Soft Computing
https://doi.org/10.1007/s00500-019-03949-w

METHODOLOGIES AND APPLICATION

A novel meta-heuristic optimization method based on golden ratio in nature

Amin Foroughi Nematollahi[1] · Abolfazl Rahiminejad[2] · Behrooz Vahidi[3]

Abstract

A novel parameter-free meta-heuristic optimization algorithm known as the golden ratio optimization method (GROM) is proposed. The proposed algorithm is inspired by the golden ratio of plant and animal growth which is formulated by the well-known mathematician Fibonacci. He introduced a series of numbers in which a number (except the first two numbers) is equal to the sum of the two previous numbers. In this series, the ratio of two consecutive numbers is almost the same for all the numbers and is known as golden ratio. This ratio can be extensively found in nature such as snail lacquer part and foliage growth of trees. The proposed approach employed this golden ratio to update the solutions in an optimization algorithm. In the proposed method, the solutions are updated in two different phases to achieve the global best answer. There is no need for any parameter tuning, and the implementation of the proposed method is very simple. In order to evaluate the proposed method, 29 well-known benchmark test functions and also 5 classical engineering optimization problems including 4 mechanical engineering problems and 1 electrical engineering problem are employed. Using several test functions, the performance of the proposed method in solving different problems including discrete, continuous, high dimension, and high constraints problems is testified. The results of the proposed method are compared with those of 11 well-regarded state-of-the-art optimization algorithms. The comparisons are made from different aspects such as the final obtained answer, the speed and behavior of convergence, and CPU time consumption. Superiority of the purposed method from different points of views can be concluded by means of comparisons.

Keywords Meta-heuristic · Golden ratio optimization method · Optimization algorithm · Constrained optimization · Optimization

1 Introduction

In actual problems, it is important to obtain the desired variables in the best way. This refers to optimization problems which need optimization methods as solutions. With regard to the non-convex and complicated nature of these problems, sophisticated methods are needed (Blum et al. 2011; BoussaïD et al. 2013; Gogna and Tayal 2013; Nematollahi et al. 2017).

Numerical methods are the early methods which obtain decision variables by finding the point at which the derivative is zero. However, lots of complicated mathematical computations are needed for these kinds of methods. Moreover, implementation of these methods for the non-convex, nonlinear, highly constrained and high variables problems is very complicated. Furthermore, these methods may be stuck in local optimum point. In summary, need for initial guess, additional mathematical calculations,

Communicated by V. Loia.

✉ Behrooz Vahidi
vahidi@aut.ac.ir

Amin Foroughi Nematollahi
amin.forooghi@aut.ac.ir

Abolfazl Rahiminejad
arahiminezhad@aut.ac.ir

1 Department of Electrical Engineering, Amirkabir University of Technology, Tehran 1591634311, Iran

2 Department of Electrical and Computer Science, Esfarayen University of Technology, Esfarayen, North Khorasan 9661998195, Iran

3 Department of Electrical Engineering, Amirkabir University of Technology, Tehran 15916343111, Iran

Published online: 03 April 2019

divergent issue, complicated implementation, difficult convergence for discrete types of optimization problem, etc., are some of drawbacks of these methods (Miettinen and Preface By-Neittaanmaki 1999; Venkataraman 2009; Statnikov and Matusov 2012; Mirjalili et al. 2018).

A bunch of optimization methods known as meta-heuristics algorithms have been developed in recent decades to overcome the difficulties of numerical methods (Mirjalili et al. 2014; Dosoglu et al. 2018; Lara et al. 2018). These methods, which are population based, decrease the possibility of being stuck in local optimum points by means of random operators. Nowadays, these methods are widely used in solving engineering problems for their simple implementation, simple mathematical operators, and low possibility of being stuck in local optimum points.

Meta-heuristic methods are classified into two main categorizes including single solution-based methods and population-based methods (Kirkpatrick et al. 1983). In the single solution-based methods, algorithms perform the search by single search agents, while the population-based methods perform the search using a set of search agents. In the population-based methods, each solution updates its position based on individual and social information. The combination of this information decreases the possibility of being stuck in local optimum points. Moreover, numerous solutions could easily search almost the entire search space; thus, better final results can be achieved in comparison to single solution-based methods.

The population-based optimization methods can be classified into 4 different classes from the inspiration point of view.

1. Evolutionary algorithms that are inspired by nature. In these kinds of methods, the solutions are produced randomly and gradually evolve. The global best answer would be obtained in the last iteration of the evolution. Genetic algorithm (Davis 1991) is the first and most well-known meta-heuristic method which simulates the Darwin's theory of evolution. Evolution strategy (ES) (Knowles and Corne 1999), genetic programming (GP) (Koza 1992), and biogeography-based optimizer (BBO) (Simon 2008) are some other evolutionary methods.
2. Physical-based optimization algorithm in which the physical rules are employed for updating the solutions. Charged system search (CSS) (Kaveh and Talatahari 2010), central force optimization (CFO) (Formato 2007), artificial chemical reaction optimization algorithm (ACROA) (Alatas 2011), black hole (BH) algorithm (Hatamlou 2013), ray optimization (RO) algorithm (Kaveh and Khayatazad 2012), small-world optimization algorithm (SWOA) (Du et al. 2006), galaxy-based search algorithm (GbSA) (Shah-Hosseini 2011), water evaporation optimization algorithm (Kaveh 2017a), multi-verse optimizer (MVO) (Mirjalili et al. 2016), lightning attachment procedure optimization (LAPO) (Nematollahi et al. 2017; Nematollahi et al. 2019), sine cosine algorithm (SCA) (Gupta and Deep 2019a, b; Mirjalili 2016; Rizk-Allah 2018), and gravitational search algorithm (GSA) (Rashedi et al. 2009; Yazdani et al. 2014) are some of the physical methods.
3. The third group includes those that mimic the social behavior of animals for obtaining a goal such as mating or finding a food source. The most well-known method of this group is particle swarm optimization (PSO) (Kennedy 2011; Naka et al. 2002). Wolf pack search algorithm (Yang et al. 2007), cuckoo search (CS) (Gandomi et al. 2013), firefly algorithm (FA) (Yang 2009; Fister et al. 2013), bird mating optimizer (BMO) (Askarzadeh 2014), monkey search algorithm (MSA) (Mucherino and Seref 2007), coral reef optimization algorithm (CRO) (Salcedo-Sanz et al. 2014; Salcedo-Sanz et al. 2013), artificial bee colony (ABC) algorithm (Karaboga and Basturk 2007; Draa and Bouaziz 2014; Saad et al. 2018), ant lion optimization algorithm (ALO) (Mirjalili 2015), gray wolf optimization algorithm (GWO) (Mirjalili et al. 2014; Saxena et al. 2019; Gupta and Deep 2018a, b, c, d; Mirjalili 2014), moth-flame optimization (MFO) algorithm (Mirjalili 2015), whale optimization algorithm (WOA) (Mirjalili and Lewis 2016), dragonfly algorithm (Mirjalili 2016), dolphin echolocation (DE) (Kaveh and Farhoudi 2013), and Krill Herd (KH) (Gandomi and Alavi 2012) are some of the other methods of this group.
4. In the fourth group, human behavior is simulated to obtain the global best answer. Teaching–learning-based optimization (TLBO) (Rao et al. 2011; Satapathy and Naik 2014; Hamzeh et al. 2018) algorithm is one of the most well-known methods of this group which simulates the enhancing procedure of a class grade. The other methods which could be classified in this group are harmony search (HS) (Geem et al. 2001), Tabu search (Glover 1989, 1990a, b; Glover and Laguna 2013), group search optimizer (GSO) (He et al. 2009), imperialist competitive algorithm (ICA) (Talatahari et al. 2012), league championship algorithm (LCA) (Kashan 2014; Kashan 2011), firework algorithm (Tan 2015a, b; Tan and Zhu 2010), colliding bodies optimization (CBO) (Kaveh and Mahdavi 2014a, b, c), tug of war optimization (TWO) (Kaveh 2017b), and interior search algorithm (ISA) (Gandomi 2014).

There are two important factors related to the population-based optimization methods. The first one is exploration, which refers to searching the whole space and having various answers in each iteration (Alba and Dorronsoro 2005). In fact, this factor shows the capability of a method in global search. For example, the crossover operator is responsible for exploration in GA (Alba and Dorronsoro 2005). The other factor is exploitation, which refers to the quality of answers in each iteration. This factor shows the capability of a method in the local search and finding the best answer around a solution. The mutation operator is responsible for local search and exploitation characteristics in GA. It should be mentioned that these two factors are in contrast to each other. In other words, focusing too much on local search, i.e., exploitation, may result in getting stuck in local optimum points, and too much focusing on global search, i.e., exploration, may cause low quality of the final best answer (Eiben and Schippers 1998).

There are numerous meta-heuristic optimization algorithms introduced in the literature. However, based on the no free lunch (NFL) theorem (Wolpert and Macready 1997), no optimization method can solve all the optimization problems. This is the motivation of introducing new methods to solve a wider range of problems. In this paper, a meta-heuristic optimization method based on the growth pattern of plants and other creatures in nature is introduced. This pattern is based on the golden ratio, which is obtained by Fibonacci. He introduced a series of numbers for which the ratio between the two consecutive numbers is the same (equal to approximately 1.618) and known as the golden ratio. This ratio is found in nature, such as the growth angle of the foliage and the snail lacquer. This ratio is employed for updating the solutions in two different phases. The proposed method is evaluated by means of 29 benchmark test functions and 6 classical engineering problems. The results of the proposed method are compared with those of 11 well-known optimization methods from different aspects including the quality of final results, the speed of convergence, and the CPU time consumption. The superiority of the proposed method can be inferred from the comparisons.

2 Golden ratio optimization method (GROM)

In this section, the proposed algorithm and its inspiration are illustrated.

2.1 Inspiration

Despite the widespread nature and the existence of a variety of creatures around humans, there is a special order over everything, which is becoming more identified by the advancement of human sciences. In nature, the growth of creatures or physical phenomena such as tornados depends on their previous situation based on a golden ratio, which causes the best performance of nature. Fibonacci was among the first people who found this golden ratio. His work resulted in the introduction of a series of numbers in which the sum of two consecutive numbers results in the next number. Moreover, the ratio between two consecutive numbers is equal to 1.618 which is the golden ratio or Φ. In the snail lacquer part, Φ angle is used. The foliage of the trees does not grow randomly in random directions. The measurements of the branches angle reveal that in their growth pattern, the order is similar to the Fibonacci sequence and the golden ratio. Trees can absorb a greater percentage of sunlight by following this type of growth pattern. In addition to nature, many artists and architects have used mathematical and geometric relationships in their works since ancient times. For example, the remains of ancient Egypt, Greece, and Rome are some of these examples. For instance, the famous Parthenon Temple is the best example of using the golden ratio (1.618). The ratios of the width to length of the rectangular windows of the temple are all equal to the golden ratio. In the Pyramids, this ratio is well respected. The ratio of the length of the Pyramid's base to its height is equal to the golden ratio. Figure 1 depicts the golden ratio in nature.

The first numbers of the Fibonacci series are as follows: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181. Except the first two numbers, the others are equal to the summation of the previous two numbers. It should also be mentioned that the ratio of every two consecutive numbers (almost after the sixth number) is almost constant which is named the Φ number.

The "Φ number" is derived from the Fibonacci sequence, a famous sequence whose reputation is not only due to the fact that each number equals the sum of the previous two sentences, but also the ratio of each two consecutive numbers is an amazing number close to 1.618, which is known as the "golden ratio." And the more surprising thing is that any Fibonacci number can be obtained through a golden number as follows:

$$x_n = \frac{\varphi^n - (1-\varphi)^n}{\sqrt{5}}. \tag{1}$$

And the answer always comes in the form of an integer, exactly equivalent to the sum of the two previous members.

For example:

$$x_6 = \frac{(1.618034\ldots)^6 - (1-1.618034\ldots)^n}{\sqrt{5}} = 8. \tag{2}$$

If a calculator is employed (when the golden number is used with 6 decimal), a response of 8.00000033 is

Fig. 1 Illustration of golden ratio in nature (Fig Ref 2019)

obtained. The result with more accurate calculations will be closer to 8.

2.1.1 Fibonacci series employment in optimization

In the optimization procedure, every answer appears as a vector and an attempt is made to put the answers in the direction of the vector with the best characteristics (i.e., the target). In the proposed method, such a process is also utilized.

First phase

In the first step of the proposed optimization method, the mean value of the population is calculated; then, the fitness of this solution is obtained and compared with the worst solution. If the mean solution has better fitness compared to the worst solution, the worst one is replaced by the mean solution. This step enhances the algorithm speed to reach convergence.

In the second step, for each vector (solution) of the population, another vector is selected randomly. In order to investigate the impact of the other solutions of the population on the movement of these two vectors, the average of the population members is used to determine the direction and extent of the new vector movement.

Now, in order to specify the direction of the new vector, it is necessary to compare the three selected vectors (the selected vector, the randomly selected vector, and the mean vector). To do this, the best vector which has the lowest value of the objective function is considered as the main vector.

$$F_{\mathrm{best}} > F_{\mathrm{medium}} > F_{\mathrm{worst}} \tag{3}$$

$$\overrightarrow{X_t} = \overrightarrow{X_{\mathrm{medium}}} - \overrightarrow{X_{\mathrm{worst}}}. \tag{4}$$

Now, to determine the amount of movement in the direction of the obtained vector, Fibonacci's formula and golden number are used. To perform the global search and then the local search during the implementation of the algorithm, Fibonacci's formula is used as follows:

$$\begin{aligned} &F_t = \mathrm{GF} \times \frac{\varphi^T - (1-\varphi)^T}{\sqrt{5}}, \quad \mathrm{GF} = 1.618 \\ &T = \frac{t}{t_{\max}}. \end{aligned} \tag{5}$$

In evolutionary optimization algorithms, the solutions should be updated mathematically in a way that the overall updating results in better objective function. This updating is mostly performed toward the best solution of the population. Moreover, in order to search the whole space of the problem, a random movement is also added to the new solution. Now, to update the solutions, the following equation is used:

$$X_{\mathrm{new}} = (1 - F_t)X_{\mathrm{best}} + \mathrm{rand} \times F_t \times X_t. \tag{6}$$

Now, the boundary conditions of variables are checked. If the constraints are satisfied, the new solution would be substituted by the old one, if the fitness of the new solutions is better than the old one.

$$\begin{array}{ll} X^i = X^i_{\mathrm{new}} & \text{if } F^i_{\mathrm{testpoint\_new}} < F^i_{\mathrm{tetpoint}} \\ X^i = X^i_{\mathrm{old}} & \quad\text{otherwise} \end{array}. \tag{7}$$

Second phase

In the second phase, every single solution attempts to come close to the best answer and refrain from the worst solution. To do this, the golden ratio is also employed as follows for all the solutions.

$$\begin{aligned} X_{\mathrm{new}} &= X_{\mathrm{old}} + \mathrm{rand} \times \left(\frac{1}{\mathrm{GF}}\right) \times (X_{\mathrm{best}} - X_{\mathrm{worst}}) \\ \frac{1}{\mathrm{GF}} &= 0.618. \end{aligned} \tag{8}$$

After updating the solutions, the upper and lower bounds of the variables are checked and the new solution would be substituted by the older one if the answer is improved, as presented below:

$$\begin{array}{ll} X^i = X^i_{\mathrm{new}} & \text{if } F^i_{\mathrm{testpoint\_new}} < F^i_{\mathrm{tetpoint}} \\ X^i = X^i_{\mathrm{old}} & \quad\text{otherwise} \end{array}. \tag{9}$$

The pseudocode of the proposed algorithm is as follows:

```
Start
    Initializing
    Calculate fitness function of all solutions
While the convergence criterion is not satisfied
    Obtain X_ave which is the mean value of all the solutions
    Set the solution with the worst fitness as X_w
    If fitness of X_ave is better than the fitness of X_w
        X_w = X_ave
    End
    For i=1: Npop (each solution)
     Select X_j randomly which is not equal to X_i
     Compare the solutions X_i, X_j, and X_ave and set the best solution as the X_best, the second best
     Solution as X_medium, and third solution as X_worst
     Update X_i based on equations 1-4
     Check the constraints and substitute the new solution with the old one based on Eq. 5
    End
    For i=1:Npop (each solution)

        For j=1:Nv (number of variables)
         Update the solutions based on 6
         Check the constraints and substitute the new solution with the old one based on Eq. 7
        End
    End
End
Stop
```

As it can be seen, the proposed method is a simple method which is free from any parameter tuning. The results and comparison show the surprising performance of the proposed method.

3 Evaluation and discussion

In this section, the performance of the proposed method is evaluated from different points of views. Twenty-nine test functions are employed to benchmark the proposed method. These test functions can be categorized in four different classes including unimodal, multimodal, fixed-dimensional multimodal (hybrid multimodal), and composite functions (Molga and Smutnicki 2005; Digalakis and Margaritis 2001; Liang et al. 2005). These four classes of test functions are listed in Tables 1, 2, 3, and 4, respectively. Moreover, the 2-D plots of these test functions are depicted in Figs. 2, 3, 4, and 5. In addition to these benchmark test functions, 5 engineering optimization problems are also used to verify the performance of the

Table 1 Unimodal benchmark functions

Function	Dim	Range	$f_{\min}$
$F_1(x) = \sum_{i=1}^{n} x^2$	30,200	[− 100, 100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	30,200	[− 10, 10]	0
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j\right)^2$	30,200	[− 100, 100]	0
$F_4(x) = \mathrm{Max}\{\lvert x_i\rvert, \quad 1 \le i \le 5\}$	30,200	[− 100, 100]	0
$F_5(x) = \sum_{i=1}^{n-1} \{100(x_{i+1} - x_i)^2 + (x_i - 1)^2\}$	30,200	[− 30, 30]	0
$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30,200	[− 100, 100]	0
$F_7(x) = \sum_{i=1}^{n} (ix_i^4 + \mathrm{random}[0, 1])$	30,200	[− 1.28, 1.28]	0

proposed method in solving real problems. The engineering test functions consist of 5 classic mechanical optimization problems including gear train design problem, cantilever beam design problem, pressure vessel design, welded beam design, and tension/compression spring design, and a high constraint electrical optimization problem known as optimal power flow (OPF).

