Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation 202 (2022) 343–373
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Original articles

Gannet optimization algorithm : A new metaheuristic algorithm for solving engineering optimization problems

Jeng-Shyang Pan[a,b], Li-Gang Zhang[a], Ruo-Bin Wang[c], Václav Snášel[d], Shu-Chuan Chu[a,*]

[a] College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao, China
[b] Department of Information Management, Chaoyang University of Technology, Taiwan, Province of China
[c] School of Information Science and Technology, North China University of Technology, Beijing, China
[d] Faculty of Electrical Engineering and Computer Science, VŠB-Technical University of Ostrava, Ostrava, Czech Republic

Received 3 March 2022; received in revised form 30 April 2022; accepted 10 June 2022
Available online 17 June 2022

Abstract

Engineering design problems are usually large-scale constrained optimization problems, and metaheuristic algorithms are vital for solving such complex problems. Therefore, this paper introduces a new nature-inspired metaheuristic algorithm: the gannet optimization algorithm (GOA). The GOA mathematizes the various unique behaviors of gannets during foraging and is used to enable exploration and exploitation. GOA's U-shaped and V-shaped diving patterns are responsible for exploring the optimal region within the search space, with sudden turns and random walks ensuring better solutions are found in this region. In order to verify the ability of the GOA to find the optimal solution, we compared it with other comparison algorithms in multiple dimensions of 28 benchmark functions. We found that the GOA has a shorter running time in high dimensions and can provide a better solution. Finally, we apply the GOA to five engineering optimization problems. The experimental results show that the GOA is suitable for many constrained engineering design problems and can provide better solutions in most cases.

Keywords: Gannet optimization algorithm; Optimization algorithms; Swarm intelligence; Evolutionary computation

1. Introduction

Optimization drives the progress of human history. From the age of steam to the age of electricity, from the age of electricity to the age of information, the process of optimization permeates every corner. Optimization is a crucial research tool in scientific research, transportation, intelligent medical and economic management. It studies the search for an optimal solution among many solutions. For example, in transportation, how to choose a reasonable travel route so that the transportation plan can meet the requirements and reduce transportation costs as much as possible; in workshop scheduling, how to allocate limited equipment resources in order to improve production efficiency; in engineering design how to choose reasonable parameters can make the design scheme

* Corresponding author.
E-mail addresses: jengshyangpan@gmail.com (J.-S. Pan), lgangzhang@126.com (L.-G. Zhang), robin945@163.com (R.-B. Wang), vaclav.snasel@vsb.cz (V. Snášel), scchu0803@gmail.com (S.-C. Chu).

https://doi.org/10.1016/j.matcom.2022.06.007

meet the design requirements and reduce the cost. Optimization is to provide theoretical methods and solutions for solving these problems. Because of the rapid development of society, various problems emerge one after another, which significantly promotes the development of optimization [31].

Many methods have been proposed to solve this constrained optimization problem, which can be roughly divided into exact and heuristic. Exact methods such as branch-and-bound and dynamic programming are not suitable for dealing with complex engineering design problems because their results often reach local optimal solutions, and the computational complexity is high. In order to solve complex problems that traditional computing methods cannot solve, metaheuristic algorithms appear. The meta-heuristic algorithm is a general algorithm of the heuristic algorithm, which usually uses random number search techniques to obtain a satisfactory answer within a reasonable time frame. It does not depend on the structure and properties of the problem to be solved, so that metaheuristics can be applied to many optimization problems [16].

To review meta-heuristics, pull the timeline back to the 1970s. Darwin's theory of evolution inspired the Genetic Algorithm (GA) proposed by Holland. It uses three operators of selection, crossover and mutation to achieve the search for the optimal solution [26]. The Ant Colony Optimization algorithm (ACO) proposed in 1991 was inspired by the process of ants searching for food. Through the signal transmission substance of pheromone, each ant will walk along with the place with high pheromone. The simple behavior of a single ant can be converted into intelligent behavior within the population, forming a mechanism similar to positive feedback [13]. The Particle Swarm Optimization (PSO) algorithm proposed in 1995 is a meta-heuristic algorithm for simulating the predation behavior of birds. The idea is that each particle has two properties of position and velocity. The particle's position update is adjusted according to the global and individual optimum. The PSO algorithm has achieved great success, and since then, the meta-heuristic algorithm has entered a stage of rapid development [34].

The Harmony Search (HS) algorithm proposed in 2001 was inspired by music performance. A band must repeatedly adjust the pitch played by each musician to obtain a beautiful harmony state. HS balances the solution space development and exploration ability by adjusting the two parameters of HMCR and PAR [20]. The Artificial Bee Colony (ABC) algorithm proposed in 2005 was inspired by the behavior of bees. The bees share the information about the food source through the "swing dance", and other bees can choose which food source to collect honey from according to the information of the dance [30]. The Cat Swarm Optimization (CSO) algorithm proposed in 2006 is inspired by the two behaviors of cat searching and tracking, and the optimal predation strategy is obtained through the transformation of the two behaviors [11]. The Gravitational Search Algorithm (GSA) proposed in 2009 was inspired by phenomena in physics and is based on the law of universal gravitation and Newton's second law. Due to the mutual attraction between particles, they move continuously in the search space and finally search for the optimal solution to the problem [52].

The teaching-based optimization algorithm (TLBO) proposed in 2011 simulates classroom teaching through two stages teacher and learner, specifically, learning from the best and learner interactive learning [51]. In 2012, a Water Cycle Algorithm (WCA) inspired by rainfall, confluence, evaporation and other phenomena was proposed to classify the water flow population into optimal, suboptimal and ordinary solutions and to introduce an indirect method of finding optimal solutions to improve the quality of solutions in the search space [15]. The sine–cosine algorithm (SCA) proposed in 2016 oscillates for optimization based on the mathematical model of the sine–cosine function [42]. The Butterfly Optimization Algorithm (BOA) proposed in 2019 was inspired by butterflies' mating and foraging behavior. Butterflies emit scents that attract each other to each other. A butterfly determines its direction by analyzing the scent of other butterflies [3].

The Archimedes Optimization Algorithm (AOA) proposed in 2020 is inspired by the classical Archimedean principle, where the individuals of the population in AOA are objects immersed in a liquid, and the optimal solution is found by adjusting the density, volume and acceleration of the objects [23]. Arithmetic Optimization Algorithm (AOA) proposed in 2021 realizes optimization according to the distribution characteristics of arithmetic operators, that is, multiplication strategy and division strategy explore the solution space by improving the dispersion of solutions, while addition strategy and subtraction strategy exploit the solution space by reducing the dispersion of solutions [1]. The Honey Badger Algorithm (HBA) proposed in 2022 simulates the foraging behavior of honey badgers. The honeypot has two behavioral modes. In the mining mode, it will use its sense of smell to locate the prey and then select a suitable location to capture the prey; in the honey-gathering mode, it will directly locate the prey by guiding the badger's position [22].

Based on the original meta-heuristic algorithm, some meta-heuristic algorithm improvement strategies have also attracted the attention of many scholars [27,28,61]. Hu et al. proposed a hybrid algorithm with the aim

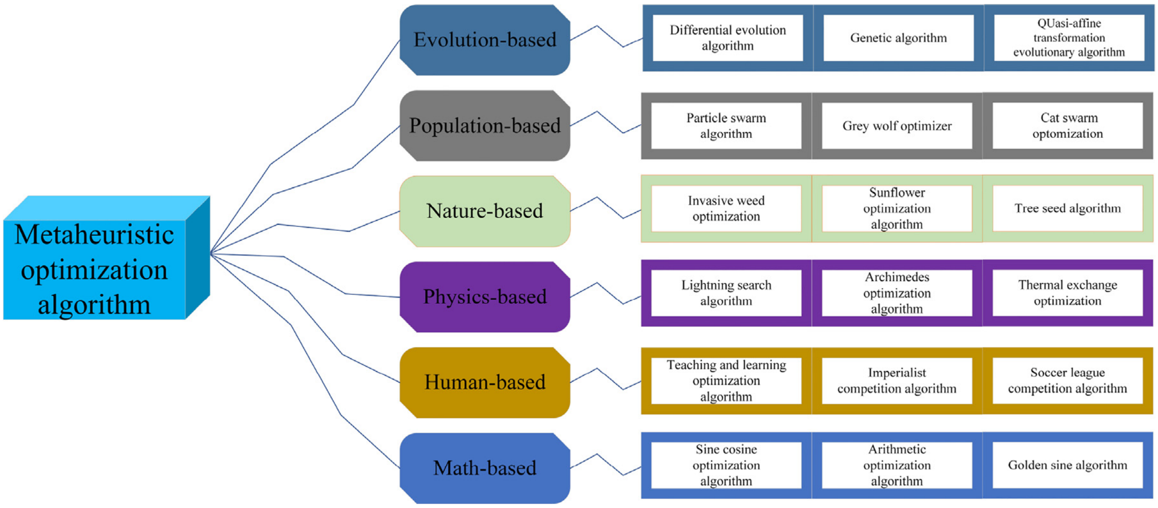


Fig. 1. Meta-heuristic algorithm classification.

of improving the performance of the arithmetic optimization algorithm and preventing it from falling into local optima. Specifically, it combines the advantages of AOA and cross-optimization algorithms, and adopts a point set initialization strategy and an optimal neighborhood strategy [27]. WANG et al. proposed a parallel Multi-Verse Optimizer (MVO) algorithm with a communication strategy. The idea is to randomly divide the initial solution into several groups and share the information of different groups after each fixed iteration. In this way, the defect of premature convergence of the original MVO and falling into local optimum can be optimized [58].

Based on the above discussion, we can classify algorithms inspired by phenomena such as biology or physics into six categories: evolution-based methods [39,48,50,59], population behavior-based methods [11,34,46], nature-based methods [21,35,38], physics-based methods [23,32,55], human-society-based methods [4,47,51], and math-based methods [1,42,56], as shown in Fig. 1. With the rapid development of the information age, the complexity of the problems people face is different from the past, and different meta-heuristic algorithms have different performances when dealing with specific problems. Therefore, it is necessary to propose new meta-heuristic algorithms to solve complex optimization problems.

Most metaheuristic algorithms are inspired by the habits of various animals, such as predation, migration, etc. [49,57]; however, no research has been conducted to analyze the predation of gannet for mathematical modeling and development of metaheuristic algorithms, so we intend to introduce a more efficient algorithm: the gannet optimization algorithm. The proposed algorithm simulates the unique behavior of gannet during predation and presents several mathematical models. In this paper, the performance of the proposed algorithm is tested using 28 benchmark functions and applied to the problems of five engineering cases. The experimental results show that the GOA is better than other optimization algorithms when dealing with different problems, and it is a robust meta-heuristic optimization algorithm.

The main contributions of this paper are as follows:

1. We propose a new swarm-based metaheuristic optimization algorithm, GOA, which imitates the predation behavior of gannets.
2. GOA has designed two phases, the exploration phase is responsible for searching for the best area through the U-shaped and V-shaped diving patterns of Gannets, and the sudden rotation and random walk in the development phase ensure that a better solution is found in the area.
3. Evaluate GOA's convergence ability, running time, and fitness statistics across multiple dimensions through state-of-the-art meta-heuristics.
4. We conduct a series of experiments to investigate the performance of the proposed algorithm on practical engineering design problems.

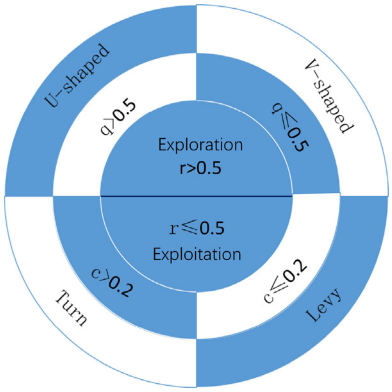


Fig. 2. Exploration and exploitation phase.

The rest of this paper is organized as follows; the GOA is introduced in detail in Section 2; the performance of the proposed algorithm is tested using 28 benchmark functions in Section 3, and the GOA is applied to 5 engineering optimizations in Section 4 Problems; Section 5 gives conclusions and future work.

2. Gannet optimization algorithm

In this section, inspired by the predation behavior of gannets, we propose a new meta-heuristic optimization algorithm called the gannet optimization algorithm. In the pond goose optimization algorithm, we propose two stages of exploration and exploitation to simulate the predatory behavior of pond geese. The exploration and exploitation stages contain a total of four different behaviors for predation, which are U-shaped dive mode, V-shaped dive mode, sudden rotation and random wandering, see Fig. 2, and the specific parameters in the figure will be explained in detail in the content of this section. In order to better explain the GOA algorithm, its pseudocode and flow chart are also given in this section, and finally, the time complexity of the GOA algorithm is analyzed.

2.1. Gannet habits

Gannets are stubby and fat, with slender necks, as shown in Fig. 3. Gannets commonly inhabit lakes and seashores in many parts of the world in flocks, feeding mainly on fish, but also on crustaceans, birds and amphibians. Gannets have sharp eyes, and even when flying at high altitudes, fish roaming in the water can never escape their eyes. Gannet moves awkwardly on land but flies and swims nimbly. If a flock of gannets finds a fish, they will form a straight line or a semi-circular array to catch it; a gannet that finds a fish will dash into the water from more than ten meters in the air a cannonball to catch the fish. Gannets also have a strong swimming ability after entering the water at high speed. They can continue to dive a few meters to find fish after the power of the dive disappears, so the predation success rate of gannets is very high.

2.2. Initialization phase

The GOA starts with a set of random solutions shown in Eq. (1), at which point the optimal solution is regarded as the optimal global solution.

