IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 August 2023, accepted 5 September 2023, date of publication 11 September 2023, date of current version 15 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3313930

RESEARCH ARTICLE

A Multi-Strategy Dung Beetle Optimization Algorithm for Optimizing Constrained Engineering Problems

WANG ZILONG AND SHAO PENG
School of Computer and Information Engineering, Jiangxi Agricultural University, Jiangxi 330045, China
Corresponding author: Shao Peng (pshao@whu.edu.cn)

This work was supported in part by the Jiangxi Provincial Social Science Planning Project under Grant 21GL12, and in part by the Technology Plan Projects of Jiangxi Provincial Education Department under Grant GJJ200424.

ABSTRACT The dung beetle optimization (DBO) algorithm is one of newly excellent swarm intelligent algorithm while its exploration capability is still insufficient. For this, a multi-strategy DBO algorithm (GODBO) by utilizing the optimal value in the current population directed shift and the opposition-based learning (OBL) is proposed. In GODBO, the OBL is used to increase the likelihood of finding a better solution in the early stage of the algorithm so that the algorithm can find the optimal solution faster. Meanwhile, the current optimal value (Gbest) is used to guide the solution to search a new solution later in the algorithm, and the improved algorithm will be searched near a better solution at the later stage to get a better solution. Therefore, both are used to enhance exploration capabilities. 29 famous mathematical benchmark functions as test objects are applied to evaluate the abilities of the GODBO algorithm, and the experimental results demonstrate that GODBO performs better in the light of convergence speed and convergence accuracy in comparison with other competitors. Furthermore, two constrained engineering optimization problems are employed in GODBO to validate the effectiveness to solve practice problems, and the experiment results show that it can make tools to tackling them.

INDEX TERMS Dung beetle optimization algorithm, opposition-based learning, Gbest, engineering optimization, CEC2017.

I. INTRODUCTION

All the time, optimization issues have always been the central point and difficulty of research, but complex optimization issues are hard to be used on top of some traditional mathematical problems. Therefore, in recent times, swarm intelligence (SI) algorithms have been extensive used due to their easy implementation and simple framework. The design of swarm intelligence algorithms is a study of the biological swarm behavior phenomena in nature. For instance, the particle swarm algorithm (PSO) with group collaboration as inspiration [1], the harris hawk algorithm(HHO) that imitates harris hawk predation [2], the dragonfly algorithm(DA) based on the daily activities of dragonflies [3], and the dung beetle optimization algorithm(DBO) based on the ball rolling, egg laying, foraging and stealing behavior of dung beetles [4] and so on.

Unfortunately, SI algorithms also have their drawbacks. Most SI algorithms have a series of demerits such as slower convergence speed and lower convergence accuracy at the early stage, which makes it difficult to be well applied in other fields. Based on these problems, scholars have proposed targeted improvements to the SI algorithm to propose a number of optimization algorithms [5], [6] so that algorithm performance can be effectively enhanced.

Nowadays, after a long period of research on SI algorithms, the capability of a large number of SI algorithms has been greatly improved, thus making the algorithms of SI play an important role in practical problems. The PSO and artificial bee swarm algorithms (ABC), in turn, have been proposed as earlier algorithms, and scholars have applied them to address a large number of practical problems such as hybrid

The associate editor coordinating the review of this manuscript and approving it for publication was Gustavo Olague.

multi-objective algorithm for blocking pipeline scheduling problem [7], software workload estimation problem on the basis of PSO [8], application of PSO in water-cooled unit control [9] and so on.

The dung beetle algorithm in this paper, however, as a newly proposed algorithm, suffers from the problems that all SI algorithms have in the early stages of establishment, and its convergence speed and accuracy still have a possibility for improvement. In view of some problems existing in dung beetle algorithm, some scholars have made some improvements to it and applied it in the following aspects, such as dung beetle optimization algorithm based on adaptive t-distribution [10], air-quality prediction based on the arima-cnn-lstm combination model optimized by dung beetle optimizer [11] and so on. However, compared with other algorithms, the dung beetle algorithm still has some shortcomings. For this, inspired by the GBest in current population enhancing the performance of ABC algorithm [12] and the PSO algorithm based on opposition-based learning [13], a multi-strategy DBO (GODBO) algorithm is proposed. These two strategies are used in the early and later stages of the DBO algorithm, respectively to enhance the convergence accuracy and accelerate the convergence speed.

The organized structure is listed as follows. Section II of the paper first introduces the original dung beetle algorithm with a detailed explanation of the process and some algorithm will be used at the later stage. Then in Section III the principle of optimal value guided and opposition-based learning and how to combine these strategies with the dung beetle algorithm are explained. Section IV is an experimental comparison of the improved dung beetle algorithm with other algorithms in various aspects, so as to explore the strengths and weakness of the improved dung beetle algorithm. Section V is an engineering application of the improved dung beetle algorithm, clearly demonstrating that the dung beetle algorithm can also work well in practical applications. Finally, the article gives a discussion and analysis in Section VI and conclusion in Section VII.

II. RELATED WORK

A. DUNG BEETLE OPTIMIZATION ALGORITHM

In nature, dung beetles roll dung into balls and then roll them up so that they move as fast as possible as a way to prevent competition from other dung beetles. This paper calls such dung beetles rolling balls. The dung beetle will navigate the direction of movement by the light in the environment, making the dung ball move in as straight a line as possible. The updated description of the dung beetle's position is defined when pushing the dung ball as the following Eq.(1).

$$X_i^{n+1} = X_i^n + a \cdot b \cdot X_i^{n-1} + B \cdot \Delta x$$
$$\Delta x = |X_i^n - X^w| \tag{1}$$

where X_i^n represents the position information of the i-th generation of dung beetles after the n-th iteration, and b is a random number of [0, 0.2] which represents the defect coefficient. B represents a constant taking values from [1, 0], and the value of a is assigned -1 or 1 which represents the various effects in nature. It is used to simulate the intensity variation of light, and X^w is the global worst position.

Furthermore, dung beetles are unable to discern the direction of their movement when there is no light in the surrounding environment or when the road is rough, in which case they will climb up the dung ball and dance, which is used to determine the direction of their next movement. The formula for the dung beetle's dance to update its position is presented as the following Eq.(2).

$$X_i^{n+1} = X_i^n + \tan\beta |X_i^n - X_i^{n-1}|$$
$$0 \le \beta \le \pi \tag{2}$$

where $|X_i^n - X_i^{n-1}|$ is the distance between the i-th generation at the n-th iteration position and the (n-1)-th iteration position.

In nature, dung beetles roll their dung balls to a safe place to hide them as a place to reproduce their offspring. Meanwhile, it is crucial for dung beetles to select a suitable location in a pile of dung balls for oviposition. Therefore, the formula for determining the boundaries of spawning is defined as the following Eq.(3).

$$LB_1 = max(X^r \cdot (1-T), LB)$$
$$UB_1 = min(X^r \cdot (1+T), UB) \tag{3}$$

where X^r is the current local optimal site, and LB_1 and UB_1 are used to determine the size of the spawning area, where LB and UB are the upper and lower bounds of the optimization issues. And T=1-n/N_{max} where N_{max} is the maximum quantity of iterations.

After the boundaries of the dung beetle spawning area are determined, females will only lay eggs in the area determined by the above formula, and only one egg will be laid per generation. When the spawning area changes, the female dung beetle can clearly sense the change in the boundary and therefore dynamically adjust her spawning location. The formula for the selection of egg-laying locations in dung beetles is as the following Eq.(4).

$$X_i^{n+1} = X^r + B_1 \cdot (X_i^n - LB_1) + B_2 \cdot (X_i^n - UB_1) \tag{4}$$

where X_i^n represents the site of the i-th generation after the n-th iteration, B_1 and B_2 are two random matrices of $1 \bullet Dim$, and the Dim is the dimensional size of the algorithm.

Dung beetles crawl out of the soil as adults in search of food, and are referred as newborn dung beetles. We set up foraging zones to guide dung beetles in their foraging, simulating dung beetles foraging in their natural environment. The foraging boundaries are determined by the following Eq.(5)

$$LB_2 = \max(X^- \cdot (1-T), LB)$$
$$UB_2 = \min(X^- \cdot (1+T), UB) \tag{5}$$

where X^- is the global optimal position, and LB_2 and UB_2 are used to determine the upper and lower boundaries of the dung beetle's search area for food. T has the same meaning as the boundary selection strategy of dung beetles when laying eggs.