Eleven well-known meta-heuristic optimization methods are used as the results of the proposed method are compared with. These methods are artificial bee colony (ABC) (Karaboga and Basturk 2007), lightning attachment procedure optimization (LAPO) (Nematollahi et al. 2017), lightning search algorithm (LSA) (Shareef et al. 2015), differential evolution (DE) (Price et al. 2006), shuffled frog leaping algorithm (SFLA) (Eusuff and Lansey 2003), imperialist competitive algorithm (ICA) (Talatahari et al. 2012), particle swarm optimization (PSO) (Kennedy 2011), ant lion optimizer (ALO) (Mirjalili 2015), gray wolf optimizer (GWO) (Mirjalili et al. 2014), cuckoo search algorithm (CSA) (Gandomi et al. 2013), and firefly optimization method (FOM) (Yang 2009).

The number of particles (size of population) for each method must be selected so that the number of objective function evaluation in all iterations would be the same. In other words, the size of the population for the methods that evaluate the objective function two times in each iteration must be half of that for methods that evaluate the objective function once in an iteration. The parameter descriptions of different methods are as follows:

Method	Function evaluation in each iteration	Populations size	Other parameter
ABC	2	40	Alimit 15
DE	1	80	$F = 0.5$, $CR = 0.5$
SFLA	1	80	–
ICA	1	80	–
PSO	1	80	$C1 = C2 = 2$, $w = w_{\max} - t \times (w_{\max} - w_{\min})/(t_{\max})$, $w_{\max} = 0.9$, $w_{\min} = 0.4$, $t_{\max}$: maximum iteration number (Hu and Eberhart 2002)
ALO	2	40	Stochastic function limit = 0.5, ratio I based on the current iteration and maximum number of iteration [based on the MATLAB code available in Mirjalili (2015)]
GWO	1	80	–
CSA	2	40	Discovery rate (Pa) 0.25
Firefly	1	80	α 0.25, β 0.2, γ 1
LSA	2	40	Maximum channel time 10, energy equation based on MATLAB code available in Shareef (2015)
LAPO (Nematollahi et al. 2017)	2	40	Based on the MATLAB code available in Vahidi et al. (2017)

Since the test functions are solved in two cases including low dimensional and high dimensional, the number of total iterations is considered 500 [function evaluations (FES) = 4000] for the former and 2000 for the later. Furthermore, since meta-heuristic methods are population based, statistical analysis is needed. Thus, 30 different trials are also applied for all the methods and the best answer among 30 different trials, mean value, and standard deviations of the

Table 2 Multimodal benchmark functions

Function	Dim	Range	$f_{\min}$
$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30,200	[− 500, 500]	− 418.982 9 * Dim
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30,200	[− 5.12, 5.12]	0
$F_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30,200	[− 32, 32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30,200	[− 600, 600]	0
$F_{12}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})]$ $+(y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4) y_i$ $= 1 + \frac{x_i+1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30,200	[− 50, 50]	0
$F_{13}(x) = 0.1\{10\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + 10\sin^2(3\pi x_i + 1)]$ $+(x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30,200	[− 50, 50]	0

Table 3 Fixed-dimensional multimodal benchmark functions

Function	Dim	Range	fmin
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{\sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[− 65, 65]	1
$F_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]$	4	[− 5, 5]	0.00030
$F_{16}(x) = 4x_x^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[− 5, 5]	− 1.0316
$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	[− 5, 5]	0.398
$F_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$ $\times[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[− 2, 2]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	3	Blum et al. (2011, Gogna and Tayal (2013)	− 3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	6	[0, 1]	− 3.32
$F_{21}(x) = -\sum_{i=1}^{5}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0, 10]	10.1531
$F_{22}(x) = -\sum_{i=1}^{7}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0, 10]	− 10.4028
$F_{23}(x) = -\sum_{i=1}^{10}[(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	[0, 10]	− 10.5363

30 obtained final results are compared. CPU time consumption is another factor which should be investigated. In this paper, all the methods are implemented in MATLAB 2012a in a Core i5 PC with 3 GHz processing frequencies of CPU and 8 GB of RAM. The convergence behavior is another aspect with which the methods are compared with each other.

3.1 Exploitation analysis

To evaluate the performance of the proposed method in terms of local search, i.e., exploitation, unimodal test functions ($F1$–$F7$) are utilized. Unimodal test functions have simple convex shape, and reaching the best global answer by means of meta-heuristic methods, which are population-based methods with lots of particles, is difficult and an excellent local search is needed. The results of different methods for solving the unimodal test functions are listed in Table 5. The results reveal that the proposed method outperforms the other methods in all unimodal test functions. For $F1$ and $F3$, the exact global answer is achieved. The final results for $F2$ and $F4$ also show the high-quality local search of the proposed method. Moreover, the average of 30 trials for each test function is very close to the best answer. This means that the proposed method is robust and the same results are almost obtained in different trials. The low values of standard deviation also demonstrate this feature of the proposed method. The CPU

Table 4 Composite benchmark functions

Function	Dim	Range	$f_{\min}$
F_{24}(CF1) $f_1, f_2, f_3, \ldots f_{10}$ = Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [\frac{5}{100}, \frac{5}{100}, \frac{5}{100}, \ldots, \frac{5}{100}]$	10	[− 5, 5]	0
F_{25}(CF1) $f_1, f_2, f_3, \ldots f_{10}$ = Griewank's Function $[\sigma_1, \sigma_2, \sigma_3, \ldots \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = \left[\frac{5}{100}, \frac{5}{100}, \frac{5}{100}, \ldots, \frac{5}{100}\right]$	10	[− 5, 5]	0
F_{26}(CF1) $f_1, f_2, f_3, \ldots f_{10}$ = Griewank's Function $[\sigma_1, \sigma_2, \sigma_3, \ldots \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots \lambda_{10}] = [1, 1, 1, \ldots, 1]$	10	[− 5, 5]	0
F_{27}(CF4) f_1, f_2 = Ackley's Function f_3, f_4 = Rastrigin's Function f_5, f_6 = Weierstras's Function f_7, f_8 = Griewank's Function $f_9 f_{10}$ = Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = \left[\frac{5}{32}, \frac{5}{32}, 1, 1, \frac{5}{0.5}, \frac{5}{0.5}, \frac{5}{100}, \frac{5}{100}, \frac{5}{100}, \frac{5}{100}\right]$	10	[− 5, 5]	0
F_{28}(CF5) f_1, f_2 = Ackley's Function f_3, f_4 = Rastrigin's Function f_5, f_6 = Weierstras's Function f_7, f_8 = Griewank's Function $f_9 f_{10}$ = Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = \left[\frac{1}{5}, \frac{1}{5}, \frac{5}{0.5}, \frac{5}{0.5}, \frac{5}{100}, \frac{5}{100}, \frac{5}{32}, \frac{5}{32}, \frac{5}{100}, \frac{5}{100}\right]$	10	[− 5, 5]	0
F_{29}(CF5) f_1, f_2 = Rastrigin's Function f_3, f_4 = Weierstras's Function f_5, f_6 = Griewank's Function f_7, f_8 = Ackley's Function $f_9 f_{10}$ = Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = \left[0.1 \times \frac{1}{5}, 0.2 \times \frac{1}{5}, 0.3 \times \frac{5}{0.5}, 0.4 \times \frac{5}{0.5}, 0.5 \times \frac{5}{100}, 0.6 \times \frac{5}{100}, 0.7 \times \frac{5}{32}, 0.8 \times \frac{5}{32}, 0.9 \times \frac{5}{100}, 1 \times \frac{5}{100}\right]$	10	[− 5, 5]	0

time consumptions show that the proposed method is faster than the other methods except for PSO and GWO. By a comparison of the proposed method results with those of other methods for these 7 test functions, it can be concluded that the proposed method is very powerful in local search and can reach the final best global answer very well. Moreover, the robustness of the proposed method can be inferred from these results.

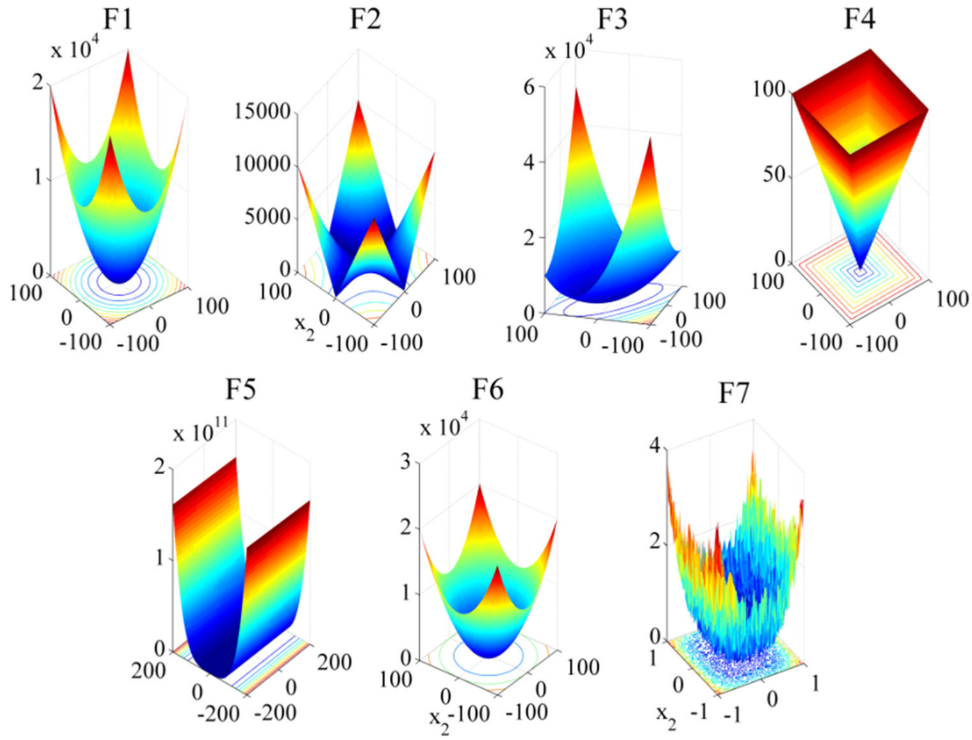


Fig. 2 2-D version of unimodal test functions

3.2 Exploration analysis

As the power of the proposed method in local search is approved, here, the performance of the proposed method in global search, i.e., exploration, is testified. It is important to find out the ability of a method such that it is not stuck in local optimum points; thus, multimodal ($F8$–$F13$) and fixed-dimensional multimodal ($F14$–$F23$) test functions are employed. Since these test functions have lots of local optimum points, the exploration feature of optimization methods can be evaluated very well by these test functions. The results of different methods for solving multimodal and fixed-dimensional multimodal test functions are listed in Tables 6 and 7, respectively. For the test function $F8$, the proposed method does not have the best results and none of the other methods could reach the best answer. However, for $F9$–$F13$ the proposed method outperforms the other methods. Moreover, the low values of standard deviations (for most of them zero is obtained) demonstrate the robustness of the proposed method. For $F14$ to $F23$, approximately all the methods reach the same best answer. Moreover, the standard deviations of the proposed method are the best for some test functions and competitive to other methods for some other test functions. Generally speaking, the proposed method has an excellent performance from an exploration point of view and could easily find the best answer among different local optimum points. This means that the proposed method rarely gets stuck in local optimum points.

3.3 Global minimum finding

The composite test functions are utilized to evaluate the performance of the proposed method in both global search and local search, simultaneously. Table 8 compares the results of different methods for composite test functions. For the test function $F24$, the best results belong to LAPO followed by the proposed method. For $F28$, both LAPO and the proposed GROM reach the exact best global answer. The best results for $F25$ belong to firefly followed by the proposed method. Generally, the proposed method reaches the best answer or second best answer with regard to 11 other optimization methods. In general, it can be concluded that not only the proposed method is able to find the overall best answer of a function among lots of local optimum points, but it can also reach nearly the exact best answer by the excellence local search.

3.4 Wilcoxon test

A statistical test is needed to evaluate and compare the results obtained in different trails. The Wilcoxon test is a nonparametric statistical hypothesis test used to compare two related samples. The *P value* is the output of this test which shows the similarity of two sets of results. For P value greater than 0.05, a bigger portion of the results are nearly the same; for P value equal to 1, all the results are the same; and for P value less than 0.05, a major portion of results are different. The P values for the results obtained

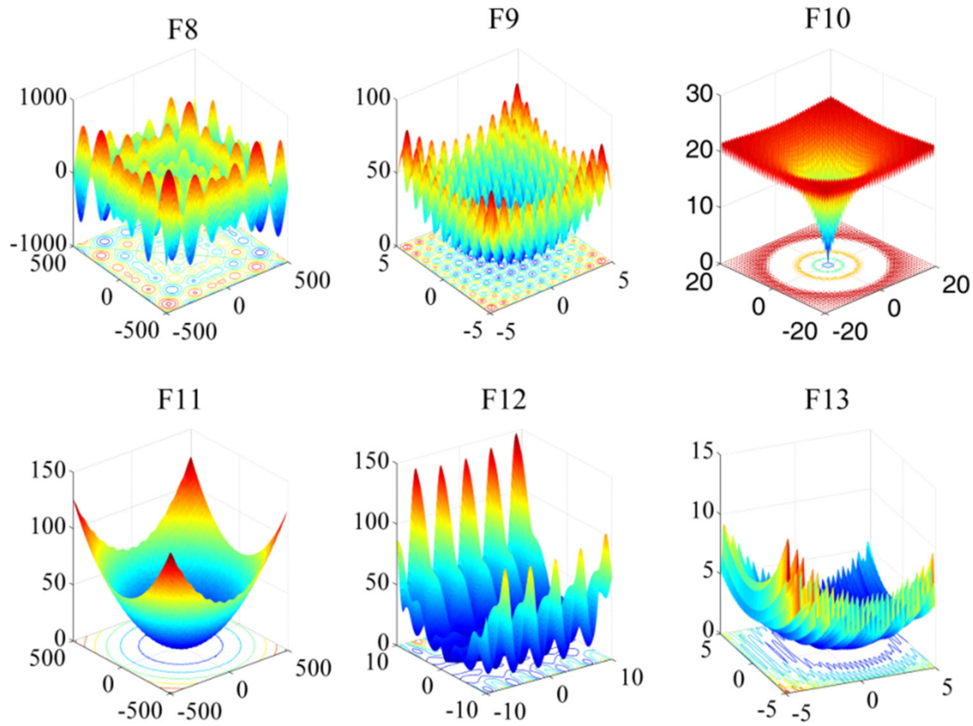


Fig. 3 2-D version of multimodal test functions

by the GROM algorithm for 29 test functions are compared with P values of the results obtained by other methods in Table 9. As it can be seen, for some test functions, the P value is greater than 0.05 and for some other, it is less than 0.05. It should be mentioned that this criterion is used to see whether the results of different trials are the same and it cannot be used to evaluate the better performance of different methods. For a comparison between the performances of different methods, the mean value, the best value, and the worst value are used.

3.5 Large-scale optimization problems

A challenging task for optimization methods is to find the optimal decision for the problems with lots of variables. Thus, large-scale optimization problems should also be used to evaluate an optimization method. To evaluate the performance of the proposed algorithm in large-scale problems, the unimodal and multimodal test functions are employed considering 200 variables for each test function. Moreover, these test functions are solved by different optimization algorithms in two cases. In the first one, the size of the population is 500 with 2000 iterations, and in the second case, the population size is considered to be 40 with total iterations of 500. Ten different trials are also run for each method.

The results of different methods for solving large-scale unimodal and multimodal test functions for the first case are listed in Tables 10 and 11, respectively, and for the second case in Tables 13 and 14. It can be observed from the results that the proposed method could reach the best answers for almost all the test functions in different cases very well. In other words, the proposed method has an excellent performance in solving large-scale test functions both with a high and with a low number of particles.

For the statistical analyses of sets of results, the Wilcoxon test is applied to both cases 1 and 2 and the results are listed in Tables 12 and 15, respectively.

3.6 Convergence behavior

As mentioned before, the CPU time consumption which is related to calculation volume in each iteration is an important factor of an optimization method. However, it is also important for a method to reach the final best global answer in lower iterations. This feature, which is known as convergence behavior, is related to the way in which solutions are updated. A number of mathematical equations are used to update the solutions and move them toward the best global answer. If the updating procedure and movement toward the best answer are performed well, the global best answer would be obtained in lower iterations.

In order to investigate the convergence behavior of the proposed method, the trajectory of the first variable of the first solution toward the best value is depicted in Fig. 6. This figure exhibits the 3-D shape of the test function, search history, trajectory of the first variable of the first solution, the average of fitness function obtained in each

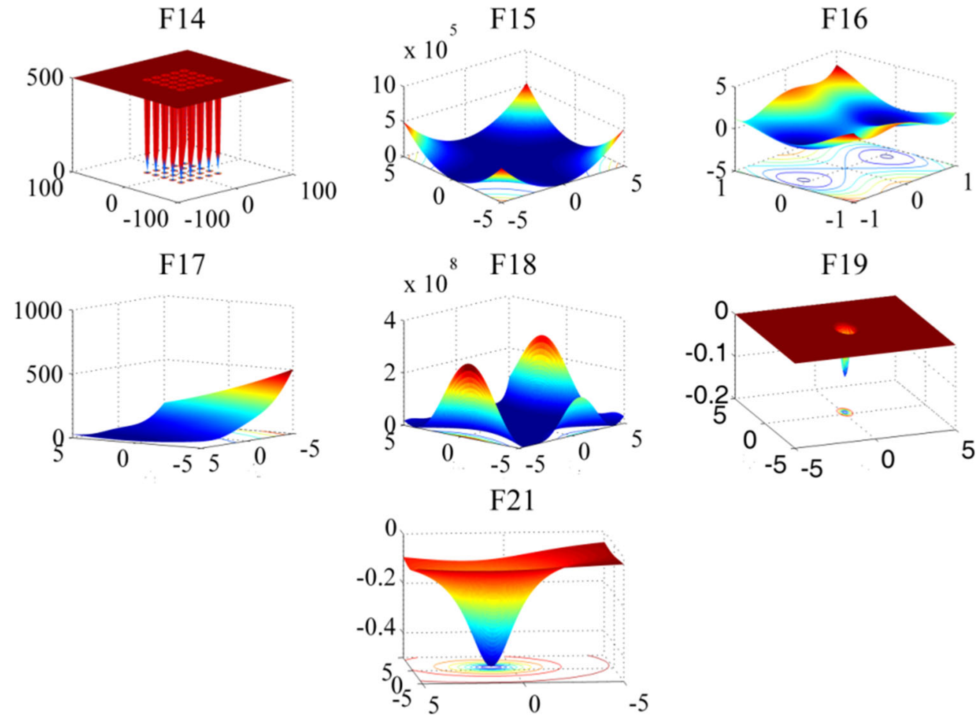


Fig. 4 2-D version of fixed-dimensional multimodal test functions

iteration, and convergence to reach the best results for some test functions. As it can be seen, the search space is searched very well, the variable reaches the best value in early iterations, and the method finds the final best result in early iterations.