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,j} & \cdots & x_{1,Dim-1} & x_{1,Dim} \\ x_{2,1} & \cdots & x_{2,j} & \cdots & x_{2,Dim-1} & x_{2,Dim} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \cdots & \cdots & x_{i,j} & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N-1,1} & \cdots & x_{N-1,j} & \cdots & x_{N-1,Dim-1} & x_{N-1,Dim} \\ x_{N,1} & \cdots & x_{N,j} & \cdots & x_{N,Dim-1} & x_{N,Dim} \end{bmatrix} \tag{1}$$

Fig. 3. Gannet.

x_i denotes the position of the ith individual. Each $x_{i,j}$ in the matrix X can be calculated by Eq. (2).

$$x_{i,j} = \mathrm{r}_1 \times \left(\mathrm{UB}_j - LB_j\right) + LB, i = 1, 2, \ldots, N, j = 1, 2, \ldots, Dim \tag{2}$$

where UB_j and LB_j are the upper and lower bounds of the jth dimension of the given problem, N is the number of individuals in the population, Dim denotes the dimensional size of the problem, and r_1 is a random number between 0 and 1.

We also define an MX matrix called the memory matrix. The initialization phase assigns the values of the X matrix to MX. In each iteration of evolution, the change of position of the gannet individuals will be recorded in the memory matrix MX. After evaluation by the fitness function, if the memory matrix MX_i individual performs better than the current solution X_i individual, then MX_i is used instead of X_i; otherwise, the solution in the X matrix continues to be used.

2.3. Exploration phase

Gannets search for prey in the water from the air, and once they find prey, they adjust their dive pattern according to the depth of the prey dive. There are two types of dives: a long and deep U-shaped dive and a short and shallow V-shaped dive [37], see Fig. 4. Here we use Eq. (4) for the U-shaped dive and Eq. (5) for the V-shaped dive.

$$t = 1 - \frac{It}{Tmax\_iter} \tag{3}$$

$$a = 2 * \cos\left(2 * \pi * \mathrm{r}_2\right) * t \tag{4}$$

$$b = 2 * \ \mathrm{V}\left(2 * \pi * \mathrm{r}_3\right) * t \tag{5}$$

$$V(x) = \begin{cases} -\frac{1}{\pi} * x + 1, x \in (0, \pi) \\ \frac{1}{\pi} * x - 1, x \in (\pi, 2\pi) \end{cases} \tag{6}$$

where It is the number of current iterations, Tmax_iter is the maximum number of iterations, r_2 and r_3 are both random numbers between 0 and 1.

The next step is to use these two dive strategies for position updating. Gannet has essentially the same probability of choosing between the two strategies when they are predating, so we define a random number q to randomly select the two dive strategies. The position update formula is given in Eq. (7).

$$MX_i(t+1) = \begin{cases} X_i(t) + u1 + u2 & , q \geq 0.5 \quad \text{(a)} \\ X_i(t) + v1 + v2 & , q < 0.5 \quad \text{(b)} \end{cases} \tag{7}$$

$$u2 = A * \left(X_i(t) - X_r(t)\right) \tag{8}$$

$$v2 = B * \left(X_i(t) - X_m(t)\right) \tag{9}$$

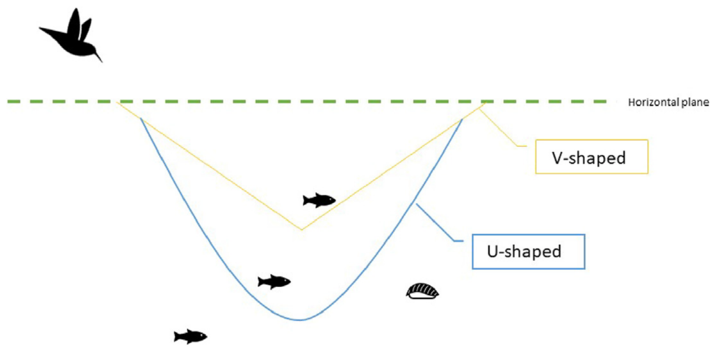


Fig. 4. U-shaped and V-shaped.

$$A = (2 * \mathrm{r}_4 - 1) * a \tag{10}$$

$$B = (2 * \mathrm{r}_5 - 1) * b \tag{11}$$

where r_4 and r_5 are both random numbers between 0 and 1, u1 is a random number between −a and a, v1 is a random number between −b and b, $\mathrm{X_i}(t)$ is the ith individual in the current population, $\mathrm{X_r}(t)$ is a randomly selected individual in the current population, $\mathrm{X_m}(t)$ denotes the average position of individuals in the current population, and $\mathrm{X_m}(t)$ can be calculated using the following equation.

$$X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \tag{12}$$

2.4. Exploitation phase

Two more actions are needed to further exploitation after the gannet rushes into the water in the above two ways. Cunning fish in the water is often accompanied by a sudden turning movement to escape the gannet's chase. The gannet also expends a tremendous amount of energy to capture the fish trying desperately to escape. Here we define a capture capacity, see Eq. (13). When the gannet has sufficient energy, i.e., a high capture capacity, the gannet will capture the fish, see the first scenario in Fig. 5. As time iterates, the gannet's energy decreases and will not be able to complete the capturing action, see the second scenario in Fig. 5.

$$Capturability = \frac{1}{R * t2} \tag{13}$$

$$t2 = 1 + \frac{It}{Tmax\_iter} \tag{14}$$

$$R = \frac{M * vel^2}{L} \tag{15}$$

$$L = 0.2 + (2 - 0.2) * \mathrm{r}_6 \tag{16}$$

where r_6 is a random number between 0 and 1, M = 2.5 Kg is the weight of the gannet, Vel = 1.5 m/s is the speed of the gannet in the water, ignoring the resistance in the water at this time.

If the catching ability of the gannet is within the range of catchable prey, the position is updated with a sudden turning; otherwise, the gannet is unable to catch this flexible fish and performs a Levy movement to search for the next target at random [60], see Eq. (17).

$$MX_i(t+1) = \begin{cases} t * delta * (X_i(t) - X_{Best}(t)) + X_i(t) & , Capturability \geq c \quad \text{(a)} \\ X_{Best}(t) - (X_i(t) - X_{Best}(t)) * P * t & , Capturability < c \quad \text{(b)} \end{cases} \tag{17}$$

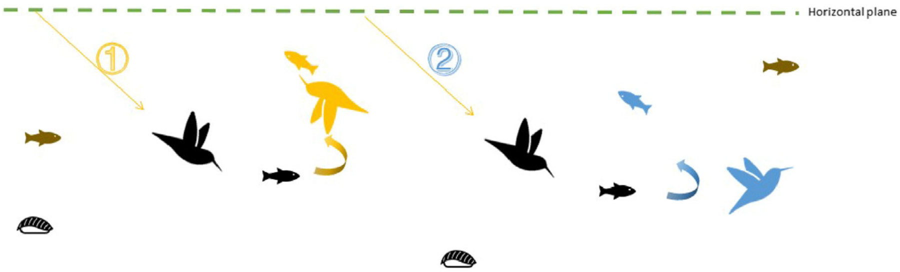


Fig. 5. Sudden turning.

$$delta = Capturability * |X_i(t) - X_{Best}(t)| \tag{18}$$

$$P = \text{Levy}(Dim) \tag{19}$$

where c = 0.2 is a constant whose value was determined after several experiments, $X_{\text{Best}}(t)$ is the best performing individual in the current population, and Levy() is the Levy flight function, which can be obtained from Eq. (20).

$$\text{Levy(Dim)} = 0.01 \times \frac{\mu \times \sigma}{|v|^{\frac{1}{\beta}}} \tag{20}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \tag{21}$$

where μ and σ are random values between (0,1) and $\beta = 1.5$ is a pre-determined constant.

2.5. GOA pseudo-code

The GOA starts the whole optimization process by generating a random set of initial solutions. During each iteration, all individuals adjust their positions according to the four position update formulas provided by the GOA. The two position update formulas are chosen with equal probability in the exploration phase, and the choice of the position update formula in the exploitation phase is determined by the size of the catching ability. Because the two diving methods and the turning search process are inseparable in the gannet feeding process, our algorithm selects either the exploration or exploitation phases with equal probability in each iteration. The whole optimization process is repeated repeatedly with the expectation of obtaining a near-optimal solution or an optimal solution. Finally, when the result satisfies the end criteria, the whole algorithm terminates the iteration. The pseudo-code of the GOA is given in Algorithm 1. The flowchart is given in Fig. 6.

2.6. Complexity analysis

The computational complexity of the GOA depends on three main processes: the initialization process, the calculation of the fitness function and the update of the gannet positions. The initial process of the GOA obtains the number of individuals N in the population and evaluates the fitness value for the positions of N individuals, so the complexity is O(N). Where the calculation of the fitness function depends on the specific problem and is not included in the analysis here. In the last process, the complexity of the GOA for location update is O(T × N) + O(T × N × D), where T represents the maximum number of iterations and D represents the dimensionality of the problem. Finally, the computational complexity of the GOA is obtained as O(N × (T × D + 1)).

Algorithm 1 Pseudo-code of the GOA

```
Input: N: population size; Dim: problem dimension; Tmax_iter: maximum number of iterations;
Output: the location of Gannet and its fitness value;
Initialize the population X randomly according to equation 1,r and q are all random numbers from 0 to 1;
Generate memory matrix MX;
Calculate the fitness value of X;
while stopping condition is not met do
    if rand > 0.5 then
        for MX_i do
            if q ≥ 0.5 then
                Update the location Gannet using Equation (7a);
            else
                Update the location Gannet using Equation (7b);
            end if
        end for
    else
        for MX_i do
            if c ≥ 0.2 then
                Update the location Gannet using Equation (17a);
            else
                Update the location Gannet using Equation (17b);
            end if
        end for
    end if
    for MX_i do
        Calculate the fitness value of MX_i;
        If the value of MX_i is better than the value of X_i, replace X_i with MX_i;
    end for
end while
```

3. Results and analysis of the experiment

In this section, in order to study the effectiveness of the proposed gannet optimization algorithm, the statistical results such as the average fitness value, standard deviation and Wilcoxon Rank-sum test are compared; the proposed algorithm is more intuitively compared through the fitness value iteration curve performance, and finally, compare the execution time of the algorithms. We use the 28 benchmark functions in the CEC2013 test set for testing and compare the results with the following algorithms. These comparison algorithms include recently proposed algorithms such as STOA and AOA, as well as classical algorithms such as PSO and SCA.

- Particle Swarm Optimization (PSO)
- Whale Optimization Algorithm (WOA)
- Sine Cosine Algorithm (SCA)
- Butterfly Optimization Algorithm (BOA)
- Sooty Tern Optimization Algorithm (STOA)
- Archimedes Optimization Algorithm (AOA)
- Black-Hole Optimization Algorithm (BH)
- Aquila Optimizer (AO)
- Harris Hawks Optimization (HHO)

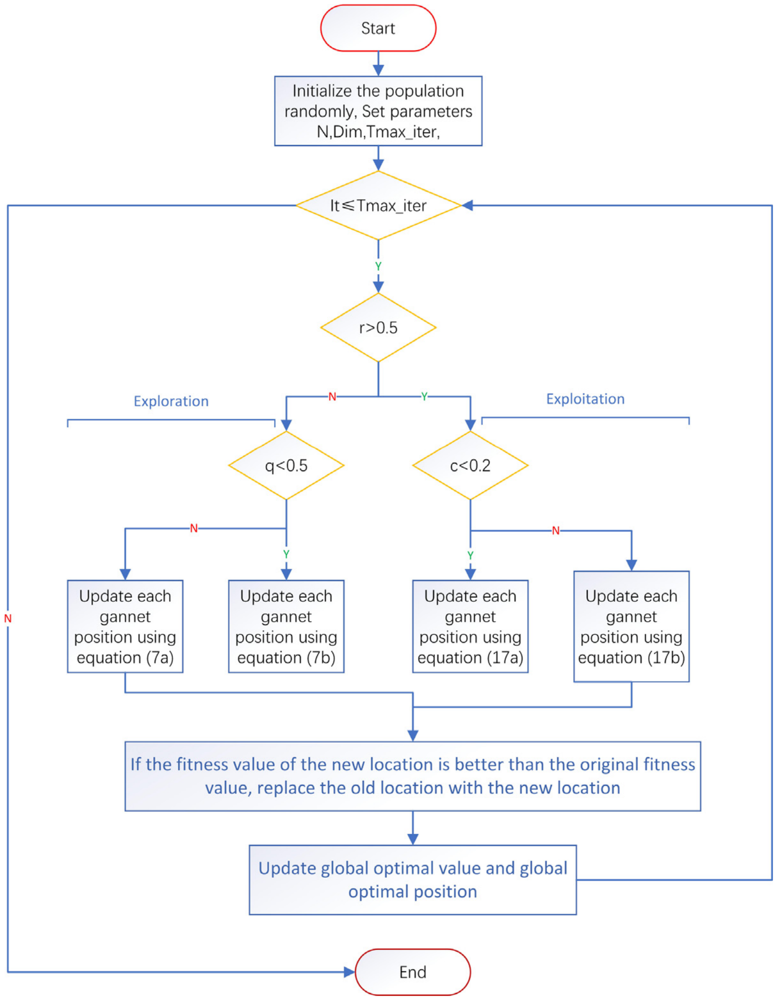


Fig. 6. Flow chart of GOA.

3.1. Parameter settings

The GOA was experimented with using a desktop computer with Windows 10 Professional 64-bit, Intel(R) Core (TM) i3-8100 CPU @ 3.60 GHz, 24.0 GB RAM, Matlab R2014b. The parameters used for the comparison algorithm can be found in Table 1.

3.2. CEC 2013 test suite analysis

In order to consider the fairness of the algorithm comparisons, we limit the number of iterations and the population size of each algorithm. The CEC2013 test set includes single-peak, basic multimodal, and combinatorial functions. The single-peaked function (F1–F5) has a sole global optimum, allowing comparison of the exploitation capabilities of different optimization algorithms. The multimodal functions (F6–F20) have multiple local optima and can be used to compare the exploration ability of various algorithms and their ability to escape from local optima. The combinatorial functions (F21–F28) are a combination of single-peaked and multimodal functions and maybe good for evaluating the combined ability of various algorithms in exploration and exploitation. All of these functions are minimization problems, and comparing the minimum values can help us evaluate optimization algorithms' properties.

Table 1
Parameter settings for each related algorithm.

