently by the proposed SCA.

In the searching path of original SCA, there is only an optimal value Pi, this is not enough when solving complicated problems. By using the empirical parameter, the knowledge and information about what had previously searched and obtained can be used as the guide for accurate iteration [52].

Therefore, empirical parameter is introduced into the search path to improve the accuracy of search. The empirical parameter had been applied in the field of antenna pattern optimization, and achieved good results [37], [53]. It was a parameter that could optimize the search path and reduce the search error, which was from the previous iteration. The final path search expression is as follows:

$$X_i^{k+1} = \begin{cases} r_1 X_i^k + c_1 \times \sin(r_2) \times (P_{best-i}^k - X_i^k) \\ \quad + c_2 \times \sin(r_2) \times (E_{best}^k - X_i^k), & r_3 < 0.5 \\ r_1 X_i^k + c_1 \times \cos(r_2) \times (P_{best-i}^k - X_i^k) \\ \quad + c_2 \times \cos(r_2) \times (E_{best}^k - X_i^k), & r_3 > 0.5 \end{cases} \tag{4}$$

In the equation (4), $r_1$ is the convergence factor, $r_2 \in [0, \pi/2]$, $r_3$ is a random number, which is used to control the choice of sine or cosine paths. $c_1$ and $c_2$ are constants which can be adjusted according to actual problems. $E$ is empirical parameters which is obtained from previous iteration. $P_i$ is the optimal solution.

Through the introduction of the linear search path, the search of the optimization algorithm will be more efficient. Through the introduction of empirical parameter, under the joint constraints of $E$ and $P$, the accuracy of the search will be improved and the global optimum can be found faster. Through such improvements, the algorithm will be better suited to complex optimization problems.

### B. IMPROVED CONVERGENCE FACTOR

In addition, in this work, the definition of related parameters is adjusted. Meanwhile, it is very important to define a convergence factor which has a better performance in the optimization algorithm.

In the equation (4), $r_1$ is defined as follows:

$$r_1 = r_{max} - (r_{max} - r_{min}) \frac{t}{T} \tag{5}$$

where $r_{max}$ and $r_{min}$, are constants, which can be set according to specific problem.

In equation (4), only one parameter m can be adjusted. Through this modification, the value of the convergence factor will be more flexible. By defining $r_{max}$ and $r_{min}$, it can be adapted to more optimization situations.

At the beginning of the iteration, a larger convergence factor is helpful for improving the global search capability. As the number of iterations increases, the convergence factor can quickly decrease to a smaller value. The choice of the smaller value and the slow decreasing speed are useful to improving the final optimization result of the algorithm.

The pseudo code of the proposed MSCA is shown as follows:

The flowchart of the proposed algorithm is shown as follows:

```
Initialize the population  X_i (i = 1,2,···,n)
Calculate the fitness of each search agent
X* = the best search agent
while (t < maximum number of iterations)
  for each search agent
  Update  r1, r2 and r3
  if (r3 < 0.5)
     Update the position of the current search agent by the Eq. (4)
     of the sine coefficient
  else if (r3 > 0.5)
     Update the position of the current search agent by the Eq. (4)
     of the cosine coefficient
  End for
  Check if any search agent goes beyond the search space and amend it
  Calculate the fitness of each search agent, Update X* if there is a better solution
  t=t+1
end while
return X*
```

**FIGURE 7.** The pseudo code of the proposed MSCA.

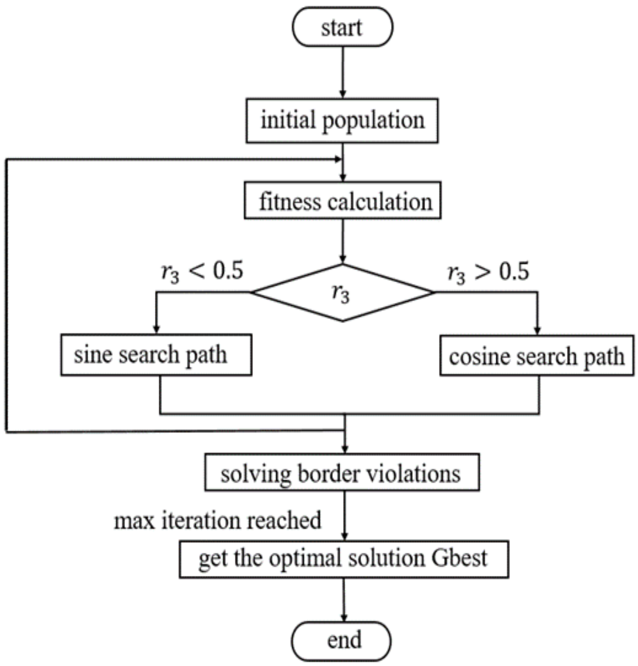


**FIGURE 8.** The flowchart of the MSCA.

## IV. NUMERICAL SIMULATION AND COMPARISON

In this part, two different types of test cases are applied to prove the capacity of the proposed MSCA, namely benchmark function test and actual engineering problem test. However, due to the randomness of the calculation results of the stochastic optimization algorithm, when using the benchmark function test, appropriate, sufficient and different types of benchmark functions should be used. According to the summary of different articles, such as [29], [34] and [38], this article uses 23 commonly used benchmark function tests with different characteristics.

In the performance verification of the stochastic optimization algorithm, except for a certain number and type of benchmark functions verification, several actual engineering verifications are also needed to ensure that the proposed algorithm can also obtain better performance in engineering applications. At the same time, the actual engineering problems are optimization problems with many constraints, which

**TABLE 1.** The Source of Inspiration for the Algorithm Used.

| Algorithm | The source of inspiration |
|---|---|
| PSO | foraging behavior of the birds |
| GA | the evolutionary laws of organisms in nature |
| DE | similar to the GA |
| WOA | predation behavior of the whale |
| GWO | the behavior of grey wolf hunting prey |
| SCA | sine function and cosine function |

are especially suitable for performance comparison among different algorithms.

In the verification of engineering problem, inverse scattering (IS) problem, pressure vessel design problem, tension/ compression spring design and welded beam design problem are used respectively. These four engineering problems belong to the fields of electromagnetic field, structural design and mechanics respectively. They are representative of a certain degree, and the inverse scattering problem is highly ill-conditioned, and is particularly suitable for verification of optimization algorithms. In addition, this is the first time that the inverse scattering problem of the electromagnetic field is used as a verification for the optimization algorithms.

### A. BENCHMARK FUNCTIONS TEST

#### 1) COMPARISON WITH OTHER RANDOM OPTIMIZATION ALGORITHMS

The 23 well-known benchmark test functions, collected from references [29], [34] and [38], are applied. *F1* to *F7* belong to the unimodal benchmark functions. *F8* to *F13* belong to the multimodal benchmark functions and *F14* to *F23* belong to the fixed-dimensions multimodal benchmark function. The selected test functions and the parameters are both summarized in Table 3,4 and 5. Several three-dimensional images of typical functions are shown in the Figure 9.

The optimal value comparison results of the selected algorithms and the proposed MSCA are presented in Table 6. Table 7 concludes the performance comparison between the different algorithms.

''Better'' means that the optimal value of the function obtained is smaller than the algorithm used for performance comparison, ''Equal'' means that the optimal value obtained by the two algorithms is equal, and ''Worst'' means that the optimal value of the function obtained is greater than the algorithm used for performance comparison.

For the purpose of solving the benchmark functions test, the population size is set to 50 and the total number of iterations is set to 1000. The proposed algorithm is compared with PSO, GA, DE, WOA, GWO and SCA.

The source of inspiration for the algorithm used are shown in the Table 1. The parameter settings of the algorithm used for performance comparison are shown in the Table 2.

First of all, from the optimal solution and statistical results, compared with PSO and GA, the performance of the proposed MSCA is significantly better. Although PSO and GA are commonly used optimization algorithms, they have not shown good performance in benchmark function tests. This is due in part to the linear search path, more flexible parameter selection method of MSCA and the introduction of empirical parameters; secondly, compared to DE and GWO, although the algorithm still has advantages, but in terms of fixed-dimensions multimodal benchmark, performance has declined. It can also be seen that compared to WOA, the proposed MSCA obtains the better performance on 14 functions, which is because that the MSCA has a simpler path search method. Although WOA also has high efficiency, its search process is relatively cumbersome.