The comparisons between the convergence behavior of the proposed method and other methods for 4 test functions are shown in Fig. 7. One can see that the proposed method obtains a better answer or a similar answer in lower iterations in comparison with other methods. To sum up, the proposed method has excellent convergence behavior.

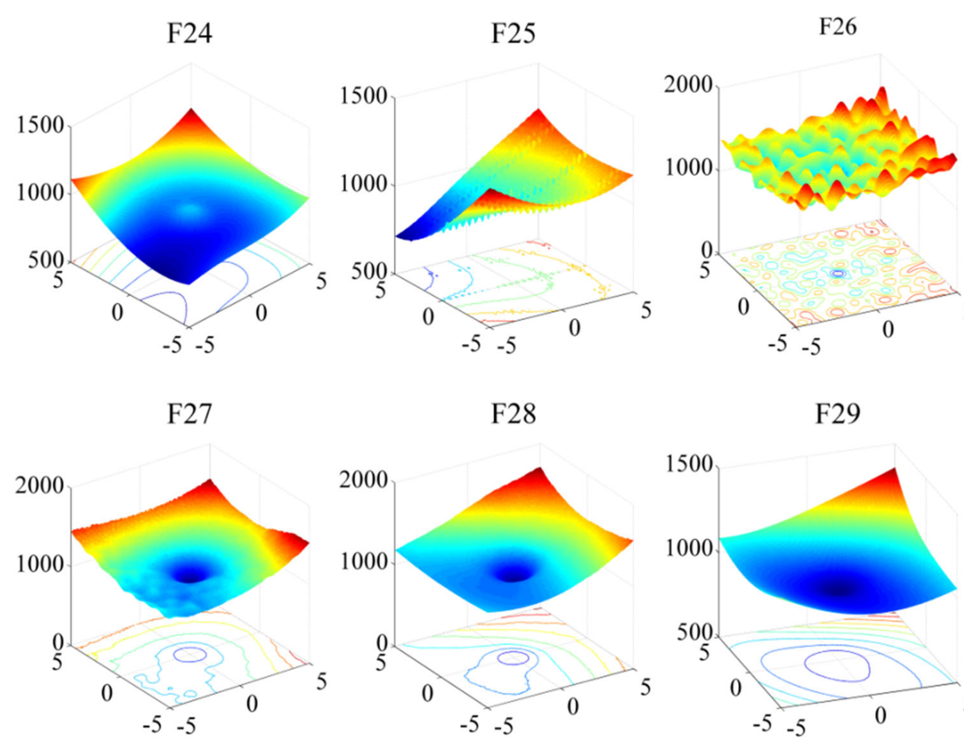


Fig. 5 2-D version of composite test functions

Table 5 Results of different methods for solving the unimodal test functions (DIM = 30)

F	GROM				ABC			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*1	0	3.1350e− 247	6.5961e−251	0.56846	5.1833	14.5733	7.5363	0.6516
*F*2	2.8836e−53	1.367e−52	9.173e−53	0.5888	0.7384	1.1845	0.2592	0.6509
*F*3	0	3.398e−87	9.6154e−87	1.3371	1.6298E4	2.5815E4	3.6430E3	1.52845
*F*4	7.155e−145	1.118e−131	3.0352e−132	0.594906	46.7749	57.2323	3.9239	0.714885
*F*5	18.46243	19.12	0.6160	0.6549	1.8295E3	5.5065E3	3.5868E3	0.777289
*F*6	1.594e− 7	3.9096e− 7	3.573e− 7	0.60241	3.1398	18.9567	10.4361	0.774391
*F*7	2.677e−5	0.000131	4.7138e−5	0.52610445	0.3353	0.4881	0.0778	0.865856
	LAPO				LSA			
	Best	AVE	Std	Time (s)	Best	Ave	Std	Time (s)
*F*1	1.3406E−15	2.0664E−13	5.5098E−13	0.8137	2.6192e−11	2.5961e−8	1.3958e−7	13.468
*F*2	4.2412E−9	2.2547E−8	1.7473E−8	0.8797	0.0015378	0.068542	0.09996	13.819
*F*3	4.1270E−7	1.1385E−5	3.6624E−5	2.656843	30.719	130.89	168.24	17.374
*F*4	2.7951E−7	4.3915E−7	1.3825E−7	1.007296	2.8447	4.5491	2.03	13.435
*F*5	19.5667	22.7427	0.6846	1.061064	21.7546	61.376	34.334	13.94
*F*6	1.3619E−6	1.1151E−5	1.0200E−5	1.035845	2.00	4.5	2.5495	3.1517
*F*7	1.3323E−4	7.1418E−4	4.3695E−4	1.2725	0.028583	0.032825	0.0035634	13.9
	DE				SFLA			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*1	2.6201E−12	1.8812E−4	7.0503E−4	0.5086	4.7749	18.1945	12.1923	1.472477
*F*2	1.6397E−7	6.1642E−4	0.0021	0.539950	1.1161	3.3016	1.0643	1.559033
*F*3	137.5353	757.1439	965.7426	1.44286	181.3897	470.3856	154.6271	2.322263
*F*4	14.1436	25.9679	6.0893	0.586869	3.8694	6.7903	1.6306	1.755044
*F*5	20.2093	334.0200	444.3179	0.640554	131.6334	558.8711	369.8338	1.536910
*F*6	5.0067E−5	1.4954E−4	4.4487E−4	0.643048	4.1500	19.0857	13.9022	1.54936
*F*7	0.0209	0.0364	0.0129	0.763583	0.0171	0.0352	0.0129	3.745837
	ICA				PSO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*1	7.2247E−6	1.3205E−4	1.7340E−4	1.48590	23.2860	76.9335	31.9150	0.355952
*F*2	3.1098E−4	0.0011	6.1653E−4	1.4804	2.3918	7.3666	3.0318	0.3660
*F*3	614.0714	1.3905E3	559.3069	2.540456	354.3102	2.6799E3	2.6157E3	1.236538
*F*4	4.1320	9.4623	2.7549	1.563988	4.3508	7.5786	1.7179	0.47463
*F*5	26.5015	205.4637	299.0098	1.584770	160.6205	1.4821E3	1.7592E3	0.461078
*F*6	3.1110E−5	2.2500E−4	1.6011E−4	1.600717	26.2675	88.1072	68.0237	0.481432
*F*7	0.0267	0.0226	0.0768	1.724640	0.0038	0.0355	0.0240	0.608523
	ALO				GWO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*1	6.0598E−5	3.5790E−4	4.3611E−4	29.4673	9.0481E−12	3.2320E−11	2.5277E−11	0.2945
*F*2	0.6943	48.7203	46.1718	28.9521	1.0069E−13	3.6499E−13	2.0871E−13	0.3168
*F*3	963.2682	2.6094E3	1.0290E3	30.720294	4.3817E−5	0.0060	0.0097	1.1417
*F*4	9.0764	15.8283	4.4187	28.354286	4.3976E−6	2.7134E−5	2.7926E−5	0.3805
*F*5	23.4972	196.5491	191.7828	28.414984	36.0767	37.0454	0.8404	0.4298
*F*6	6.9376E−5	2.2413E−4	1.1807E−4	28.238345	1.0001	1.4185	0.4020	0.4354

Table 5 (continued)

	ALO				GWO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*7	0.0966	0.1734	0.0512	28.42491	0.0016	0.0028	9.1592E−4	0.5522

	CSA				Firefly			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*1	3.6952	12.8494	6.8743	0.6817	0.0049	0.0118	0.0045	2.208220
*F*2	2.3404	4.9688	2.1806	0.596314	0.1556	0.3733	0.1033	2.2510
*F*3	632.8637	1.5724E3	446.1137	1.421566	920.4498	2.0394E3	835.4943	3.065636
*F*4	7.0618	14.1226	3.7242	0.635692	0.0501	0.0807	0.0164	2.331268
*F*5	248.7507	1.3990E3	972.8326	0.686697	26.5929	133.8007	161.0121	2.33665
*F*6	2.5103	9.2432	5.0420	0.680190	0.0046	0.0130	0.0051	2.308121
*F*7	0.0504	0.1476	0.0723	0.797525	0.0099	0.0320	0.0172	2.446989

Table 6 Results of different methods for solving the multimodal test functions

F	GROM				ABC			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 9467.296	− 9070.8	347.215	0.4655	− 5.421E3	− 5.037E3	168.1494	0.67172
*F*9	0	0	0	1.671	24.3595	30.9360	4.8363	0.77650
*F*10	8.8817e−16	8.8817e−16	0	1.573	3.4924	4.3770	0.5014	0.81702
*F*11	0	0	0	1.6902	1.0335	1.1260	0.0567	0.85534
*F*12	3.8742e−9	9.980e−9	5.7890e−9	0.8982	0.0582	0.3590	0.1777	1.26490
*F*13	5.2761e−7	0.003506	0.006430	0.874	0.3822	1.2131	0.6178	1.26163
	LAPO				LSA			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 1.0613E+4	− 1.036E+4	1.9994E3	0.927065	− 8502.8	− 8131.6	281.98	13.175
*F*9	0	1.53344	3.70144	1.095045	46.763	64.274	13.636	13.704
*F*10	9.5009E−9	5.8694E−8	5.0496E−8	1.130083	1.3404	2.8488	0.78898	13.836
*F*11	1.7764E−15	1.5914E−13	2.9758E−13	1.217667	2.0206e−14	0.0066515	0.007159	14.139
*F*12	6.8458E−9	0.0104	0.0311	2.197772	5.287e−6	0.18718	0.3595	15.362
*F*13	5.5452E−7	0.0098	0.0240	2.048834	3.0594e−3	0.16322	0.30357	15.358
	DE				SFLA			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 1.104E4	− 9.916E3	690.4544	0.570069	− 9.2376E3	7.5146E3	709.8488	1.84541
*F*9	18.9042	37.6106	11.9314	0.677520	25.7164	47.2634	15.0827	1.96860
*F*10	0.9313	2.6724	1.1995	0.712424	4.1208	6.5725	1.5489	1.81110
*F*11	2.3758E−4	0.0491	0.0595	0.821489	0.9253	1.1603	0.1557	1.73998
*F*12	0.0038	3.4061	3.1716	1.134543	2.4217	7.0897	2.3381	2.28280
*F*13	3.1951E−4	11.7909	10.4224	1.119140	1.7653	30.4043	19.1206	2.61907

Table 6 continued

	ICA				PSO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 1.126E4	− 9.671E3	824.1584	1.465802	− 9.3168E3	− 7.771E3	1.0051E3	1.02069
*F*9	5.9779	14.3925	3.5594	1.546710	44.6941	84.6695	22.4398	1.12109
*F*10	0.0011	0.0061	0.0057	1.570927	2.8692	4.6089	1.0713	1.15249
*F*11	2.7347E−5	0.0466	0.0339	1.638158	1.0935	1.6135	0.4053	1.19303
*F*12	2.4743E−8	1.1269E−5	4.2876E−5	2.10093	0.6399	3.1318	1.7733	1.61608
*F*13	1.5961E−6	0.0015	0.0037	2.190709	2.3482	14.4184	11.1284	1.60883
	ALO				GWO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 5.5636E3	− 5.4872E3	59.3746	28.954543	− 8.5581E3	− 6.964E3	1.665E3	0.4936
*F*9	40.7934	75.9153	20.8421	29.172624	5.6843E−5	3.1699	6.6764	0.4701
*F*10	1.1567	2.5447	1.0413	28.910295	5.7643E−8	1.5083E−7	6.5140E−8	0.4932
*F*11	0.0061	0.0290	0.0131	29.139031	0.0012	0.0060	0.0109	0.5270
*F*12	7.4959	12.2652	5.9904	30.796473	0.0438	0.0753	0.0255	0.9347
*F*13	0.0344	15.5303	22.1428	30.290157	0.9754	1.2997	0.1932	0.9449
	CSA				Firefly			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 8.909E3	− 8.5464E3	172.9980	3.620904	− 7.6344E3	− 6.216E3	697.3859	2.28836
*F*9	93.0875	116.9196	10.9529	3.449429	16.1872	28.3631	5.8445	2.35327
*F*10	10.5414	13.1248	1.2356	3.637010	0.0294	0.0523	0.0157	2.39521
*F*11	1.3221	1.5594	0.1394	3.770781	0.0037	0.0057	0.0013	2.42128
*F*12	3.9863	5.5999	0.9444	4.234347	8.0382E−5	2.2886E−3	1.2389E−3	3.12889
*F*13	13.5315	19.3205	2.6419	4.307192	0.0013	0.0023	7.1625E−4	3.15469

Table 7 Results of different methods for solving the fixed-dimensional multimodal test functions

F	GROM				ABC			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*14	0.998	1.1964	0.5952	1.253	0.9980	0.9980	4.0101E−8	0.4385
*F*15	0.0003074	0.0003074	2.83762e−19	0.3364	8.0855E−4	0.0017	8.9604E−4	0.1375
*F*16	− 1.0316	− 1.03162	3.315148e−7	0.28188	− 1.0316	2.2037E−8	0.0706	0.1224
*F*17	0.397887	0.39788	6.678e−7	0.1975	0.3979	0.3979	2.9441E−5	0.1174
*F*18	2.999	2.999	7.02166e−16	0.2625	3.0000	3.0061	0.0078	0.1178
*F*19	− 3.8627	− 3.862	8.88178e−16	0.4084	− 3.8628	− 3.8628	1.6310E−5	0.1549
*F*20	− 3.3219	− 3.29821	0.0475558	0.4489	− 3.3219	− 3.3211	8.4270E−4	0.1573
*F*21	− 10.153	− 10.1531	7.944109e−16	0.4620	− 10.1476	− 10.0188	0.0974	0.1801
*F*22	− 10.402	− 10.402	1.4862e−15	0.5032	− 10.4007	− 10.2614	0.1857	0.1910
*F*23	− 10.536	− 10.53	1.776e−15	0.578	− 10.5074	− 10.1530	0.4089	0.2108

Table 7 continued

	LAPO				LSA			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*14	0.9980	0.9980	5.7495E−8	0.767174	0.998	1.1968	0.41912	3.7214
*F*15	3.0749E−4	5.5811E−4	2.2495E−4	0.167664	0.000307	0.00053523	0.00043114	3.1152
*F*16	− 1.0316	− 1.0316	1.4460E−7	0.134042	− 1.031628	− 1.031628	1.95824e−16	0.6379
*F*17	0.3979	0.3983	4.8405E−4	0.123149	0.3979	0.3979	0	0.1835
*F*18	3.0000	3.0000	7.5626E−16	0.132780	3.001	3.001	1.03620e−15	1.788
*F*19	− 3.8628	− 3.8628	8.5422E−16	0.198560	− 0.305	− 0.3004	7.40148e−17	1.270
*F*20	− 3.3220	− 3.2729	0.0571	0.201384	− 3.321	− 3.27443	0.061396	1.383
*F*21	− 10.1532	− 9.6960	0.8042	0.245951	− 10.053	− 8.385	2.9225	1.632
*F*22	− 10.4029	− 10.1728	0.6905	0.276474	− 10.402	− 6.0452	3.101	2.000
*F*23	− 10.5364	− 10.2295	0.6352	0.316722	− 10.536	− 7.7078	3.730	2.457

	DE				SFLA			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*14	0.9980	1.4568	1.6536	0.4269	0.9980	2.4509	1.3675	0.5811
*F*15	3.0749E−4	0.0060	0.0087	0.1007	3.0764E−4	7.4043E−4	4.9286E−4	0.2595
*F*16	− 1.0316	− 1.0316	5.5880E−16	0.0838	− 1.0316	− 1.0316	2.1065E−16	0.3186
*F*17	0.3979	0.3979	0	0.0787	0.3979	0.3979	0	0.3166
*F*18	3.0000	3.0000	1.7123E−15	0.0830	3.0000	3.0000	3.7891E−15	0.3135
*F*19	− 3.8628	− 3.8628	2.5252E−15	0.1172	− 3.8628	− 3.8628	4.8648E−16	0.2631
*F*20	− 3.3220	− 3.2625	0.0594	0.1190	− 3.3220	− 3.2830	0.0584	0.2748
*F*21	− 10.1532	− 5.1345	2.9569	0.1501	− 10.1532	− 6.6493	3.5714	0.2787
*F*22	− 10.4029	− 6.0078	3.4431	0.1587	− 10.4029	− 8.3021	3.2189	0.2885
*F*23	− 10.5364	− 7.2796	3.7594	0.1758	− 10.5364	− 5.7586	3.9045	0.3049

	ICA				PSO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*14	0.9980	0.9980	4.3414E−14	0.635135	0.9980	1.0311	0.1784	0.379351
*F*15	4.0430E−4	0.0021	0.0028	0.311820	3.0750E−4	0.0056	0.0084	0.075062
*F*16	− 1.0316	− 1.0316	5.0634E−16	0.288696	− 1.0316	− 1.0316	2.7133E−11	0.069478
*F*17	0.3981	0.3983	3.8523E−3	0.287647	0.3979	0.3979	6.1259E−12	0.055647
*F*18	3.0000	3.0000	2.1353E−11	0.285817	3.0000	5.7000	14.5399	0.059684
*F*19	− 3.8628	− 3.8628	3.1377E−12	0.329280	− 3.8628	− 3.8625	0.0014	0.091313
*F*20	− 3.3220	− 3.3220	1.2090E−8	0.318563	− 3.3220	− 3.2755	0.0681	0.093961
*F*21	− 10.1532	− 7.4016	3.4354	0.350146	− 10.1532	− 6.9746	3.4764	0.119316
*F*22	− 10.4029	− 9.6392	2.2911	0.367494	− 10.4029	− 7.8672	3.3898	0.133108
*F*23	− 10.5364	− 8.4297	3.2277	0.386303	− 10.5364	− 7.0843	3.8065	0.152079