After the dung beetle's foraging area is determined, the dung beetle will determine its foraging location based on the boundaries of the foraging area and thus the location update formula is listed as the following Eq.(6).

$$X_i^{n+1} = X_i^n + K_1 \cdot (X_i^n - LB_2) + K_2 \cdot (X_i^n - UB_2) \quad (6)$$

where X_i^n is the position of the i-th generation of small dung beetles at the n-th iteration, and K_1 is a number obeying gaussian distribution, and K_2 represent a set belonging to [0, 1].

In addition, there are thieves in dung beetle colonies that steal the dung balls of other dung beetles, which are called thieving dung beetles. From the previous equation, it is clear that X^s is the best place to get food. Therefore, it is assumed that X^s is the most suitable place to contend. Therefore, the updated description of the location of the thieves is determined as the following Eq.(7).

$$X_i^{n+1} = X^s + P \cdot f \cdot (|X_i^n - X^r| + |X_i^n - X^-|) \quad (7)$$

where X_i^n is the position of the i-th burglar at the t-th iteration, and f is a stochastic set obeying a normal distribution with size $1 \bullet Dim$, and P is denoted as a constant value

B. HARRIS HAWK OPTIMIZATION

Harris hawk optimization (HHO) [2] is one of swarm intelligence optimization algorithms proposed by Ali Asghar Heidari and his colleagues in 2019 based on harris hawk's hunting processes. The algorithm divided its hunting processes into two parts: global search and local development. The global search was defined as follow.

$$\text{X(t+1)} = \begin{cases} X_{rand}(t) - r_1|X_{rand}(t) - 2r_2X(t)|\text{q} \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) \\ \text{q<0.5} \end{cases} \quad (8)$$

where $X(t+1)$is the position vector of hawks in the next iteration t, X_{rabbit} (t) is the position of rabbit, $X(t)$ is the current position vector of hawks, r_1, r_2, r_3, r_4, and q are random numbers inside (0,1), which are updated in each iteration, LB and UB show the upper and lower bounds of variables, X_{rand} $(t)$is a randomly selected hawk from the current population, and X_m is the average position of the current population of hawks.

The local development was divided into four parts, soft besiege, hard besiege, soft besiege with progressive rapid dives and hard besiege with progressive rapid dives. When the escaping energy $|E| < 1$, the local development was performed. The four-part formulas for local development are presented as follows.

(1) Soft besiege

$$\begin{aligned} X(t+1) &= \Delta X(t) - E|JX_{rabbit}(t) - X(t)| \\ \Delta X(t) &= X_{rabbit}(t) - X(t) \end{aligned} \quad (9)$$

where j is a random number of [1] and [2].

(2) Hard besiege

$$X(t+1) = X_{rabbit}(t) - E|\Delta X(t)| \quad (10)$$

(3) Soft besiege with progressive rapid dives

$$\text{Y} = X_{rabbit}(t) - E|JX_{rabbit}(t) - X(t)| \quad (11)$$

(4) Hard besiege with progressive rapid dives

$$\begin{aligned} Y &= X_{rabbit}(t) - E|JX_{rabbit}(t) - X_m(t)| \\ X_m(t) &= \frac{1}{N}\sum_{i=1}^{N} X_i(t) \end{aligned} \quad (12)$$

where $X_i(t)$ indicates the location of each hawk in iteration t and N denotes the total number of hawks

C. WHALE OPTIMIZATION ALGORITHM

Whale optimization Algorithm(WOA) [14] is a humpback whale special hunting method model algorithm, it abstracts the three behaviors of encircling prey, bubble-net attacking, and search for prey in the hunting process. The algorithm obtains the global optimal solution by updating the position of the whale. WOA is widely used in many fields because of its simple mechanism, few parameters and strong optimization abilities [15], [16]. The three-part formula as are described as follows.

(1) Encircling prey

$$\begin{aligned} D &= |C \cdot X^*(t) - X(t)| \\ X(t+1) &= X^*(t) - A \cdot D \\ A &= 2a \cdot r - a \\ C &= 2r \end{aligned} \quad (13)$$

where t indicates the current iteration, A and C are coefficient vectors, X^* is the position vector of the best solution obtained so far, X is the position vector, a is linearly decreased from 2 to 0 over the course of iterations, and r is a random vector in [1, 0].

(2) Bubble-net attacking

$$\begin{aligned} X(t+1) &= D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t) \\ D' &= |X^*(t) - X(t)| \end{aligned} \quad (14)$$

where b is a constant for defining the shape of the logarithmic spiral, and l is a random number in [−1,1].

(3) Search for prey

$$\begin{aligned} D'' &= |C \cdot X_{rand} - X| \\ X(t+1) &= X_{rand} - A \cdot D \end{aligned} \quad (15)$$

where X_{rand} is a random position vector chosen from the current population.

D. BUTTERFLY OPTIMIZATION ALGORITHM

Butterfly Optimization Algorithm(BOA) [17] is an algorithm that simulates the foraging and mating behavior of butterflies and solves the global optimization problem. It is mainly divided into global search and local search, and its location update formula is presented as follows.

(1) Global search

$$x_i^{t+1} = x_i^t + (r^2 \bullet g^* - x_i^t) \bullet f_i \tag{16}$$

where x_i^t is the solution vector x_i for i-th butterfly in iteration number t. Here, g^* represents the current best solution found among all the solutions in current iteration. Fragrance of i-th butterfly is represented by f_i and r is a random number in [0, 1].

(2) Local search

$$x_i^{t+1} = x_i^t + (r^2 \bullet x_j^t - x_k^t) \bullet f_i \tag{17}$$

where x_j^t and x_k^t are j-th and k-th butterflies from the solution space

E. SEAGULL OPTIMIZATION ALGORITHM

Seagulls are social animals that migrate with the seasons. During the migration, they will change their direction towards the best position that makes them likely to collide during migration and sometimes the gulls will attack prey by spiraling. Seagull optimization algorithm (SOA) [18] is an algorithm inspired by the migration and attacks of seagulls. The migration (exploration) and attacking (exploitation) formula are described as follows. (1) Migration

$$\begin{aligned} C_s &= A \cdot P_s(x) \\ A &= f_c - (x \cdot (f_c/Max_{iteration})) \\ M_s &= B \cdot (P_{bs}(x) - P_s(x)) \\ B &= 2A^2 \cdot rd \\ D_s &= |C_s + M_s| \end{aligned} \tag{18}$$

where C_s represents the position of search agent which does not collide with other search agent, P_s represents the current position, x is the current iteration, f_c which goes from 2 to 0 is to control changes in A, $Max_{iteration}$ is the max number of iterations, M_s is the best position, P_{bs} is the current best position, P_s is the current position, r is a random of [1, 0] and D_s is the new position.

(2) Attacking

$$\begin{aligned} x' &= r \cdot \cos(k) \\ y' &= r \cdot \sin(k) \\ z' &= r \cdot k \\ r &= u \cdot e^{kv} \\ P_s(x) &= (D_s \cdot x' \cdot y' \cdot z') + P_{bs}(x) \end{aligned} \tag{19}$$

where r is the radius of each turn of the spiral, k is a random number in range $[0 \leq \mathrm{k} \leq 2\pi]$. u and v are constants to define the spiral shape, $P_s(x)$save the best solution and updates the position of others search agent.

III. THE IMPROVED DUNG BEETLE ALGORITHM

A. OPPOSITION-BASED LEARNING DESIGN IN GODBO

As well known that exploration is an important part in any algorithm, it is a major indicator of the performance of the DBO algorithm. Therefore, in order to improve the performance of the DBO algorithm, opposition-based learning is introduced into the algorithm. The opposition-based learning has been used in some algorithm [19] and its opposite solution is defined as follows.

$$x^t = m + n - x \tag{20}$$

where x is a random value in [m, n], and x^t is the opposite number. In the high dimensional problems, the above formula can be extended as follows.

$$x_j^t = lb + ub - x_j \tag{21}$$

where x_j is the number of n-dimensional vector, and $lb$and $ub$are the boundaries of DBO algorithm, and x_j^t is the number of opposite vector.

In the Eq (1) and Eq (2), the algorithm update position by previous position, that find the time to get the better solution will be longer. The opposition-based learning is used to obtain the opposite vector, then renew the position as the following Eq.(22).

$$x_i^{n+1} = \begin{cases} x_i^{n+1} x_j^t \geq x_i^{n+1} \\ x_j^t x_j^t < x_i^{n+1} \end{cases} \tag{22}$$

B. THE ROLE OF OPTIMAL VALUE IN GODBO

Meanwhile, in the Eq. (6), the generation of candidate solutions is affected by two random numbers (K_1 and K_2), which makes the probability of producing better and worse candidate solutions equal. In the particle swarm optimization algorithm, each particle changes its speed and position as it moves according to the optimal value it finds and the nowadays global best value of the swarm, and inspired by the motion of the PSO, its idea that each particle is influenced by the optimal value during its motion is applied to the ABC, resulting in an improved artificial bee swarm algorithm with improved performance.