According to the research results of the researchers, the excellent results of DE are mainly due to its flexible coding method and excellent performance to solve 0-1 problems. The two new natural heuristic optimization algorithms, WOA and GWO, are inspired by the predation laws of the biological world, and they are also of great application value. Finally, through the performance comparison with traditional SCA, only three benchmark functions have test performance worse than traditional SCA. The performance comparison with traditional SCA in detail is shown in the next part. To summarize, the performance of the proposed MSCA algorithm in benchmark function testing is relatively superior.

#### 2) COMPARISON WITH TRADITIONAL SCA IN DIFFERENT DIMENSION

In addition to compare with other optimization algorithms, it is essential to compare the proposed MSCA with original SCA from the two aspects of optimal function value and convergence rate. Table 8 reports the comparison results of the optimal function values between SCA and MSCA.

In order to fairly compare the performance of the proposed MSCA and original SCA, the same algorithm parameters are set for SCA and MSCA. For each benchmark function, the overall size is set to 50. And verification is from four different dimensions, namely dimensions 30, 50, 100 and 500, and the maximum number of iterations in all simulations is 1000.

**TABLE 2.** **The Parameter Settings of the Algorithm Used.**

| Algorithm | parameter settings | N |
|---|---|---|
| PSO | w=0.68<br>c1=0.5<br>c2=0.5 | 50 |
| GA | Pc = 0.8<br>Pm = 0.2<br>gap = 0.9 | 50 |
| DE | Mutation rate: F0 = 0.5<br>Cross probability: CR = 0.9 | 50 |
| WOA | b=1<br>p $\in$(0,1) | 50 |
| GWO | r1$\in$ (0,1)<br>r2$\in$(0,1) | 50 |
| SCA | a = 2 | 50 |
| Proposed Algorithm | $r_{max}$ = 0.9<br>$r_{min}$ = 0.4<br>c1=c2=0.5 | 50 |

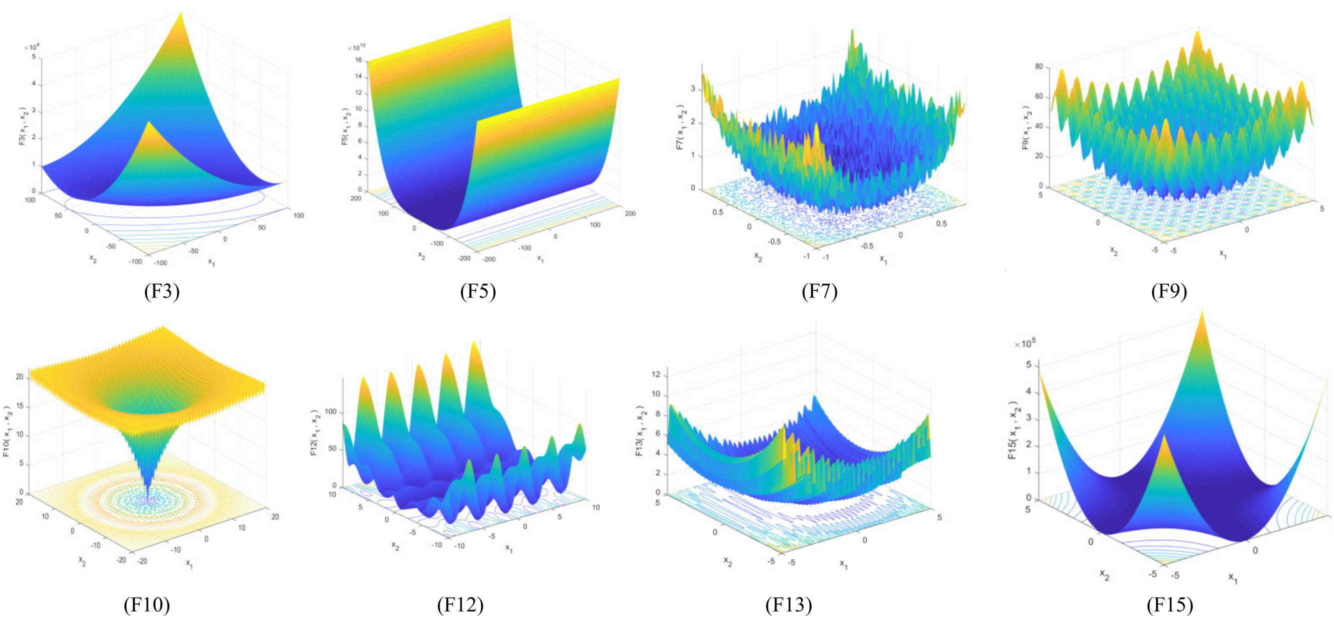


**FIGURE 9.** **Several three-dimensional images of typical functions.**

From the results in Table 8 of benchmark test functions, for D = 30, (*F1-F13, F15-F18*). For the *F8* and *F18* tests, the performance of the traditional SCA is better than the proposed MSCA. The test results of the two algorithms are the same on *F16*. In the remaining benchmark function tests, the performance obtained by MSCA is better than original SCA. The optimal function values obtained by MSCA are closer to the theoretical solution.

For D = 50, for the *F16* and *F18* tests, the performance of the traditional SCA is better than the proposed MSCA. In the remaining benchmark function tests, the performance obtained by MSCA is better than original SCA.

**TABLE 3.** Unimodal Benchmark Functions Used for Validation.

| Function | dim | range | $f_{\min}$ |
|---|---|---|---|
| $F_1(x)=\sum_{i=1}^{n} x_i^2$ | 10 | [-100,100] | 0 |
| $F_2(x)=\sum_{i=1}^{n} \lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$ | 10 | [-10,10] | 0 |
| $F_3(x)=\sum_{i=1}^{n}(\sum_{j-1}^{i} x_j)^2$ | 10 | [-100,100] | 0 |
| $F_4(x)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$ | 10 | [-100,100] | 0 |
| $F_5(x)=\sum_{i=1}^{n-1}\left[100(x_{i+1}+x_i^2)^2+(x_i-1)^2\right]$ | 10 | [-30,30] | 0 |
| $F_6(x)=\sum_{i=1}^{n}([x_i+0.5])^2$ | 10 | [-100,100] | 0 |
| $F_7(x)=\sum_{i=1}^{n} ix_i^4+random[0,1)$ | 10 | [-1.28,1.28] | 0 |

**TABLE 4.** Multimodal Benchmark Functions Used for Validation.

| Function | dim | range | $f_{\min}$ |
|---|---|---|---|
| $F_8(x)=\sum_{i=1}^{n}-x_i\sin(\sqrt{\lvert x_i\rvert})$ | 10 | [-500,500] | $-418.9829\times5$ |
| $F_9(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right]$ | 10 | [-5.12,5.12] | 0 |
| $F_{10}(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$ | 10 | [-32,32] | 0 |
| $F_{11}(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$ | 10 | [-600,600] | 0 |
| $F_{12}(x)=\frac{\pi}{n}\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)[1+10\sin^2(\pi y_{i+1})]+(y_n-1)^2\}+\sum_{i=1}^{n}u(x_i,10,100,4)$ | 10 | [-50,50] | 0 |
| $y_i=1+\frac{x_i+1}{4}\quad u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m, x_i>a\\ 0,-a<x_i<a\\ k(x_i-a)^m, x_i<-a\end{cases}$ | | | |
| $F_{13}(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{n}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+(x_n-1)^2[1+\sin^2(2\pi x_n)]\}+\sum_{i=1}^{n}u(x_i,5,100,4)$ | 10 | [-50,50] | 0 |

For D = 100, for the *F16* and *F18* tests, the performance of the traditional SCA is better than the proposed MSCA. In the remaining benchmark function tests, the performance obtained by SCA is worse than MSCA.