Algorithm	Parameter	Value
PSO	Vmax	6
	c1 and c2	2
	w	0.3
WOA	a	Linearly decreases from 2 to 0
	a2	Linearly decreases from −1 to −2
	Spiral factor b	1
SCA	a	2
AO	alpha	0.1
	delta	0.1
	u,	0.0265
	r0	10
HHO	E0	Random number between −1 and 1
	E1	Linearly decreases from 2 to 0
BOA	Probability switch	0.6
	power_exponent	0.1
	sensory_modality	0.01
STOA	Sa	Linearly decreases from 2 to 0
AOA	MOP_Max	1
	MOP_Min	0.2
	Alpha	5
	Mu	0.499

3.2.1. Statistical results

In this experiment, we use 10-dimensional, 30-dimensional and 50-dimensional three dimensions for testing. Record each algorithm's average fitness value and standard deviation in three dimensions. Tables 2–4 show the statistical results of each algorithm. At the end of each table, use the win to represent the number of wins for the GOA. As more wins are obtained, the better the performance of the pond goose optimization algorithm is proven to be. From the table, we can see that the GOA shows excellent results in all dimensions, and the GOA also gets better as the dimension increases.

In order to statistically verify the excellent results obtained by GOA, we use the nonparametric Wilcoxon Rank-sum test to determine whether the GOA algorithm is significantly different from other comparison algorithms. The significance level alpha is taken as 0.05, the p-values of the tests are given in Tables 5–7, and the data with p-values greater than 0.05 have been shown in bold. We can see that in the three dimensions of the eighth function, the GOA algorithm is not very different from other algorithms, which means that most algorithms can find the best solution. Compared to GOA, most algorithms also achieve good results on the 50-dimension of the 16th function. In addition to the above cases, GOA has obvious gaps with other comparison algorithms.

3.2.2. Convergence analysis

To further evaluate the superiority of the GOA relative to other algorithms, we use the convergence curves of all algorithms for visual comparison. We can obtain the differences between the GOA and other algorithms in terms of convergence speed and convergence ability through the convergence curves. The iteration plots of each algorithm on 28 benchmark functions are shown in Figs. 7–9. The horizontal axis is the number of iterations. In order to display the iteration curve smoothly, we make a point every 50 generations, so there are twenty points shown in the graph for one thousand iterations. The vertical axis is the average of the fitness values of the GOA and other comparison algorithms run 30 times. From the statistics in Tables 5–7, we can see that the GOA is significantly superior in all three dimensions, and here we only list the iteration curves of the algorithm for 30dim.

In the single-peaked functions in Fig. 7, we can see that the GOA finds a relatively good solution at the beginning of the iteration compared to other algorithms and maintains this advantage until the final stage. The fact that the

Table 2
10 Dim simulation results of CEC 2013 benchmark function.

Function	Metric	GOA	PSO	WOA	SCA	BH	AO	BOA	STOA	AOA	HHO
F1	AVG	−1.40E+03	−1.20E+03	−1.40E+03	−7.15E+02	−1.30E+03	−1.40E+03	6.81E+03	−1.01E+03	5.54E+03	−1.40E+03
	STD	1.77E−12	3.15E+02	1.77E+00	2.41E+02	3.95E+01	3.99E−01	2.42E+03	2.66E+02	2.50E+03	1.25E−01
F2	AVG	1.98E+05	3.46E+05	6.48E+06	5.55E+06	2.91E+06	6.11E+06	9.94E+06	2.37E+06	2.30E+07	2.71E+06
	STD	1.42E+05	1.36E+06	3.34E+06	2.57E+06	4.53E+05	3.12E+06	5.10E+06	2.02E+06	1.64E+07	1.48E+06
F3	AVG	5.00E+07	5.71E+08	1.50E+09	1.14E+09	7.78E+08	6.06E+08	4.38E+10	5.21E+08	8.48E+11	7.39E+08
	STD	1.05E+08	1.96E+09	1.28E+09	7.13E+08	5.06E+08	7.25E+08	4.94E+10	4.76E+08	4.50E+12	8.52E+08
F4	AVG	2.80E+03	9.45E+03	3.52E+04	8.88E+03	7.74E+03	1.07E+04	1.31E+04	7.58E+03	1.43E+04	1.05E+04
	STD	3.36E+03	6.13E+03	1.75E+04	2.66E+03	3.00E+03	2.58E+03	3.13E+03	3.44E+03	3.22E+03	1.88E+03
F5	AVG	−1.00E+03	−8.57E+02	−9.44E+02	−8.14E+02	−9.53E+02	−9.69E+02	1.92E+03	−9.19E+02	−1.86E+02	−1.00E+03
	STD	2.83E−08	2.14E+02	2.21E+01	5.89E+01	1.34E+01	2.37E+01	1.80E+03	8.28E+01	2.79E+02	1.14E−01
F6	AVG	−8.87E+02	−7.96E+02	−8.39E+02	−8.44E+02	−8.20E+02	−8.64E+02	−3.50E+02	−8.49E+02	−5.84E+02	−8.48E+02
	STD	1.73E+01	7.66E+01	4.34E+01	2.69E+01	6.28E+00	3.19E+01	1.78E+02	3.78E+01	2.15E+02	3.34E+01
F7	AVG	−7.57E+02	−7.04E+02	−7.03E+02	−7.51E+02	−7.50E+02	−7.53E+02	5.36E+02	−7.64E+02	−3.76E+02	−6.86E+02
	STD	3.71E+01	5.53E+01	3.87E+01	1.41E+01	2.50E+01	1.52E+01	3.83E+03	1.90E+01	5.39E+02	5.25E+01
F8	AVG	−6.80E+02	−6.80E+02	−6.80E+02	−6.80E+02	−6.80E+02	−6.80E+02	−6.80E+02	−6.80E+02	−6.80E+02	−6.80E+02
	STD	7.74E−02	1.08E−01	8.39E−02	7.12E−02	7.32E−02	8.67E−02	9.70E−02	6.74E−02	7.13E−02	1.08E−01
F9	AVG	−5.95E+02	−5.93E+02	−5.91E+02	−5.91E+02	−5.93E+02	−5.94E+02	−5.92E+02	−5.93E+02	−5.90E+02	−5.91E+02
	STD	1.68E+00	1.62E+00	1.43E+00	1.12E+00	1.57E+00	1.25E+00	7.09E−01	1.15E+00	9.83E−01	1.75E+00
F10	AVG	−5.00E+02	−3.59E+02	−4.79E+02	−4.00E+02	−4.77E+02	−4.86E+02	3.60E+02	−4.45E+02	1.43E+02	−4.97E+02
	STD	2.02E−01	1.04E+02	1.32E+01	3.68E+01	7.09E+00	7.43E+00	2.48E+02	3.62E+01	2.81E+02	1.38E+00
F11	AVG	−3.90E+02	−3.45E+02	−3.33E+02	−3.41E+02	−3.33E+02	−3.60E+02	−2.84E+02	−3.64E+02	−3.04E+02	−3.56E+02
	STD	7.50E+00	2.43E+01	2.62E+01	9.49E+00	2.27E+01	1.86E+01	2.25E+01	1.21E+01	3.34E+01	1.29E+01
F12	AVG	−2.75E+02	−2.31E+02	−2.04E+02	−2.38E+02	−2.35E+02	−2.59E+02	−1.85E+02	−2.65E+02	−2.01E+02	−1.96E+02
	STD	1.24E+01	2.98E+01	2.67E+01	1.06E+01	2.18E+01	1.66E+01	2.05E+01	1.33E+01	2.99E+01	3.55E+01
F13	AVG	−1.56E+02	−1.21E+02	−1.13E+02	−1.38E+02	−1.35E+02	−1.36E+02	−8.22E+01	−1.54E+02	−5.81E+01	−8.82E+01
	STD	1.72E+01	2.52E+01	2.34E+01	9.47E+00	1.62E+01	1.97E+01	2.60E+01	1.22E+01	2.86E+01	3.35E+01
F14	AVG	2.45E+02	6.37E+02	1.07E+03	1.25E+03	1.18E+03	6.21E+02	1.66E+03	8.78E+02	7.71E+02	6.25E+02
	STD	1.78E+02	2.35E+02	3.78E+02	2.02E+02	2.46E+02	3.04E+02	1.58E+02	3.35E+02	2.46E+02	2.55E+02
F15	AVG	7.91E+02	1.16E+03	1.27E+03	1.72E+03	1.18E+03	7.34E+02	1.70E+03	1.39E+03	1.33E+03	1.09E+03
	STD	2.94E+02	3.08E+02	3.85E+02	2.44E+02	3.71E+02	2.58E+02	2.37E+02	4.00E+02	2.73E+02	2.54E+02
F16	AVG	2.01E+02	2.01E+02	2.01E+02	2.01E+02	2.01E+02	2.01E+02	2.02E+02	2.01E+02	2.01E+02	2.01E+02
	STD	2.33E−01	3.28E−01	3.08E−01	2.31E−01	2.07E−01	3.94E−01	2.69E−01	2.25E−01	3.10E−01	2.46E−01
F17	AVG	3.23E+02	3.37E+02	3.77E+02	3.71E+02	3.65E+02	3.59E+02	4.17E+02	3.53E+02	4.16E+02	4.07E+02
	STD	5.58E+00	1.80E+01	2.20E+01	1.34E+01	1.28E+01	1.10E+01	1.74E+01	1.03E+01	3.04E+01	3.15E+01
F18	AVG	4.26E+02	4.58E+02	4.89E+02	4.75E+02	4.62E+02	4.69E+02	5.17E+02	4.57E+02	5.18E+02	5.09E+02
	STD	6.53E+00	2.18E+01	2.54E+01	1.02E+01	1.25E+01	1.34E+01	1.28E+01	1.30E+01	2.27E+01	3.32E+01

(*continued on next page*)

Table 2 (*continued*).

F19	AVG	5.01E+02	5.49E+02	5.10E+02	5.12E+02	5.10E+02	5.06E+02	6.55E+03	5.06E+02	2.32E+03	5.07E+02
	STD	4.65E−01	1.83E+02	5.47E+00	4.71E+00	2.52E+00	2.24E+00	5.33E+03	1.15E+01	1.45E+03	2.54E+00
F20	AVG	6.03E+02	6.04E+02	6.04E+02	6.04E+02	6.04E+02	6.04E+02	6.04E+02	6.04E+02	6.05E+02	6.04E+02
	STD	5.72E−01	4.76E−01	3.33E−01	3.04E−01	3.54E−01	2.92E−01	4.36E−01	2.77E−01	4.32E−01	3.49E−01
F21	AVG	1.09E+03	1.11E+03	1.08E+03	1.12E+03	1.10E+03	1.10E+03	1.26E+03	1.09E+03	1.21E+03	1.09E+03
	STD	5.72E+01	1.15E+01	6.53E+01	8.62E+00	1.57E−02	1.07E−02	5.51E+01	4.78E+01	5.63E+01	3.49E+01
F22	AVG	1.49E+03	2.02E+03	2.27E+03	2.57E+03	2.34E+03	1.90E+03	2.79E+03	1.98E+03	2.24E+03	1.96E+03
	STD	3.27E+02	3.60E+02	3.57E+02	2.53E+02	4.46E+02	3.75E+02	1.84E+02	2.81E+02	3.91E+02	3.32E+02
F23	AVG	1.82E+03	2.40E+03	2.56E+03	2.75E+03	2.48E+03	2.15E+03	2.90E+03	2.34E+03	2.60E+03	2.36E+03
	STD	3.40E+02	3.71E+02	3.79E+02	2.33E+02	3.10E+02	3.14E+02	1.93E+02	3.94E+02	3.06E+02	3.21E+02
F24	AVG	1.22E+03	1.23E+03	1.23E+03	1.23E+03	1.22E+03	1.22E+03	1.18E+03	1.22E+03	1.24E+03	1.23E+03
	STD	5.66E+00	7.97E+00	3.61E+00	2.31E+00	2.42E+01	3.05E+00	1.64E+01	5.19E+00	9.31E+00	1.42E+01
F25	AVG	1.32E+03	1.33E+03	1.33E+03	1.32E+03	1.31E+03	1.32E+03	1.29E+03	1.32E+03	1.34E+03	1.33E+03
	STD	1.97E+01	5.93E+00	3.76E+00	1.37E+01	3.15E+01	3.29E+00	2.39E+01	2.75E+00	1.12E+01	4.12E+00
F26	AVG	1.40E+03	1.47E+03	1.42E+03	1.40E+03	1.38E+03	1.38E+03	1.39E+03	1.40E+03	1.45E+03	1.45E+03
	STD	5.46E+01	7.28E+01	6.82E+01	1.05E+01	2.02E+01	2.92E+01	1.57E+01	4.95E−01	4.99E+01	6.82E+01
F27	AVG	1.83E+03	1.84E+03	1.94E+03	1.93E+03	1.71E+03	1.75E+03	1.92E+03	1.89E+03	2.04E+03	1.95E+03
	STD	9.52E+01	1.42E+02	6.71E+01	2.51E+01	3.27E+01	7.79E+01	6.51E+01	4.04E+01	1.00E+02	1.26E+02
F28	AVG	1.96E+03	2.19E+03	2.28E+03	2.15E+03	2.19E+03	2.09E+03	2.19E+03	2.01E+03	2.46E+03	2.30E+03
	STD	2.24E+02	1.78E+02	9.55E+01	9.75E+01	1.07E+02	1.41E+02	2.32E+02	1.51E+02	1.16E+02	1.85E+02
Win		/	26	27	27	24	24	24	26	28	27

Table 3
30 Dim simulation results of CEC 2013 benchmark function.