	ALO				GWO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*14	0.9980	5.6102	3.5498	0.689122	0.998	4.3865	3.7196	0.745630
*F*15	0.0011	0.0041	0.0065	0.602195	3.0846E−4	0.0033	0.0069	0.155647
*F*16	− 1.0316	− 1.0316	3.4120E−13	0.398864	− 1.0316	− 1.0316	2.2117E−4	0.122084
*F*17	0.3979	0.3979	4.7070E−13	0.293945	0.3979	0.3979	3.6777E−5	0.123547
*F*18	3.0000	3.0000	5.5923E−12	0.397778	3.0000	3.0005	4.8378E−4	0.120457
*F*19	− 3.8628	− 3.8526	0.0302	0.524160	− 3.8628	− 3.8610	0.0025	0.182076
*F*20	− 3.3220	− 3.2622	0.0784	0.809817	− 3.3219	− 3.2421	0.0853	0.219921
*F*21	− 5.1008	− 3.8729	1.2007	0.642313	− 10.1502	− 8.6959	2.6781	0.235932

Table 7 (continued)

	ALO				GWO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*22	− 10.4029	− 7.2446	3.2225	0.650708	− 10.4003	− 9.6799	2.1149	0.258903
*F*23	− 10.5364	− 5.9964	3.7391	0.666060	− 10.5326	− 10.0440	1.7398	0.297190

	CSA				Firefly			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*14	0.9980	1.0530	0.1915	0.407883	0.9980	5.0897	2.8480	0.713667
*F*15	5.4387E−4	0.0013	4.9224E−4	0.106910	7.720E−4	0.0054	0.0045	0.419765
*F*16	− 1.0316	− 1.0316	1.2697E−7	0.095428	− 1.0316	− 1.0315	5.5489E−4	0.412668
*F*17	0.3979	0.3979	7.2995E−6	0.090751	0.3979	0.4016	0.0199	0.403244
*F*18	3.0000	3.0001	1.8279E−4	0.092633	3.0000	3.0000	2.1246E−7	0.401207
*F*19	− 3.8628	− 3.8627	1.0269E−4	0.124587	− 3.8628	− 3.8578	0.0081	0.433102
*F*20	− 3.3179	− 3.2888	0.0258	0.129720	− 3.3217	− 3.2364	0.0953	0.447982
*F*21	− 10.1029	− 9.0731	0.9759	0.149500	− 10.1532	− 5.6266	3.5789	0.456901
*F*22	− 10.3858	− 9.4701	0.7201	0.167090	− 10.4029	− 9.3773	2.6150	0.476818
*F*23	− 10.4778	− 9.1012	0.8841	0.181062	− 10.5364	− 10.5361	0.0015	0.487832

Table 8 Results of different methods for solving the composite test functions

F	GROM				ABC			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*24	1.3254E−29	4.9821E−21	9.8425E−25	265.2489	0.1811	1.4526	1.6171	129.0839
*F*25	1.85201	23.75416	19.4826	251.2815	25.3678	50.7242	14.2701	115.6244
*F*26	124.5176	165.9840	38.4128	254.104	155.3707	188.6470	26.3783	115.2801
*F*27	211.48966	263.7109	48.9548	284.2548	343.8729	401.3431	46.5083	139.5159
*F*28	0	34.41452	29.4751	295.4145	8.3465	26.1306	13.6009	137.6828
*F*29	440.9783	495.41875	98.4126	274.2268	406.9140	490.3227	40.3188	137.5745
	LAPO				**LSA**			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*24	0	7.7214E−23	4.2143E−23	264.797980	12.14	120.000	103.279	141.483
*F*25	2.7340	29.87214	32.23484	244.215891	4.446	148.26	137.7999	285.57
*F*26	116.0883	175.86457	90.15781	247.254191	147.312	257.40	115.45	268.814
*F*27	265.2541	312.1458	67.9681	277.169869	286.999	483.162	136.21	309.031
*F*28	0	45.8928	56.4218	279.241119	3.4655	155.64	154.408	312.72
*F*29	500.000	544.8186	119.6597	254.226818	500.411	782.051	193.86	323.642
	DE				**SFLA**			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*24	4.7080E−24	52.2576	69.0770	131.1335	2.3809E−6	70.0002	78.1025	102.3544
*F*25	7.4468	94.8094	91.3860	114.5313	52.1968	198.5301	120.2203	101.3876
*F*26	163.9931	223.8490	81.1158	111.2575	273.8043	400.3250	116.3799	95.6668
*F*27	284.2209	326.3868	43.1065	143.6823	408.1973	443.0856	35.3076	137.0135
*F*28	4.3712	92.2381	83.2146	140.4139	12.9486	82.5981	85.9617	182.5330
*F*29	500.0000	741.9855	196.6948	136.8673	501.9177	697.4181	188.0246	114.8238

Table 8 continued

	ICA				PSO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*24	1.5873E−18	70.0000	100.4988	160.0497	0.3053	186.3420	138.9956	132.5921
*F*25	22.9368	174.3427	78.8243	133.2649	28.2429	203.1709	159.6724	114.8665
*F*26	160.9427	363.9297	122.3701	122.9039	271.7938	396.7644	86.3684	111.5002
*F*27	471.4093	599.9630	69.9404	143.7568	337.6973	474.7387	148.4452	139.3360
*F*28	12.4696	59.2129	28.9372	152.4287	40.5131	241.2266	123.3227	136.3697
*F*29	506.3825	784.8621	181.8203	149.5143	515.2240	833.4793	150.7666	136.9043
	ALO				GWO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*24	1.2055E−9	1.6480E−9	6.4372E−10	148.6606	80.8012	224.3565	128.8061	34.4861
*F*25	140.0910	162.7753	32.4937	135.2122	172.4628	302.5475	108.3346	125.3465
*F*26	229.6219	439.6753	236.0181	130.2807	150.6736	303.6108	182.9697	118.2997
*F*27	334.1691	485.3973	136.6477	139.3494	287.2757	388.5935	138.7823	137.3359
*F*28	3.4804	4.8742	1.5446	138.7856	11.1285	73.1323	65.5494	138.5388
*F*29	500.3315	769.7003	233.2831	139.4281	902.3750	905.2132	2.3131	135.0984
	CSA				Firefly			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*24	0.0034	2.0362	3.3849	132.3139	0.1841	40.0000	48.9898	138.7930
*F*25	5.0744	42.4796	50.4552	124.8886	1.7922	78.4052	78.7343	123.2573
*F*26	214.0799	253.2691	38.7039	115.3250	109.3281	146.0378	29.8617	117.3834
*F*27	287.6394	335.8235	24.9870	130.8379	268.6814	284.2349	14.8796	139.4033
*F*28	5.8431	9.7506	2.2546	139.9418	1.1924	52.6011	49.0204	141.2526
*F*29	411.1958	525.5262	97.3582	142.2612	500.4599	782.2253	184.0946	143.8334

Table 9 Results of Wilcoxon test for different methods in solving 29 test functions

	ABC	LAPO	LSA	DE	SFLA	ICA	PSO	ALO	GWO	CSA	Firefly
*F*1	9.4E−14	3.1E−5	5.5E−1	2.4E−1	1.5E−8	1.1E−3	7.3E−12	1.1E−3	2.6E−9	7.3E−12	8.7E−13
*F*2	9.4E−14	5.6E−11	1.7E−2	2.9E−3	9.4E−14	8.7E−13	9.4E−14	4.0E−10	8.7E−13	9.4E−14	9.4E−14
*F*3	9.4E−14	2.4E−1	8.3E−8	1.1E−4	9.4E−14	9.4E−14	3.1E−5	9.4E−14	7.9E−6	9.4E−14	9.4E−14
*F*4	9.4E−14	9.4E−14	8.7E−13	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	1.9E−6	9.4E−14	9.4E−14
*F*5	4.0E−10	9.4E−14	1.4E−8	3.5E−4	1.5E−8	1.4E−1	7.9E−6	3.1E−5	9.4E−14	8.7E−13	7.9E−6
*F*6	7.3E−12	6.2E−7	8.7E−13	3.7E−2	7.3E−12	5.6E−11	4.1E−7	8.7E−13	9.4E−14	4.0E−10	9.4E−14
*F*7	9.4E−14	5.6E−9	9.4E−14	9.4E−14	9.4E−14	2.4E−1	7.3E−12	9.4E−14	8.7E−13	8.7E−13	9.4E−14
*F*8	3.1E−13	9.9E−3	6.2E−4	1.2E−2	9.4E−14	6.8E−2	2.8E−5	3.6E−12	4.7E−6	4.6E−2	7.8E−12
*F*9	9.4E−14	3.4E−4	9.4E−14	9.4E−14	9.4E−14	1.1E−13	9.4E−14	9.4E−14	4.0E−3	9.4E−14	9.4E−14
*F*10	9.4E−14	3.2E−6	9.4E−14	9.4E−14	9.4E−14	4.0E−10	9.4E−14	9.4E−14	2.5E−12	9.4E−14	9.4E−14
*F*11	9.4E−14	3.1E−3	3.1E−5	1.1E−3	9.4E−14	4.0E−10	9.4E−14	8.7E−13	7.4E−2	9.4E−14	9.4E−14
*F*12	9.4E−14	9.2E−2	2.3E−4	2E−6	9.4E−14	3.3E−1	7.3E−12	7.3E−12	9.3E−13	9.4E−14	6.4E−1
*F*13	9.6E−12	2.3E−1	1.4E−4	7.3E−12	4.0E−10	9.3E−1	8.3E−8	4.1E−7	9.4E−14	9.4E−14	9.1E−1
*F*14	4.7E−2	4.7E−2	9.6E−1	2.4E−1	2.9E−8	4.7E−2	8.7E−2	3.1E−9	1.2E−7	1.5E−1	4.7E−8
*F*15	3.9E−10	3.9E−10	2.9E−3	4.1E−7	2.5E−9	1.1E−4	7.4E−3	1.1E−4	1.1E−3	9.4E−14	5.6E−11
*F*16	9.4E−14	6.9E−1	6.2E−1	6.2E−1	6.2E−1	6.2E−1	6.2E−1	6.2E−1	3.7E−2	9.0E−1	2.4E−1

Table 9 (continued)

	ABC	LAPO	LSA	DE	SFLA	ICA	PSO	ALO	GWO	CSA	Firefly
*F*17	1.1E−3	3.1E−5	1.1E−17	1.1E−17	1.1E−17	3.7E−1	9.4E−14	9.4E−14	3.5E−4	9.4E−14	2.4E−1
*F*18	7.4E−6	6.7E−14	7.2E−14	7.3E−14	7.8E−14	7.9E−14	2.3E−1	7.9E−14	7.9E−14	7.9E−14	7.9E−14
*F*19	8.8E−14	6.6E−14	5.5E−14	8.3E−14	2E−14	8.8E−14	1.1E−4	3.7E−2	2.9E−3	8.8E−14	3.0E−5
*F*20	4.7E−2	1.6E−1	5.3E−3	1.6E−3	1.4E−1	4.7E−2	1.5E−1	2E−1	1.7E−3	1.9E−1	1.5E−3
*F*21	4.9E−12	9.5E−5	1.6E−2	5.9E−14	3.3E−7	3.3E−7	6.7E−6	5.9E−14	1.6E−6	1.2E−8	1.9E−9
*F*22	7.3E−3	2.3E−1	6.5E−12	1.8E−6	7.5E−6	7.7E−1	2.9E−5	3.9E−7	1.4E−1	1.4E−8	1.0E−3
*F*23	1.0E−3	7.3E−3	1.1E−4	3.0E−5	7.8E−8	3.4E−4	3.0E−5	2.4E−9	5.5E−1	3.7E−10	8.4E−14
*F*24	4.1E−7	9.4E−14	2.6E−9	1.1E−4	8.3E−8	3.7E−2	8.3E−8	9.4E−14	9.4E−14	3.7E−2	1.9E−6
*F*25	9.9E−7	8.2E−1	2.7E−7	8.8E−2	5.5E−12	1.3E−12	1.2E−9	1.1E−13	9.4E−14	1.7E−2	1.8E−4
*F*26	6.7E−4	4.4E−1	2.3E−4	3.4E−4	2.9E−13	5.2E−11	9.4E−14	6.8E−11	7.6E−3	1.7E−11	4.0E−3
*F*27	5.6E−13	1.3E−2	1.4E−11	9.2E−5	1.2E−13	9.4E−14	5.1E−10	1.5E−8	7.2E−6	2.6E−9	7.4E−2
*F*28	1.2E−1	1.5E−2	1.7E−7	1.1E−4	1.6E−5	1.6E−6	1.1E−12	5.4E−5	5.3E−4	2.7E−4	3.2E−2
*F*29	1.3E−1	4.6E−4	6.4E−11	2E−8	8.3E−8	3.9E−9	4.8E−13	1.3E−8	9.4E−14	3.2E−1	1.3E−9

Table 10 Results of different methods for solving the 200-dimensional version of unimodal test functions, with the population size of 200 and the 2000 iterations

F	GROM				ABC			
					Best	Ave	Std	Time (s)
*F*1	4.0197e−283	5.0162e−283	0	66.119	8.9499E3	1.4148E4	1.9222E3	27.5773
*F*2	1.1623e−156	1.6258e−156	3.2777e−157	63.418	73.0196	81.6573	5.8675	28.9298
*F*3	1.2908e−207	1.7886e−191	0	207.35	4.7612E5	6.5662E5	7.3296E4	163.0073
*F*4	6.8726e−122	7.808e−122	7.3555e−123	65.461	190.5305	192.4391	0.7630	28.6786
*F*5	189.62	190.04	0.30596	66.456	7.2876E6	1.1209E7	1.9375E6	30.9131
*F*6	0.14347	0.203	0.064899	65.846	1.0346E4	1.3273E4	2.1238E3	30.4218
*F*7	1.9492e−5	5.4613e−5	1.1139e−6	49.699	23.0871	32.1411	5.5446	43.1471
	LAPO				Firefly			
	Best	Ave	Std	Time (s)	Time (s)	Ave	Std	Time (s)
*F*1	3.7643E−29	8.1296E−29	3.2204E−29	40.1957	0.1528	196.5869	0.0068	381.8076
*F*2	1.4339E−15	1.8098E−15	2.7520E−16	41.3853	2.5763	1.0000E+10	0.1915	385.0120
*F*3	0.4160	2.6872	3.5371	315.1508	6.5049E4	1.9185E5	5.8542E3	527.6232
*F*4	2.0925E−10	7.2703E−10	3.3558E−10	45.4820	52.1716	14.0077	4.3816	372.9640
*F*5	192.6322	193.0862	0.3316	47.1548	203.4103	6.7598E3	115.6933	383.8150
*F*6	0.0283	0.0412	0.0072	44.7716	0.1427	192.7601	0.0079	378.1490
*F*7	2.8619E−5	1.8284E−4	1.3097E−4	71.8251	0.0859	0.6048	0.0261	393.2078
	DE				SFLA			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*1	0.0094	3.4267	2.9338	18.4324	164.9749	275.9424	62.8101	14.1161
*F*2	0.5247	5.9106	6.6286	19.3465	17.2986	21.7154	1.9108	15.8777
*F*3	3.8709E5	4.6516E5	4.7067E4	154.4452	5.5593E3	7.1881E3	1.0812E3	82.5222
*F*4	46.7385	52.1769	2.8675	20.8550	10.0687	11.7461	0.7967	32.3306
*F*5	3.6550E3	2.7380E4	3.0000E4	21.3643	3.3207E3	4.2391E3	664.7247	20.5108
*F*6	0.0137	1.7010	3.0727	20.9296	184.4612	270.8978	74.0124	16.4177
*F*7	0.2162	0.6486	0.5584	34.2304	0.2077	0.2268	0.0140	62.8454

	ICA				PSO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*1	3.6177E−5	4.9125E−5	1.107E−5	29.8373	22.2761	42.9502	10.0862	9.0901
*F*2	0.0018	0.0024	4.8728E−4	30.3026	3.1852	8.2847	2.8917	10.0612
*F*3	2.5280E4	3.4281E4	4.7509E3	187.6643	1.9931E3	6.5972E3	4.6773E3	147.541
*F*4	11.5323	14.6959	1.9066	29.9090	2.2303	2.8657	0.6449	11.4063
*F*5	546.6862	685.3896	85.6767	32.8787	340.4599	540.7680	108.3468	11.8282
*F*6	2.7313E−5	4.4321E−5	1.2661E−5	30.2679	47.2150	80.7581	16.8446	12.0781
*F*7	0.1920	0.2377	0.0330	45.2809	4.4161E−4	0.0114	0.0098	25.0383
	ALO				GWO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*1	3.4918E−3	5.3074E−3	1.8514E−3	7943.24	4.5190E−38	2.2323E−37	1.2855E−37	22.4423
*F*2	652.3152	889.2314	312.2145	7542.31	1.8924E−20	5.2932E−20	3.0364E−20	23.2889
*F*3	2.857E3	3.2144E3	410.3142	24231.23	2.4381E−8	2.0677E−5	4.1389E−5	156.0946
*F*4	50.1289	52.4897	3.32458	8023.197	9.2622E−8	0.0014	0.0037	23.5751
*F*5	183.5943	203.1251	42.2674	8415.12	194.8496	196.6516	0.7670	26.1214
*F*6	2.124E−3	0.01437	0.0097	7625.23	17.5594	19.9656	1.2912	25.2861
*F*7	1.3985	3.2457	1.9875	8326.15	1.2338E−4	3.2376E−4	1.0256E−4	38.8676

Table 11 Results of different methods for solving the 200-dimensional version of multimodal test functions, with the population size of 200 and the 2000 iterations