For D = 500, for the *F17* and *F18* tests, the performance of the traditional SCA is better than the proposed MSCA. The test results of the two algorithms are the same on *F16*. In the remaining benchmark function tests, the performance obtained by SCA is worse than MSCA.

To summarize, *F16*, *F17* and *F18* are fixed-dimensions multimodal benchmark functions. In the tests of different dimensions, the performance of traditional SCA is better than the proposed MSCA. But in other benchmark functions, MSCA has achieved better performance.

For further illustration, the convergence speed of the SCA and MSCA on eight test functions (*F3, F5, F7, F9, F10, F12, F13 and F15*) are shown in Figure 10.

From Figure 10, It can be concluded that, compared with the original SCA, the proposed MSCA can obtain a better convergence speed. This also proves that the original SCA does have the drawback of decreasing convergence rate as the dimensionality increases (as introduced in the part III. A). By introducing the linear search path and empirical parameter, the convergence efficiency of original SCA has been improved.

From Tab. 8 and Fig 9, it can be concluded that the proposed MSCA can obtain a better function optimal value and converge rate. This can also illustrate the advantages of the modified linear position searching equation and empirical parameter which were introduced.

### B. SEVERAL ENGINEERING PROBLEMS

#### 1) INVERSE SCATTERING PROBLEM

The theory of inverse scattering has been widely used in many engineering fields, such as medical imaging, ground-penetrating radar and others. Inverse scattering belongs to the inverse problem. However, due to the nonlinearity and highly ill-conditioned nature of inverse scattering problem, it is easy to obtain the local optimum by using traditional methods.

the Lippman-Schwinger equation is as follows:

$$E(r)=E_{inc}(r)+i\omega\mu_0\int_D G(r,r')J(r')dr \tag{6}$$

where $J(r')=-i\omega\varepsilon_0[\varepsilon_r(r')-1]E(r')$.

**TABLE 5. Fixed-Dimensions Multimodal Benchmark Function Used for Validation.**

| Function | dim | range | $f_{\min}$ |
|---|---|---|---|
| $F_{14}(x)=(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6})^{-1}$ | 2 | [-65.536,65.536] | 1 |
| $F_{15}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}\right]^2$ | 4 | [-5,5] | 0.00030 |
| $F_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$ | 2 | [-5,5] | −1.0316 |
| $F_{17}(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$ | 2 | [-15,15] | 0.398 |
| $F_{18}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times[30+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$ | 2 | [-2,2] | 3.0000 |
| $F_{19}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})^2)$ | 3 | [0,1] | −3.86 |
| $F_{20}(x)=-\sum_{i=1}^{4}c_i\exp(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})^2)$ | 6 | [0,1] | −3.32 |
| $F_{21}(x)=-\sum_{i=1}^{5}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$ | 4 | [0,10] | −10.1532 |
| $F_{22}(x)=-\sum_{i=1}^{7}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$ | 4 | [0,10] | −10.4028 |
| $F_{23}(x)=-\sum_{i=1}^{10}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$ | 4 | [0,10] | −10.5363 |

**TABLE 6. The Optimal Solution of the Benchmark Function.**

| F | PSO | GA | DE | WOA | GWO | SCA | Proposed algorithm |
|---|---|---|---|---|---|---|---|
| *F1* | 1.3913E-04 | 2.0561E-08 | 5.6793E-23 | 8.1819E-178 | 7.8393E-147 | 2.1749E-32 | **4.3280E-154** |
| *F2* | 0.0029 | 2.0942 | 3.5840E-12 | 5.4076E-119 | 5.2849E-81 | 9.1887E-21 | **5.5052E-73** |
| *F3* | 2.2345 | 74.0978 | 2.6863E-12 | 1.7304 | 7.1560-67 | 1.8916E-15 | **6.3984E-141** |
| *F4* | 3.4385 | 13.2390 | 2.1594E-08 | 2.6777 | 7.2759E-48 | 2.3561E-10 | **3.3269E-36** |
| *F5* | 8.3497E+02 | 20.9672 | 0.5759 | 5.0475 | 6.232 | 6.3353 | **0.0000** |
| *F6* | 2.4025E-07 | 3.1117E-08 | 3.1923E-12 | 1.0655E-06 | 3.7257E-07 | 0.3425 | **0.0000** |
| *F7* | 0.0459 | 0.2060 | 0.0128 | 4.2038E-05 | 2.3381E-04 | 0.0039 | **7.4453E-07** |
| *F8* | -2.7685E+03 | -867.9147 | -4.1898E+03 | -3062.1843 | -2640.5536 | -2447.3544 | **-4.1898E+03** |
| *F9* | 24.8739 | 26.8638 | 24.3654 | 0.0000 | 0.0000 | 0.0000 | **0.0000** |
| *F10* | 3.6305E-05 | 19.8723 | 1.0563E-11 | 4.4409E-15 | 4.4409E-15 | 4.4409E-15 | **8.8818E-16** |
| *F11* | 0.0541 | 1.0504 | 0.4794 | 0.0000 | 0.0000 | 0.0000 | **0.0000** |
| *F12* | 1.1038E-05 | 0.0026 | 5.6473E-24 | 1.3546e-05 | 2.0347E-07 | 0.0442 | **4.7116E-32** |
| *F13* | 4.7847 | 10.7892 | 2.0775E-23 | 2.2492e-05 | 6.6773E-07 | 0.1841 | **1.3498E-32** |
| *F14* | 10.7632 | 7.8740 | 0.9980 | 0.9980 | 2.9821 | 0.9980 | **0.9980** |
| *F15* | 0.0083 | 7.6363E-04 | 3.0749E-04 | 7.7468E-04 | 3.0749E-04 | 5.4538E-04 | **3.1036E-04** |
| *F16* | -1.0316 | -0.9999 | -1.0316 | -1.0316 | -1.0316 | -1.0316 | **-1.0316** |
| *F17* | 0.3979 | 0.3979 | 0.3979 | 0.3979 | 0.3979 | 0.39871 | **0.3979** |
| *F18* | 3.0000 | 3.000 | 3.0000 | 3.0000 | 3.0000 | 3.0000 | **3.0007** |
| *F19* | -3.8628 | -1.4856 | -3.8628 | -3.8610 | -3.8628 | -3.8541 | **-3.8531** |
| *F20* | -3.3220 | -1.0000 | -3.3220 | -3.3220 | -3.3220 | -3.0012 | **-3.2645** |
| *F21* | -2.6829 | -1.0000 | -10.1532 | -10.1531 | -10.1531 | -0.49729 | **-10.1532** |
| *F22* | -10.4029 | -1.0000 | -10.4029 | -10.4029 | -10.4026 | -4.4104 | **-2.7519** |
| *F23* | -2.8711 | -1.0000 | -10.5364 | -10.4029 | -10.5362 | -5.6398 | **-10.5364** |

**TABLE 7.** **The Statistics of Comparison for the Performance With Different Algorithms.**

| Algorithm | Better | Equal | Worst |
|---|---|---|---|
| Proposed Algorithm versus PSO | 17 | 2 | 4 |
| Proposed Algorithm versus GA | 22 | 1 | 1 |
| Proposed Algorithm versus DE | 12 | 6 | 5 |
| Proposed Algorithm versus WOA | 14 | 3 | 6 |
| Proposed Algorithm versus GWO | 12 | 7 | 4 |
| Proposed Algorithm versus SCA | 16 | 4 | 3 |