Function	Metric	GOA	PSO	WOA	SCA	BH	AO	BOA	STOA	AOA	HHO
F1	AVG	−1.40E+03	2.38E+04	−5.78E+02	1.62E+04	1.41E+04	−8.94E+02	4.95E+04	6.50E+03	5.00E+04	−1.38E+03
	STD	2.32E−04	6.02E+03	4.10E+02	3.29E+03	1.47E+03	2.68E+02	4.54E+03	4.29E+03	7.17E+03	6.24E+00
F2	AVG	9.37E+06	3.19E+08	1.12E+08	2.61E+08	1.17E+08	7.74E+07	8.06E+08	5.12E+07	5.35E+08	4.37E+07
	STD	4.60E+06	1.47E+08	4.29E+07	8.29E+07	1.09E+07	3.01E+07	3.26E+08	1.91E+07	2.11E+08	1.20E+07
F3	AVG	3.32E+09	1.06E+15	1.44E+11	9.54E+10	4.39E+12	3.84E+10	8.59E+20	2.08E+10	1.28E+18	2.14E+10
	STD	3.59E+09	2.70E+15	2.29E+11	5.22E+10	5.45E+12	1.46E+10	2.66E+21	9.19E+09	4.02E+18	1.08E+10
F4	AVG	3.55E+04	5.00E+04	1.04E+05	5.51E+04	4.82E+04	5.01E+04	5.86E+04	4.69E+04	6.24E+04	4.53E+04
	STD	8.70E+03	9.88E+03	2.84E+04	1.03E+04	5.01E+03	4.28E+03	3.52E+03	8.27E+03	4.32E+03	4.58E+03
F5	AVG	−1.00E+03	3.99E+03	−6.38E+01	2.69E+03	1.44E+03	−2.46E+02	2.46E+04	1.21E+03	1.05E+04	−9.01E+02
	STD	5.23E−03	1.62E+03	2.71E+02	1.02E+03	3.37E+02	3.53E+02	9.12E+03	1.32E+03	2.80E+03	3.78E+01
F6	AVG	−8.36E+02	2.77E+03	−5.86E+02	5.01E+02	1.12E+03	−6.31E+02	1.06E+04	−5.77E+02	8.30E+03	−7.57E+02
	STD	2.60E+01	1.44E+03	1.26E+02	3.31E+02	2.76E+02	6.25E+01	2.46E+03	1.36E+02	2.95E+03	3.78E+01
F7	AVG	−6.72E+02	9.85E+03	6.86E+03	−5.41E+02	3.63E+03	3.39E+02	1.09E+06	−6.69E+02	6.27E+05	5.88E+03
	STD	4.39E+01	2.53E+04	2.75E+04	1.06E+02	5.10E+03	4.29E+03	2.29E+06	2.61E+01	1.13E+06	1.57E+04
F8	AVG	−6.79E+02	−6.79E+02	−6.79E+02	−6.79E+02	−6.79E+02	−6.79E+02	−6.79E+02	−6.79E+02	−6.79E+02	−6.79E+02
	STD	5.52E−02	7.72E−02	6.79E−02	5.62E−02	7.14E−02	4.22E−02	4.75E−02	5.61E−02	4.28E−02	7.14E−02
F9	AVG	−5.69E+02	−5.63E+02	−5.62E+02	−5.59E+02	−5.62E+02	−5.63E+02	−5.60E+02	−5.66E+02	−5.60E+02	−5.61E+02
	STD	4.36E+00	4.02E+00	2.93E+00	1.91E+00	2.57E+00	3.31E+00	1.74E+00	4.04E+00	2.07E+00	2.41E+00
F10	AVG	−4.99E+02	3.14E+03	1.40E+02	1.89E+03	1.49E+03	−1.38E+01	8.30E+03	4.59E+02	6.04E+03	−4.00E+02
	STD	4.56E−01	9.11E+02	1.87E+02	4.84E+02	2.20E+02	1.46E+02	1.40E+03	2.88E+02	1.35E+03	3.76E+01
F11	AVG	−3.13E+02	3.27E+01	2.16E+02	3.08E+01	1.07E+02	−5.55E+01	3.94E+02	−1.04E+02	4.06E+02	9.52E+00
	STD	2.56E+01	8.24E+01	1.44E+02	4.74E+01	7.27E+01	6.16E+01	7.53E+01	5.11E+01	1.41E+02	7.33E+01
F12	AVG	−1.02E+02	1.71E+02	3.26E+02	1.33E+02	2.05E+02	8.30E+01	5.00E+02	1.04E+01	4.37E+02	3.30E+02
	STD	6.90E+01	8.31E+01	1.25E+02	3.72E+01	5.03E+01	4.43E+01	7.87E+01	4.46E+01	1.04E+02	9.79E+01
F13	AVG	5.35E+01	4.00E+02	4.05E+02	2.45E+02	3.34E+02	2.72E+02	6.00E+02	1.57E+02	4.97E+02	4.96E+02
	STD	5.65E+01	9.10E+01	1.04E+02	4.41E+01	5.41E+01	6.43E+01	7.08E+01	3.22E+01	9.59E+01	1.15E+02
F14	AVG	2.87E+03	3.61E+03	5.82E+03	7.59E+03	6.53E+03	4.45E+03	8.03E+03	6.18E+03	7.07E+03	3.98E+03
	STD	6.73E+02	6.83E+02	7.12E+02	3.04E+02	6.86E+02	5.01E+02	3.33E+02	6.91E+02	6.40E+02	9.51E+02
F15	AVG	4.82E+03	4.61E+03	6.75E+03	7.94E+03	6.67E+03	5.48E+03	8.13E+03	6.61E+03	7.60E+03	5.50E+03
	STD	9.77E+02	7.66E+02	7.74E+02	3.02E+02	8.16E+02	7.41E+02	3.69E+02	6.24E+02	4.06E+02	7.62E+02
F16	AVG	2.02E+02	2.02E+02	2.02E+02	2.03E+02	2.03E+02	2.03E+02	2.03E+02	2.03E+02	2.03E+02	2.02E+02
	STD	3.88E−01	6.94E−01	4.82E−01	4.53E−01	4.24E−01	5.36E−01	3.27E−01	3.89E−01	4.67E−01	4.93E−01
F17	AVG	4.45E+02	6.90E+02	9.79E+02	8.91E+02	8.17E+02	7.99E+02	1.12E+03	8.30E+02	1.21E+03	1.10E+03
	STD	3.44E+01	8.65E+01	8.78E+01	6.72E+01	5.98E+01	6.88E+01	4.79E+01	6.50E+01	6.97E+01	8.32E+01
F18	AVG	5.84E+02	9.12E+02	1.07E+03	9.75E+02	9.27E+02	9.57E+02	1.23E+03	9.31E+02	1.27E+03	1.21E+03
	STD	3.42E+01	8.77E+01	1.16E+02	5.92E+01	6.13E+01	6.62E+01	5.19E+01	7.53E+01	6.41E+01	9.58E+01

(*continued on next page*)

Table 3 (*continued*).

F19	AVG	5.11E+02	5.75E+04	7.33E+02	1.53E+04	7.63E+03	5.87E+02	4.78E+05	4.59E+03	3.85E+05	5.44E+02
	STD	3.78E+00	4.81E+04	2.01E+02	1.04E+04	1.31E+03	3.66E+01	1.74E+05	5.91E+03	1.10E+05	1.06E+01
F20	AVG	6.13E+02	6.15E+02	6.15E+02	6.14E+02	6.15E+02	6.15E+02	6.15E+02	6.14E+02	6.15E+02	6.15E+02
	STD	1.33E+00	1.83E−01	2.22E−01	4.31E−01	1.77E−01	2.33E−01	6.02E−06	4.89E−01	4.67E−02	1.16E−01
F21	AVG	1.02E+03	2.78E+03	1.64E+03	2.86E+03	2.48E+03	1.30E+03	3.20E+03	2.52E+03	3.25E+03	1.10E+03
	STD	9.26E+01	1.00E+02	3.98E+02	8.42E+01	6.51E+01	1.73E+02	5.39E+01	2.73E+02	7.71E+01	5.50E+01
F22	AVG	3.85E+03	6.01E+03	7.91E+03	9.09E+03	8.72E+03	6.34E+03	9.41E+03	7.17E+03	8.78E+03	6.81E+03
	STD	9.73E+02	9.08E+02	9.66E+02	4.82E+02	5.72E+02	7.17E+02	3.50E+02	7.07E+02	6.52E+02	8.51E+02
F23	AVG	6.24E+03	6.44E+03	8.12E+03	9.33E+03	8.92E+03	7.35E+03	9.65E+03	7.68E+03	9.44E+03	7.86E+03
	STD	6.85E+02	8.98E+02	6.83E+02	3.19E+02	6.55E+02	7.99E+02	3.24E+02	8.12E+02	4.36E+02	7.27E+02
F24	AVG	1.28E+03	1.40E+03	1.32E+03	1.33E+03	1.36E+03	1.31E+03	1.35E+03	1.29E+03	1.43E+03	1.34E+03
	STD	1.06E+01	5.02E+01	1.15E+01	6.49E+00	1.18E+01	8.25E+00	2.17E+01	7.86E+00	5.82E+01	2.54E+01
F25	AVG	1.40E+03	1.51E+03	1.44E+03	1.44E+03	1.48E+03	1.43E+03	1.45E+03	1.41E+03	1.52E+03	1.46E+03
	STD	1.38E+01	2.55E+01	1.37E+01	6.25E+00	1.09E+01	9.55E+00	1.72E+01	9.33E+00	1.96E+01	2.07E+01
F26	AVG	1.53E+03	1.51E+03	1.60E+03	1.44E+03	1.41E+03	1.50E+03	1.45E+03	1.51E+03	1.58E+03	1.58E+03
	STD	7.71E+01	9.77E+01	3.70E+01	5.61E+01	1.85E+00	9.79E+01	4.49E+01	9.19E+01	4.51E+01	7.08E+01
F27	AVG	2.36E+03	2.72E+03	2.68E+03	2.72E+03	2.73E+03	2.58E+03	3.11E+03	2.52E+03	2.86E+03	2.75E+03
	STD	9.57E+01	1.03E+02	8.53E+01	5.57E+01	8.34E+01	7.13E+01	1.58E+02	8.37E+01	1.19E+02	8.63E+01
F28	AVG	2.06E+03	5.59E+03	6.32E+03	4.63E+03	5.57E+03	4.97E+03	6.82E+03	3.69E+03	6.82E+03	6.23E+03
	STD	8.86E+02	4.12E+02	8.60E+02	2.99E+02	5.55E+02	4.11E+02	5.12E+02	4.66E+02	5.64E+02	5.31E+02
		/	25	27	26	27	26	26	27	27	27

Table 4
50 Dim simulation results of CEC 2013 benchmark function.

Function	Metric	GOA	PSO	WOA	SCA	BH	AO	BOA	STOA	AOA	HHO
F1	AVG	−1.40E+03	4.24E+04	2.88E+03	3.82E+04	2.66E+04	1.35E+03	7.33E+04	2.19E+04	7.60E+04	−1.26E+03
	STD	3.36E−03	7.46E+03	1.59E+03	3.93E+03	1.45E+03	9.92E+02	4.76E+03	4.54E+03	5.34E+03	2.88E+01
F2	AVG	2.15E+07	8.78E+08	1.87E+08	7.84E+08	2.51E+08	1.62E+08	2.29E+09	1.25E+08	1.73E+09	7.79E+07
	STD	1.15E+07	4.61E+08	6.23E+07	1.63E+08	3.31E+07	5.00E+07	8.20E+08	5.40E+07	6.41E+08	2.63E+07
F3	AVG	1.28E+10	3.15E+12	3.23E+11	3.78E+11	5.91E+10	6.20E+10	9.72E+19	4.93E+10	2.38E+14	3.61E+10
	STD	9.78E+09	7.39E+12	1.03E+12	5.04E+11	6.54E+09	1.30E+10	2.42E+20	9.04E+09	4.13E+14	1.22E+10
F4	AVG	6.21E+04	7.67E+04	9.91E+04	9.21E+04	7.09E+04	1.11E+05	9.72E+04	7.13E+04	8.81E+04	7.19E+04
	STD	1.22E+04	1.48E+04	1.95E+04	1.38E+04	5.22E+03	1.54E+04	1.20E+04	9.49E+03	8.58E+03	8.28E+03
F5	AVG	−1.00E+03	5.47E+03	5.07E+02	4.25E+03	1.34E+03	−5.08E+01	1.99E+04	2.88E+03	1.08E+04	−7.94E+02
	STD	1.67E−02	1.84E+03	2.10E+02	1.20E+03	2.10E+02	2.39E+02	2.66E+03	1.59E+03	1.57E+03	3.68E+01
F6	AVG	−8.00E+02	3.23E+03	−2.33E+02	2.11E+03	8.77E+02	−4.32E+02	8.82E+03	5.10E+02	7.11E+03	−6.62E+02
	STD	3.73E+01	1.12E+03	1.67E+02	6.19E+02	1.63E+02	1.03E+02	1.46E+03	4.14E+02	1.62E+03	5.76E+01
F7	AVG	−6.71E+02	−2.53E+02	2.13E+03	−4.99E+02	−5.47E+02	−4.57E+02	2.98E+04	−6.67E+02	8.40E+03	2.00E+02
	STD	2.51E+01	5.37E+02	4.88E+03	1.18E+02	9.24E+01	1.66E+02	3.06E+04	2.00E+01	2.01E+04	2.02E+03
F8	AVG	−6.79E+02	−6.79E+02	−6.79E+02	−6.79E+02	−6.79E+02	−6.79E+02	−6.79E+02	−6.79E+02	−6.79E+02	−6.79E+02
	STD	4.46E−02	4.43E−02	4.80E−02	4.09E−02	3.72E−02	3.60E−02	3.55E−02	3.25E−02	5.21E−02	3.64E−02
F9	AVG	−5.39E+02	−5.34E+02	−5.28E+02	−5.24E+02	−5.29E+02	−5.34E+02	−5.27E+02	−5.34E+02	−5.27E+02	−5.31E+02
	STD	5.32E+00	5.16E+00	3.55E+00	1.85E+00	4.17E+00	5.38E+00	1.72E+00	5.48E+00	2.81E+00	4.41E+00
F10	AVG	−4.64E+02	6.05E+03	8.60E+02	5.30E+03	2.91E+03	6.69E+02	1.27E+04	2.09E+03	1.04E+04	−1.90E+02
	STD	2.10E+01	1.50E+03	3.22E+02	1.09E+03	2.07E+02	2.73E+02	1.69E+03	5.46E+02	2.43E+03	9.12E+01
F11	AVG	−2.12E+02	2.25E+02	4.82E+02	4.25E+02	4.32E+02	1.84E+02	7.00E+02	2.27E+02	7.80E+02	2.58E+02
	STD	4.42E+01	8.44E+01	9.18E+01	5.73E+01	9.35E+01	7.65E+01	5.67E+01	6.50E+01	1.13E+02	5.99E+01
F12	AVG	7.23E+01	4.50E+02	7.92E+02	5.56E+02	5.68E+02	4.81E+02	8.82E+02	3.44E+02	8.90E+02	7.29E+02
	STD	8.60E+01	8.95E+01	1.33E+02	7.63E+01	9.95E+01	8.58E+01	6.66E+01	7.13E+01	9.66E+01	8.36E+01
F13	AVG	3.63E+02	7.19E+02	8.84E+02	6.71E+02	7.72E+02	6.94E+02	1.03E+03	5.20E+02	9.85E+02	1.00E+03
	STD	7.73E+01	9.76E+01	1.39E+02	6.90E+01	9.61E+01	1.00E+02	5.71E+01	7.87E+01	1.27E+02	7.69E+01
F14	AVG	5.97E+03	6.67E+03	1.11E+04	1.40E+04	1.31E+04	8.67E+03	1.49E+04	1.20E+04	1.38E+04	8.23E+03
	STD	1.15E+03	9.92E+02	1.19E+03	4.99E+02	8.40E+02	1.27E+03	3.82E+02	8.72E+02	6.60E+02	1.27E+03
F15	AVG	1.14E+04	8.51E+03	1.31E+04	1.53E+04	1.39E+04	1.12E+04	1.55E+04	1.32E+04	1.52E+04	1.16E+04
	STD	1.43E+03	1.23E+03	1.04E+03	4.33E+02	9.61E+02	1.18E+03	3.60E+02	1.06E+03	4.14E+02	1.34E+03
F16	AVG	2.03E+02	2.03E+02	2.03E+02	2.04E+02	2.03E+02	2.04E+02	2.04E+02	2.04E+02	2.04E+02	2.03E+02
	STD	5.21E−01	6.70E−01	4.38E−01	4.07E−01	5.63E−01	7.82E−01	3.19E−01	3.73E−01	4.27E−01	6.01E−01
F17	AVG	6.52E+02	1.08E+03	1.53E+03	1.47E+03	1.32E+03	1.30E+03	1.67E+03	1.30E+03	1.70E+03	1.61E+03
	STD	7.49E+01	1.18E+02	1.00E+02	9.69E+01	1.12E+02	9.74E+01	6.41E+01	9.56E+01	5.98E+01	7.62E+01
F18	AVG	8.80E+02	1.42E+03	1.61E+03	1.56E+03	1.44E+03	1.47E+03	1.77E+03	1.39E+03	1.78E+03	1.74E+03
	STD	8.31E+01	1.34E+02	1.32E+02	8.68E+01	1.29E+02	8.19E+01	5.30E+01	7.39E+01	4.66E+01	6.19E+01