F	GROM				ABC			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 55179	− 53752	1010.1	42.868	− 5.87116E4	− 5.799E4	516.7504	34.319
*F*9	0	0	0	48.868	5.5206E2	581.1772	15.0524	33.449
*F*10	8.8817E−16	8.8818E−16	0	57.541	12.057293	12.5316	0.2296	35.306
*F*11	0	0	0	61.278	1.10033E2	122.2712	9.9774	35.508
*F*12	1.4254E−5	5.3593E−5	7.2927E−5	88.792	7.99036E5	2.765E6	9.0463E5	65.586
*F*13	0.09547	1.0157	1.4678	93.0254	2.84168E6	1.919E7	1.0240E7	65.177
	LAPO				Firefly			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 5.617E4	− 5.5361E4	527.2355	52.582	− 4.0562E4	− 3.492E4	3.1571E3	380.843
*F*9	0	0	0	53.500	166.6876	194.7440	15.6631	378.833
*F*10	8.8817E−16	8.8818E−16	0	54.894	0.0905	0.1003	0.0061	385.306
*F*11	0	0	0	57.355	0.0260	0.0300	0.0023	386.951
*F*12	8.64884E−5	1.1944E−4	2.2692E−5	118.133	2.1867E−4	0.0036	0.0100	415.116
*F*13	0.071735	0.1748	0.0712	117.209	0.0151	0.0181	0.0026	412.845
	DE				SFLA			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 6.106E4	− 5.819E+4	2.16E3	25.2330	− 4.868E4	− 3.88E+4	4.8031E3	39.526
*F*9	450.733	570.464	86.3149	24.4353	405.393	474.705	49.512	27.892
*F*10	5.5220	8.0100	1.5294	24.9623	5.4943	6.9422	0.7936	33.824
*F*11	0.0844	0.7206	0.4426	25.8570	3.0548	3.5738	0.5035	20.739

Table 11 (continued)

	DE				SFLA			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*12	8.9614	10.634	1.6729	56.7757	4.0529	5.4245	0.6846	64.482
*F*13	187.8197	229.396	27.495	56.5316	180.854	226.089	26.383	66.7729

	ICA				PSO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 5.432E4	− 5.0741E4	2.390E3	33.7845	− 4.1848E4	− 3.885E+4	3.4655E3	29.3711
*F*9	292.5286	422.3477	59.2422	33.0526	221.1672	424.0131	99.6931	28.2308
*F*10	0.0123	0.0620	0.0834	33.9011	0.5767	1.4667	0.3718	29.3400
*F*11	1.0037E−5	7.5631E−4	0.0022	35.6621	1.0995	1.3244	0.1599	30.7325
*F*12	9.2675E−5	0.1099	0.3190	69.7339	0.3038	0.4822	0.1204	61.7301
*F*13	2.3728E−5	5.0033E−5	2.6147E−5	70.2588	20.7460	28.2002	3.7339	61.9184

	ALO				GWO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 3.8756E4	− 3.754E4	231.24	8.976E3	− 3.4619E4	− 3.046E4	1.8936E3	25.1740
*F*9	310.2421	572.89	110.158	8.4474E3	0	0	0	27.2447
*F*10	7.53781	8.19321	1.3152	8.759E3	2.9310E−14	3.6060E−14	3.7091E−15	27.9993
*F*11	0.02389	0.0312	0.1134	8.8255E3	0	1.1102E−17	3.3307E−17	29.7272
*F*12	4.32151	6.5876	3.5487	1.8551E4	0.2177	0.2606	0.0264	61.0197
*F*13	0.8965	1.3985	1.3215	1.9231E4	11.7505	12.8678	0.5411	60.2916

3.7 Solving engineering problems by GROM

In order to evaluate the performance of the proposed method in solving real engineering optimization problems, five classic mechanical engineering problems are employed. These problems are tension/compression spring, welded beam, pressure vessel designs, gear train design, and cantilever beam design. In addition to these test functions, an electrical optimization problem that is associated with lots of constraints is also taken into consideration. It should be noted that all constraints are handled by means of penalty factors. In other words, whenever a constraint is violated, a large value is added to the objective function.

3.7.1 Tension/compression spring design

This problem, which is depicted in Fig. 8, is minimization of the weight of a tension/compression spring (Hu and Eberhart 2002; Mirjalili 2015; Shareef 2015). There are three variables and a number of constraints including shear stress, surge frequency, and minimum deflection, which must be satisfied while solving the procedure. The formulation of the problem is as follows:

$$\text{Function}\quad f(x) = (x_3+2)x_2x_1^2 \tag{10}$$

$$\text{Inequality constraints}\quad \begin{aligned} g_1(x) &= 1-\frac{x_2^3x_3}{71785x_1^4}\le 0\\ g_2(x) &= \frac{4x_2^2-x_1x_2}{12566(x_2x_1^3-x_1^4)}+\frac{1}{5108x_1^4}\le 0\\ g_3(x) &= 1-\frac{140.45x_1}{x_2^2x_3}\le 0\\ g_4(x) &= \frac{x_1+x_2}{1.5}-1\le 0\end{aligned} \tag{11}$$

$$\text{Variable range}\quad \begin{aligned} 0.05 \le x_1 \le 2.00\\ 0.25\le x_2\le 1.30\\ 2.00\le x_3\le 15.0\end{aligned}. \tag{12}$$

In these equations, x_1, x_2, and x_3 are wire diameter (d), mean coil diameter (D), and the number of active coils (N). The results of solving this problem with different methods are listed in Table 16. It is obvious that the best results belong to LAPO followed by the proposed method.

3.7.2 Welded beam design

The second problem is minimization of the fabrication cost of a welded beam as illustrated in Fig. 9 (Coello 2000). The thickness of weld (h), length of attached part of bar (l),

Table 12 Results of Wilcoxon test for different methods in solving the 200-dimensional version of multimodal test functions, with the population size of 200 and 2000 iterations

	ABC		LAPO		Firefly		DE		SFLA		ICA		PSO		ALO		GWO	
*F*1	2.E−14	2.E−14	2.E−14	4.4E−6	2E−14	2E−14	2E−14	2E−14	1.9E−12	2E−14	2E−14	2.E−14	4.4E−6	2E−14	2E−14	2E−14	2E−14	1.9E−12
*F*2	9.4E−14	9.4E−14	9.4E−14	1.1E−4	9.4E−14	9.4E−14	9.4E−14	9.4E−14	8.7E−13	9.4E−14	9.4E−14	9.4E−14	1.1E−4	9.4E−14	9.4E−14	9.4E−14	9.4E−14	8.7E−13
*F*3	2E−14	5.9E−3	2E−14	2E−14	2E−14	2E−14	1.3E−10	2E−14	5.9E−3	2E−14	5.9E−3	2E−14	2E−14	2E−14	2E−14	1.3E−10	2E−14	5.9E−3
*F*4	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	7.4E−3	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	7.4E−3
*F*5	9.4E−14	9.4E−14	9.4E−14	2.6E−9	9.4E−14	9.4E−14	9.4E−14	2.4E−1	9.4E−14	9.4E−14	9.4E−14	9.4E−14	2.6E−9	9.4E−14	9.4E−14	9.4E−14	2.4E−1	9.4E−14
*F*6	9.4E−14	9.4E−14	9.4E−14	3.6E−2	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	3.6E−2	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14
*F*7	9.4E−14	2.3E−6	9.4E−14	4.0E−10	9.4E−14	9.4E−14	3.1E−5	4.0E−10	9.4E−14	9.4E−14	2.3E−6	9.4E−14	4.0E−10	9.4E−14	9.4E−14	3.1E−5	4.0E−10	9.4E−14
*F*8	1.0E−13	7.9E−10	9.4E−14	9.0E−12	9.4E−14	4.2E−8	9.4E−14	9.4E−14	9.4E−14	1.0E−13	7.9E−10	9.4E−14	9.0E−12	9.4E−14	4.2E−8	9.4E−14	9.4E−14	9.4E−14
*F*9	2E−14	Na	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	Na	2E−14	Na	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	Na
*F*10	2E−14	Na	2E−14	2E−14	2E−14	1.8E−5	2E−14	2E−14	2E−14	2E−14	Na	2E−14	2E−14	2E−14	1.8E−5	2E−14	2E−14	2E−14
*F*11	2E−14	Na	2E−14	1.9E−12	2E−14	1.5E−2	2E−14	1.5E−2	7.7E−4	2E−14	Na	2E−14	1.9E−12	2E−14	1.5E−2	2E−14	1.5E−2	7.7E−4
*F*12	9.4E−14	2.3E−4	7.3E−2	9.4E−14	9.4E−14	7.4E−3	9.4E−14	5.6E−11	9.4E−14	9.4E−14	2.3E−4	7.3E−2	9.4E−14	9.4E−14	7.4E−3	9.4E−14	5.6E−11	9.4E−14
*F*13	4.0E−10	1.7E−5	6.9E−7	9.4E−14	9.4E−14	6.9E−7	9.4E−14	9.5E−1	9.4E−14	4E−10	1.7E−5	6.9E−7	9.4E−14	9.4E−14	6.9E−7	9.4E−14	9.5E−1	9.4E−14

the height of the bar (t), and thickness of the bar (b) are optimization variables. The constraints are also shear stress (s), bending stress in the beam (h), buckling load on the bar (Pc), end deflection of the beam (d), and side constraints. The mathematical formulation of this problem is as follows:

$$\text{Function} \quad f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \tag{13}$$

$$\text{Inequality constraints} \quad \begin{aligned} &g_1(x) = \tau(x) - \tau_{\max} \le 0 \\ &g_2(x) = \sigma(x) - \sigma_{\max} \le 0 \\ &g_3(x) = \delta(x) - \delta_{\max} \le 0 \\ &g_4(x) = x_1 - x_4 \le 0 \\ &g_5(x) = P - P_c(x) \le 0 \\ &g_6(x) = 0.125 - x_1 \le 0 \\ &g_7(x) = 1.10471x_1 + 0.04811x_3x_4(14 + x_2) \\ &\qquad -5 \le 0 \end{aligned} \tag{14}$$

$$\text{Variable range} \quad \begin{aligned} 0.1 \le x_1 \le 2.00 \\ 0.1 \le x_2 \le 10 \\ 0.1 \le x_3 \le 10 \\ 0.1 \le x_4 \le 2 \end{aligned} \tag{15}$$

$$\text{where} \quad \begin{aligned} &\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2} \\ &\tau' = \frac{P}{\sqrt{2}x_1x_2}, \tau'' = \frac{\mathrm{MR}}{J}, M = P\left(L + \frac{x_2}{2}\right) \\ &R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{x}\right)^2} \\ &J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{x}\right)^2\right]\right\} \\ &\sigma(x) = \frac{6PL}{x_4x_3^2}, \delta(x) = \frac{6PL^3}{Ex_4x_3^2} \\ &P_c(x) = \frac{4.013\sqrt{\frac{x_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \\ &P = 6000 \text{ lb}, \ L = 14 \text{ in.}, \ \delta_{\max} = 0.25 \text{ in.}, \\ &E = 30 \times 10^6 \text{ psi}, \\ &G = 12 \times 10^6 \text{ psi}, \tau_{\max} = 13600 \text{ psi}, \sigma_{\max} = 30000 \text{ psi} \end{aligned} \tag{16}$$

The results of different method for solving this problem are shown in Table 17. It is obvious that the proposed method has the best results followed by LAPO.

3.7.3 Pressure vessel design

The minimization of the total cost including material, forming, and welding of a cylindrical vessel is the third problem which is shown in Fig. 10. Decision variables of this problem are thickness of the shell (Ts), thickness of the head (Th), inner radius (R), and length of the cylindrical

Table 13 Results of different methods for solving the 200-dimensional version of multimodal test functions, with the population size of 40 and the 500 iterations

F	GROM				ABC			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*1	3.1809e−65	5.3021e−65	1.6155e−65	3.2856	2.4360E5	2.7219E5	1.2199E4	0.856961
*F*2	3.7697e−36	5.2158e−36	1.0254e−36	3.199	1.4795E+14	2.9094E+22	1.0439E+23	6.386335
*F*3	1.1586e−62	2.8633e−57	2.553e−57	10.584	1.1942E6	1.2362E6	4.2002E4	7.589553
*F*4	3.8682e−30	4.0382e−30	1.4237e−31	3.3257	93.1279	95.5376	1.2093	0.852726
*F*5	195.44	197.6	0.55796	3.4171	8.1614E8	9.4098E8	7.299E7	0.977426
*F*6	16.756	17.942	1.1032	3.3551	2.3291E5	2.7077E5	1.4489E4	0.931721
*F*7	1.7976E−4	2.8908E−4	1.1115E−4	2.5216	2.3945E3	3.0210E3	310.5942	1.525904
	LAPO				Firefly			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*1	1.0288E−12	1.8090E−9	6.1216E−9	1.871493	0.9584	2.3924	1.6209	3.808463
*F*2	2.0087E−7	3.5023E−6	4.3034E−6	1.901820	42.5075	69.0379	13.9463	3.836813
*F*3	0.9660	6.7051	5.7391	15.430342	1.6087E5	2.3346E5	3.5424E4	11.036595
*F*4	9.8895E−6	2.4845E−5	1.4955E−5	2.108931	70.0208	74.2297	2.4547	3.604922
*F*5	196.3227	197.3273	0.5454	2.148707	1.4679E3	3.6955E3	2.5851E3	4.065498
*F*6	13.7919	16.5038	1.3669	2.098711	0.9638	1.9872	0.8689	4.118451
*F*7	3.5309E−4	0.0010	4.5410E−4	3.332803	1.1936	2.2279	0.5156	4.537479
	DE				SL*F*A			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*1	3.0715E4	4.4865E4	8.7919E3	0.984315	1.8473E3	2.8660E3	550.3588	2.794788
*F*2	171.2055	232.9325	36.3980	1.010780	49.9520	57.6806	3.8804	3.256660
*F*3	5.4828E5	6.8268E5	8.2453E4	7.839305	2.4304E4	3.2983E4	6.0560E3	21.028389
*F*4	52.8730	59.1089	3.6240	1.049797	12.0156	15.4449	1.2661	5.204148
*F*5	2.6482E7	5.8002E7	1.8757E7	1.142108	9.0164E4	1.4844E5	4.3592E4	3.199952
*F*6	3.0321E4	4.8823E4	1.1315E4	1.094466	2.1183E3	2.8351E3	533.9642	3.164061
*F*7	64.1714	151.7041	55.7205	1.646033	0.4357	0.6017	0.0838	23.22358
	ICA				PSO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*1	1.5421E3	2.4932E3	427.4432	1.562873	2.3606E3	4.6042E3	1.6342E3	0.542811
*F*2	51.3144	73.8622	9.1490	1.562673	50.9413	88.4166	24.2348	0.554422
*F*3	1.0483E5	2.4815E4	2.3915E4	8.660863	4.9300E4	1.7849E5	9.6867E4	6.996546
*F*4	36.5269	43.1785	3.5339	1.614045	12.3977	20.2241	3.2475	0.606078
*F*5	1.3918E5	2.7371E5	8.9298E4	1.659135	1.1221E5	4.1913E5	3.2195E5	0.638427
*F*6	1.8802E3	2.4839E3	598.6150	1.612699	1.3346E3	5.8691E3	2.5229E3	0.645103
*F*7	4.2427	0.7137	5.6604	2.310599	0.4911	1.9675	1.5525	1.178825
	ALO				GWO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*1	1.9446E02	2.8545E02	7.8596E01	187.21457	5.0773E−9	1.8412E−8	1.1280E−8	1.0913
*F*2	273.293	325.324	91.257	190.15789	7.0949E−6	9.0642E−6	1.4596E−6	1.1181
*F*3	2.6191E4	3.1931E4	1.0249E4	182.12503	6.3143E3	1.5097E4	8.3781E3	7.8298
*F*4	20.2158	175.21458	142.5678	184.174021	14.4764	21.5119	5.7528	1.2095
*F*5	3.9246E6	7.8716E6	6.0127E6	194.995556	195.8662	197.7457	0.7064	1.3368
*F*6	1.8288E4	2.9352E4	6.2965E3	185.243149	26.2291	27.9390	1.3802	1.2796
*F*7	14.7902	25.6137	9.0479	197.237721	0.0060	0.0105	0.0029	1.9543

Table 14 Results of different methods for solving the 200-dimensional version of multimodal test functions, with the population size of 40 and the 500 iterations

F	GROM				ABC			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 15765	− 15167	430.67	2.1547	− 3.4035E4	− 3.294E4	734.7127	0.9800
*F*9	0	0	0	12.037	1.8525E3	1.9515E3	45.5173	1.0466
*F*10	8.8818e−16	8.8818e−16	0	10.181	19.5318	19.6937	0.0774	1.1107
*F*11	0	0	0	11.648	2.1671E3	2.4488E3	111.3545	1.1884
*F*12	0.15496	0.1663	0.00805	4.4994	1.6032E9	2.0864E9	2.1788E8	2.7197
*F*13	12.803	12.822	0.024253	4.518	3.3108E9	4.0474E9	3.1469E8	2.566407
	LAPO				Firefly			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 1.5279E4	−1.324E4	815.8747	2.23657	− 4.4063E4	− 3.6144E+4	3.7575E3	4.155578
*F*9	4.5475E−13	9.2776E−10	3.6949E−10	2.242719	482.2428	566.8940	49.7806	4.085035
*F*10	7.6647E−8	4.2081E−6	2.8691E−6	2.267153	1.0800	1.7067	0.2831	4.144103
*F*11	2.8019E−12	5.6723E−10	2.8363E−10	2.431825	0.2400	0.3097	0.0597	4.206908
*F*12	0.0995	0.0149	0.1178	5.632202	6.4607	7.8803	0.8738	5.606936
*F*13	12.7156	1.9412	15.7881	5.562711	189.4229	252.7130	36.7520	5.661512
	DE				SL*FA*			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 5.4017E4	− 4.977E4	2.0663E3	1.1282	− 3.0403E4	− 2.649E4	2.3060E3	5.7415
*F*9	731.6366	964.7553	114.3569	1.2444	880.8990	1.0968E3	98.0936	4.1798
*F*10	14.5240	16.1965	0.7866	1.2590	6.7096	7.8612	0.7120	4.5330
*F*11	235.9151	440.5763	122.8757	1.3054	19.9710	30.0968	5.4912	4.0709
*F*12	2.6572E6	2.6120E+7	1.6150E7	2.8559	7.6205	11.2009	1.8581	9.4083
*F*13	3.9146E7	1.3105E8	7.2073E7	2.7561	236.0627	335.8872	108.7458	9.890213
	ICA				PSO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 4.7091E4	− 4.055E4	3.9585E3	1.60234	− 2.9405E4	− 2.4118E4	3.0779E3	1.307349
*F*9	367.8463	527.6486	78.9449	1.704648	652.9414	1.2740E3	154.8052	1.441813
*F*10	7.3976	9.1514	0.7618	1.767502	5.0312	8.4714	3.2569	1.454020
*F*11	17.7185	25.3039	3.1858	1.858962	22.3847	50.5061	19.7840	1.540186
*F*12	12.5943	18.2755	3.5619	3.554902	4.4008	10.9512	3.7138	3.138827
*F*13	849.3433	8.4779E+3	9.2190E3	3.469389	171.5073	4.7033E3	6.5763E3	3.061198
	ALO				GWO			
	Best	Ave	Std	Time (s)	Best	Ave	Std	Time (s)
*F*8	− 3.6118E4	− 3.6118E+4	0	189.5486	− 3.4022E4	− 2.5747E+4	8.6739E3	1.2000
*F*9	910.1357	1.0018E+3	171.3214	189.5022	1.0633	17.9953	11.9642	1.2994
*F*10	12.9351	14.0165	1.1293	189.7388	6.0842E−6	8.4374E−6	1.5815E−6	1.3449
*F*11	173.2457	232.9929	88.5350	189.8364	1.3757E−9	4.0790E−9	3.2534E−9	1.4355
*F*12	9.7521E4	1.541E5	1.4595E5	191.5433	0.4078	0.4781	0.0417	3.0040
*F*13	5.5724E5	5.3616E+6	3.4425E6	199.0284	15.7488	16.3107	0.4155	2.9299