**TABLE 8.** **Comparison With Traditional SCA in Different Dimensions.**

| F | D=30 | | D=50 | | D=100 | | D=500 | |
|---|---|---|---|---|---|---|---|---|
| | SCA | Proposed algorithm | SCA | Proposed algorithm | SCA | Proposed algorithm | SCA | Proposed algorithm |
| *F1* | 8.4046E-05 | **1.2933E-73** | 0.22191 | **1.8139E-80** | 1885.5243 | **1.9094E-60** | 1.3811E+05 | **6.9985E-30** |
| *F2* | 1.1124E-06 | **1.4234E-71** | 0.0073 | **1.5440E-51** | 0.27042 | **9.9180E-54** | 125.9699 | **2.0270E-24** |
| *F3* | 1.9548E+03 | **2.3605E-91** | 9109.6261 | **5.7509E-92** | 2.3568E+05 | **7.7030E-50** | 5.3577E+06 | **1.0663E-22** |
| *F4* | 26.7163 | **5.4836E-08** | 51.5433 | **4.7829E-06** | 82.7196 | **5.3759E-06** | 99.0895 | **0.4715** |
| *F5* | 33.5098 | **0.0000** | 3.1184E+05 | **48.4953** | 8.4868E+07 | **9.9436E-18** | 1.2176E+09 | **2.0734E-18** |
| *F6* | 4.5926 | **0.0000** | 9.4791 | **0.0000** | 2.1335E+03 | **0.0000** | 2.0473E+05 | **0.0000** |
| *F7* | 0.023733 | **1.0012E-04** | 0.079571 | **2.3172E-05** | 32.3317 | **1.4162E-04** | 8504.9833 | **7.0122E-05** |
| *F8* | -3881.3488 | **-5.4177E+03** | -5007.1743 | **-2.0949E+04** | -7049.4654 | **-4.1898E+04** | -1.7662E+04 | **-2.0949E+05** |
| *F9* | 9.0125 | **0.0000** | 119.8821 | **0.0000** | 400.1272 | **0.0000** | 816.3688 | **497.4795** |
| *F10* | 0.0025 | **4.4409E-16** | 20.3685 | **8.8818E-16** | 20.5735 | **4.4409E-15** | 20.7361 | **4.4409E-15** |
| *F11* | 0.0072 | **0.0000** | 0.041677 | **0.0000** | 58.9534 | **0.0000** | 227.7139 | **0.0000** |
| *F12* | 0.3732 | **1.5705E-32** | 3.9337E+05 | **9.4233E-33** | 2.4297E+08 | **4.7116E-33** | 3.7681E+09 | **9.4233E-34** |
| *F13* | 2.3252 | **1.3498E-32** | 11.4874 | **1.3498E-32** | 2.5002E+08 | **1.3498E-32** | 7.0251E+09 | **1.3498E-32** |
| *F15* | 5.7174E-04 | **3.4147E-04** | 8.1117E-04 | **3.1047E-04** | 6.3321E-04 | **3.1743E-04** | 4.3516E-04 | **3.0976E-04** |
| *F16* | -1.0316 | **-1.0316** | -1.0316 | **-1.0315** | -1.0316 | **-1.0315** | -1.0316 | **-1.0316** |
| *F17* | 0.3987 | **0.3979** | 0.39921 | **0.3980** | 0.39869 | **0.3979** | 0.3980 | **0.3984** |
| *F18* | 3.0000 | **3.0163** | 3.0000 | **3.0065** | 3.0000 | **3.0146** | 3.0000 | **3.0074** |

In the equation (6), $\varepsilon_r$ is unknown that needs to be solved. Other physical quantities are known.

The model of inverse scattering problem is shown in Figure 11. The data of scattering field is obtained by several receiving antennas, and then the permittivity distribution of the scatterer can be calculated by the data of scattering field.

Consider a problem of two-dimensional inverse scattering. The scatterer is a square object, which has uniform permittivity distribution. The conductor center point of coordinate is located in (0, 0). In order to measure the performance of the proposed algorithm in different dimensions, the scatterer is divided into small pieces of 3*3, 4*4 and 6*6

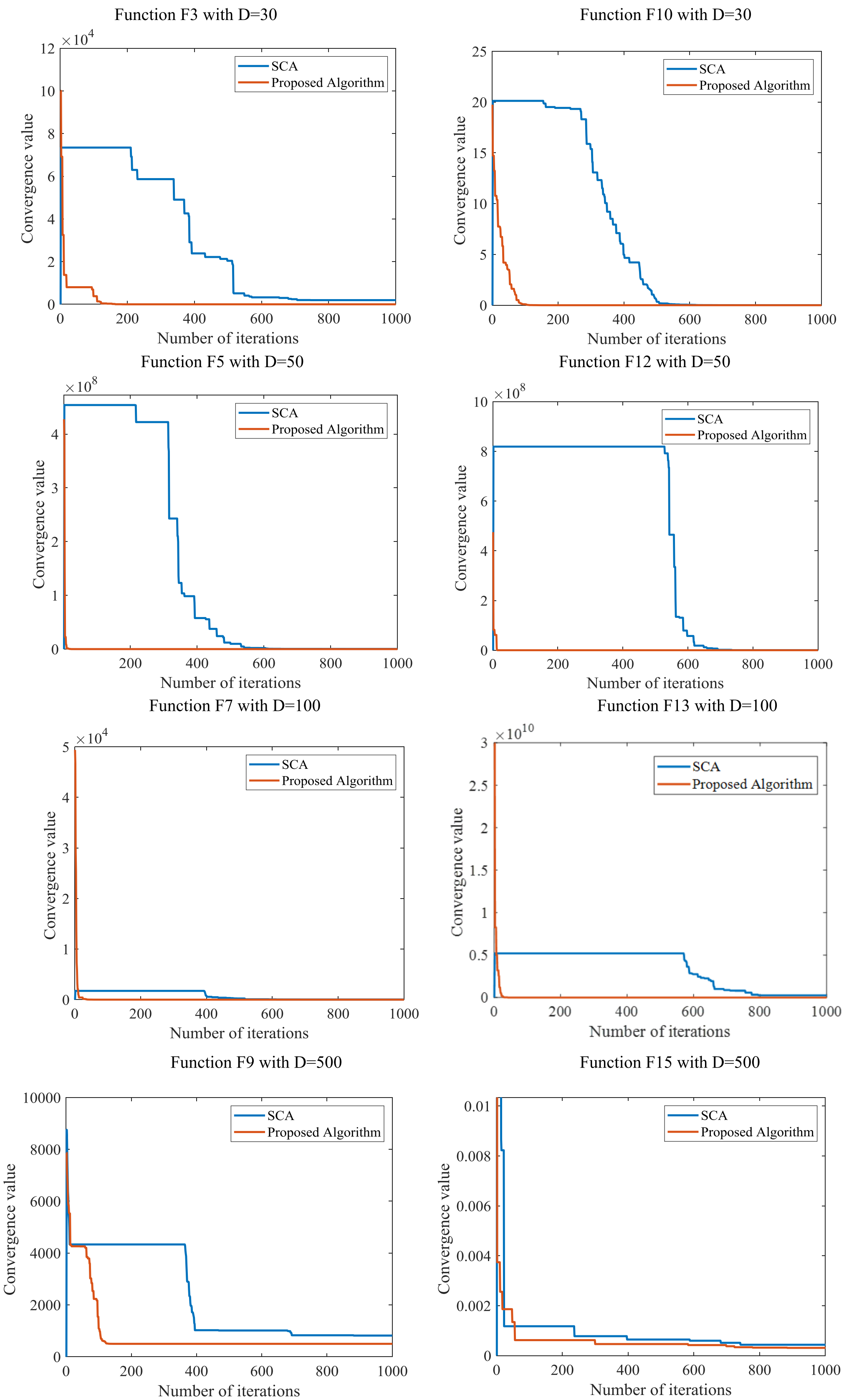


**FIGURE 10.** **The convergence value of SCA and proposed algorithm for typical functions.**

**TABLE 9.** The Results of Optimal Solution and Optimal Function Values.

| Dim | algorithm | max | min | $f_{\min}$ | MSE |
|---|---|---|---|---|---|
| 9 | DE | 3.0647 | 2.9429 | $2.7652\times 10^{-8}$ | $9.9790\times 10^{-4}$ |
| | GA | 4.5347 | 1.0098 | 0.0185 | 1.3434 |
| | PSO | 4.7115 | 1.4601 | 0.0040 | 0.9110 |
| | SCA | 4.3473 | 2.0740 | 0.0252 | 0.4827 |
| | **MSCA** | **3.0138** | **2.9880** | $\mathbf{7.3203\times 10^{-5}}$ | $\mathbf{8.5876\times10^{-5}}$ |
| 16 | DE | 4.6879 | 1.5817 | $4.6206\times 10^{-7}$ | 0.5671 |
| | GA | 4.3230 | 1.0000 | 0.0381 | 0.9541 |
| | PSO | 4.5014 | 1.3863 | 0.0013 | 0.6605 |
| | SCA | 4.7315 | 1.4890 | 0.0587 | 1.1980 |
| | **MSCA** | **3.0000** | **3.0000** | $\mathbf{8.9913\times 10^{-7}}$ | $\mathbf{5.5780\times10^{-10}}$ |
| 36 | DE | 4.3075 | 1.2902 | $1.3355\times 10^{-6}$ | 0.5351 |
| | GA | 5.0000 | 1.0000 | 0.0533 | 3.1065 |
| | PSO | 4.8897 | 1.1209 | 0.0548 | 1.0911 |
| | SCA | 4.9900 | 1.0000 | 0.1520 | 1.6690 |
| | **MSCA** | **3.0009** | **2.9988** | $\mathbf{2.6222\times 10^{-5}}$ | $\mathbf{2.1421\times10^{-7}}$ |