(*continued on next page*)

Table 4 (*continued*).

F19	AVG	5.28E+02	1.56E+05	2.44E+03	1.03E+05	2.83E+04	1.07E+03	9.27E+05	2.14E+04	1.04E+06	5.94E+02
	STD	7.06E+00	1.12E+05	1.12E+03	7.22E+04	3.95E+03	2.87E+02	3.18E+05	1.81E+04	2.69E+05	1.41E+01
F20	AVG	6.23E+02	6.24E+02	6.25E+02	6.25E+02	6.25E+02	6.25E+02	6.25E+02	6.24E+02	6.25E+02	6.25E+02
	STD	1.06E+00	2.70E−01	2.26E−01	3.32E−01	2.64E−01	2.71E−01	1.95E−01	9.40E−01	1.65E−01	2.17E−01
F21	AVG	1.55E+03	4.67E+03	3.68E+03	4.85E+03	4.08E+03	3.24E+03	5.20E+03	4.57E+03	5.28E+03	1.92E+03
	STD	2.89E+02	1.62E+02	5.07E+02	1.62E+02	4.28E+01	4.04E+02	8.17E+01	1.50E+02	7.33E+01	4.22E+02
F22	AVG	7.23E+03	1.10E+04	1.45E+04	1.59E+04	1.58E+04	1.23E+04	1.68E+04	1.33E+04	1.65E+04	1.16E+04
	STD	1.11E+03	1.30E+03	1.05E+03	5.85E+02	7.82E+02	1.02E+03	5.28E+02	1.04E+03	7.70E+02	9.79E+02
F23	AVG	1.25E+04	1.25E+04	1.53E+04	1.68E+04	1.64E+04	1.43E+04	1.73E+04	1.44E+04	1.73E+04	1.47E+04
	STD	1.54E+03	1.35E+03	9.37E+02	4.51E+02	4.15E+02	1.19E+03	4.80E+02	1.27E+03	5.61E+02	1.48E+03
F24	AVG	1.36E+03	1.60E+03	1.42E+03	1.44E+03	1.57E+03	1.41E+03	1.67E+03	1.38E+03	1.69E+03	1.46E+03
	STD	1.43E+01	1.21E+02	1.46E+01	7.73E+00	3.90E+01	1.36E+01	1.17E+02	1.37E+01	1.31E+02	4.00E+01
F25	AVG	1.49E+03	1.72E+03	1.55E+03	1.56E+03	1.67E+03	1.54E+03	1.60E+03	1.50E+03	1.72E+03	1.58E+03
	STD	1.65E+01	3.93E+01	1.95E+01	7.91E+00	2.58E+01	1.33E+01	1.49E+01	1.35E+01	3.76E+01	3.95E+01
F26	AVG	1.63E+03	1.68E+03	1.65E+03	1.66E+03	1.60E+03	1.65E+03	1.57E+03	1.66E+03	1.69E+03	1.68E+03
	STD	4.49E+01	5.54E+01	1.05E+02	8.45E+01	1.28E+02	9.33E+01	8.92E+01	4.81E+01	5.54E+01	5.34E+01
F27	AVG	3.07E+03	3.94E+03	3.64E+03	3.71E+03	3.90E+03	3.49E+03	4.56E+03	3.32E+03	4.07E+03	3.85E+03
	STD	1.64E+02	1.97E+02	1.18E+02	6.61E+01	1.07E+02	1.17E+02	3.88E+02	1.54E+02	1.65E+02	1.52E+02
F28	AVG	3.29E+03	8.70E+03	1.04E+04	7.73E+03	9.01E+03	7.93E+03	1.06E+04	6.40E+03	1.08E+04	1.07E+04
	STD	2.01E+03	7.57E+02	1.53E+03	4.78E+02	5.38E+02	9.14E+02	7.03E+02	8.33E+02	6.91E+02	9.57E+02
		/	25	27	27	26	27	27	28	27	27

Table 5
P-Values of the Wilcoxon rank-sum test for CEC2013 functions (10 dim).

Function	AO	BOA	HHO	SCA	WOA	STOA	PSO	BH	AOA
F1	2.08E−11	2.08E−11	2.08E−11	2.08E−11	2.08E−11	2.08E−11	2.08E−11	2.08E−11	2.08E−11
F2	3.02E−11	3.02E−11	1.09E−10	3.02E−11	3.02E−11	4.08E−11	0.00557	3.02E−11	3.02E−11
F3	3.35E−08	3.02E−11	2.19E−08	3.69E−11	9.92E−11	3.35E−08	0.000856	1.33E−10	3.02E−11
F4	3.50E−09	2.87E−10	5.97E−09	8.35E−08	8.99E−11	1.19E−06	2.68E−06	7.60E−07	1.33E−10
F5	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F6	1.10E−08	3.02E−11	9.51E−06	2.92E−09	2.20E−07	9.83E−08	9.92E−11	3.02E−11	3.02E−11
F7	0.014412	1.21E−10	4.80E−07	0.003183	9.53E−07	**0.982307**	2.68E−06	0.045146	3.50E−09
F8	**0.958731**	**0.1809**	**0.437641**	**0.371077**	**0.283778**	**0.523297**	**0.911709**	**0.946956**	**0.911709**
F9	**0.063533**	1.41E−09	1.07E−07	9.76E−10	2.02E−08	2.00E−05	0.001597	3.83E−05	3.34E−11
F10	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F11	1.17E−09	3.02E−11	1.33E−10	3.02E−11	5.49E−11	6.12E−10	1.46E−10	4.50E−11	3.02E−11
F12	0.000213	3.02E−11	8.15E−11	8.99E−11	4.08E−11	0.00557	8.48E−09	2.23E−09	1.09E−10
F13	9.79E−05	7.39E−11	6.72E−10	8.29E−06	2.02E−08	**0.403538**	1.36E−07	2.49E−06	4.50E−11
F14	1.61E−06	3.02E−11	5.09E−08	3.02E−11	9.92E−11	3.82E−10	5.09E−08	3.02E−11	5.07E−10
F15	**0.610008**	1.33E−10	0.000141	1.46E−10	7.74E−06	5.19E−07	3.59E−05	9.79E−05	9.83E−08
F16	0.002157	5.07E−10	**0.559231**	1.55E−09	**0.258051**	3.50E−09	**0.610008**	7.60E−07	0.012732
F17	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	4.08E−11	0.000111	3.02E−11	3.02E−11
F18	4.08E−11	3.02E−11	3.69E−11	3.02E−11	4.50E−11	6.07E−11	1.10E−08	4.50E−11	3.02E−11
F19	3.34E−11	3.02E−11	3.02E−11	3.02E−11	3.69E−11	4.98E−11	4.98E−11	3.02E−11	3.02E−11
F20	0.000691	4.18E−09	1.07E−07	7.20E−05	1.49E−06	0.003339	0.00077	0.003034	1.41E−09
F21	2.49E−11	2.49E−11	4.28E−10	2.49E−11	7.35E−07	6.81E−07	4.69E−09	2.49E−11	2.49E−11
F22	6.36E−05	3.69E−11	1.03E−06	9.92E−11	4.18E−09	1.87E−07	4.80E−07	1.31E−08	1.20E−08
F23	0.000141	3.82E−10	1.07E−07	9.76E−10	1.31E−08	2.00E−06	1.87E−07	1.10E−08	2.03E−09
F24	7.66E−05	4.20E−10	6.01E−08	4.57E−09	5.97E−09	0.015014	1.86E−09	0.000903	1.10E−08
F25	0.000201	0.000253	1.69E−09	3.52E−07	2.03E−07	**0.065671**	8.35E−08	**0.176128**	3.82E−09
F26	**0.05012**	**0.695215**	1.43E−05	2.13E−05	**0.728265**	1.11E−06	0.001442	**0.482517**	2.77E−05
F27	0.006097	9.79E−05	4.42E−06	6.01E−08	2.20E−07	0.012212	**0.841801**	0.000132	3.20E−09
F28	0.00557	7.66E−05	1.01E−08	0.000141	7.77E−09	**0.610008**	2.00E−06	2.68E−06	3.69E−11

algorithm wins in all five single-peaked functions and has a large advantage in functions F2 and F4 shows the strong exploitation ability of the GOA.

In the multimodal functions in Fig. 8 we can see that in functions F6, F7, F10, F17, and F18, the GOA finds a good solution at the beginning and maintains a lead over the other algorithms. In function F8, most algorithms beat the GOA, but their difference in values is small. In functions F9, F11, F12, F13, F14, and F20, the GOA shows a huge advantage over the other algorithms, which shows the strong ability of the GOA to explore and escape from the local optimum. In function F15, although the GOA finally did not exceed PSO, the curve of the GOA continues to decline while PSO has converged, indicating that the GOA still has great potential in the late exploration stage of the algorithm, function F16 is the best example. Most of the algorithms in function F19 can find the optimal value quickly.

We can see in the combined functions in Fig. 9 that the GOA in functions F21, F24, F25, F27, and F28 converges faster and gets the best solution compared to the other algorithms. In functions F22 and F23, the convergence speed of the GOA is not the fastest in the early stage, but his convergence ability is very strong, and it shows a great advantage in the later stage. In function F26, the GOA shows poor results.

3.2.3. Comparison of algorithm time

To evaluate the usability of the GOA relative to other algorithms, we conduct time tests on all three dimensions. The average time, the best time and standard deviation of each algorithm running 30 times on the CEC2013 test set are given in the Tables 8–10 . It can be seen that the GOA algorithm is more effective in higher dimensions. For example, the average running time of WOA is 32.88 s on 28 functions of 10dim and 117.71 s on 50dim, while the average running time of the GOA algorithm is 28.27 on 28 functions of 10dim and 55.90 on 50dim s; most importantly, the GOA algorithm beats the WOA algorithm 27 times in the average fitness values of both 10 and

Table 6
P-Values of the Wilcoxon rank-sum test for CEC2013 functions (30 dim).