Inspired by the improved artificial bee colony algorithm, this paper introduces the current optimal value to guide the production of candidate solutions. The newly formulate is presented as the following Eq.(23).

$$\begin{aligned} X_i^{n+1} = X_i^n + K_1 \bullet (X_i^n - LB_1) + K_2 \bullet (X_i^n - UB_2) \\ + \lambda(X^r - X_i^n) \end{aligned} \tag{23}$$

where λ is a random value in [0, M], and M is a number greater than zero. X^r is the current optimal value and X_i^n represents the place information of the i-th generation of dung beetles after the n-th iteration. Their difference represents the direction of the current position and the optimal position, while λ is used to control the amount of influence received by the optimal value so that the probability of producing better will be larger. As the value of M increases from 0, the new position is influenced by the current optimal position and thus the mining ability is enhanced, but if the value of C is too large, the new solution is influenced by the optimal solution too much and thus the mining ability of the algorithm is reduced.

C. GODBO ALGORITHM DESCRIPTION

The performing process of the GODBO algorithm is described as the following Algorithm 1.

Algorithm 1 GODBO algorithm

Input parameters: maximum quantity of iterations N_{max}, population size Pop**;**

Result: the optimal site X^r and its adaptation value f_b;

1. Set the population, max iteration and so on;
2. **while** the current frequency of iterations is less than N_{max} **do**
3. **for** from the first population to pop **do**
4. **if** Rolling dung beetle **then**
5. Define the random quantity δ**;**
6. **if** random value δ is less than 0.9 **then**
7. Generate α for Eq. (1)
8. Updating the dung beetle position using Eq. (1).
9. **else**
10. Updating the dung beetle position by using Eq. (2).
11. **end if**
12. Acquire the opposite dung beetle position.
13. Update the dung beetle position by using Eq. (22).
14. **end if**
15. **if** female dung beetle **then**
16. Updating the dung beetle position by using Eq. (4).
17. **end if**
18. **if** newborn dung beetle **then**
19. Updating the dung beetle position by using Eq. (23).
20. **end if**
21. **if** thief dung beetle **then**
22. Updating the dung beetle position by using Eq. (7).
23. **end if**
24. **end for**
25. **if** the new location is better **then**
26. Updating.
27. **end if**
28. the number of iterations plus one.
29. **end while**
30. **return** the optimal site X^r and its adaptation value f_b

IV. EXPERIMENT

A. RELATED PARAMETER AND TEST FUNCTION

CEC2017 functions are employed in the original dung beetle algorithm and a large number of other optimization algorithms are utilized to test how good the algorithm is, hence they are also used to be as test the algorithms in this paper. Since the second function of CEC2017 has been removed, this paper uses the remaining 29 functions for testing. The dimension (*Dim*) of the experiments is set to 10, the initial population size (*Pop*) is 30, the peak value of iterations is set to 500 and 30 runs are performed on each test function.

B. EXPERIMENTAL METHODS

1) COMPARISON OF PARAMETER VALUES

Optimal movement is where the dung beetle is influenced by the current optimal value during its movement, updating the position of the dung beetle, which λ is used to determine the magnitude of the influence of the optimal position during the movement of the dung beetle. To determine the effect of λ on the algorithm, function 1 (Shifted and Rotated Bent Cigar Function) from cec2017 [20] is used for comparison in this paper. This paper compares the Minimum(Min), Maximum(Max), Average(Avg) and Variance(Var) to evaluation the algorithm advantages and disadvantages. The *Dim* is set as 10 and the initial *Pop* is set as 30 to determine the effect of λ on the GODBO, and the experimental results are recorded in Table 1.

TABLE 1. Comparison of parameters.

Different values	Min	Max	Avg	Var
0.75	1.02E+02	6.10E+07	3.10E+06	1.29E+14
1.0	1.02E+02	1.76E+06	6.33E+04	1.02E+11
1.25	1.50E+02	**1.27E+04**	**4.75E+03**	**1.29E+07**
1.5	1.06E+02	1.13E+07	3.82E+05	4.27E+12
1.75	6.38E+02	6.80E+08	2.27E+07	1.54E+16

From the Table 1, it is clear that the performance of GODBO is better when the value of λ is taken as 1.25, hence this paper sets its value as 1.25 for subsequent experiments.

2) COMPARISON OF DIFFERENT DIMENSIONS

To test another initialization parameter (*Dim*) of the algorithm, this paper uses some test functions of cec2017 for testing. The *Pop* was set to 30, and the Dimensions are 10, 30, 50 and 100. The detailed results are shown in Table 2.

Through the table 2, it is well known that GODBO can also be applied to higher dimensional problems.

3) ACCURACY TESTING

In this section, the algorithm is evaluated by maximum value, minimum value, mean value and variance. To test the accuracy of algorithm, many algorithms are used which contains DBO algorithm, HHO algorithm, WOA algorithm, SOA algorithm and BOA algorithm. The analysis results of various algorithms on test function are shown in Table 3.

From the above data, the GODBO algorithm is a bit better than all the algorithms tested in all respects of some functions. However, the GODBO algorithm does not perform so well on some functions, indicating that GODBO is still inadequate in some aspects and needs further improvement.

In the comparison between GODBO algorithm and DBO algorithm, from the mean value, 24 GODBO algorithms are slightly better than the original algorithm, 2 are equal to

TABLE 2. Dimension analysis.

Function name	Algorithm	Min	Max	Avg	Var
		Dim=10			
F1	DBO	7.33E+02	1.50E+07	1.16E+06	1.37E+13
	GODBO	**1.50E+02**	**1.27E+04**	**4.75E+03**	**1.29E+07**
F3	DBO	3.00E+02	4.72E+03	8.94E+02	1.18E+06
	GODBO	**3.00E+02**	**1.02E+03**	**4.08E+02**	**2.82E+04**
F8	DBO	8.15E+02	8.59E+02	8.31E+02	1.23E+02
	GODBO	**8.08E+02**	**8.44E+02**	**8.23E+02**	**8.04E+01**
F12	DBO	4.43E+03	2.31E+07	2.51E+06	2.52E+13
	GODBO	**2.85E+03**	**8.78E+06**	**1.96E+06**	**1.01E+13**
F20	DBO	2.03E+03	2.24E+03	2.12E+03	5.27E+03
	GODBO	**2.02E+03**	**2.20E+03**	**2.10E+03**	**3.09E+03**
		Dim=30			
F1	DBO	2.46E+07	9.30E+08	3.09E+08	5.83E+16
	GODBO	**5.35E+04**	**1.44E+08**	**3.86E+07**	**1.36E+15**
F3	DBO	7.34E+04	1.90E+05	9.70E+04	9.51E+08
	GODBO	**6.00E+04**	**1.08E+05**	**7.96E+04**	**1.22E+08**
F8	DBO	9.16E+02	1.15E+03	1.04E+03	3.12E+03
	GODBO	**8.77E+02**	**1.02E+03**	**9.38E+02**	**1.06E+03**
F12	DBO	2.84E+06	2.13E+08	5.58E+07	3.10E+15
	GODBO	**1.32E+06**	**1.74E+08**	**3.24E+07**	**1.31E+15**
F20	DBO	2.26E+03	3.04E+03	2.72E+03	4.99E+04
	GODBO	2.39E+03	3.13E+03	**2.68E+03**	**3.17E+04**
		Dim=50			
F1	DBO	2.75E+08	5.02E+10	5.36E+09	7.61E+19
	GODBO	**2.42E+08**	**3.56E+09**	**1.06E+09**	**5.40E+17**
F3	DBO	1.63E+05	4.35E+05	2.65E+05	4.58E+09
	GODBO	1.73E+05	**3.70E+05**	**2.49E+05**	**2.82E+09**
F8	DBO	1.03E+03	1.53E+03	1.31E+03	1.39E+04
	GODBO	1.04E+03	**1.30E+03**	**1.14E+03**	**3.72E+03**
F12	DBO	2.64E+08	2.87E+09	1.03E+09	4.20E+17
	GODBO	**3.67E+07**	**2.35E+09**	**6.00E+08**	**4.00E+17**
F20	DBO	3.09E+03	4.29E+03	3.80E+03	9.40E+04
	GODBO	2.94E+03	4.48E+03	**3.78E+03**	2.07E+05
		Dim=100			
F1	DBO	1.99E+10	2.57E+11	8.48E+10	5.10E+21
	GODBO	**1.68E+10**	**5.22E+10**	**3.15E+10**	**8.33E+19**
F3	DBO	3.54E+05	1.29E+06	6.43E+05	6.77E+10
	GODBO	3.81E+05	1.45E+06	**6.32E+05**	7.22E+10
F8	DBO	1.84E+03	2.56E+03	2.14E+03	5.53E+04
	GODBO	**1.69E+03**	**2.17E+03**	**1.87E+03**	**1.59E+04**
F12	DBO	3.05E+09	1.46E+10	7.75E+09	6.11E+18
	GODBO	**1.51E+09**	**4.55E+09**	**2.68E+09**	**6.13E+17**
F20	DBO	5.39E+03	8.70E+03	7.49E+03	5.42E+05
	GODBO	**4.94E+03**	**8.60E+03**	**7.25E+03**	1.07E+06

the original algorithm, and 3 are a bit worse than the DBO. And in variance, there are 21 slightly better than the original algorithm and 8 a bit worse than the DBO.