**TABLE 10.** The Experimental Results of MSCA and Other Algorithms for Pressure Vessel Design Problem.

| Algorithm | Optimal values for variables | | | | Optimal cost |
|---|---|---|---|---|---|
| | *Ts* | *Th* | *R* | *L* | |
| WOA (Mirjalili & Lewis, 2016) | 0.8125 | 0.4375 | 42.0982699 | 176.638998 | 6059.7410 |
| MVO (Mirjalili et al., 2016) | 0.8125 | 0.4375 | 42.090738 | 176.73869 | 6060.8066 |
| CPSO (He & Wang, 2007) | 0.8125 | 0.4375 | 42.091266 | 176.7465 | 6061.0777 |
| GSA (Rashedi et al., 2009) | 1.125 | 0.625 | 55.9886598 | 84.4542025 | 8538.8359 |
| GA (Coello & Mezura-Montes, 2002) | 0.8125 | 0.4375 | 42.097398 | 176.65405 | 6059.9463 |
| ES (Mezura-Montes & Coello, 2008) | 0.8125 | 0.4375 | 42.098087 | 176.640518 | 6059.7456 |
| ACO (Kaveh & Talatahari, 2010) | 0.8125 | 0.4375 | 42.098353 | 176.637751 | 6059.7258 |
| CGDA (Baykasoğlu, 2012) | 0.8125 | 0.4375 | 42.0975 | 176.6484 | 6059.8391 |
| CSA (Gandomi et al., 2013) | 0.8125 | 0.4375 | 42.0984 | 176.6366 | 6059.7140 |
| BA (Gandomi et al., 2013) | 0.8125 | 0.4375 | 42.0984 | 176.6366 | 6059.7143 |
| MFO (Mirjalili, 2015) | 0.8125 | 0.4375 | 42.098445 | 176.636596 | 6059.7143 |
| HPSODE (Liu et al., 2010) | 0.8125 | 0.4375 | 42.098446 | 176.636596 | 6059.714335 |
| CDE (Hang et al., 2007) | 0.8125 | 0.4375 | 42.0984 | 176.6376 | 6059.7340 |
| UABC (Brajevic & Tuba, 2013) | 0.8125 | 0.4375 | 42.098446 | 176.636596 | 6059.714335 |
| AFA (Baykasoğlu & Ozsoydan, 2015) | 0.8125 | 0.4375 | 42.0984 | 176.6366 | 6059.7143 |
| CSA (Askarzadeh, 2016) | 0.8125 | 0.4375 | 42.0984 | 176.6366 | 6059.7144 |
| TEO (Kaveh & Dadras, 2017) | 0.8125 | 0.4375 | 42.0984 | 176.6366 | 6059.71 |
| GWO (Mirjalili et al., 2014) | 0.8125 | 0.4375 | 42.0989181 | 176.758731 | 6051.5639 |
| EEGWO (Long et al., 2018) | 13.09291 | 6.792196 | 42.09758 | 176.6495 | 6059.8704 |
| SCA (Mirjalili, 2016) | 0.817577 | 0.417932 | 41.74939 | 183.5727 | 6137.3724 |
| ISCA(Wen Long et al, 2018) | 12.96419 | 7.150134 | 42.09829 | 176.6392 | 6059.7489 |
| ROL-GWO(Wen Long et al, 2019) | 12.73387 | 6.781898 | 42.09825 | 176.6397 | 6059.7528 |
| **MSCA (this work)** | **0.8125** | **0.4375** | **42.0984** | **175.0000** | **6011.400** |

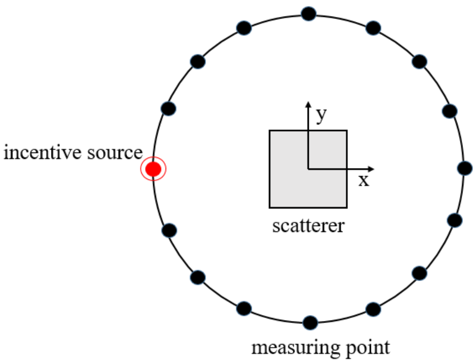


**FIGURE 11.** The model of scattered field.

(Dimension=9,16 and 36). The relative permittivity of the conductor is the parameter waiting to be reconstructed.

An incentive source with the frequency of 3GHz is adopted. Measuring points are evenly arranged on a circle, which the radius is 0.5m, as shown in the Figure 11. The relative permittivity of the scatterer is 3 and the relative permittivity of the surrounding free space is 1.

The object function is defined as follows:

$$f = \frac{\sum_{l=1}^{L}\sum_{m=1}^{M}\left|E_s^l(\vec{r_m}) - E_m^l(\vec{r_m})\right|}{\sum_{l=1}^{L}\sum_{m=1}^{M}\left|E_m^l(\vec{r_m})\right|} \tag{7}$$

where $E_m^1$ is the known solution of the scattered field and $E_s^l$ is the value of the scattered field calculated from during the algorithm iteration process.

Mean squared error (MSE) can be used as an indicator to measure the robustness and error of the calculation results. It is defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}[\varepsilon_i - \varepsilon_r]^2 \tag{8}$$

where $\varepsilon_i$ is optimal solution of permittivity, $\varepsilon_r$ is a known constant which equals 3.

The performance of the proposed MSCA is compared with DE, GA, PSO and SCA. The results of optimal solution maximums, minimums, optimal function values and mean squared error (MSE) are listed in Table 9. The selection of the maximum and minimum values is to measure the stability of the results. The convergence curve of Dim=9,16 and 36 are shown in Figure 12,13 and 14. The number of iterations is set to 1000, and the number of populations is set to 50. (The results in the table retain four significant numbers)

From the results of optimal function values, it can be observed that although the optimal function value of DE is better than the proposed algorithm, the solution obtained by the proposed algorithm MSCA is closer to the theoretical value compared with the other four algorithms, considering from the stability and error of the solution in the case of three different dimensions.

From the results of convergence speed and mean squared error (MSE), it is obvious that PSO and SCA have a faster convergence rate. However, the PSO and SCA are stuck in algorithm premature, the MSE of the proposed MSCA is better than other four algorithms. To summarize, the proposed MSCA can solve the inverse scattering problem well.