Table 15 Results of Wilcoxon test for different methods in solving the 200-dimensional version of multimodal test functions, with the population size of 40 and 500 iterations

	ABC		LAPO		Firefly		DE		SFLA		ICA		PSO		ALO		GWO	
*F*1	9.4E−14	1.4E−1	2.6E−9	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	5.6E−11	9.4E−14	1.4E−1	2.6E−9	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	5.6E−11
*F*2	1.0E0	4.1E−7	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	1.0E0	4.1E−7	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14
*F*3	9.4E−14	2.6E−9	9.4E−14	9.4E−14	9.4E−14	8.3E−8	9.4E−14	9.4E−14	8.7E−13	9.4E−14	2.6E−9	9.4E−14	9.4E−14	9.4E−14	8.3E−8	9.4E−14	9.4E−14	8.7E−13
*F*4	9.4E−14	7.3E−12	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	1.5E−8	9.4E−14	9.4E−14	7.3E−12	9.4E−14	9.4E−14	9.4E−14	9.4E−14	9.4E−14	1.5E−8	9.4E−14
*F*5	9.4E−14	2.4E−1	5.6E−11	9.4E−14	9.4E−14	9.4E−14	2.6E−9	8.3E−8	2.4E−1	9.4E−14	2.4E−1	5.6E−11	9.4E−14	9.4E−14	9.4E−14	2.6E−9	8.3E−8	2.4E−1
*F*6	9.4E−14	1.8E−7	9.4E−14	9.4E−14	9.4E−14	9.4E−14	7.3E−12	9.4E−14	9.4E−14	9.4E−14	1.8E−7	9.4E−14	9.4E−14	9.4E−14	9.4E−14	7.3E−12	9.4E−14	9.4E−14
*F*7	9.4E−14	2.7E−9	9.4E−14	9.4E−14	9.4E−14	3.7E−1	5.6E−11	9.4E−14	9.4E−14	9.4E−14	2.7E−9	9.4E−14	9.4E−14	9.4E−14	3.7E−1	5.6E−11	9.4E−14	9.4E−14
*F*8	9.4E−14	1.5E−13	9.4E−14	9.4E−14	9.4E−14	9.4E−14	8.0E−13	1.1E−17	2.5E−8	9.4E−14	1.5E−13	9.4E−14	9.4E−14	9.4E−14	9.4E−14	8.0E−13	1.1E−17	2.5E−8
*F*9	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	1.9E−12	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	1.9E−12
*F*10	2E−14	9.3E−10	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	9.3E−10	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14
*F*11	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	2E−13	1.7E−11	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	2E−14	2E−13	1.7E−11
*F*12	9.4E−14	1.4E−6	9.4E−14	5.6E−11	9.4E−14	9.4E−14	9.4E−14	8.3E−8	9.4E−14	9.4E−14	1.4E−6	9.4E−14	5.6E−11	9.4E−14	9.4E−14	9.4E−14	8.3E−8	9.4E−14
*F*13	9.4E−14	2.9E−3	9.4E−14	5.6E−11	8.7E−13	1.9E−6	1.9E−6	2.6E−9	9.4E−14	9.4E−14	2.9E−3	9.4E−14	5.6E−11	8.7E−13	1.9E−6	1.9E−6	2.6E−9	9.4E−14

section without considering the head (L). This problem is formulated as follows:

$$\text{Function} f(x) = 0.6224x_1x_3x_4 + 1.7781x_3^2x_2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \tag{17}$$

$$\text{Inequality constraints} \quad \begin{aligned} g_1 &= -x_1 + 0.0193x_3 \le 0 \\ g_2 &= -x_3 + 0.0095x_3 \le 0 \\ g_3 &= -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0 \\ g_4 &= x_4 - 240 \le 0 \end{aligned} \tag{18}$$

$$\text{Variable range} \quad \begin{aligned} 0 \le x_1 \le 99 \\ 0 \le x_2 \le 99 \\ 10 \le x_3 \le 200 \\ 10 \le x_4 \le 200 \end{aligned} . \tag{19}$$

The results of solving this problem by different methods are illustrated in Table 18. It is obvious that the proposed method outperforms all other optimization methods.

3.7.4 Cantilever beam design problem

The fourth problem is to minimize the weight of the beam in the cantilever beam as explained in Fig. 11 (Chickermane and Gea 1996). This problem consists of 5 variables and also 1 vertical displacement constraint. The problem is formulated as follows:

$$\text{Function} \quad f(x) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5) \tag{20}$$

$$\text{Inequality constraints} \quad g(x) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} \le 1 \tag{21}$$

$$\text{Variable range} \quad 0.01 \le x_1, x_2, x_3, x_4, x_5 \le 100 . \tag{22}$$

The results of the proposed method for solving this problem are compared with those of other methods in Table 19. As it can be seen, the proposed method has the best results among other methods.

3.7.5 Gear train design problem

In this problem, which is a discrete one, the goal is to find the optimal number of teeth for four gears of a train with the aim of minimizing the gear ratio (Gandomi 2014; Sandgren 1990). The problem is exhibited in Fig. 12, and the mathematical formulation of this problem is as follows:

$$\text{Function} \quad f(x) = \left(\frac{1}{6.931} - \frac{x_3x_2}{x_1x_4}\right)^2 \tag{23}$$

$$\text{Variable range} \quad 12 \le x_1, x_2, x_3, x_4 \le 60 \tag{24}$$

$$\text{Where} \quad x_1, x_2, x_3, x_4 \ \text{are discrete.} \tag{25}$$

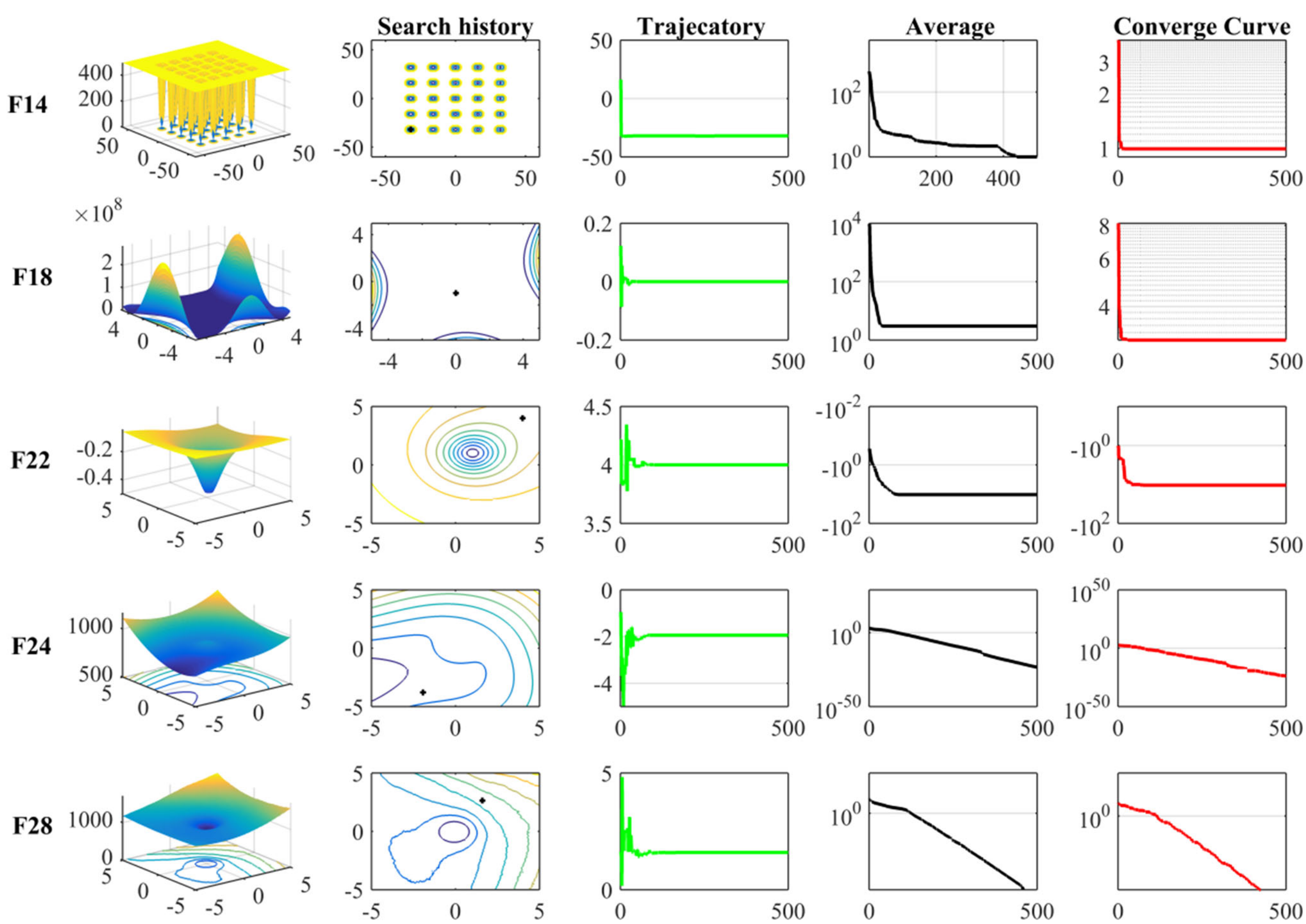


Fig. 6 Search history and trajectory of the first variable of the first solution

The results of different methods are listed in Table 20. LAPO and the proposed method have the best results. Thus, it can be concluded that the proposed method has acceptable performance in solving discrete problems.

3.7.6 Optimal power flow

The problem of optimal power flow (OPF) is the determination of generator power output such that an objective function is minimized or maximized. Different objective functions for this problem can be considered. For example, cost minimization, power loss minimization (Mirjalili et al. 2018; Rahiminejad et al. 2014; Foroughi Nematollahi et al. 2016; Moosavi et al. 2017; Foroughi Nematollahi et al. 2018), emission reduction, etc., are some of the well-known objective functions for this problem. Among these objective functions, cost minimization is the most common objective which is used for this problem. In this paper, the problem is solved based on cost minimization. This problem is illustrated in Nematollahi et al. (2017).

In this paper, the problem is solved for the IEEE 14-bus test case as a test function of which the single-line diagram is depicted in Fig. 13. The optimal results of different methods including the proposed method are listed in Table 21. As it can be seen, the proposed method and LAPO have the best results among the other methods.

Figure 14 shows the convergence behavior of GROM algorithm in solving all the engineering optimization algorithm.

Evaluation of the golden ration

In the proposed method, the golden ratio of the Fibonacci series is used to update the solutions. In this section, it is shown that this ratio is the best number which can be used in the proposed method to reach the best answer. To do so, the test functions $F5$ and $F12$ and also the problem of pressure vessel design are solved by the proposed method with different numbers instead of the golden ratio and the results are compared, as listed in Table 22. As it can be seen, the best results and the best convergence behavior belong to the method in which the golden ratio is used.

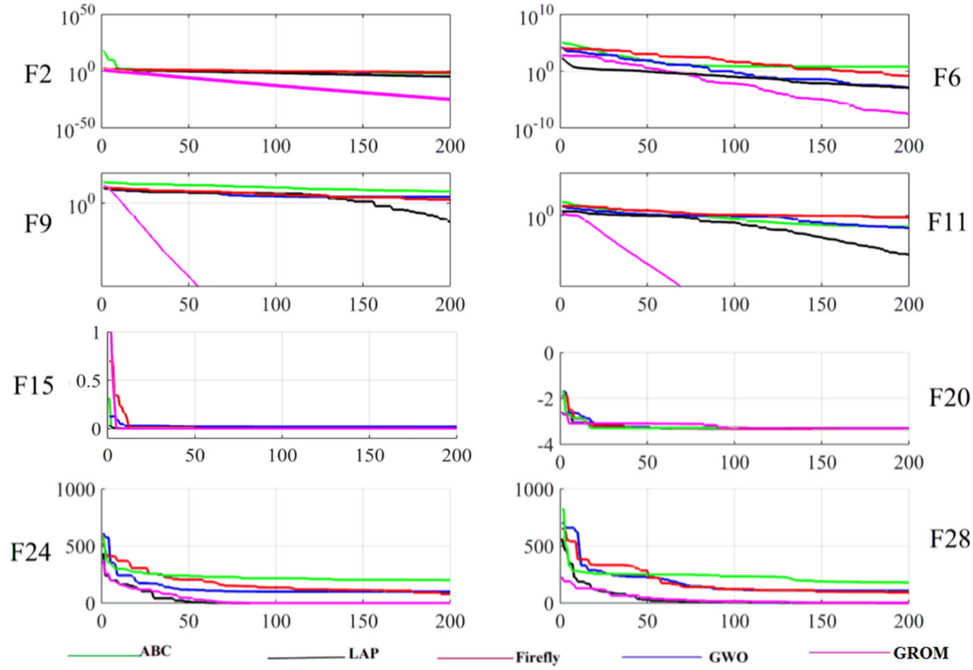


Fig. 7 Comparison of the convergence behavior of different methods

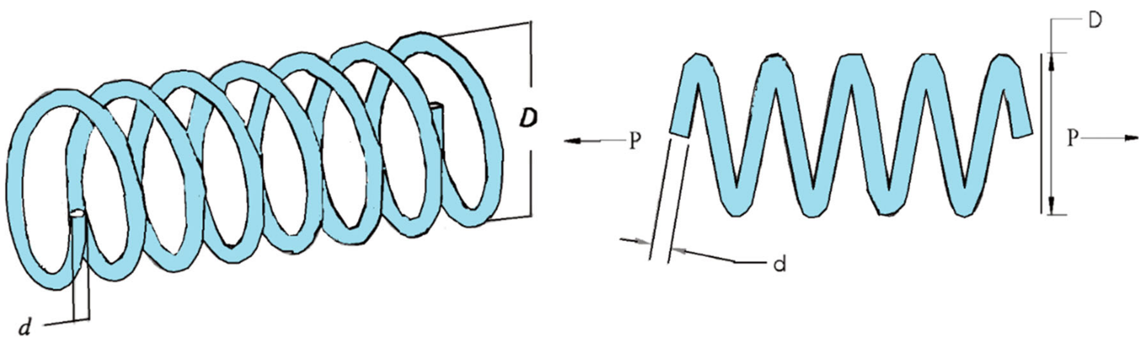


Fig. 8 Problem of tension/compression spring design

4 Conclusion

This paper proposes a new meta-heuristic optimization algorithm known as golden ratio optimization method (GROM) that is based on natural growth. The proposed method uses the golden ratio of Fibonacci series to update the solutions in two different phases. The proposed method is parameter free with simple implementation. Moreover, it is very robust and almost similar results are obtained in different trials. In order to evaluate the proposed method, four different types of test functions including unimodal, multimodal, fixed-dimensional multimodal, and composite test function are employed. Using the unimodal test functions, the ability of local search of the proposed (exploitation) is investigated, and then by the multimodal and fixed-dimensional multimodal test functions, the power of global search (exploration) of the proposed method is evaluated. Finally, these two features, i.e., exploration and exploitation, are testified simultaneously using composite test functions. In addition to these benchmark test functions, some engineering optimization problems including five mechanical and a high constraint electrical optimization problem are also employed for evaluation of the proposed method's performance. The performance of the proposed method is compared with 11 well-regarded optimization methods. To investigate the proposed method statistically, 30 trials are applied for each test function. The best results among 30 different trials reveal that the proposed method could reach a better answer in comparison with other methods. Moreover, the low values of standard deviations demonstrate the robustness of the proposed method. In addition, the CPU time consumption of the

Table 16 Optimal results of tension/compression spring design

Method	Optimal values for variables			$F_{\min}$
	d	D	N	
GROM	0.05172715179	0.357630345	11.23614371	0.01265809
LAPO (Nematollahi et al. 2017)	0.0519038638	0.361890902	11.28885	0.01265722
GWO (Mirjalili et al. 2014)	0.05169	0.356737	13.525410	0.012666
GSA (Rashedi et al. 2009)	0.050276	0.323680	11.244543	0.0127022
PSO (He and Wang 2007)	0.051728	0.357644	10.6531340	0.0126747
GA (Coello 2000)	0.051480	0.351661	11.632201	0.0127048
ES (Coello and Montes 2002)	0.051989	0.363965	10.890522	0.0126810
HS (Mahdavi et al. 2007)	0.051154	0.349871	12.076432	0.0126706
DE (Huang et al. 2007)	0.051609	0.354714	11.410831	0.0126702
Mathematical optimization (Belegundu and Arora 1985)	0.053396	0.399180	9.1854000	0.0127303
Constraint correction (Arora 2004)	0.050000	0.315900	14.250000	0.0128334