4) CONVERGENT ANALYSIS

Fig. 1 displays the results of the above six algorithms on the test functions of 29 of CEC2017 benchmark functions.

From the above images, the GODBO algorithm has a faster convergence than the others in some cases, which means that it can find the same optimal value with the least number of iterations. Unfortunately, GODBO is worse than DBO at f17 and f25, indicating that GODBO has a slow convergence and will be attracted to the local optimal solution, thus not finding the optimal solution quickly.

5) FRIEDMAN TEST

To reflect differences between multiple samples, the Friedman [21] test is used. The average ranking is an indicator

TABLE 3. Accuracy test.

Function Name	Algorithm name	Min	Max	Avg	Var
F1	DBO	7.33E+02	1.50E+07	1.16E+06	1.37E+13
	GODBO	**1.50E+02**	**1.27E+04**	**4.75E+03**	**1.29E+07**
	HHO	3.89E+05	2.29E+07	3.85E+06	3.97E+13
	WOA	6.54E+06	3.52E+08	6.26E+07	7.11E+15
	BOA	6.42E+09	2.18E+10	1.44E+10	1.25E+19
	SOA	6.11E+06	9.37E+08	3.07E+08	5.25E+16
F3	DBO	3.00E+02	4.72E+03	8.94E+02	1.18E+06
	GODBO	**3.00E+02**	**1.02E+03**	**4.08E+02**	**2.82E+04**
	HHO	3.34E+02	1.93E+03	7.40E+02	1.95E+05
	WOA	1.13E+03	2.97E+04	9.43E+03	5.48E+07
	BOA	9.72E+03	5.65E+04	1.89E+04	6.17E+07
	SOA	4.07E+02	7.54E+03	3.21E+03	5.63E+06
F4	DBO	4.01E+02	5.13E+02	4.30E+02	1.44E+03
	GODBO	**4.00E+02**	5.16E+02	**4.15E+02**	**7.56E+02**
	HHO	4.01E+02	4.98E+02	4.33E+02	1.11E+03
	WOA	4.06E+02	6.42E+02	4.51E+02	3.13E+03
	BOA	9.37E+02	3.97E+03	2.22E+03	7.76E+05
	SOA	4.15E+02	6.22E+02	4.59E+02	1.49E+03
F5	DBO	5.19E+02	5.68E+02	5.40E+02	2.15E+02
	GODBO	**5.11E+02**	5.59E+02	**5.34E+02**	1.23E+02
	HHO	5.23E+02	5.93E+02	5.57E+02	3.85E+02
	WOA	5.15E+02	5.90E+02	5.52E+02	4.10E+02
	BOA	5.86E+02	7.11E+02	6.59E+02	9.16E+02
	SOA	5.11E+02	5.57E+02	5.34E+02	1.09E+02
F6	DBO	6.01E+02	6.37E+02	6.13E+02	9.48E+01
	GODBO	6.01E+02	**6.17E+02**	**6.07E+02**	**2.62E+01**
	HHO	6.09E+02	6.58E+02	6.39E+02	1.97E+02
	WOA	6.15E+02	6.79E+02	6.42E+02	2.60E+02
	BOA	6.64E+02	7.19E+02	6.92E+02	2.11E+02
	SOA	6.05E+02	6.35E+02	6.15E+02	4.36E+01
F7	DBO	7.28E+02	7.88E+02	7.53E+02	2.90E+02
	GODBO	**7.22E+02**	7.92E+02	**7.44E+02**	**2.44E+02**
	HHO	7.36E+02	8.33E+02	7.91E+02	6.04E+02
	WOA	7.42E+02	8.33E+02	7.85E+02	5.28E+02
	BOA	9.16E+02	1.12E+03	1.00E+03	3.48E+03
	SOA	7.41E+02	8.03E+02	7.66E+02	2.65E+02
F8	DBO	8.15E+02	8.59E+02	8.31E+02	1.23E+02
	GODBO	**8.08E+02**	**8.44E+02**	**8.23E+02**	**8.04E+01**
	HHO	8.12E+02	8.47E+02	8.32E+02	6.52E+01
	WOA	8.20E+02	8.91E+02	8.45E+02	2.20E+02
	BOA	8.76E+02	9.63E+02	9.24E+02	4.70E+02
	SOA	8.12E+02	8.48E+02	8.31E+02	8.33E+01
F9	DBO	9.01E+02	1.42E+03	9.63E+02	1.14E+04
	GODBO	**9.01E+02**	**1.02E+03**	**9.20E+02**	**7.32E+02**
	HHO	1.09E+03	2.03E+03	1.53E+03	5.27E+04
	WOA	9.96E+02	3.43E+03	1.63E+03	2.98E+05
	BOA	2.80E+03	8.28E+03	5.04E+03	1.28E+06
	SOA	9.21E+02	1.44E+03	1.08E+03	2.32E+04
F10	DBO	1.03E+03	2.75E+03	1.99E+03	1.10E+05
	GODBO	1.18E+03	2.61E+03	**1.81E+03**	1.16E+05
	HHO	1.69E+03	2.57E+03	2.09E+03	6.06E+04
	WOA	1.62E+03	2.85E+03	2.23E+03	9.52E+04
	BOA	2.88E+03	4.04E+03	3.48E+03	1.00E+05
	SOA	1.45E+03	2.49E+03	1.99E+03	8.23E+04
F11	DBO	1.11E+03	1.70E+03	1.23E+03	1.80E+04
	GODBO	**1.11E+03**	1.48E+03	1.20E+03	1.21E+04
	HHO	1.13E+03	1.37E+03	1.19E+03	3.78E+03
	WOA	1.15E+03	1.47E+03	1.26E+03	7.95E+03
	BOA	1.59E+03	2.12E+04	5.60E+03	1.89E+07
	SOA	1.14E+03	1.44E+03	1.30E+03	7.37E+03
F12	DBO	4.43E+03	2.31E+07	2.51E+06	2.52E+13
	GODBO	**2.85E+03**	**8.78E+06**	**1.96E+06**	**1.01E+13**
	HHO	1.58E+04	1.78E+07	4.96E+06	2.54E+13
	WOA	4.79E+04	2.35E+07	5.16E+06	3.33E+13
	BOA	1.96E+08	3.10E+09	1.36E+09	6.49E+17
	SOA	4.27E+04	1.45E+07	3.69E+06	1.30E+13
F13	DBO	1.92E+03	4.40E+04	1.15E+04	1.56E+08
	GODBO	2.04E+03	**3.48E+04**	**9.07E+03**	**9.47E+07**
	HHO	1.71E+03	5.60E+04	1.71E+04	1.41E+08
	WOA	2.43E+03	6.75E+04	1.87E+04	1.98E+08
	BOA	1.04E+05	9.50E+08	1.07E+08	4.26E+16
	SOA	2.39E+03	1.18E+05	2.74E+04	5.79E+08
F14	DBO	1.47E+03	3.60E+03	1.93E+03	4.44E+05
	GODBO	**1.45E+03**	**3.44E+03**	**1.74E+03**	**1.92E+05**
	HHO	1.51E+03	5.21E+03	2.23E+03	9.76E+05
	WOA	1.47E+03	6.14E+03	2.83E+03	2.84E+06
	BOA	2.04E+03	4.48E+06	2.72E+05	6.70E+11
	SOA	1.50E+03	8.41E+03	3.63E+03	5.17E+06
F15	DBO	1.68E+03	3.27E+04	5.31E+03	3.36E+07
	GODBO	**1.67E+03**	**1.03E+04**	**3.76E+03**	**3.63E+06**