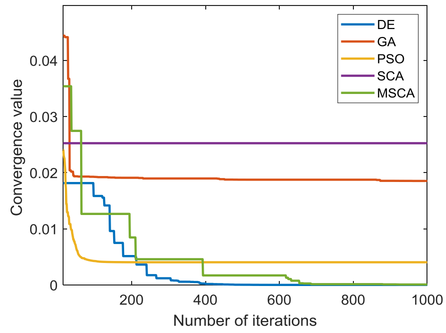


**FIGURE 12.** Convergence curve of D=9.

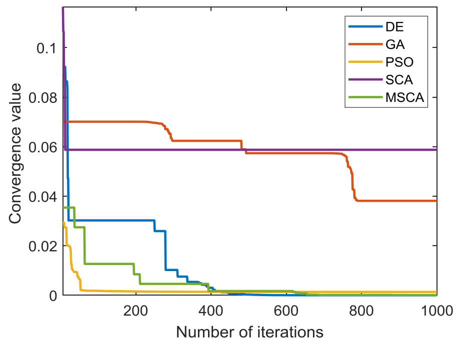


**FIGURE 13.** Convergence curve of D=16.

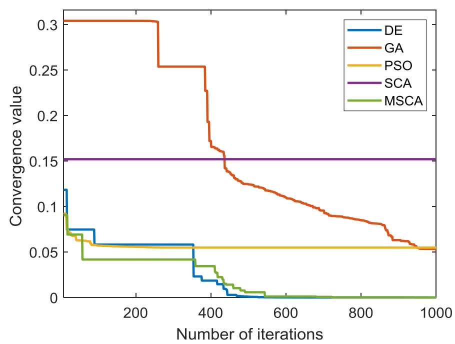


**FIGURE 14.** Convergence curve of D=36.

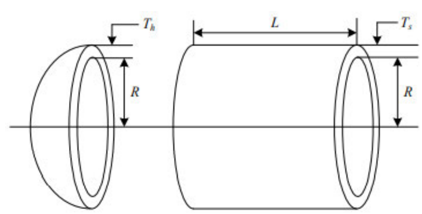


**FIGURE 15.** The model of pressure vessel design problem.

### 2) PRESSURE VESSEL DESIGN PROBLEM

Figure 15 shows the model of this design problem. This problem is a common test in the optimization algorithm. It is involved in many literatures about optimization algorithm.

**TABLE 11. The Experimental Results of MSCA and Other Algorithms for Tension/Compression Spring Design Problem.**

| Algorithm | Optimal values for variables | | | Optimal cost |
|---|---|---|---|---|
| | $d$ | $D$ | $P$ | |
| WOA (Mirjalili & Lewis, 2016) | 0.051207 | 0.345215 | 12.54854 | 0.0126763 |
| MVO (Mirjalili et al., 2016) | 0.05251 | 0.37602 | 10.33513 | 0.012790 |
| CPSO (He & Wang, 2007) | 0.051728 | 0.357644 | 11.244543 | 0.0126747 |
| GSA (Rashedi et al., 2009) | 0.0500 | 0.317312 | 14.22867 | 0.0128739 |
| GA (Coello & Mezura-Montes, 2002) | 0.051989 | 0.363965 | 10.890522 | 0.012681 |
| ES (Mezura-Montes & Coello, 2008) | 0.051643 | 0.3556 | 11.397926 | 0.012698 |
| HS (Mahdavi et al, 2007) | 0.051154 | 0.349871 | 12.076432 | 0.0126706 |
| CGDA (Baykasoğlu, 2012) | 0.0516925 | 0.3568108 | 11.2835059 | 0.012665 |
| BA (Gandomi et al., 2013) | 0.05169 | 0.35673 | 11.2885 | 0.0126652 |
| MFO (Mirjalili, 2015) | 0.051994457 | 0.36410932 | 10.868421862 | 0.0126669 |
| HPSODE (Liu et al., 2010) | 0.0516888101 | 0.3567117001 | 11.289319935 | 0.0126652329 |
| CDE (Hang et al., 2007) | 0.051609 | 0.354714 | 11.410831 | 0.0126702 |
| UABC (Brajevic & Tuba, 2013) | 0.051691 | 0.356769 | 11.285988 | 0.012665 |
| AFA (Baykasoğlu & Ozsoydan, 2015) | 0.051667 | 0.356198 | 11.319561 | 0.0126653 |
| CSA (Askarzadeh, 2016) | 0.051689 | 0.356717 | 11.289012 | 0.0126652 |
| TEO (Kaveh & Dadras, 2017) | 0.051775 | 0.358792 | 11.168390 | 0.012665 |
| GWO (Mirjalili et al., 2014) | 0.05169 | 0.356737 | 11.28885 | 0.012665 |
| EEGWO (Long et al., 2018) | 0.051673 | 0.35634 | 11.3113 | 0.012665 |
| SCA (Mirjalili, 2016) | 0.05078 | 0.334779 | 12.72269 | 0.0127097 |
| ISCA(Wen Long et al, 2018) | 0.0520217 | 0.364768 | 10.8323 | 0.012667 |
| ROL-GWO(Wen Long et al, 2019) | 0.0517234 | 0.357538 | 11.2416 | 0.012666 |
| **MSCA (this work)** | **0.0509** | **0.3111** | **10.1592** | **0.0098** |

It is a four-variable optimization problem: $Ts$, $Th$, $R$ and $L$, as shown in Figure 15. The description of related issues and the optimization results of other algorithms in Table 10 are from references [51,*5.7.1*] and [37].

$Ts$ is the shell thickness. $Th$ is the head thickness. $R$ is the inner radius and $L$ is cylindrical part length.

The mathematical formulation is defined as follows:

$$
\begin{aligned}
&Consider\ X = [\,x_1 \quad x_2 \quad x_3 \quad x_4\,] = [\,T_s \quad T_h \quad R \quad L\,],\\
&Minimize\ f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2\\
&+3.1661x_1^2x_4 + 19.84x_1^2x_3\\
&Subject\ to\ g_1(X) = -x_1 + 0.00193x_3 \leq 0\\
&\qquad g_2(X) = -x_3 + 0.00954x_3 \leq 0\\
&\qquad g_3(X) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^2 + 129600 \leq 0\\
&\qquad g_4(X) = x_4 - 240 \leq 0
\end{aligned}
$$

where $0 \leq x_1, x_2 \leq 99$, $10 \leq x_3, x_4 \leq 200$.

The results of proposed MSCA and other 22 optimization algorithms, which were proposed previously, are listed in the Table 10. As shown in Table 10, the MSCA can obtain better results compared to the other optimization algorithms listed. The proposed MSCA can obtain the smallest optimal cost under the premise of satisfying the constraints of the problem.

### 3) TENSION/COMPRESSION SPRING DESIGN PROBLEM

Figure 16 shows the model of this design problem. This question belongs to the scope of mechanics. This problem has three variables: $D$, $P$, $d$. The definition of variables, the description of related issues and the optimization results of other algorithms in Table 11 are from references [51,*5.7.2*] and [37].

The mathematical definition of it is shown as follows:

$$
\begin{aligned}
&Consider\ X = [\,x_1 \quad x_2 \quad x_3\,] = [\,d \quad D \quad P\,]\\
&Minimize\ f(X) = (x_3 + 2)x_2x_1^2
\end{aligned}
$$

**TABLE 12.** The Experimental Results of MSCA and Other Algorithms for Welded Beam Design Problem.

| Algorithm | Optimal values for variables | | | | Optimal cost |
|---|---|---|---|---|---|
| | *h* | *l* | *t* | *b* | |
| GA (Deb, 2000) | 0.2489 | 6.1730 | 8.1789 | 0.2533 | 2.4331 |
| HS (Lee & Geem, 2005) | 0.2442 | 6.2231 | 8.2915 | 0.2443 | 2.3807 |
| SA (Atiqullah & Rao, 2000) | 0.2471 | 6.1451 | 8.2721 | 0.2495 | 2.4138 |
| SBM (Akhtar et al., 2002) | 0.2407 | 6.4851 | 8.2399 | 0.2497 | 2.4426 |
| SCA (Ray & Liew, 2003) | 0.2444 | 6.2380 | 8.2886 | 0.2446 | 2.3854 |
| IPSO (He et al., 2004) | 0.2444 | 6.2175 | 8.2915 | 0.2444 | 2.3810 |
| NOSA (Liu, 2005) | 0.2444 | 6.2175 | 8.2915 | 0.2444 | 2.3810 |
| DFSA (Hedar & Fukushima, 2006) | 0.2444 | 6.2158 | 8.2939 | 0.2444 | 2.3811 |
| DEDS (Zhang et al., 2008) | 0.2444 | 6.2175 | 8.2915 | 0.2444 | 2.3810 |
| EMEA (Zhang et al., 2009) | 0.2443 | 6.2201 | 8.2940 | 0.2444 | 2.3816 |
| ISA (Gandomi, 2014) | 0.2443 | 6.2199 | 8.2915 | 0.2443 | 2.3812 |
| MTSA (Babalik, et al., 2018) | 0.24415742 | 6.22306595 | 8.29555011 | 0.24440474 | 2.38241101 |
| ISCA (Wen Long,et al., 2018) | 0.24435 | 6.2178 | 8.2919 | 0.24437 | 2.3810 |
| EEGWO (Long et al., 2018) | 0.2444 | 6.2170 | 8.2928 | 0.2444 | 2.3813 |
| ROL-GWO (Wen Long,et al., 2019) | 0.24434 | 6.2188 | 8.2916 | 0.24437 | 2.3811 |
| **MSCA (this work)** | **0.1250** | **0.8130** | **8.5336** | **0.2764** | **1.6948** |