No.	AO	BOA	HHO	SCA	WOA	STOA	PSO	BH	AOA
F1	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F2	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	4.50E−11	3.02E−11	3.02E−11	3.02E−11
F3	3.69E−11	3.02E−11	2.37E−10	3.02E−11	3.02E−11	1.46E−10	3.02E−11	3.02E−11	3.02E−11
F4	1.43E−08	1.61E−10	4.74E−06	1.70E−08	3.02E−11	8.88E−06	5.60E−07	9.06E−08	7.39E−11
F5	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F6	3.02E−11	3.02E−11	3.47E−10	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F7	8.89E−10	3.02E−11	4.98E−11	6.12E−10	6.07E−11	**0.363222**	1.17E−09	3.02E−11	3.02E−11
F8	**0.818746**	**0.222573**	0.014412	**0.304177**	**0.074827**	**0.830255**	0.015638	**0.222573**	**0.297272**
F9	8.20E−07	1.33E−10	1.69E−09	6.07E−11	1.85E−08	0.039167	1.64E−05	1.10E−08	1.09E−10
F10	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F12	4.20E−10	3.02E−11	3.34E−11	1.96E−10	3.34E−11	5.97E−09	8.15E−11	4.08E−11	3.02E−11
F13	5.49E−11	3.02E−11	3.02E−11	3.69E−11	4.08E−11	4.18E−09	3.02E−11	3.02E−11	3.02E−11
F14	8.15E−11	3.02E−11	1.02E−05	3.02E−11	3.02E−11	3.02E−11	0.000201	3.02E−11	3.02E−11
F15	0.003183	3.02E−11	0.006097	3.02E−11	5.46E−09	2.44E−09	**0.217017**	7.12E−09	3.69E−11
F16	0.000178	1.78E−10	0.029205	1.55E−09	**0.085**	6.70E−11	**0.05012**	4.35E−05	2.67E−09
F17	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F18	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F19	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F20	1.25E−06	1.67E−07	4.13E−08	3.16E−04	8.07E−08	0.006352	1.60E−06	5.53E−05	6.71E−05
F21	3.02E−11	3.02E−11	1.58E−04	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F22	3.82E−10	3.02E−11	1.21E−10	3.02E−11	4.50E−11	3.34E−11	3.50E−09	3.02E−11	3.02E−11
F23	3.09E−06	3.02E−11	3.20E−09	3.02E−11	2.15E−10	5.53E−08	**0.318304**	6.07E−11	3.02E−11
F24	1.46E−10	3.02E−11	3.34E−11	3.34E−11	4.08E−11	9.53E−07	3.02E−11	3.02E−11	3.02E−11
F25	4.62E−10	6.07E−11	3.02E−11	3.02E−11	7.39E−11	0.000301	3.02E−11	3.02E−11	3.02E−11
F26	**0.096263**	0.006097	1.73E−07	3.39E−02	2.87E−10	0.035137	**0.077272**	0.001953	0.003183
F27	1.86E−09	3.02E−11	3.69E−11	4.08E−11	1.09E−10	6.01E−08	6.07E−11	4.50E−11	3.02E−11
F28	5.07E−10	3.02E−11	3.34E−11	5.57E−10	5.49E−11	3.35E−08	9.92E−11	1.33E−10	3.02E−11

50 dimensions. Although GAO is inferior to PSO, SCA, BH and STOA in terms of running speed, comprehensive consideration of running speed and obtaining optimal values shows that the GAO algorithm is trustworthy.

4. Engineering design problems

The GOA is applied to five constrained engineering design problems. These five engineering problems are speed reducer design, three-bar truss design, cantilever beam design, tubular column design, and welded beam design.

4.1. Speed reducer design

The reducer is widely used in modern machinery and is often used to reduce transmission between the original moving parts and the working machine. They are divided into gear reducers, worm gear reducers, etc. In order to reduce the weight of the reducer, seven variables of the reducer have to be optimized. These seven variables are the width of the face b (x_1), the number of tooth modules m (x_2), the number of pinions z (x_3), the length of the first shaft between bearings l1 (x_4), the length of the second shaft between bearings l2 (x_5), the diameter of the first shaft d1 (x_6) and the diameter of the second shaft d2 (x_7), as shown in Fig. 10.

The mathematical description of the problem is as follows.

$$\begin{aligned} f(X) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) \\ -1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right) \end{aligned} \tag{22}$$

Table 7
P-Values of the Wilcoxon rank-sum test for CEC2013 functions (50 dim).

No.	AO	BOA	HHO	SCA	WOA	STOA	PSO	BH	AOA
F1	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F2	3.02E−11	3.02E−11	9.92E−11	3.02E−11	3.02E−11	6.70E−11	3.02E−11	3.02E−11	3.02E−11
F3	4.50E−11	3.02E−11	4.18E−09	3.02E−11	3.02E−11	6.07E−11	3.02E−11	3.02E−11	3.02E−11
F4	4.08E−11	1.46E−10	6.55E−04	1.69E−09	1.29E−09	0.001767	0.000201	0.00062	3.20E−09
F5	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F6	3.02E−11	3.02E−11	2.61E−10	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F7	4.98E−11	3.02E−11	3.69E−11	3.34E−11	3.02E−11	**0.673495**	1.96E−10	3.82E−10	3.02E−11
F8	0.043584	**0.520145**	**0.270705**	**0.501144**	**0.1809**	**0.935192**	**0.096263**	**0.304177**	**0.706171**
F9	0.00077	4.08E−11	1.47E−07	3.02E−11	1.29E−09	0.001174	0.00137	3.50E−09	1.96E−10
F10	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F12	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	8.99E−11	3.69E−11	3.02E−11	3.02E−11
F13	4.50E−11	3.02E−11	3.02E−11	3.69E−11	3.02E−11	9.26E−09	3.02E−11	3.02E−11	3.02E−11
F14	2.92E−09	3.02E−11	7.69E−08	3.02E−11	3.02E−11	3.02E−11	0.019112	3.02E−11	3.02E−11
F15	**0.264326**	3.02E−11	**0.923442**	3.34E−11	2.32E−06	2.00E−06	1.01E−08	2.67E−09	3.02E−11
F16	**0.141278**	5.57E−10	**0.684323**	4.57E−09	**0.695215**	1.69E−09	**0.072446**	**0.355472**	1.43E−08
F17	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.69E−11	3.02E−11	3.02E−11
F18	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F19	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F20	2.80E−11	3.02E−11	2.98E−11	7.39E−11	2.26E−11	1.62E−05	1.21E−10	6.07E−11	3.02E−11
F21	3.02E−11	3.02E−11	1.43E−05	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11
F22	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	3.69E−11	3.02E−11	3.02E−11
F23	5.27E−05	3.02E−11	2.88E−06	3.02E−11	2.67E−09	2.28E−05	**0.717189**	3.02E−11	3.02E−11
F24	3.69E−11	3.02E−11	3.02E−11	3.02E−11	3.02E−11	2.83E−08	3.02E−11	3.02E−11	3.02E−11
F25	4.50E−11	3.02E−11	3.34E−11	3.02E−11	4.08E−11	0.000125	3.02E−11	3.02E−11	3.02E−11
F26	8.20E−07	0.00557	5.07E−10	4.64E−05	6.74E−06	1.07E−07	7.12E−09	**0.053685**	5.60E−07
F27	2.15E−10	3.02E−11	3.02E−11	3.02E−11	3.02E−11	1.39E−06	3.02E−11	3.02E−11	3.02E−11
F28	8.99E−11	3.02E−11	3.02E−11	3.34E−11	3.02E−11	7.69E−08	3.02E−11	3.02E−11	3.02E−11

Table 8
Time comparison results for GOA vs. other algorithms (dim = 10).

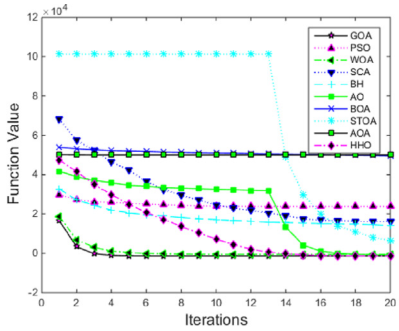
Metric	GOA	PSO	WOA	SCA	BH	AO	BOA	STOA	AOA	HHO
AVG	27.96	17.42	34.63	17.35	16.67	47.62	31.36	16.85	46.98	51.40
BEST	27.72	16.82	32.95	16.54	16.60	47.39	31.05	16.76	46.79	49.67
STD	0.23	0.42	1.43	0.76	0.04	0.12	0.28	0.05	0.48	1.33

Table 9
Time comparison results for GOA vs. other algorithms (dim = 30).

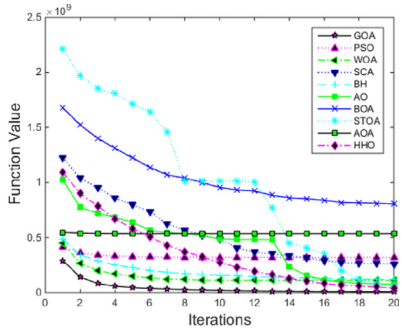
Metric	GOA	AO	WOA	AOA	STOA	BH	HHO	PSO	SCA	BOA
AVG	40.69	75.58	73.82	115.21	30.81	28.37	76.57	29.39	29.54	53.80
BEST	39.84	70.95	73.69	114.95	30.73	28.30	75.91	29.21	29.50	53.18
STD	0.79	5.25	0.07	0.16	0.05	0.04	0.30	0.09	0.03	0.60

Table 10
Time comparison results for GOA vs. other algorithms (dim = 50).

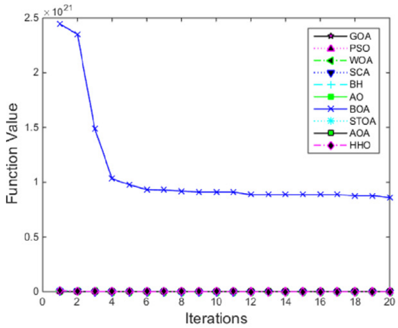
Metric	GOA	AO	WOA	AOA	STOA	BH	HHO	PSO	SCA	BOA
AVG	55.34	99.44	117.71	187.40	48.10	42.69	114.33	45.00	46.81	78.97
BEST	55.08	97.35	116.70	184.97	46.36	41.78	107.27	44.15	45.02	78.29
STD	0.14	1.68	0.66	1.88	1.44	0.92	6.13	0.64	1.71	0.53



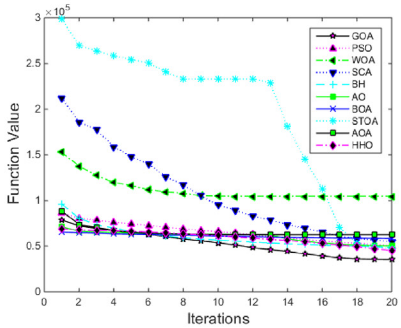
(a) F1



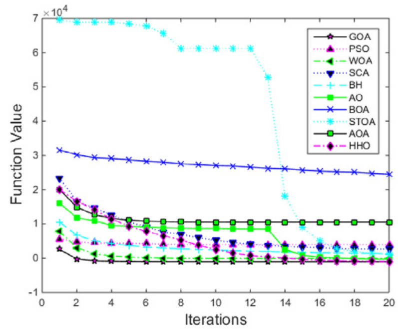
(b) F2



(c) F3



(d) F4



(e) F5

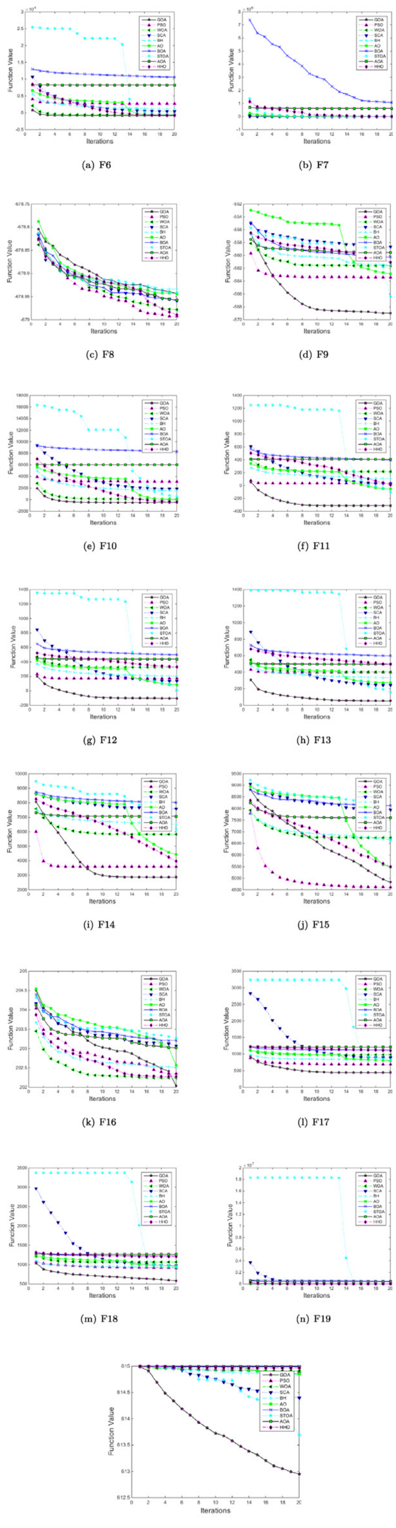
Fig. 7. Single-peak function iteration curve.

The above equation obeys the following constraint.

$$g_1(X) = \frac{27}{x_1 x_2^2 x_3} - 1 \le 0$$

$$g_2(X) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0$$

$$g_3(X) = \frac{1.93 x_4^3}{x_2 x_6^4 x_3} - 1 \le 0$$



(a) F6 (b) F7

(c) F8 (d) F9

(e) F10 (f) F11

(g) F12 (h) F13

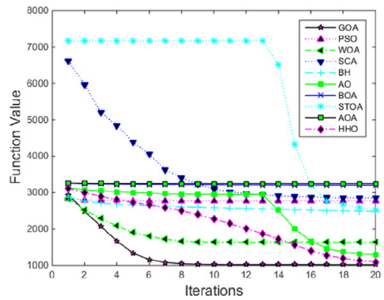
(i) F14 (j) F15

(k) F16 (l) F17

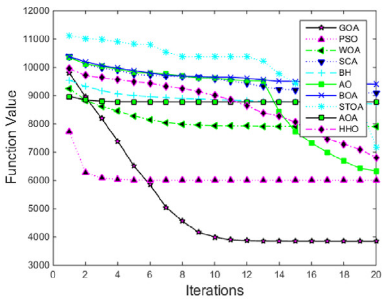
(m) F18 (n) F19

(o) F20

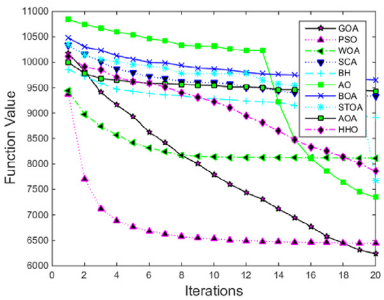
Fig. 8. Multimode function iteration curve.