TABLE 3. ***(Continued.)*** **Accuracy test.**

	HHO	1.92E+03	1.53E+04	7.04E+03	1.34E+07
	WOA	1.70E+03	2.62E+04	8.82E+03	5.67E+07
	BOA	8.92E+03	3.89E+05	1.20E+05	9.62E+09
	SOA	1.66E+03	1.36E+04	5.71E+03	1.15E+07
F16	DBO	1.60E+03	2.09E+03	1.78E+03	1.65E+04
	GODBO	**1.60E+03**	**1.99E+03**	**1.75E+03**	**1.25E+04**
	HHO	1.61E+03	2.19E+03	1.93E+03	1.77E+04
	WOA	1.67E+03	2.38E+03	1.96E+03	2.60E+04
	BOA	1.98E+03	3.16E+03	2.70E+03	7.81E+04
	SOA	1.62E+03	2.03E+03	1.79E+03	1.61E+04
F17	DBO	1.72E+03	1.90E+03	1.79E+03	1.75E+03
	GODBO	1.73E+03	**1.88E+03**	**1.77E+03**	**1.37E+03**
	HHO	1.74E+03	1.94E+03	1.79E+03	2.74E+03
	WOA	1.74E+03	1.90E+03	1.81E+03	2.05E+03
	BOA	1.94E+03	2.51E+03	2.18E+03	2.40E+04
	SOA	1.75E+03	1.98E+03	1.81E+03	3.08E+03
F18	DBO	2.65E+03	4.06E+04	2.12E+04	1.85E+08
	GODBO	2.49E+03	4.96E+04	1.76E+04	2.03E+08
	HHO	2.22E+03	4.61E+04	1.85E+04	2.08E+08
	WOA	1.93E+03	4.40E+04	1.71E+04	1.79E+08
	BOA	2.86E+06	2.34E+09	7.05E+08	5.44E+17
	SOA	1.83E+04	2.05E+05	5.73E+04	1.35E+09
F19	DBO	1.95E+03	1.12E+05	2.21E+04	8.78E+08
	GODBO	**1.93E+03**	**2.62E+04**	**7.50E+03**	**4.49E+07**
	HHO	3.03E+03	8.57E+04	2.01E+04	3.44E+08
	WOA	2.56E+03	6.33E+05	1.27E+05	3.64E+10
	BOA	2.07E+05	1.50E+08	1.26E+07	8.29E+14
	SOA	2.63E+03	5.92E+04	2.08E+04	1.31E+08
F20	DBO	2.03E+03	2.24E+03	2.12E+03	5.27E+03
	GODBO	**2.02E+03**	**2.20E+03**	**2.10E+03**	**3.09E+03**
	HHO	2.08E+03	2.30E+03	2.18E+03	3.41E+03
	WOA	2.06E+03	2.40E+03	2.20E+03	9.00E+03
	BOA	2.26E+03	2.74E+03	2.49E+03	1.39E+04
	SOA	2.05E+03	2.30E+03	2.14E+03	4.48E+03
F21	DBO	2.20E+03	2.34E+03	2.22E+03	8.34E+02
	GODBO	**2.20E+03**	2.36E+03	2.29E+03	3.83E+03
	HHO	2.21E+03	2.40E+03	2.33E+03	3.80E+03
	WOA	2.23E+03	2.40E+03	2.35E+03	1.37E+03
	BOA	2.25E+03	2.53E+03	2.43E+03	3.10E+03
	SOA	2.20E+03	2.34E+03	2.21E+03	1.04E+03
F22	DBO	2.30E+03	2.34E+03	2.31E+03	4.47E+01
	GODBO	**2.27E+03**	**2.32E+03**	**2.31E+03**	6.22E+01
	HHO	2.31E+03	2.33E+03	2.31E+03	2.10E+01
	WOA	2.27E+03	4.02E+03	2.45E+03	1.79E+05
	BOA	2.95E+03	5.02E+03	3.83E+03	2.76E+05
	SOA	2.32E+03	3.96E+03	2.88E+03	4.14E+05
F23	DBO	2.62E+03	2.68E+03	2.65E+03	2.87E+02
	GODBO	**2.61E+03**	2.71E+03	**2.64E+03**	3.51E+02
	HHO	2.62E+03	2.75E+03	2.68E+03	1.07E+03
	WOA	2.62E+03	2.69E+03	2.65E+03	2.34E+02
	BOA	2.69E+03	3.13E+03	2.87E+03	8.29E+03
	SOA	2.61E+03	2.67E+03	2.64E+03	1.95E+02
F24	DBO	2.50E+03	2.80E+03	2.74E+03	7.48E+02
	GODBO	2.53E+03	2.79E+03	2.76E+03	2.01E+02
	HHO	2.50E+03	2.96E+03	2.82E+03	1.32E+04
	WOA	2.60E+03	2.84E+03	2.77E+03	2.61E+03
	BOA	2.85E+03	3.34E+03	3.02E+03	1.04E+04
	SOA	2.74E+03	2.78E+03	2.76E+03	1.22E+02
F25	DBO	2.90E+03	2.97E+03	2.94E+03	5.89E+02
	GODBO	**2.90E+03**	2.99E+03	2.94E+03	7.53E+02
	HHO	2.62E+03	3.03E+03	2.93E+03	5.12E+03
	WOA	2.91E+03	3.04E+03	2.96E+03	9.31E+02
	BOA	3.35E+03	4.89E+03	4.09E+03	9.74E+04
	SOA	2.91E+03	3.04E+03	2.95E+03	8.67E+02
F26	DBO	2.90E+03	3.76E+03	3.15E+03	3.20E+04
	GODBO	**2.80E+03**	4.23E+03	**3.11E+03**	7.89E+04
	HHO	2.82E+03	4.66E+03	3.55E+03	3.23E+05
	WOA	2.92E+03	4.84E+03	3.75E+03	3.87E+05
	BOA	4.07E+03	5.81E+03	4.90E+03	1.72E+05
	SOA	2.93E+03	4.17E+03	3.37E+03	2.19E+05
F27	DBO	3.09E+03	3.20E+03	3.11E+03	6.77E+02
	GODBO	**3.09E+03**	3.17E+03	3.10E+03	2.13E+02
	HHO	3.10E+03	3.31E+03	3.17E+03	3.46E+03
	WOA	3.10E+03	3.28E+03	3.15E+03	2.22E+03
	BOA	3.21E+03	3.72E+03	3.44E+03	1.89E+04
	SOA	3.09E+03	3.10E+03	3.09E+03	3.95E+00
F28	DBO	3.20E+03	3.41E+03	3.31E+03	8.25E+03

TABLE 3. ***(Continued.)*** **Accuracy test.**

	GODBO	3.20E+03	3.74E+03	3.37E+03	1.84E+04
	HHO	3.10E+03	3.77E+03	3.47E+03	2.84E+04
	WOA	3.12E+03	3.75E+03	3.50E+03	2.86E+04
	BOA	3.64E+03	4.50E+03	4.05E+03	3.66E+04
	SOA	3.17E+03	3.73E+03	3.34E+03	1.52E+04
F29	DBO	3.15E+03	3.39E+03	3.26E+03	5.30E+03
	GODBO	**3.15E+03**	3.41E+03	3.26E+03	**4.94E+03**
	HHO	3.22E+03	3.63E+03	3.36E+03	8.34E+03
	WOA	3.23E+03	3.72E+03	3.40E+03	1.64E+04
	BOA	3.55E+03	4.33E+03	3.84E+03	4.17E+04
	SOA	3.16E+03	3.44E+03	3.24E+03	5.99E+03
F30	DBO	7.87E+03	6.02E+06	1.08E+06	2.00E+12
	GODBO	**5.80E+03**	**2.67E+06**	7.32E+05	6.49E+11
	HHO	6.68E+04	9.95E+06	2.10E+06	6.38E+12
	WOA	3.08E+04	6.07E+06	1.99E+06	3.28E+12
	BOA	4.11E+07	2.85E+08	1.30E+08	4.80E+15
	SOA	6.49E+04	8.86E+05	3.88E+05	8.68E+10
	Results	22/29	15/29	20/29	16/29

TABLE 4. **Average ranking.**

Algorithms	Average Ranking
DBO	2.28
GODBO	**1.48**
HHO	3.66
WOA	4.41
BOA	6.00
SOA	3.17

to access the differences in the samples. The smaller the average ranking is, the larger the differences between the examples are. For algorithms, the less the mean difference, the more significantly different the designed algorithm is from the algorithm involved. That means the designed algorithm is better. The following algorithms with comparison are tested by the Friedman and their rankings are listed in Table 4.

From Table 4 we can see that it shows that the GODBO algorithm has the smallest average ranking in comparison with other competitors, which is an indication that the GODBO algorithm is completely different from others.