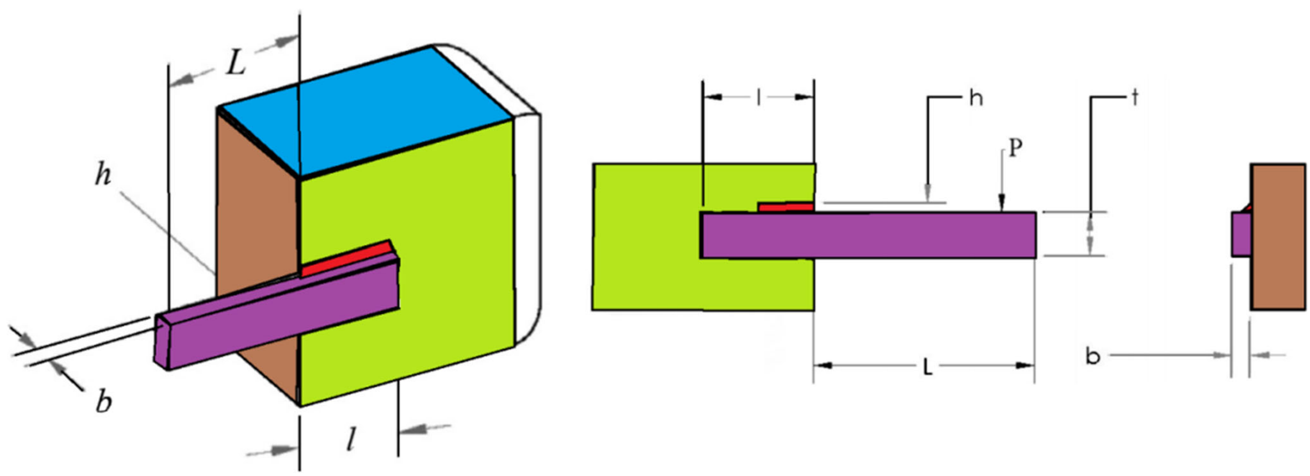


Fig. 9 Illustration of welded beam design problem

Table 17 Optimal results of welded beam design

Method	Optimal values for variables				$f_{\min}$
	h	l	t	b	
GROM	0.205530838237860	3.39469488081047	9.07663928640037	0.20553083824796	**1.7196019906343**
LAPO	0.205528028615	3.394773587424	9.076635213428	0.2055309791245	1.71960676580
GWO (Mirjalili et al. 2014)	0.205676	3.478377	9.03681	0.205778	1.72624
GSA (Mirjalili et al. 2014)	0.182129	3.856979	10.00000	0.202376	1.879952
GA (Deb 1991; Deb and Goyal 1996)	0.2489	6.1730	8.1789	0.2533	2.4331
HS (Lee and Geem 2005)	0.2442	6.2231	8.2915	0.2443	2.3807
RANDOM (Ragsdell and Phillips 1976)	0.4575	4.7313	5.0853	0.6600	2.5307
SIMPLEX (Ragsdell and Phillips 1976)	0.2792	5.6256	7.7512	0.2796	2.5307
APPROX (Ragsdell and Phillips 1976)	0.2444	6.2189	8.2915	0.2444	2.3815

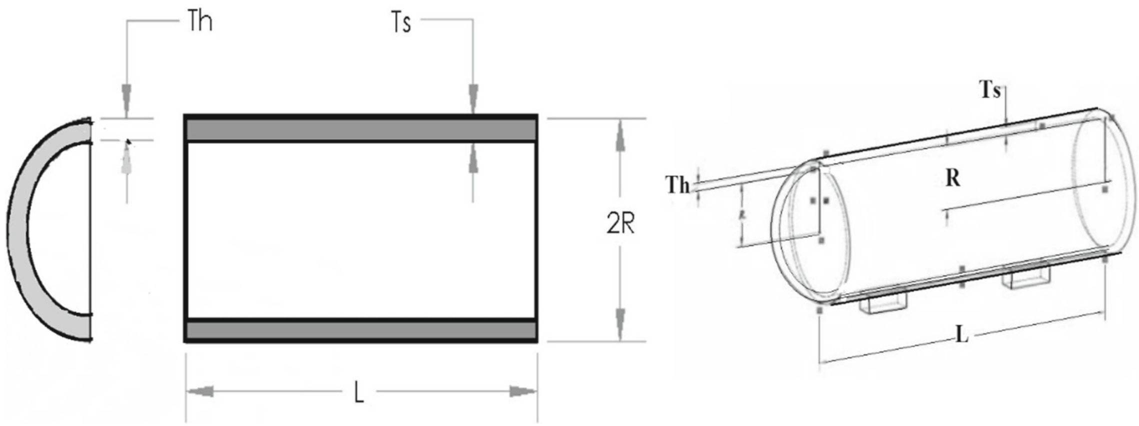


Fig. 10 Problem of pressure vessel design

Table 18 Optimal results of pressure vessel design

Method	Optimal values for variables				F_{min}
	Ts	Th	R	L	
GROM	0.778168641372626	0.384649162633450	40.3196187241064	200	**5885.33277364205**
LAPO (Nematollahi et al. 2017)	0.786283665	0.39160673845	40.758736322	194.296457539	**5916.1935786**
GWO (Mirjalili et al. 2014)	0.812500	0.434500	42.089181	176.758731	8538.8359
GSA (Mirjalili et al. 2014)	1.125000	0.625000	55.9886598	84.4542025	6061.0777
PSO (He and Wang 2007)	0.812500	0.437500	42.091266	176.746500	6288.7445
GA (Coello and Montes 2002)	0.812500	0.434500	40.323900	200.000000	6059.7456
ES (Mezura-Montes and Coello 2008)	0.812500	0.437500	42.098087	176.640518	6410.3811
DE (Huang et al. 2007)	0.812500	0.437500	42.098411	176.637690	6059.7340
ACO (Kaveh and Talatahari 2010)	0.812500	0.437500	42.103624	176.572656	6059.0888
Lagrangian multiplier (Kannan and Kramer 1994)	1.125000	0.625000	58.291000	43.6900000	7198.0428
Branch and bound (Sandgren 1990)	1.125000	0.625000	47.700000	117.701000	8129.1036

Fig. 11 Cantilever beam design problem

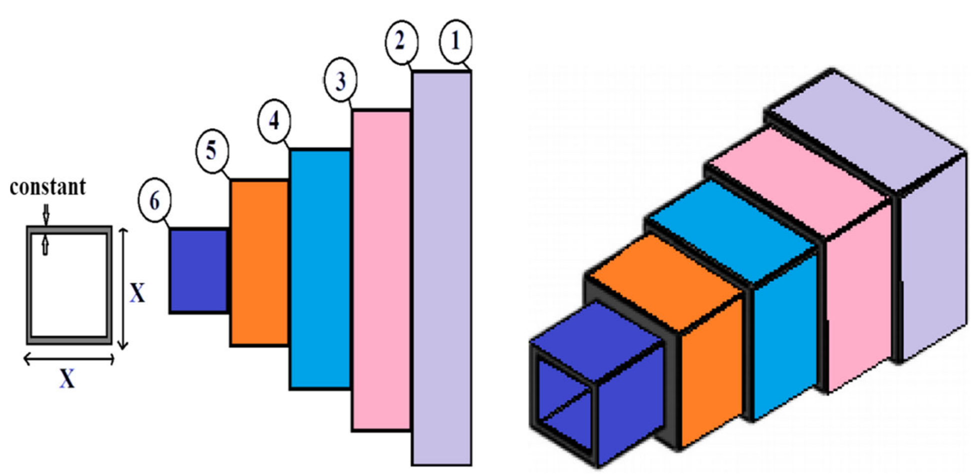


Table 19 Optimal results of cantilever beam design

Method	Optimal values for variables					f_{min}
	X_1	X_2	X_3	X_4	X_5	
GROM	6.01540111331018	5.30998470907654	4.4953671259842	3.5006352767383	2.1522728718473	**1.336520666674**
LAPO (Nematollahi et al. 2017)	6.01243634	5.314870556	4.4959135494	3.4993942765	2.151154796	**1.336521415**
ALO (Mirjalili 2015)	6.01812	5.31142	4.48836	3.49751	2.158329	1.33995
MMA (Chickermane and Gea 1996)	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
GCA_I (Chickermane and Gea 1996)	6.0100	5.30400	4.4900	3.4980	2.1500	1.3400
GCA_II (Chickermane and Gea 1996)	6.0100	5.3000	4.4900	3.4900	2.1500	1.3400
SOS (Cheng and Prayogo 2014)	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996
CS (Gandomi et al. 2013)	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999

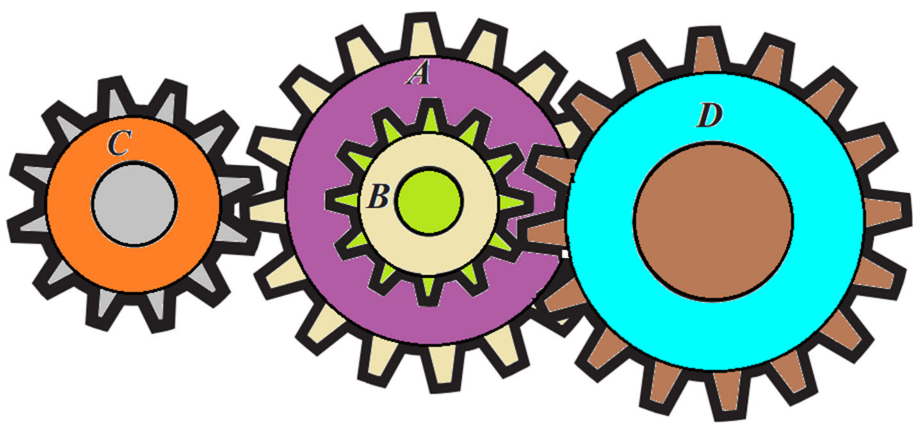


Fig. 12 Gear train design problem

Table 20 Optimal results of gear train design

Method	Optimal values for variables				f_{min}
	n_A	n_B	n_C	n_D	
GROM	43	16	19	49	***2.7008571E−12***
LAPO (Nematollahi et al. 2017)	49	16	19	43	**2.700857E−12**
ALO (Mirjalili 2015)	49	19	16	43	2.7009e012
CS (Gandomi et al. 2013)	43	16	19	49	2.7009e012
MBA (Sadollah et al. 2013)	43	16	19	49	2.7009e012
ISA (Gandomi 2014)	N/A	N/A	N/A	N/A	2.7009e012
ABC (Sharma et al. 2012)	19	16	44	49	2.78e11
GA (Deb and Goyal 1996)	33	14	17	50	1.362E−9
ALM (Kannan and Kramer 1994)	33	15	13	41	2.1469E−8

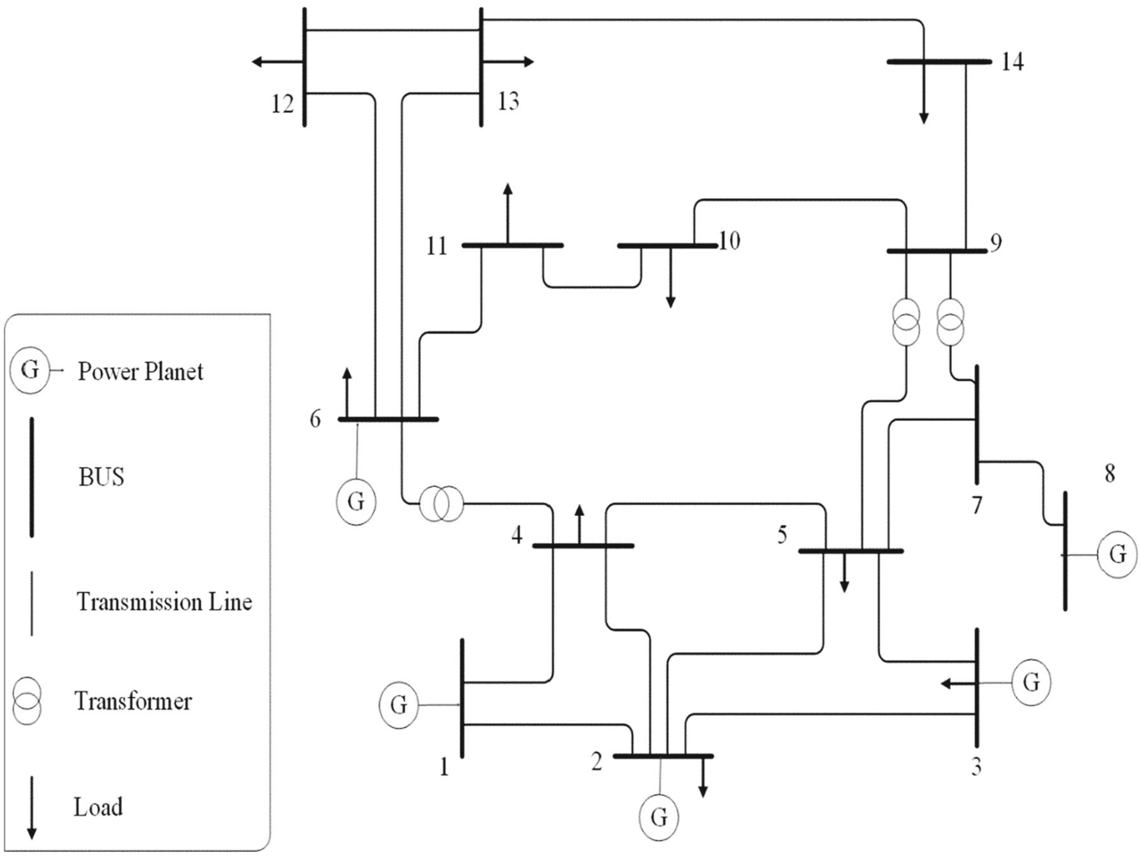


Fig. 13 Single-line diagram of IEEE-14 bus test system

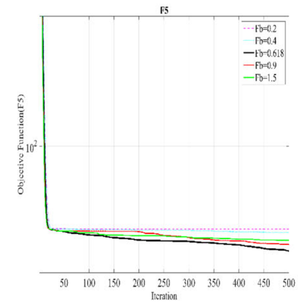
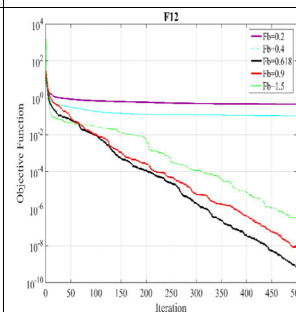
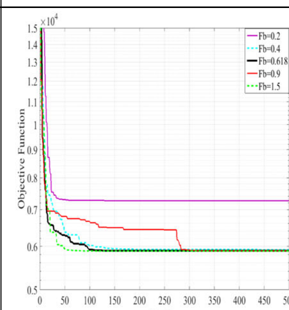
Table 21 Optimal results of optimal power flow

Method	Optimum variables								Optimum cost	Time
	P_{G1}	P_{G2}	P_{G3}	P_{G4}	V_1	V_2	V_3	V_4		
GROM	0.36748	0.29017	0	**0.08009**	1.0384	1.0155727	1.051794	1.05947947	8079.9264	88.521
LAPO (Nematollahi et al. 2017)	0.368	0.290	0.000	0.080	1.0385	1.0156	1.0599	1.0594	8079.9264	85.882
ABC (Nematollahi et al. 2017)	0.347	0.219	0.048	0.123	1.0379	1.0157	1.043	1.0067	8086.4	56.81
DE (Nematollahi et al. 2017)	0.367	0.287	0	0.088	1.0403	1.0167	1.0196	0.9500	8083.3881	75.01
PSO (Nematollahi et al. 2017)	0.367	0.283	0	0.09	1.0395	1.06	1.0256	0.95	8085.43	79.25
CSA (Nematollahi et al. 2017)	0.034	0	0.9	0.404	1.0401	1.0025	1.0097	0.99887	8772.4419	78.322
Firefly (Nematollahi et al. 2017)	0.350	0.338	0.090	0.108	1.0403	1.0428	1.0194	0.99184	8103.4	72.32
GA (Nematollahi et al. 2017)	0.368	0.298	5.1E−4	0.078	1.0384	1.0156	1.0599	1.05	8080.019	98.21
Pattern search (Nematollahi et al. 2017)	0.350	0.467	0.0	0.0	1.0417	1.0164	0.9968	1.031	8096.0007	14.11

proposed method is better than the other nine methods and competitive to two other methods. The comparison reveals that the proposed method could reach the best answer very well. Furthermore, the convergence behavior of the proposed method shows that the proposed method could obtain the final best results in the early iterations.

To sum up, the following characteristics can be concluded from the results of the proposed method.