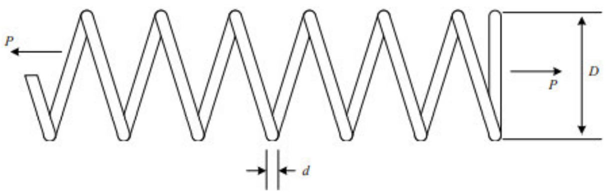


**FIGURE 16.** The model of tension/compression spring design problem/.

$$\text{Subject to } g_1(X) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0$$
$$g_2(X) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - x_1^4)} + \frac{1}{5108 x_1^2} \le 0$$
$$g_3(X) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0$$
$$g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \le 0$$

where $0.05 \le x_1 \le 2, 0.25 \le x_2 \le 1.30, 2.00 \le x_3 \le 15$.

The results of proposed MSCA and other 21 optimization algorithms, which were proposed previously, are listed in the Table 11. As shown in Table 11, the MSCA can obtain better results compared to the other optimization algorithms listed. The proposed MSCA can obtain the smallest optimal cost under the premise of satisfying the constraints of the problem.

#### 4) WELDED BEAM DESIGN PROBLEM

Figure 17 shows the model of this design problem. This problem has four variables: *h, l, t and b*. More details on this design problem are in [45] and the optimization results of other algorithms in Table 12 are from references [51,*5.7.3*] and [37].

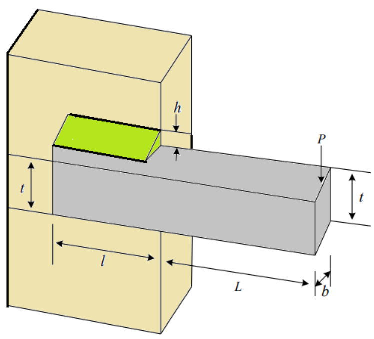


**FIGURE 17.** The model of welded beam design problem.

The results of proposed MSCA and other 15 optimization algorithms, which were proposed previously, are listed in the Table 12. As shown in Table 12, the MSCA can obtain better results compared to the other optimization algorithms listed. The proposed MSCA can obtain the smallest optimal cost under the premise of satisfying the constraints of the problem.

## V. CONCLUSION

In this article, a modified SCA is proposed. The linear search path and empirical parameter have been introduced for the first time to improve the accuracy of search direction for optimization problems with different dimensions and highly ill-posed actual problems.

The numerical results of benchmark function tests for the proposed MSCA are compared with original SCA and other widespread algorithms. From these numerical results, it can be proved that the MSCA has the better performance than the original SCA and other population-based algorithms. Additionally, the results of the MSCA in solving actual problems are compared with several algorithms and obtains better results.

In addition, except the algorithms used in the article, some other intelligence algorithms can also be used to solve the actual engineering problems, such as monarch butterfly optimization (MBO), earthworm optimization algorithm (EWA), elephant herding optimization (EHO), moth search algorithm (MSA) and rhino herd (RH). And the good performance can be obtained.

In future work, continuous learning and inspiration from the algorithms mentioned above is the driving force for improvement. And more applications such as signal processing, image processing, big data analysis, neural network and engineering structure design need to be applied to prove the performance of MSCA and continuously improve existing algorithms.

## ACKNOWLEDGMENT

The authors would like to thank the reviewers and the editors for their laborious work and the constructive comments sincerely.

## REFERENCES

[1] D. Tansui and A. Thammano, "Hybrid nature-inspired optimization algorithm: Hydrozoan and sea turtle foraging algorithms for solving continuous optimization problems," *IEEE Access*, vol. 8, pp. 65780–65800, 2020.

[2] F. Wahid, M. S. Zia, R. N. B. Rais, M. Aamir, U. M. Butt, M. Ali, A. Ahmed, I. A. Khan, and O. Khalid, "An enhanced firefly algorithm using pattern search for solving optimization problems," *IEEE Access*, vol. 8, pp. 148264–148288, 2020.

[3] W. Fu, B. Wang, X. Li, L. Liu, and Y. Wang, "Ascent trajectory optimization for hypersonic vehicle based on improved chicken swarm optimization," *IEEE Access*, vol. 7, pp. 151836–151850, 2019.

[4] S. Zhao, L. Gao, J. Tu, and D. Yu, "A novel modified Tree-Seed algorithm for high-dimensional optimization problems," *Chin. J. Electron.*, vol. 29, no. 2, pp. 337–343, Mar. 2020.

[5] Y. L. Karnavas, "Application of recent nature-inspired meta-heuristic optimisation techniques to small permanent magnet DC motor parameters identification problem," *J. Eng.*, vol. 2020, no. 10, pp. 877–888, Oct. 2020.

[6] K. Choi, D.-H. Jang, S.-I. Kang, J.-H. Lee, T.-K. Chung, and H.-S. Kim, "Hybrid algorithm combing genetic algorithm with evolution strategy for antenna design," *IEEE Trans. Magn.*, vol. 52, no. 3, Mar. 2016, Art. no. 7209004.

[7] Q. Zhao and C. Li, "Two-stage multi-swarm particle swarm optimizer for unconstrained and constrained global optimization," *IEEE Access*, vol. 8, pp. 124905–124927, 2020.

[8] Q. Fan and X. Yan, "Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 219–232, Jan. 2016.

[9] M. W. Guo, J. S. Wang, L. F. Zhu, S. S. Guo, and W. Xie, "An improved grey wolf optimizer based on tracking and seeking modes to solve function optimization problems," *IEEE Access*, vol. 8, pp. 69861–69893, 2020.

[10] X. Wu, S. Zhang, W. Xiao, and Y. Yin, "The exploration/exploitation tradeoff in whale optimization algorithm," *IEEE Access*, vol. 7, pp. 125919–125928, 2019, doi: 10.1109/ACCESS.2019.2938857.

[11] S. Padmanaban, N. Priyadarshi, J. B. Holm-Nielsen, M. S. Bhaskar, F. Azam, A. K. Sharma, and E. Hossain, "A novel modified sine-cosine optimized MPPT algorithm for grid integrated PV system under real operating conditions," *IEEE Access*, vol. 7, pp. 10467–10477, 2019.

[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, Perth, WA, Australia, Nov./Dec. 1995, vol. 4. no. 8, pp. 1942–1948.

[13] J. Kennedy and R. Eberhard, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, 2002, pp. 1942–1948.

[14] N. K. Jain, U. Nangia, and J. Jain, "A review of particle swarm optimization," *J. Inst. Eng.*, vol. 99, no. 4, pp. 1–5, 2018.

[15] M. Gao, X. Fu, G. Dong, and H. Li, "An adaptive mutation multi-particle swarm optimization for traveling salesman problem," in *Proc. 3rd Int. Conf. Mater., Mech. Manuf. Eng.*, 2015, pp. 1–5.

[16] Y. Marinakis, A. Migdalas, and A. Sifaleras, "A hybrid particle swarm optimization–variable neighborhood search algorithm for constrained shortest path problems," *Eur. J. Oper. Res.*, vol. 261, no. 3, pp. 819–834, Sep. 2017.

[17] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.

[18] Y. Cao, H. Zhang, W. Li, M. Zhou, Y. Zhang, and W. A. Chaovalitwongse, "Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 718–731, Aug. 2019.