V. ENGINEERING APPLICATIONS

A. PRESSURE VESSL DESIGN ISSUES

Pressure vessels are designed to reduce the cost of vessel fabrication, which is controlled by four parameters (s_1,s_2,s_3,s_4) and four constraints (g_1,g_2,g_3,g_4) during the calculation of the cost and it can be presented as the following Eq.(24).

$$\begin{aligned}
\min f(s) &= 0.6224 s_1 s_3 s_4 + 1.7781 s_2 s_3^3 + 3.1661 s_1^2 s_4 \\
&\quad + 19.84 s_1^2 s_3 \\
\text{s.t. } g_1 &= -s_1 + 0.0193 s_3 \leq 0, \\
g_2 &= -s_2 + 0.00954 s_3 \leq 0, \\
g_3 &= -\pi s_3^2 s_4 - \frac{4}{3}\pi s_3^3 + 1296000 \leq 0, \\
g_4 &= s_4 - 240 \leq 0
\end{aligned}$$

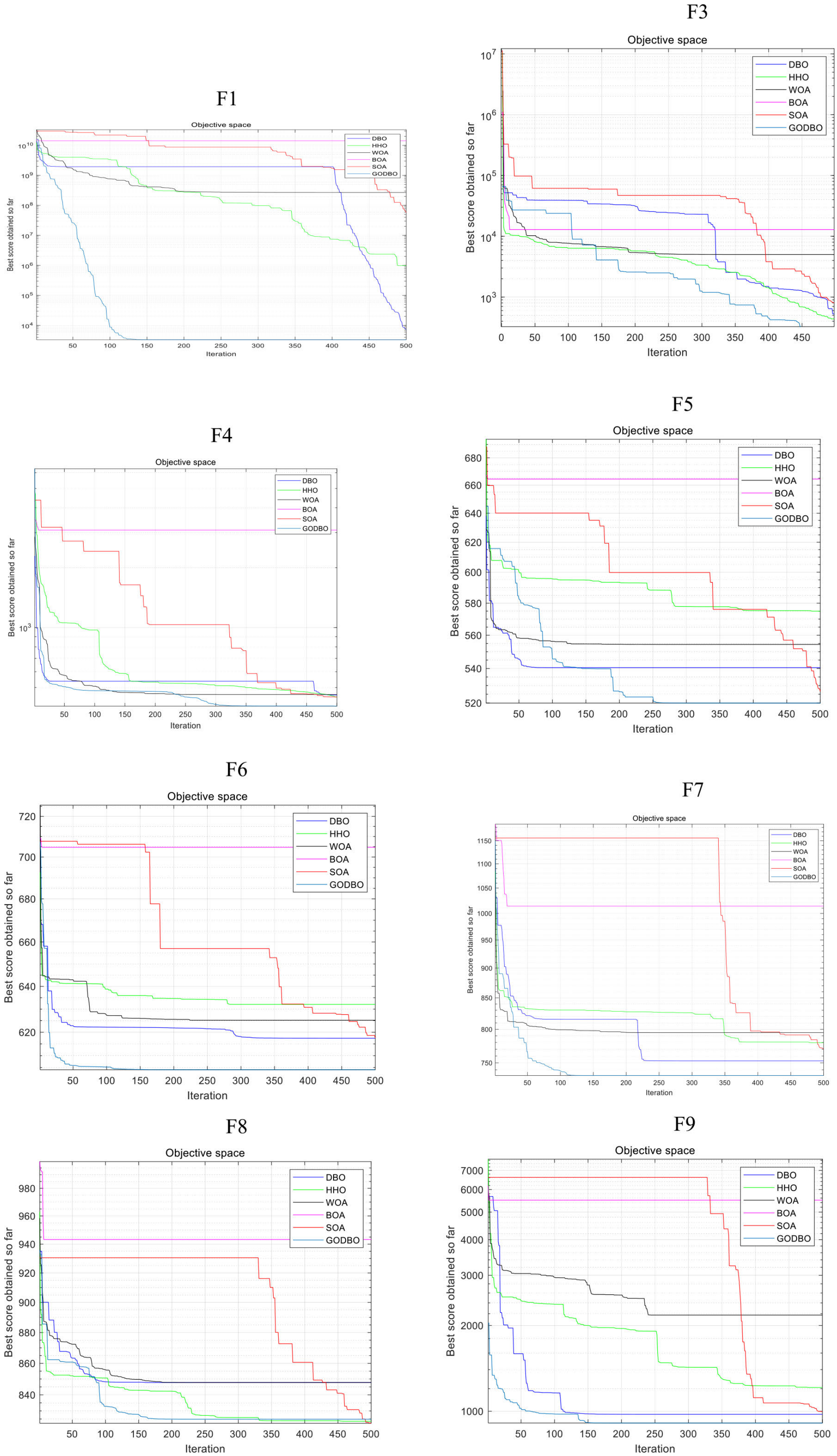

FIGURE 1. Convergence analysis.