Table 22 Results with different numbers instead of golden ratio

		Best	Ave	Std	Convergence
*F*5	$\frac{1}{GF} = 0.2$	28.718	28.825	0.072088	
	$\frac{1}{GF} = 0.4$	26.812	27.28	0.39557	
	$\frac{1}{GF} = 0.618$	18.323	19.348	0.887	
	$\frac{1}{GF} = 0.9$	21.455	22.226	0.51586	
	$\frac{1}{GF} = 1.5$	24.376	24.602	0.20605	
*F*12	$\frac{1}{GF} = 0.2$	0.18422	0.2755	0.10806	
	$\frac{1}{GF} = 0.4$	0.0076275	0.10965	0.074928	
	$\frac{1}{GF} = 0.618$	7.1115e−10	0.025917	0.04489	
	$\frac{1}{GF} = 0.9$	1.756e−9	0.051835	0.05183	
	$\frac{1}{GF} = 1.5$	1.117e−7	2.1245e−6	3.306e−6	
Pressure vessel design	$\frac{1}{GF} = 0.2$	6509.5	6956.7	322.47	
	$\frac{1}{GF} = 0.4$	5917.9	6293.4	343.85	
	$\frac{1}{GF} = 0.618$	5885.3	5885.3	2.6034e−6	
	$\frac{1}{GF} = 0.9$	5885.3	5885.3	0.00081678	
	$\frac{1}{GF} = 1.5$	5885.3	6637.4	1302.6	

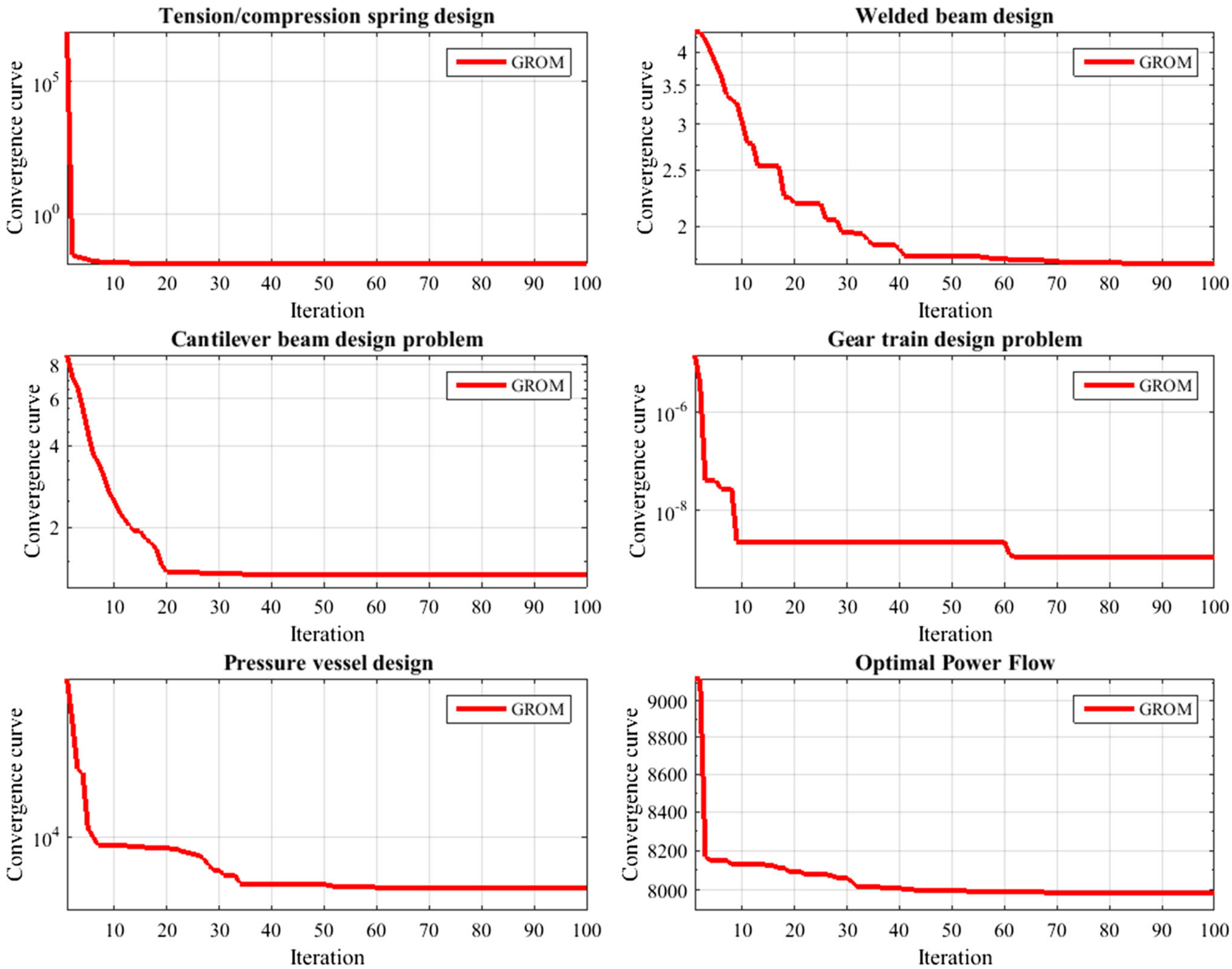


Fig. 14 Convergence behavior of GROM for engineering problems

- The proposed method is a parameter-free algorithm.
- The volume of calculation in each iteration is not big, and the proposed method is a fast algorithm.
- The solution updating is performed very well which results in fast convergence to the best final results.
- The golden ratio causes fast convergence and reaching the best global answer.
- The proposed method is a powerful algorithm in both local and global searches.
- The proposed method is very robust and different trials reach almost the same results.
- The proposed method has excellent performance in solving large-scale problems, discrete problems, actual engineering problems, and high constraints problems.

Compliance with ethical standards

Conflict of interest Author Amin Foroughi Nematollahi declares that he has no conflict of interest. Author Abolfazl Rahiminejad declares that he has no conflict of interest. Author Behrooz Vahidi declares that he has no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Alatas B (2011) ACROA: artificial chemical reaction optimization algorithm for global optimization. Expert Syst Appl 38:13170–13180

Alba E, Dorronsoro B (2005) The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. IEEE Trans Evol Comput 9:126–142

Arora J (2004) Introduction to optimum design. Academic Press, Cambridge

Askarzadeh A (2014) Bird mating optimizer: an optimization algorithm inspired by bird mating strategies. Commun Nonlinear Sci Numer Simul 19:1213–1228

Belegundu AD, Arora JS (1985) A study of mathematical programming methods for structural optimization. Part I: theory. Int J Numer Methods Eng 21:1583–1599

Blum C, Puchinger J, Raidl GR, Roli A (2011) Hybrid metaheuristics in combinatorial optimization: A survey. Appl Soft Comput 11:4135–4151

BoussaïD I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. Inf Sci (NY) 237:82–117

Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112

Chickermane H, Gea HC (1996) Structural optimization using a new local approximation method. Int J Numer Methods Eng 39:829–846

Coello CAC (2000) Use of a self-adaptive penalty approach for engineering optimization problems. Comput Ind 41:113–127

Coello CAC, Montes EM (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inf 16:193–203

Davis L (1991) Handbook of genetic algorithms. Van Nostrand Reinhold, New York

Deb K (1991) Optimal design of a welded beam via genetic algorithms. AIAA J 29:2013–2015

Deb K, Goyal M (1996) A combined genetic adaptive search (GeneAS) for engineering design. Comput Sci Inf 26:30–45

Digalakis JG, Margaritis KG (2001) On benchmarking functions for genetic algorithms. Int J Comput Math 77:481–506

Dosoglu MK, Guvenc U, Duman S, Sonmez Y, Kahraman HT (2018) Symbiotic organisms search optimization algorithm for economic/emission dispatch problem in power systems. Neural Comput Appl 29:721–737

Draa A, Bouaziz A (2014) An artificial bee colony algorithm for image contrast enhancement. Swarm Evol Comput 16:69–84

Du H, Wu X, Zhuang J (2006) Small-world optimization algorithm for function optimization. In: International conference on computer. Springer, pp 264–273

Eiben AE, Schippers CA (1998) On evolutionary exploration and exploitation. Fundam Inform 35:35–50

Eusuff MM, Lansey KE (2003) Optimization of water distribution network design using the shuffled frog leaping algorithm. J Water Resour Plan Manag 129:210–225

Fig Ref (2019) https://www.canva.com/learn/what-is-the-golden-ratio/. Accessed 17 Feb 2019

Fister I, Fister I Jr, Yang X-S, Brest J (2013) A comprehensive review of firefly algorithms. Swarm Evol Comput 13:34–46

Formato RA (2007) Central force optimization: a new metaheuristic with applications in applied electromagnetics. Prog Electromagn Res 77:425–491

Foroughi Nematollahi A, Dadkhah A, Asgari Gashteroodkhani O, Vahidi B (2016) Optimal sizing and siting of DGs for loss reduction using an iterative-analytical method. J Renew Sustain Energy 8:55301

Foroughi Nematollahi A, Rahiminejad A, Vahidi B, Askarian H, Safaei A (2018) A new evolutionary-analytical two-step optimization method for optimal wind turbine allocation considering maximum capacity. J Renew Sustain Energy 10:43312

Gandomi AH (2014) Interior search algorithm (ISA): a novel approach for global optimization. ISA Trans 53:1168–1183

Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17:4831–4845

Gandomi AH, Yang X-S, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35

Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76:60–68

Glover F (1989) Tabu search—part I. ORSA J Comput 1:190–206

Glover F (1990a) Tabu search—part II. ORSA J Comput 2:4–32

Glover F (1990b) Tabu search: a tutorial. Interfaces (Providence) 20:74–94

Glover F, Laguna M (2013) Tabu Search⁎. Springer, New York

Gogna A, Tayal A (2013) Metaheuristics: review and application. J Exp Theor Artif Intell 25:503–526

Gupta S, Deep K (2018a) An opposition-based chaotic Grey Wolf Optimizer for global optimisation tasks. J Exp Theor Artif Intell 30:1–29

Gupta S, Deep K (2018b) Random walk grey wolf optimizer for constrained engineering optimization problems. Comput Intell 34:1025–1045

Gupta S, Deep K (2018c) Cauchy Grey Wolf Optimiser for continuous optimisation problems. J Exp Theor Artif Intell 30:1051–1075

Gupta S, Deep K (2018d) A novel random walk grey wolf optimizer. Swarm Evol Comput 44:101–112

Gupta S, Deep K (2019a) Improved sine cosine algorithm with crossover scheme for global optimization. Knowl Based Syst 165:374–406

Gupta S, Deep K (2019b) A hybrid self-adaptive sine cosine algorithm with opposition based learning. Expert Syst Appl 119:210–230

Hamzeh M, Vahidi B, Nematollahi AF (2018) Optimizing configuration of cyber network considering graph theory structure and teaching-learning-based optimization (GT-TLBO). IEEE Trans Ind Inf. https://doi.org/10.1109/TII.2018.2860984

Hatamlou A (2013) Black hole: a new heuristic optimization approach for data clustering. Inf Sci (NY) 222:175–184

He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20:89–99

He S, Wu QH, Saunders JR (2009) Group search optimizer: an optimization algorithm inspired by animal searching behavior. IEEE Trans Evol Comput 13:973–990

Hu X, Eberhart R (2002) Solving constrained nonlinear optimization problems with particle swarm optimization. In: Proceedings of sixth world multiconference on Systemics, Cybernetics and Informatics. Citeseer, pp 203–206

Huang F, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 186:340–356

Kannan BK, Kramer SN (1994) An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 116:405–411

Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39:459–471

Kashan AH (2011) An efficient algorithm for constrained global optimization and application to mechanical engineering design: League championship algorithm (LCA). Comput Des 43:1769–1792

Kashan AH (2014) League Championship Algorithm (LCA): an algorithm for global optimization inspired by sport championships. Appl Soft Comput 16:171–200

Kaveh A (2017a) Water evaporation optimization algorithm. In: Advances in metaheuristic algorithms for optimal design of structures. Springer, Cham, pp 489–509

Kaveh A (2017b) Tug of war optimization. In: Advances in metaheuristic algorithms for optimal design of structures. Springer, pp 451–487

Kaveh A, Farhoudi N (2013) A new optimization method: Dolphin echolocation. Adv Eng Softw 59:53–70

Kaveh A, Khayatazad M (2012) A new meta-heuristic method: ray optimization. Comput Struct 112:283–294

Kaveh A, Mahdavi VR (2014a) Colliding bodies optimization method for optimum design of truss structures with continuous variables. Adv Eng Softw 70:1–12

Kaveh A, Mahdavi VR (2014b) Colliding bodies optimization method for optimum discrete design of truss structures. Comput Struct 139:43–53

Kaveh A, Mahdavi VR (2014c) Colliding bodies optimization: a novel meta-heuristic method. Comput Struct 139:18–27

Kaveh A, Talatahari S (2010a) A novel heuristic optimization method: charged system search. Acta Mech 213(3-4):267–289

Kaveh A, Talatahari S (2010b) An improved ant colony optimization for constrained engineering design problems. Eng Comput 27:155–182

Kennedy J (2011) Particle swarm optimization. In: Sammut C, Webb GI (eds.) Encyclopedia of machine learning. Springer, pp 760–766

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simmulated annealing. Science 80(220):671–680

Knowles J, Corne D (1999) The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. In: Proceedings of 1999 Congress Evolutionary Computation 1999. CEC 99. IEEE

Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge

Lara CL, Trespalacios F, Grossmann IE (2018) Global optimization algorithm for capacitated multi-facility continuous location-allocation problems. J Glob Optim 71:1–19

Lee KS, Geem ZW (2005) A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Methods Appl Mech Eng 194:3902–3933

Liang J-J, Suganthan PN, Deb K (2005) Novel composition test functions for numerical global optimization. In: Proceedigs of 2005 IEEE swarm intelligence symposium. SIS 2005. IEEE, pp 68–75

Mahdavi M, Fesanghary M, Damangir E (2007) An improved harmony search algorithm for solving optimization problems. Appl Math Comput 188:1567–1579

Mezura-Montes E, Coello CAC (2008) An empirical study about the usefulness of evolution strategies to solve constrained optimization problems. Int J Gen Syst 37:443–473

Miettinen K, Preface By-Neittaanmaki P (1999) Evolutionary algorithms in engineering and computer science: recent advances in genetic algorithms, evolution strategies, evolutionary programming, GE. Wiley, New York

Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61

Mirjalili S (2015a) The ant lion optimizer. Adv Eng Softw 83:80–98

Mirjalili S (2015b) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowled Based Syst 89:228–249

Mirjalili S (2015) ALO MATLAB code

Mirjalili S (2016a) SCA: a sine cosine algorithm for solving optimization problems. Knowl Based Syst 96:120–133

Mirjalili S (2016b) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27:1053–1073

Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67

Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27:495–513

Mirjalili SZ, Mirjalili S, Saremi S, Faris H, Aljarah I (2018) Grasshopper optimization algorithm for multi-objective optimization problems. Appl Intell 48:805–820

Molga M, Smutnicki C (2005) Test functions for optimization needs. Test Funct Optim Needs 101 (2005)

Moosavi K, Vahidi B, Askarian Abyaneh H, Foroughi Nematollahi A (2017) Intelligent control of power sharing between parallel-connected boost converters in micro-girds. J Renew Sustain Energy 9:65504

Mucherino A, Seref O (2007) Monkey search: a novel metaheuristic search for global optimization. In: Data mining, systems analysis, and optimization in biomedicine. AIP Publishing, pp 162–173

Naka S, Genji T, Yura T, Fukuyama Y (2002) Hybrid particle swarm optimization based distribution state estimation using constriction factor approach. In: Proceedings of International Conference SCIS ISIS, 2002, pp 1083–1088

Nematollahi AF, Rahiminejad A, Vahidi B (2017) A novel physical based meta-heuristic optimization method known as Lightning Attachment Procedure Optimization. Appl Soft Comput 59:596–621

Nematollahi AF, Rahiminejad A, Vahidi B (2019) A novel multi-objective optimization algorithm based on Lightning Attachment Procedure Optimization algorithm. Appl Soft Comput 75:404–427

Price K, Storn RM, Lampinen JA (2006) Differential evolution: a practical approach to global optimization. Springer, New York

Ragsdell KM, Phillips DT (1976) Optimal design of a class of welded structures using geometric programming. J Eng Ind 98:1021–1025

Rahiminejad A, Alimardani A, Vahidi B, Hosseinian SH (2014) Shuffled frog leaping algorithm optimization for AC–DC optimal power flow dispatch. Turk J Electr Eng Comput Sci 22:874–892

Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Des 43:303–315

Rashedi E, Nezamabadi-Pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci (NY) 179:2232–2248

Rizk-Allah RM (2018) An improved sine–cosine algorithm based on orthogonal parallel information for global optimization. Soft Comput. https://doi.org/10.1007/s00500-018-3355-y

Saad A, Khan SA, Mahmood A (2018) A multi-objective evolutionary artificial bee colony algorithm for optimizing network topology design. Swarm Evol Comput 38:187–201

Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13:2592–2612

Salcedo-Sanz S, Pastor-Sánchez A, Gallo-Marazuela D, Portilla-Figueras A (2013) A novel coral reefs optimization algorithm for multi-objective problems. In: International conference on intelligent data engineering and automated learning. Springer, pp 326–333

Salcedo-Sanz S, Del Ser J, Landa-Torres I, Gil-López S, Portilla-Figueras JA (2014) The coral reefs optimization algorithm: a novel metaheuristic for efficiently solving optimization problems. Sci World J. https://doi.org/10.1155/2014/739768

Sandgren E (1990) Nonlinear integer and discrete programming in mechanical design optimization. J Mech Des 112:223–229

Satapathy SC, Naik A (2014) Modified teaching–learning-based optimization algorithm for global numerical optimization—a comparative study. Swarm Evol Comput 16:28–37

Saxena A, Kumar R, Das S (2019) β-Chaotic map enabled Grey Wolf Optimizer. Appl Soft Comput 75:84–105

Shah-Hosseini H (2011) Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. Int J Comput Sci Eng 6:132–140

Shareef H (2015) LSA MATLAB code

Shareef H, Ibrahim AA, Mutlag AH (2015) Lightning search algorithm. Appl Soft Comput 36:315–333

Sharma TK, Pant M, Singh VP (2012) Improved local search in artificial bee colony using golden section search. arXiv Prepr. arXiv:1210.6128

Simon D (2008) Biogeography-based optimization. IEEE Trans Evol Comput 12:702–713

Statnikov R, Matusov JB (2012) Multicriteria optimization and engineering. Springer, New York

Talatahari S, Azar BF, Sheikholeslami R, Gandomi AH (2012) Imperialist competitive algorithm combined with chaos for global optimization. Commun Nonlinear Sci Numer Simul 17:1312–1319

Tan Y (2015a) Hybrid fireworks algorithms. In: Fireworks algorithm. Springer, Berlin, Heidelberg, pp 151–161

Tan Y (2015b) Discrete firework algorithm for combinatorial optimization problem. In: Fireworks algorithm. Springer, pp 209–226

Tan Y, Zhu Y (2010) Fireworks algorithm for optimization. In: Interantional conference on swarm intelligence. Springer, pp 355–364

Vahidi B, Foroughi A, Rahiminejad A (2017) Lightning attachment procedure optimization (LAPO) source codes demo version 1.0

Venkataraman P (2009) Applied optimization with MATLAB programming. Wiley, New York

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1:67–82

Yang X-S (2009) Firefly algorithms for multimodal optimization. In: International symposium on stochastic algorithms. Springer, pp 169–178

Yang C, Tu X, Chen J (2007) Algorithm of marriage in honey bees optimization based on the wolf pack search. In: Intelligence pervasive computing 2007. IPC. 2007 international conference. IEEE, pp 462–467

Yazdani S, Nezamabadi-pour H, Kamyab S (2014) A gravitational search algorithm for multimodal optimization. Swarm Evol Comput 14:1–14