[19] D. N. Mudaliar and N. K. Modi, "Applying m-mutation operator in genetic algorithm to solve permutation problems," in *Proc. IEEE Int. Conf. Syst., Comput., Autom. Netw. (ICSCAN)*, Mar. 2019, pp. 1–5.

[20] A. Panichella, R. Oliveto, M. D. Penta, and A. De Lucia, "Improving multi-objective test case selection by injecting diversity in genetic algorithms," *IEEE Trans. Softw. Eng.*, vol. 41, no. 4, pp. 358–383, Apr. 2015.

[21] A. Jafari, T. Khalili, E. Babaei, and A. Bidram, "A hybrid optimization technique using exchange market and genetic algorithms," *IEEE Access*, vol. 8, pp. 2417–2427, 2020.

[22] D. Gong, J. Sun, and Z. Miao, "A set-based genetic algorithm for interval many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 47–60, Feb. 2018.

[23] B. Doerr and C. Doerr, "Optimal static and self-adjusting parameter choices for the $(1 + (\lambda, \lambda))$ genetic algorithm," *Algorithmica*, vol. 80, no. 5, pp. 1658–1709, 2018.

[24] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[25] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," in *Advances in Differential Evolution* (Studies in Computational Intelligence), vol. 143, U. K. Chakraborty, Ed. Berlin, Germany: Springer, 2008, doi: 10.1007/978-3-540-68830-3_6.

[26] Y.-L. Li, Z.-H. Zhan, Y.-J. Gong, W.-N. Chen, J. Zhang, and Y. Li, "Differential evolution with an evolution path: A DEEP evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 9, pp. 1798–1810, Sep. 2015.

[27] Z. Zhang, Y. Cai, and D. Zhang, "Solving ordinary differential equations with adaptive differential evolution," *IEEE Access*, vol. 8, pp. 128908–128922, 2020.

[28] Z. Hong, Z.-G. Chen, D. Liu, Z.-H. Zhan, and J. Zhang, "A multi-angle hierarchical differential evolution approach for multimodal optimization problems," *IEEE Access*, vol. 8, pp. 178322–178335, 2020.

[29] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[30] G. I. Sayed, A. Darwish, and A. E. Hassanien, "A new chaotic whale optimization algorithm for features selection," *J. Classification*, vol. 35, no. 2, pp. 300–344, Jul. 2018.

[31] S. H. Hashemi Mehne and S. Mirjalili, "A direct method for solving calculus of variations problems using the whale optimization algorithm," *Evol. Intell.*, vol. 12, no. 4, pp. 677–688, Dec. 2019.

[32] Q. Zhang and L. Liu, "Whale optimization algorithm based on lamarckian learning for global optimization problems," *IEEE Access*, vol. 7, pp. 36642–36666, 2019.

[33] J. Zhang and J. S. Wang, "Improved whale optimization algorithm based on nonlinear adaptive weight and golden sine operator," *IEEE Access*, vol. 8, pp. 77013–77048, 2020.

[34] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[35] C. P. Igiri, Y. Singh, and R. C. Poonia, "A review study of modified swarm intelligence: Particle swarm optimization, firefly, bat and gray wolf optimizer algorithms," in *Recent Advances in Computer Science and Communications* (Formerly: Recent Patents on Computer Science). vol. 13, no. 1. Sharjah, U.A.E.: Bentham Science, 2020, pp. 5–18, doi: 10.2174/2213275912666190101120202.

[36] Q. Tu, X. Chen, and X. Liu, "Hierarchy strengthened grey wolf optimizer for numerical optimization and feature selection," *IEEE Access*, vol. 7, pp. 78012–78028, 2019.

[37] W. Long, J. Jiao, X. Liang, S. Cai, and M. Xu, "A random opposition-based learning grey wolf optimizer," *IEEE Access*, vol. 7, pp. 113810–113825, 2019.

[38] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.

[39] J. Wang, W. Yang, P. Du, and T. Niu, "A novel hybrid forecasting system of wind speed based on a newly developed multi-objective sine cosine algorithm," *Energy Convers. Manage.*, vol. 163, pp. 134–150, May 2018.

[40] Y. Wan, A. Ma, Y. Zhong, X. Hu, and L. Zhang, "Multiobjective hyperspectral feature selection based on discrete sine cosine algorithm," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 5, pp. 3601–3618, May 2020.

[41] M. A. A. Al-qaness, M. A. Elaziz, and A. A. Ewees, "Oil consumption forecasting using optimized adaptive neuro-fuzzy inference system based on sine cosine algorithm," *IEEE Access*, vol. 6, pp. 68394–68402, 2018.

[42] B. Wang, T. Xiang, N. Li, W. He, W. Li, and X. Hei, "A symmetric sine cosine algorithm with adaptive probability selection," *IEEE Access*, vol. 8, pp. 25272–25285, 2020.

[43] G. G. Wang, S. Deb, and Z. Cui, "Monarch butterfly optimization," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 1995–2014, 2019.

[44] A. Khan, N. Mushtaq, S. H. Faraz, O. A. Khan, M. A. Sarwar, and N. Javaid, "Genetic algorithm and earthworm optimization algorithm for energy management in smart grid," in *Proc. 12th Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput. (PGCIC)*, Madrid, Spain. Cham, Switzerland: Springer, 2017, pp. 447–459.

[45] G.-G. Wang, S. Deb, and L. D. S. Coelho, "Elephant herding optimization," in *Proc. 3rd Int. Symp. Comput. Bus. Intell. (ISCBI)*, Dec. 2015, pp. 1–5.

[46] Y.-H. Feng and G.-G. Wang, "Binary moth search algorithm for discounted 0-1 knapsack problem," *IEEE Access*, vol. 6, pp. 10708–10719, 2018.

[47] G.-G. Wang, A. H. Gandomi, A. H. Alavi, and D. Gong, "A comprehensive review of krill herd algorithm: Variants, hybrids and applications," *Artif. Intell. Rev.*, vol. 51, no. 1, pp. 119–148, Jan. 2019.

[48] E. Tuba and Z. Stanimirovic, "Elephant herding optimization algorithm for support vector machine parameters tuning," in *Proc. 9th Int. Conf. Electron., Comput. Artif. Intell. (ECAI)*, Jun. 2017, pp. 1–4.

[49] H. Nenavath and R. K. Jatoth, "Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking," *Appl. Soft Comput.*, vol. 62, pp. 1019–1043, Jan. 2018.

[50] N. Singh and S. B. Singh, "A novel hybrid GWO-SCA approach for optimization problems," *Eng. Sci. Technol., Int. J.*, vol. 20, no. 6, pp. 1586–1601, Dec. 2017.

[51] W. Long, T. Wu, X. Liang, and S. Xu, "Solving high-dimensional global optimization problems using an improved sine cosine algorithm," *Expert Syst. Appl.*, vol. 123, pp. 108–126, Jun. 2019.

[52] J. J. Jamian, M. N. Abdullah, H. Mokhlis, M. W. Mustafa, and A. H. A. Bakar, "Global particle swarm optimization for high dimension numerical functions analysis," *J. Appl. Math.*, vol. 2014, pp. 1–14, Feb. 2014.

[53] G. Lu and Z. Cao, "Radiation pattern synthesis with improved high dimension PSO," in *Proc. Prog. Electromagn. Res. Symp. Fall (PIERS-FALL)*, Singapore, Nov. 2017, pp. 2160–2165.


**MENG WANG** received the M.S. degree in engineering from the North China University of Science and Technology, Tangshan, China, in 2019. He is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering, Communication University of China, Beijing, China.

His current research interests include stochastic optimization algorithm and electromagnetic inverse problem.

**GUIZHEN LU** received the M.S. degree in science from Peking University, Beijing, China, in 1984, and the Ph.D. degree in engineering from Beihang University, Beijing, in 2002.

He is currently a Professor with the School of Information and Communication Engineering, Communication University of China; and the member of the Radio Wave Propagation Committee, Chinese Institute of Electronics, and of the Electromagnetic Compatibility Committee, Chinese Institute of Electronics. His current research interests include mathematical optimization problem, Terahertz technology, and computational electromagnetics.