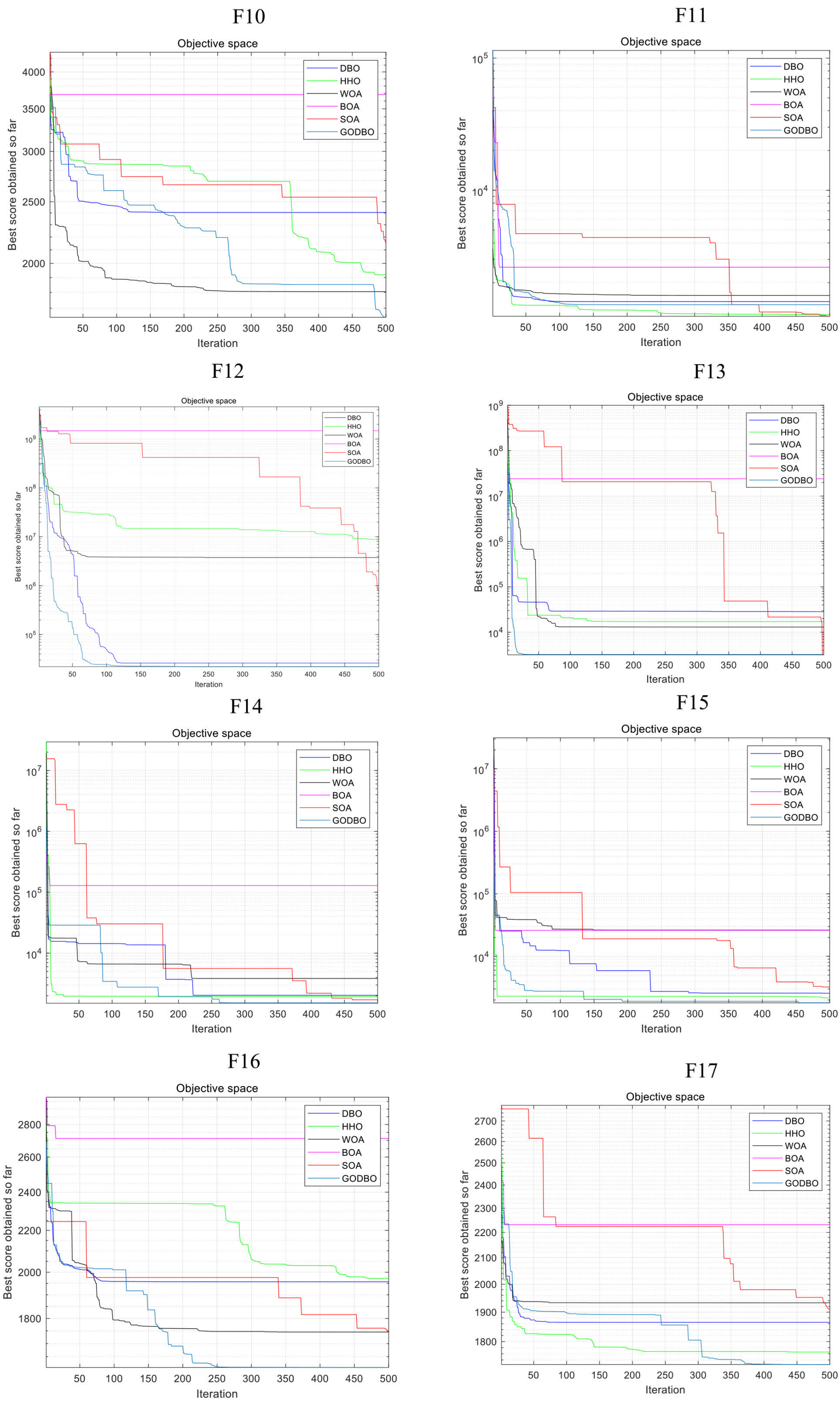

FIGURE 1. ***(Continued.)*** **Convergence analysis.**

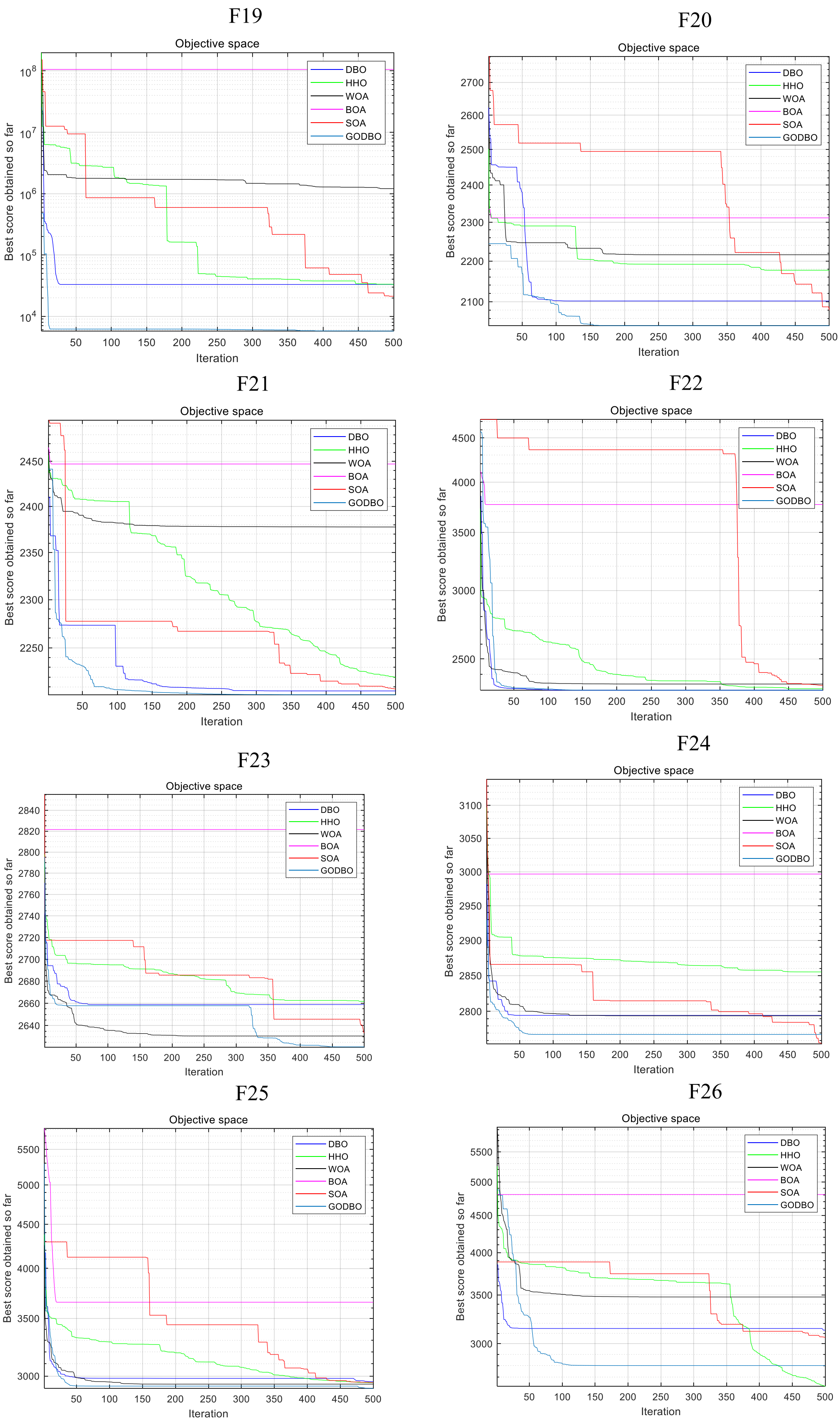

FIGURE 1. ***(Continued.)*** **Convergence analysis.**

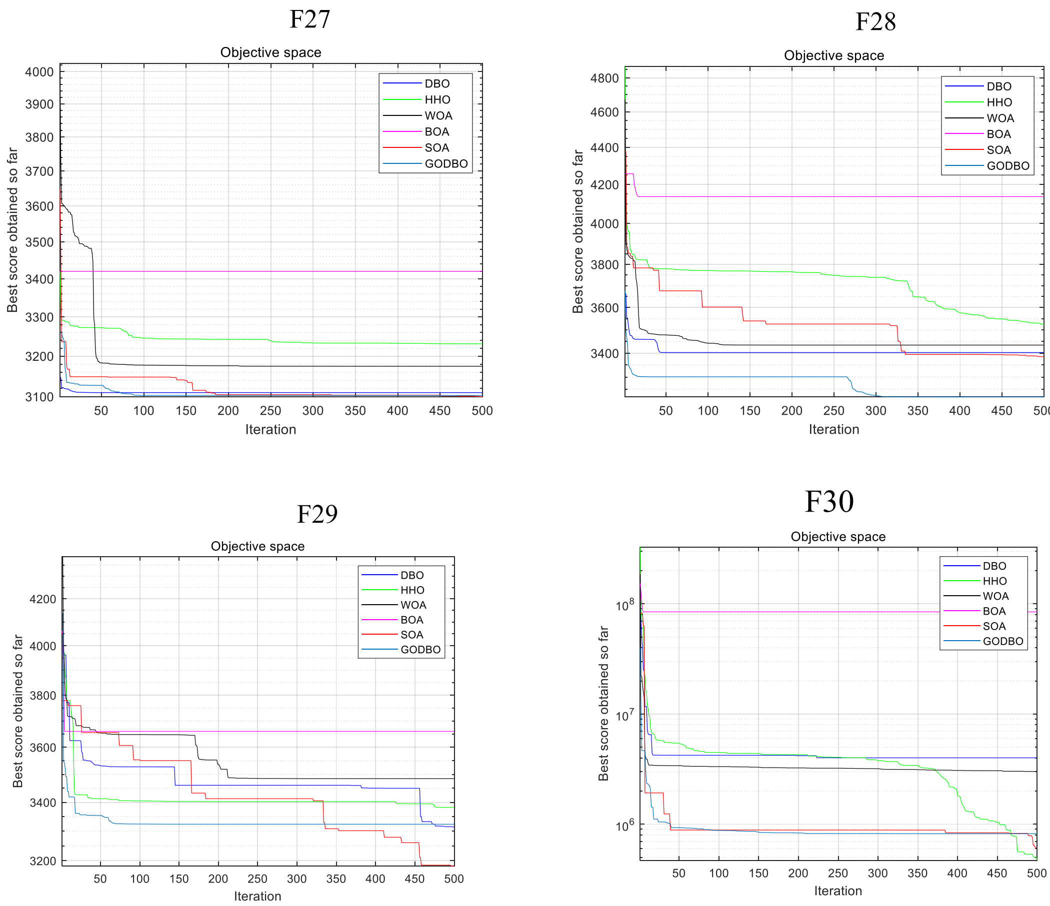

FIGURE 1. *(Continued.)* **Convergence analysis.**

$$0 \le s_1, s_2 \le 99,$$
$$0 \le s_3, s_4 \le 200. \tag{24}$$

The GODBO algorithm was compared with DBO and other optimization algorithms, such as HHO, WOA, DA [3]. Through the data in Table 5, the GODBO has smaller manufacturing cost and is more suitable compared to other algorithms. Therefore, GODBO can be optimal cost when it addresses such problems.

B. WELDED BEAM DESIGN ISSUES

The welded beam design problem is to reduce the cost in the welding beam design process. The welded beam problem has four parameters (s_1,s_2,s_3,s_4) and seven constraints condition(y_1,y_2,y_3,y_4,y_5,y_6,y_7).The formula for constraint and manufacturing costs is as the following Eq.(25).

$$\cos tf(s) = 1.1047 s_1^2 s_2 + 0.04811 s_3 s_4 (14.0 + s_2)$$
$$s.t.\ y_1(s) = \tau(s) - 13600 \le 0,$$
$$y_2(s) = \sigma(s) - 30000 \le 0,$$
$$y_3(s) = \delta(s) - 0.25 \le 0,$$
$$y_4 = s_1 - s_4 \le 0,$$
$$y_5 = p - p_c \le 0,$$
$$y_6 = 0.125 - s_1 \le 0,$$
$$y_7 = 1.10471 s_1^2 + 0.04811 s_3 s_4 (14.0 + s_2) - 5.0 \le 0,$$
$$0.1 \le s_1, s_4 \le 2.0,$$
$$0.1 \le s_2, s_3 \le 10.0 \tag{25}$$

Among them

$$\tau = \sqrt{\tau_1 + 2\tau_1 \tau_2 (\frac{s_2}{2r}) + \tau_2^2}, \tau_1 = \frac{p_d}{s_1 s_2 \sqrt{2}},$$
$$m = p_d (i + \frac{s_2}{2}), j = 2\{\sqrt{2}\, s_1 s_2 [\frac{s_2^2}{12} + \left(\frac{s_1 + s_3}{2}\right)^2]\},$$
$$r = \sqrt{\frac{s_2^2}{4} + \left(\frac{s_1 + s_3}{2}\right)^2}, \sigma = \frac{6 p_d i}{s_4 s_3^2}, \delta = \frac{6 p_d i^3}{n s_3^2 s_4},$$
$$p_c = \frac{4.013 n \sqrt{\frac{s_3^2 s_4^6}{36}}}{i^2} (1 - \frac{s_3}{2i} \sqrt{\frac{n}{4m}}),$$
$$m = 12 \times 10^6, n = 30 \times 10^6,$$
$$p_d = 6000\ lb, i = 14\ in, \tau_2 = \frac{mr}{j}.$$

Nowadays, many scholars have applied the algorithms to the welded volume problem. The detailed results of the experiment are recorded in Table 6. It can be seen form Table 6 that

TABLE 5. Pressure vessel design issues.