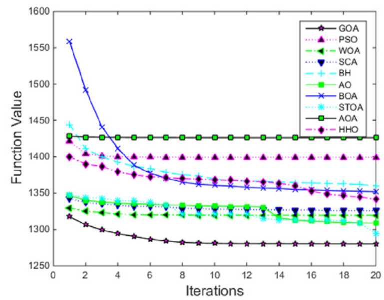
(a) F21



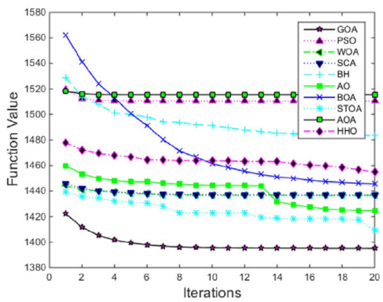
(b) F22



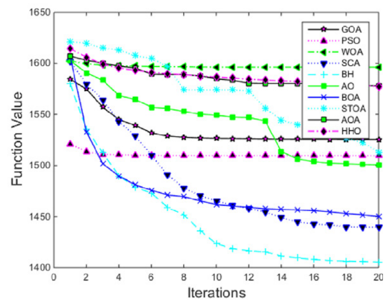
(c) F23



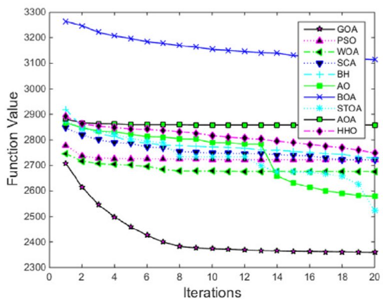
(d) F24



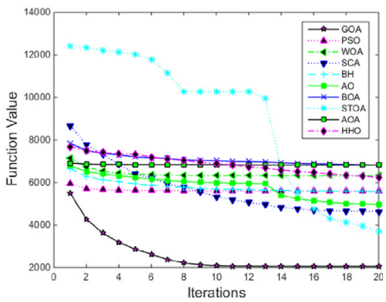
(e) F25



(f) F26



(g) F27



(h) F28

Fig. 9. Combinatorial function iteration curve.

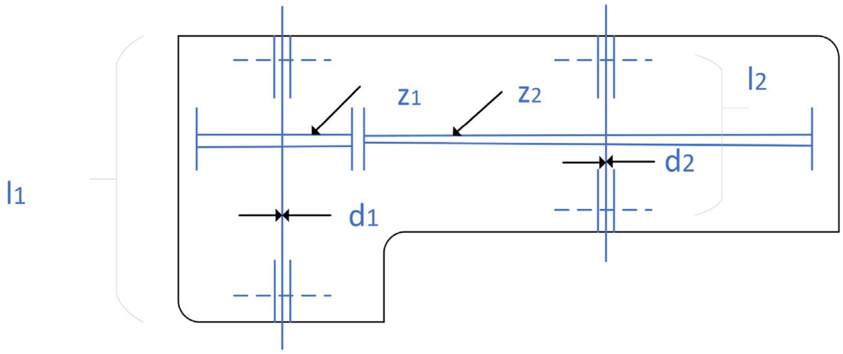


Fig. 10. Speed reducer.

$$
\begin{aligned}
g_4(X) &= \frac{1.93x_5^3}{x_2x_7^4x_3} - 1 \le 0 \\
g_5(X) &= \frac{\sqrt{(745x_4/x_2x_3)^2 + 16.9 \times 10^6}}{110x_6^3} - 1 \le 0 \\
g_6(X) &= \frac{\sqrt{(745x_5/x_2x_3)^2 + 157.5 \times 10^6}}{85x_7^3} - 1 \le 0 \\
g_7(X) &= \frac{x_2x_3}{40} - 1 \le 0 \\
g_8(X) &= \frac{5x_2}{x_1} - 1 \le 0 \\
g_9(X) &= \frac{x_1}{12x_2} - 1 \le 0 \\
g_{10}(X) &= \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0 \\
g_{11}(X) &= \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0
\end{aligned} \tag{23}
$$

The range of variables in the above equation is shown below.

$$
\begin{aligned}
&2.6 \le x_1 \le 3.6 \\
&0.7 \le x_2 \le 0.8 \\
&17 \le x_3 \le 28 \\
&7.3 \le x_4 \le 8.3 \\
&7.3 \le x_5 \le 8.3 \\
&2.9 \le x_6 \le 3.9 \\
&5 \le x_7 \le 5.5
\end{aligned} \tag{24}
$$

We give the optimal values of the GOA and other optimization algorithms in Table 11, and from the data in the table we can compare and find that the results of the GOA are better than the other algorithms.

4.2. Three-bar truss design problem

The prototype of this engineering optimization problem is a truss structure with three bars, and the goal is to reduce the weight of this structure by optimizing two cross-sectional areas A1(x_1) and A2(x_2). As shown in Fig. 11.

Table 11
Results of the comparative algorithms for solving the speed reducer design problem.

Algorithm	x_1	x_2	x_3	x_4	x_5	x_6	x_7	f_{cost}
AAO [14]	3.4990	0.6999	17.0000	7.3000	7.8000	3.3502	5.2872	2996.7830
CS [19]	3.5015	0.7000	17.0000	7.6050	7.8181	3.3520	5.2875	3000.9810
AFA [7]	3.5000	0.7000	17.0000	7.3025	7.8001	3.3502	5.2867	2996.3727
AOA [1]	3.5038	0.7000	17.0000	7.3000	7.7293	3.3565	5.2867	2997.9157
ABC [2]	3.5000	0.7000	17.0000	7.3000	7.8000	3.3502	5.2867	2997.0584
SNS [8]	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2867	2994.4711
PSO-DE [36]	3.5000	0.7000	17.0000	7.3000	7.8000	3.3502	5.2867	2996.3481
WSA [6]	3.5000	0.7000	17.0000	7.3000	7.8000	3.3502	5.2867	2996.3482
WCA [15]	3.5000	0.7000	17.0000	7.3000	7.7153	3.3502	5.2867	2994.4711
GOA	3.5000	0.7000	17.0000	7.3000	7.7153	3.3505	5.2867	2994.4245

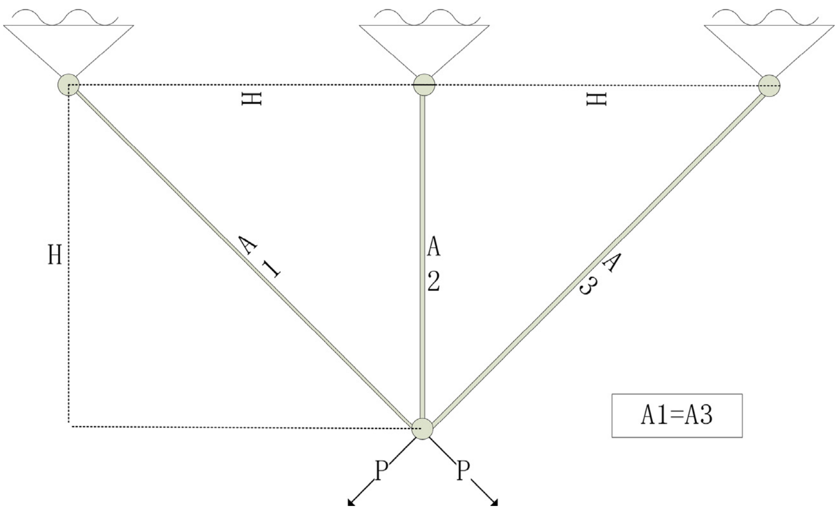


Fig. 11. Three-Bar.

The mathematical representation of the Three-Bar truss is as follows.

$$f(X) = \left(2\sqrt{2}x_1 + x_2\right) \times l \tag{25}$$

The above equation obeys the following constraint.

$$\begin{aligned} g_1(X) &= \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0, \\ g_2(X) &= \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 \\ g_3(X) &= \frac{1}{\sqrt{2}x_2 + x_1} P - \sigma \leq 0 \\ l &= 100 \quad \text{cm} \\ P &= 2 \ \text{kN/cm}^3 \\ \sigma &= 2 \ \text{kN/cm}^3 \end{aligned} \tag{26}$$

Table 12
Results of the comparative algorithms for solving the tree-bar truss design problem.

Algorithm	x_1	x_2	f_{cost}
WSA [33]	0.788683	0.408227	263.895843
MVO [45]	0.788603	0.408453	263.895850
CS [19]	0.788670	0.409020	263.971600
PSO-DE [36]	0.788675	0.408248	263.895843
MBA [53]	0.788565	0.408560	263.895852
AOA [1]	0.793690	0.394260	263.915400
GOA [54]	0.788898	0.407620	263.895881
MFO [41]	0.788245	0.409467	263.895980
ALO [40]	0.788663	0.408283	263.895843
GOA	0.788675	0.408249	263.895843

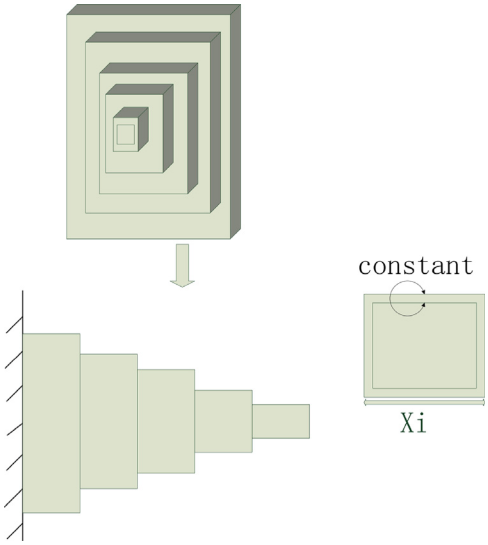


Fig. 12. Cantilever beam.

The range of variables in the above equation is shown below.

$$\begin{aligned} &0 \le x_1 \le 1 \\ &0 \le x_2 \le 1 \end{aligned} \tag{27}$$

The optimal solution obtained by the GOA is compared with the optimal solutions obtained in other papers, and it is obtained that the GOA has good results in the solution of the Tree-Bar truss design problem. The specific data are shown in Table 12.

4.3. Cantilever beam design problem

This engineering optimization problem is the weight optimization of a cantilever beam arm. The cantilever arm in this case is composed of five hollow squares, and the lengths of these five squares are the design parameters for this problem, as shown in Fig. 12.

The mathematical representation of the cantilever beam arm design is as follows.

$$f(X) = 0.0624\,(x_1 + x_2 + x_3 + x_4 + x_5) \tag{28}$$

Table 13
Results of the comparative algorithms for solving the cantilever beam design problem.

Algorithm	x_1	x_2	x_3	x_4	x_5	f_{cost}
MFO [41]	5.98487	5.31673	4.49733	3.51362	2.16162	1.33999
CS [19]	6.00890	5.30490	4.50230	3.50770	2.15040	1.33999
GCA [10]	6.01000	5.30000	4.49000	3.49000	2.15000	1.34000
SOS [9]	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996
GOA [54]	6.01167	5.31297	4.48307	3.50279	2.16333	1.33996
ALO [40]	6.01812	5.31142	4.48836	3.49751	2.15833	1.33995
SNS [8]	6.01545	5.31066	4.48800	3.50528	2.15428	1.33995
GOA	6.02076	5.30592	4.49171	3.50272	2.15257	1.33995

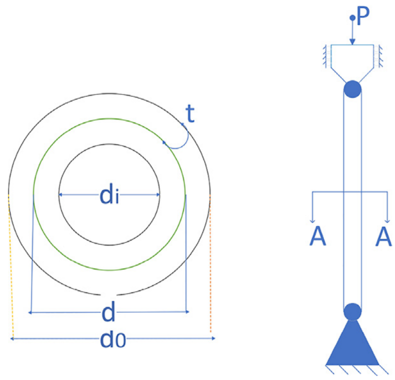


Fig. 13. Tubular column.

The above equation obeys the following constraint.

$$g(X) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \tag{29}$$

where the range of variables is as follows.

$$0.01 \leq x_i \leq 100, i = 1, \ldots, 5 \tag{30}$$

We compare the optimal solutions obtained by the GOA with those obtained in other articles and get that the GOA has good results in the solution of the cantilever beam design problem. The specific data are shown in Table 13.

4.4. Tubular column design

This is an engineering optimization problem to design a uniform tubular column at minimum cost and make it able to withstand the pressure. As shown in Fig. 13, the average diameter of the column d (x_1) and the thickness of the tube t (x_2) will be used as optimization variables for this problem. The object has a yield stress of 500 kgf/cm^2, a modulus of elasticity of 0.85 × 106 kgf/cm^2, and a density of 0.0025 kgf/cm^3. The mathematical representation of this engineering optimization is as follows.

$$f(X) = 9.8x_1x_2 + 2x_1 \tag{31}$$

Table 14
Results of the comparative algorithms for solving the tubular column design.

Algorithm	x_1	x_2	f_{cost}
CS [19]	5.4514	0.2920	26.5322
SNS [8]	5.4512	0.2920	26.4995
ISA [18]	5.4512	0.2920	26.5313
AOS [5]	5.4512	0.2920	26.5314
GOA	5.4522	0.2916	26.4864

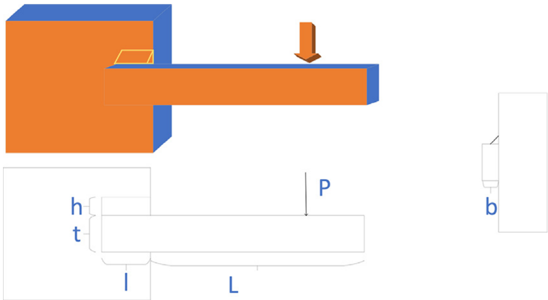


Fig. 14. Welded beam.

In this problem, there are also constraints, such as the stresses contained in the columns, as follows:

$$\begin{aligned} g_1(X) &= \frac{P}{\pi x_1 x_2 \sigma_y} - 1 \le 0 \\ g_2(X) &= \frac{8PL^2}{\pi^3 E x_1 x_2 \left(x_1^2 + x_2^2\right)} - 1 \le 0 \\ g_3(X) &= \frac{2.0}{x_1} - 1 \le 0 \\ g_4(X) &= \frac{x_1}{14} - 1 \le 0 \\ g_5(X) &= \frac{0.2}{x_2} - 1 \le 0 \\ g_6(X) &= \frac{x_2}{8} - 1 \le 0 \end{aligned} \tag{32}$$

where the range of variables is as follows.

$$\begin{aligned} 2 \le x_1 \le 14 \\ 0.2 \le x_2 \le 0.8 \end{aligned} \tag{33}$$

We compare the optimal solution obtained by the GOA with the optimal solution obtained in other papers and get that the GOA has good results in tubular column design problem-solving. The specific data are shown in Table 14.

4.5. Welded beam design

This is an engineering optimization problem for designing a welded beam at minimum cost. This problem has four optimizable variables and seven constraints, as shown in Fig. 14. The mathematical representation of this engineering optimization is as follows.

$$f(X) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2) \tag{34}$$

The seven constraints for the engineering optimization problem of the welded beam are as follows.

$$\begin{aligned} g_1(X) &= \tau(X) - \tau_{\max} \leq 0 \\ g_2(X) &= \sigma(X) - \sigma_{\max} \leq 0 \\ g_3(X) &= \delta(X) - \delta_{\max} \leq 0 \\ g_4(X) &= x_1 - x_4 \leq 0 \\ g_5(X) &= P - P_c(X) \leq 0 \\ g_6(X) &= 0.125 - x_1 \leq 0 \\ g_7(X) &= 1.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \leq 0 \end{aligned} \tag{35}$$

where x_1, x_2, x_3 and x_4 denote the weld thickness h, height l, length t and reinforcement thickness b, respectively.

$$\begin{aligned} \tau(X) &= \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \\ \tau' &= \frac{P}{\sqrt{2}x_1x_2} \\ \tau'' &= \frac{MR}{J} \\ M &= P\left(L + \frac{x_2}{2}\right) \\ R &= \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} \\ J &= 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \\ P &= 6000 \text{ lb} \\ \delta &= 14\text{in} \\ \delta(\vec{X}) &= \frac{6PL}{x_4x_3^2} \\ \delta(\vec{X}) &= \frac{6PL^3}{Ex_3^2x_4} \\ & \frac{4.013E\sqrt{x_3^2x_4^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \\ E &= 30 \times 10^6 \text{ psi} \\ G &= 12 \times 10^6 \text{ psi} \\ \tau_{\max} &= 13,600 \text{ psi} \\ \sigma_{\max} &= 30,000 \text{ psi} \end{aligned} \tag{36}$$

where the range of variables is as follows.

$$\begin{aligned} 0.1 &\leq x_1 \leq 2 \\ 0.1 &\leq x_2 \leq 10 \\ 0.1 &\leq x_3 \leq 10 \\ 0.1 &\leq x_4 \leq 2 \end{aligned} \tag{37}$$

The results of the GOA for the optimization of welded beams are compared with those in other articles in Table 15, which shows the superiority of the GOA for solving this problem.

Table 15
Results of the comparative algorithms for solving the welded beam design problem.

Algorithm	x_1	x_2	x_3	x_4	f_{cost}
HHO [25]	0.2040	3.5311	9.0275	0.2061	1.7320
WOA [44]	0.2054	3.4843	9.0374	0.2063	1.7305
CDE [29]	0.2031	3.5430	9.0335	0.2062	1.7335
GA [12]	0.2088	3.4205	8.9975	0.2100	1.7483
WCA [15]	0.2057	3.4705	9.0366	0.2057	1.7249
SSA [43]	0.2057	3.4714	9.0366	0.2057	1.7249
WSA [33]	0.2057	3.4705	9.0366	0.2057	1.7249
ABC [2]	0.2057	3.4705	9.0366	0.2057	1.7249
EO [17]	0.2057	3.4705	9.0366	0.2057	1.7249
MFO [41]	0.2057	3.4703	9.0364	0.2057	1.7245
CPSO [24]	0.2024	3.5442	9.0482	0.2057	1.7280
MVO [45]	0.2055	3.4732	9.0445	0.2057	1.7265
GOA	0.2057	3.4705	9.0366	0.2057	1.7249

5. Conclusion

In this paper, we propose a novel metaheuristic optimization algorithm called the gannet optimization algorithm. The GOA is inspired by the predation behavior of gannets in nature. By mathematizing the gannet's U-shaped and V-shaped dive patterns for effective exploration of the search space, exploitation with the help of sudden rotation and random wandering inspiration. The design of the memory matrix improves the convergence speed of the algorithm. The feasibility of the proposed algorithm is tested by 28 benchmark functions, and it can be seen from the statistical results that the proposed GOA is a very competitive meta-heuristic algorithm, outperforming other comparison algorithms in all three dimensions. The iterative curve also more intuitively shows the powerful ability of GOA to explore and escape from local optima. As the dimension increases, the GOA algorithm also has excellent advantages in running time, so GOA has a good processing ability for large-dimensional problems. We also apply the GOA to five engineering design problems, and the experimental results show that it outperforms many existing algorithms on the selected evaluation criteria. The limitation of the GOA is that there are multiple parameters, whether better parameter settings can be found or parameter adaptation will be the direction of our future improvement. Adding binary and multi-objective versions in future research will be a valuable direction to solving other practical optimization problems.

CRediT authorship contribution statement

Jeng-Shyang Pan: Conceptualization, Methodology, Project administration. **Li-Gang Zhang:** Data curation, Writing – original draft, Writing – review & editing. **Ruo-Bin Wang:** Visualization, Investigation. **Václav Snášel:** Supervision, Formal analysis. **Shu-Chuan Chu:** Software, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding statement

No funding support in this article.

References

[1] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Comput. Methods Appl. Mech. Engrg. 376 (2021) 113609.
[2] B. Akay, D. Karaboga, Artificial bee colony algorithm for large-scale problems and engineering design optimization, J. Intell. Manuf. 23 (4) (2012) 1001–1014.
[3] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, Soft Comput. 23 (3) (2019) 715–734.

[4] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, 2007, pp. 4661–4667.
[5] M. Azizi, S. Talatahari, A. Giaralis, Optimization of engineering design problems using atomic orbital search algorithm, IEEE Access 9 (2021) 102497–102519.
[6] A. Baykasoğlu, Ş. Akpinar, Weighted superposition attraction (WSA): A swarm intelligence algorithm for optimization problems–part 1: Unconstrained optimization, Appl. Soft Comput. 56 (2017) 520–540.
[7] A. Baykasoğlu, F.B. Ozsoydan, Adaptive firefly algorithm with chaos for mechanical design optimization problems, Appl. Soft Comput. 36 (2015) 152–164.
[8] H. Bayzidi, S. Talatahari, M. Saraee, C.-P. Lamarche, Social network search for solving engineering optimization problems, Comput. Intell. Neurosci. 2021 (2021).
[9] M.-Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112.
[10] H. Chickermane, H.C. Gea, Structural optimization using a new local approximation method, Internat. J. Numer. Methods Engrg. 39 (5) (1996) 829–846.
[11] S.-C. Chu, P.-W. Tsai, J.-S. Pan, Cat swarm optimization, in: Pacific Rim International Conference on Artificial Intelligence, Springer, 2006, pp. 854–858.
[12] C.A.C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2) (2000) 113–127.
[13] A. Colorni, M. Dorigo, V. Maniezzo, et al., Distributed optimization by ant colonies, in: Proceedings of the First European Conference on Artificial Life, Paris, France, Vol. 142, 1991, pp. 134–142.
[14] J.M. Czerniak, H. Zarzycki, D. Ewald, AAO as a new strategy in modeling and simulation of constructional problems optimization, Simul. Model. Pract. Theory 76 (2017) 22–33.
[15] H. Eskandar, A. Sadollah, A. Bahreininejad, M. Hamdi, Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems, Comput. Struct. 110 (2012) 151–166.
[16] A.E. Ezugwu, A.K. Shukla, R. Nath, A.A. Akinyelu, J.O. Agushaka, H. Chiroma, P.K. Muhuri, Metaheuristics: a comprehensive overview and classification along with bibliometric analysis, Artif. Intell. Rev. 54 (6) (2021) 4237–4316.
[17] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: A novel optimization algorithm, Knowl.-Based Syst. 191 (2020) 105190.
[18] A.H. Gandomi, D.A. Roke, Engineering optimization using interior search algorithm, in: 2014 IEEE Symposium on Swarm Intelligence, IEEE, 2014, pp. 1–7.
[19] A.H. Gandomi, X.-S. Yang, A.H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Eng. Comput. 29 (1) (2013) 17–35.
[20] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2) (2001) 60–68.
[21] G.F. Gomes, S.S. da Cunha, A.C. Ancelotti, A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates, Eng. Comput. 35 (2) (2019) 619–626.
[22] F.A. Hashim, E.H. Houssein, K. Hussain, M.S. Mabrouk, W. Al-Atabany, Honey badger algorithm: New metaheuristic algorithm for solving optimization problems, Math. Comput. Simulation 192 (2022) 84–110.
[23] F.A. Hashim, K. Hussain, E.H. Houssein, M.S. Mabrouk, W. Al-Atabany, Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems, Appl. Intell. 51 (3) (2021) 1531–1551.
[24] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (1) (2007) 89–99.
[25] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.
[26] J.H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, MIT Press, 1992.
[27] G. Hu, J. Zhong, B. Du, G. Wei, An enhanced hybrid arithmetic optimization algorithm for engineering applications, Comput. Methods Appl. Mech. Engrg. 394 (2022) 114901.
[28] G. Hu, X. Zhu, G. Wei, C.-T. Chang, An improved marine predators algorithm for shape optimization of developable ball surfaces, Eng. Appl. Artif. Intell. 105 (2021) 104417.
[29] F.-z. Huang, L. Wang, Q. He, An effective co-evolutionary differential evolution for constrained optimization, Appl. Math. Comput. 186 (1) (2007) 340–356.
[30] D. Karaboga, et al., An Idea Based on Honey Bee Swarm for Numerical Optimization, Tech. rep., Technical report-tr06, Erciyes University, Engineering Faculty, Computer …, 2005.
[31] A. Kaveh, T. Bakhshpoori, Metaheuristics: Outlines, MATLAB Codes and Examples, Springer, 2019.
[32] A. Kaveh, A. Dadras, A novel meta-heuristic optimization algorithm: thermal exchange optimization, Adv. Eng. Softw. 110 (2017) 69–84.
[33] A. Kaveh, A.D. Eslamlou, Water strider algorithm: A new metaheuristic and applications, in: Structures, Vol. 25, Elsevier, 2020, pp. 520–541.
[34] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, IEEE, 1995, pp. 1942–1948.
[35] M.S. Kiran, TSA: Tree-seed algorithm for continuous optimization, Expert Syst. Appl. 42 (19) (2015) 6686–6698.
[36] H. Liu, Z. Cai, Y. Wang, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, Appl. Soft Comput. 10 (2) (2010) 629–640.

[37] G.E. Machovsky-Capuska, Hunting Between the Air and the Water: the Australasian Gannet (Morus Serrator): a Thesis Presented in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy in Ecology at Massey University, Auckland, New Zealand (Ph.D. thesis), Massey University, 2012.
[38] A.R. Mehrabian, C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, Ecol. Inform. 1 (4) (2006) 355–366.
[39] Z. Meng, J.-S. Pan, Quasi-affine transformation evolutionary (QUATRE) algorithm: A parameter-reduced differential evolution algorithm for optimization problems, in: 2016 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2016, pp. 4082–4089.
[40] S. Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) 80–98.
[41] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.
[42] S. Mirjalili, SCA: a sine cosine algorithm for solving optimization problems, Knowl.-Based Syst. 96 (2016) 120–133.
[43] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.
[44] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[45] S. Mirjalili, S.M. Mirjalili, A. Hatamlou, Multi-verse optimizer: a nature-inspired algorithm for global optimization, Neural Comput. Appl. 27 (2) (2016) 495–513.
[46] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[47] N. Moosavian, B.K. Roodsari, Soccer league competition algorithm: A novel meta-heuristic algorithm for optimal design of water distribution networks, Swarm Evol. Comput. 17 (2014) 14–24.
[48] J.-S. Pan, N. Liu, S.-C. Chu, A hybrid differential evolution algorithm and its application in unmanned combat aerial vehicle path planning, IEEE Access 8 (2020) 17691–17712.
[49] J.-S. Pan, P.-W. Tsai, Y.-B. Liao, Fish migration optimization based on the fishy biology, in: 2010 Fourth International Conference on Genetic and Evolutionary Computing, IEEE, 2010, pp. 783–786.
[50] K. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization, Springer Science & Business Media, 2006.
[51] R.V. Rao, V.J. Savsani, D. Vakharia, Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput. Aided Des. 43 (3) (2011) 303–315.
[52] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inform. Sci. 179 (13) (2009) 2232–2248.
[53] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems, Appl. Soft Comput. 13 (5) (2013) 2592–2612.
[54] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: theory and application, Adv. Eng. Softw. 105 (2017) 30–47.
[55] H. Shareef, A.A. Ibrahim, A.H. Mutlag, Lightning search algorithm, Appl. Soft Comput. 36 (2015) 315–333.
[56] E. Tanyildizi, G. Demir, Golden sine algorithm: A novel math-inspired algorithm, Adv. Electr. Comput. Eng. 17 (2) (2017) 71–78.
[57] G.-G. Wang, S. Deb, Z. Cui, Monarch butterfly optimization, Neural Comput. Appl. 31 (7) (2019) 1995–2014.
[58] X. Wang, J.-S. Pan, S.-C. Chu, A parallel multi-verse optimizer for application in multilevel image segmentation, IEEE Access 8 (2020) 32018–32030.
[59] D. Whitley, A genetic algorithm tutorial, Stat. Comput. 4 (2) (1994) 65–85.
[60] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2010.
[61] L.-G. Zhang, F. Fan, S.-C. Chu, A. Garg, J.-S. Pan, Hybrid strategy of multiple optimization algorithms applied to 3-d terrain node coverage of wireless sensor network, Wirel. Commun. Mob. Comput. 2021 (2021).