Algorithm	s_1	s_2	s_3	s_4	Optimal Cost
GODBO	0.778169	0.384649	40.31962	200	5885.333
DBO	0.81254	0.402322	42.098439	176.6367	5949.13541
HHO	0.982363	0.481957	50.42212	95.0319	6364.267
WOA	1.258149	0.521658	52.87878	77.0291	7829.96
SOA	0.865917	0.427362	41.73767	184.6834	6537.428
DA	0.788358	0.443888	40.47234	200	6157.191

TABLE 6. Welded beam problem.

Algorithms	s_1	s_2	s_3	s_4	Optimal Cost
GODBO	0.205734	3.25304	9.036624	0.20573	1.695245
DBO	0.20163	3.3477	9.0405	0.20606	1.7050958
HHO	0.146605	4.695901	9.163467	0.206891	1.81673
WOA	0.211444	2.941222	9.908907	0.221792	1.936495
SOA	0.173676	4.049001	9.031598	0.206151	1.751655
DA	0.189313	3.573864	9.052871	0.205961	1.717927

GODBO has achieved better result than other algorithms, and hence it can be better applied.

VI. DISCUSSION AND ANALYSIS

Through the above experiments, this paper explores different GBest impact factors and finds the most suitable GBest impact factor. In addition, this paper also compares GODBO with other algorithms in different dimensions to explore the performance of GODBO algorithm at higher latitude. In this paper, we use 29 test functions of cec2017 to compare GODBO with other algorithms and thus explore the exploitation and exploration capabilities of GODBO. Finally, Friedman tests are used to compare GODBO with other algorithms, and GODBO is applied to some simple engineering problems.

From the above experimental results, it can be seen that GODBO still achieves better results compared to DBO as the dimensionality becomes larger, and surface GODBO can also achieve better results when facing problems of higher dimensionality. In addition, it is clear that the exploration capability of GODBO was significantly enhanced in certain test functions. The GODBO introduces the Gbest guided and opposite-based Learning to enhance the exploration power of GODBO by making it stronger in certain subpopulations. In this paper, Friedman tests are performed, showing that GODBO has significant differences with other compared algorithms. Finally, in simple physical problems, GODBO can achieve better results compared to other algorithms, indicating that GODBO has better practical value.

However, from the images of GODBO we clearly see that in the early stage of the algorithm, the convergence speed of GODBO in some test functions is not significantly improved compared to the DBO algorithm, which indicates that the exploitation ability of GODBO is still insufficient, making the algorithm cannot converge quickly. In addition, in some cases, the stability of GODBO is not so ideal, making the algorithm may achieve poor results with a smaller number of experiments. Therefore, how to improve the exploitation capacity and stability of the algorithm and apply it to practical problems will be the next goal.

VII. CONCLUSION

Based on Opposite-Based Learning and the current optimal value(GBest) guided dung beetle algorithm, an enhanced DBO algorithm called GODBO is proposed. In GODBO, the Opposite-Based Learning and the current optimal value is employed to enhance exploration as soon as possible. The 29 test functions of CEC2017 are employed and the results indicate that GODBO has improved in convergence accuracy and convergence speed. Meanwhile, the design of the GODBO is better than other algorithms as illustrated by the Friedman test. In addition, the GODBO algorithm is successfully applied to two constrained engineering design problems, which verifies the practicality of the GODBO algorithm. However, there are still some shortcomings in the GODBO algorithm, how to make the algorithm be better will be the future work.

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Perth, WA, Australia, Dec. 1995, pp. 1942–1948.
[2] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.
[3] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, 2015.
[4] J. Xue and B. Shen, "Dung beetle optimizer: A new meta-heuristic algorithm for global optimization," *J. Supercomput.*, vol. 79, no. 7, pp. 7305–7336, May 2023.
[5] S. S. Choong, L.-P. Wong, and C. P. Lim, "A dynamic fuzzy-based dance mechanism for the bee colony optimization algorithm," *Comput. Intell.*, vol. 34, no. 4, pp. 999–1024, Nov. 2018.
[6] Q. Lu, Han Q.-L. Han, and S. Liu, "A finite-time particle swarm optimization algorithm for odor source localization," *Inf. Sci.*, vol. 277, pp. 40–111, Sep. 2014.
[7] D. Gong Y. Han and J. Sun, "A novel hybrid multi-objective artificial bee colony algorithm for blocking lot-streaming flow shop scheduling problems," *Knowl.-Based Syst.*, vol. 148, pp. 30–115, May 2018.
[8] D. Wu, J. Li, and C. Bao, "Case-based reasoning with optimized weight derived by particle swarm optimization for software effort estimation," *Soft Comput.*, vol. 22, no. 16, pp. 5299–5310, 2017.
[9] M. Karami and L. Wang, "Particle Swarm Optimization for control operation of an all-variable speed water-cooled chiller plant," *Appl. Thermal Eng.*, vol. 130, pp. 962–978, Nov. 2018.
[10] S. Li and J. Li, "Chaotic dung beetle optimization algorithm based on adaptive t-Distribution," in *Proc. IEEE 3rd Int. Conf. Inf. Technol., Big Data Artif. Intell. (ICIBA)*, vol. 3, May 2023, pp. 925–933.
[11] J. Duan, Y. Gong, J. Luo, and Z. Zhao, "Air-quality prediction based on the ARIMA-CNN-LSTM combination model optimized by dung beetle optimizer," *Sci. Rep.*, vol. 13, no. 1, Jul. 2023.
[12] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," *Appl. Math. Comput.*, vol. 217, no. 7, pp. 3166–3173, 2010.
[13] H. Tizhoosh, "Opposition-based Learning: A new scheme for machine intelligence," in *Proc. Int. Conf. Comput. Intell. Modeling Control Autom.*, Vienna, Austria, 2005, pp. 695–701.
[14] S. Mirjalili and A. Lewis, "The Whale Optimization Algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2019.
[15] R. Mahadeva, M. Kumar, V. Gupta, G. Manik, and S. P. Patole, "Modified whale optimization algorithm based ANN: A novel predictive model for RO desalination plant," *Sci. Rep.*, vol. 13, no. 1, p. 2901, Feb. 2023.

[16] J. Wang, J. Bei, H. Song, H. Zhang, and P. Zhang, "A whale optimization algorithm with combined mutation and removing similarity for global optimization and multilevel thresholding image segmentation," *Appl. Soft Comput.*, vol. 137, Apr. 2023, Art. no. 110130.

[17] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, 2018.

[18] G. Dhiman and V. Kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems," *Knowl.-Based Syst.*, vol. 165, pp. 96–169, Feb. 2019.

[19] L. Han and X. He, "A novel opposition-based particle swarm optimization for noisy problems," in *Proc. 3rd Int. Conf. Natural Comput. (ICNC)*, 2007, pp. 624–629.

[20] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization," Nat. Univ. Defense Technol., Changsha, China, Kyungpook Nat. Univ., Daegu, South Korea, Nanyang Technol. Univ., Singapore, Tech. Rep., 2017.

[21] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, Dec. 1937.

WANG ZILONG is currently pursuing the bachelor's degree in computer science with Jiangxi Agricultural University. His current research interests include artificial intelligence, algorithms, and related real-world applications.

SHAO PENG received the M.Sc. degree from the School of Computer and Information Engineering, Jiangxi Agricultural University (JXAU), China, in 2011, and the Ph.D. degree from Wuhan University, China, in June 2016. He is currently with the School of Computer and Information Engineering, JXAU. His current research interests include artificial intelligence, large-scale optimization, image processing, and related real-world applications. He has published more than ten research articles in these fields.