Soft Computing (2019) 23:6617–6631
https://doi.org/10.1007/s00500-018-3310-y

METHODOLOGIES AND APPLICATION

An improved hybrid grey wolf optimization algorithm

Zhi-jun Teng[1] · Jin-ling Lv[1] · Li-wen Guo[1]

Published online: 25 June 2018

Abstract

The existing grey wolf optimization algorithm has some disadvantages, such as slow convergence speed, low precision and so on. So this paper proposes a grey wolf optimization algorithm combined with particle swarm optimization (PSO_GWO). In this new algorithm, the Tent chaotic sequence is used to initiate the individuals' position, which can increase the diversity of the wolf pack. And the nonlinear control parameter is used to balance the global search and local search ability of the algorithm and improve the convergence speed of the algorithm. At the same time, the idea of PSO is introduced, which utilize the best value of the individual and the best value of the wolf pack to update the position information of each grey wolf. This method preserves the best position information of the individual and avoids the algorithm falling into a local optimum. To verify the performance of this algorithm, the proposed method is tested on 18 benchmark functions and compared with some other improved algorithms. The simulation results show that the proposed algorithm can better search global optimal solution and better robustness than other algorithm.

Keywords Grey wolf optimizer · Particle swarm optimization · Chaotic sequence · Dynamic adaptation · Global optimization

1 Introduction

One of the salient features of the development of modern science and technology is that life sciences and engineering sciences cross, interpenetrate, and influence each other. The vigorous development of swarm intelligence algorithm reflects this characteristic and trend of scientific development. In recent years, swarm intelligence algorithm has gradually become the focus of scholars. Due to its simplicity, flexibility, non-derivation mechanism and avoidance of local optimality, swarm intelligence algorithm is applied to not only computers but also agriculture (Zou et al. 2016), metallurgy (Reihanian et al. 2011), military (Zheng et al. 2017), civil, and hydraulic engineering (Quiniou et al. 2014), etc.

Recently, many nature inspired optimization algorithm are proposed. Artificial fish swarm algorithm (AFSA) (Xian et al. 2017) imitates the foraging, gathering, and rear-tailing behaviour of fish group by constructing artificial fish to achieve optimal results; ant colony optimization (ACO) (Chen et al. 2017) is inspired by the behaviour of ants in finding the path during the search for food; butterfly optimization algorithm (BOA) (O'Neil et al. 2010) is a natural heuristic algorithm that mimics butterfly foraging behaviour; cuckoo search (CS) (Rakhshani and Rahati 2017) effectively solves the optimization problem by simulating some of the cuckoo species' brood parasitism. At the same time, CS also use the relevant Levy flight search mechanism; firefly algorithm (FA) (Nekouie and Yaghoobi 2016) is mainly to use the characteristics of firefly luminescence for random optimization; krill herd algorithm (KH) (Gandomi and Alavi 2012) simulates the response behaviour of krill for the biochemical processes and environments evolve; fruit fly optimization algorithm (FOA) (Pan 2012) is a method for seeking global optimization based on fruit fly foraging behaviour; flower pollination algorithm (FPA) (Yang 2013) is a stochastic global optimization algorithm developed to mimic the biological characteristics of self-pollination and cross-pollination of flowering plants in nature; chicken optimization algorithm (COA) (Meng et al. 2014) simulates the chicken hierarchy and chicken behaviour.

The above metaheuristic algorithms show that many swarm intelligence technologies have been proposed. Some of them simulate animal predatory behaviour. However, they ignored the grey wolf, which is a predator in the top of

Communicated by V. Loia.

✉ Jin-ling Lv
1114130692@qq.com

[1] Department of Information Engineering, Northeast Electric Power University, Jilin City 132012, Jilin Province, China

the food chain. Researchers have not established a mathematical model of grey wolf social hierarchy and predatory behaviour. Therefore, Mirjalili et al. (2014) proposed a novel metaheuristic algorithm, which imitated leadership hierarchy and hunting process of grey wolves. Based on nature of grey wolves, they proposed a grey wolf optimization algorithm (GWO). GWO is a heuristic algorithm based on population. It mainly simulates the leadership hierarchy of wolves and grey wolf predation behaviour. Through a series of standard test functions, GWO algorithm can converge to a better quality near-optimal solution, possesses better convergence characteristics than other prevailing population-based techniques, such as genetic algorithm (GA), particle swarm optimization (PSO), firefly algorithm (FA), artificial fish swarm algorithm (AFSA) and differential evolution algorithms (DE). At the same time, the GWO algorithm is simple to operate in the optimization process, easy to implement, and has few adjustment parameters. Therefore, the GWO algorithm is widely concerned and is applied to solve many practical problems, such as multilevel thresholding problem (Khairuzzaman and Chaudhury 2017), chemistry experiment (Bian et al. 2018), multi-objective optimal reactive power dispatch (Nuaekaew et al. 2017), etc.

Like other swarm intelligent optimization algorithms, the grey wolf optimization algorithm also has some disadvantages. For example, the original GWO algorithm is easy to fall into stagnation when attacking prey, and the speed of convergence gradually slows down in the late search period. Therefore, Saremi et al. (2015) and Mirjalili et al. (2016) proposed the combination of dynamic evolutionary population and grey wolf algorithm to improve the local search ability of the algorithm. But it neglects the algorithm's global search ability. Mittal et al. (2016) presented a modified GWO(mGWO) to balance exploration and exploitation ability. Long et al. (2016) proposed a hybrid grey wolf optimization (HGWO), which utilizes chaotic sequence to strengthen the diversity of global searching. Zhu et al. (2015) and Yao and Wang (2016) introduced differential evolution algorithm to improve the searching ability of wolf algorithm. In order to increase the diversity of the population, Long and Wu (2017), Zuo et al. (2017) and Guo et al. (2017) proposed the good point set theory to update the grey wolves individuals' position, so that the initial population is evenly distributed. Kohli and Arora (2017) presented the chaos theory into the GWO algorithm and aimed at accelerating its global convergence speed. Tawhid and Ali (2017) combined grey wolf optimization and genetic algorithm to improve the convergence performance of the algorithm. Jitkongchuen (2016) proposed a hybrid differential evolution algorithm with grey wolf optimizer, which solves function optimization problem. But, the above algorithms do not consider the influence of the individual wolves' experience on the whole population when studying the grey wolf predation process. So Singh and Singh (2017) presented a newly hybrid nature inspired algorithm called HPSOGWO. This algorithm improves the ability of exploitation in particle swarm optimization with the ability of exploration in grey wolf optimizer to produce both variants' strength.

On this basis, this paper presents a grey wolf optimization algorithm combined with particle swarm optimization (PSO_GWO). This algorithm contains three main improvement concepts. Firstly, it initializes the population by using Tent mapping; secondly, it adopts nonlinear control parameter strategy to coordinate the exploration and exploitation ability; thirdly, inspired by the particle swarm optimization (PSO) algorithm, a new position update equation of individuals by incorporating the information of individual historical best solution into the position update equation is designed to speed up convergence.

The rest of this paper is organized as follows. Section 2 presents a brief description of GWO. Section 3 illustrates the proposed approach. Section 4 comprises benchmark test functions and experimental results. Section 5 concludes the work and outlines some ideas for future works.

2 Grey wolf optimization algorithm

2.1 Grey wolf social rank and hunting behaviour

Grey wolf is a predator in the top of the food chain. Most of the wolves live in groups, which has 5–12 wolves per population on average. And each wolf has its own role in the population, so they have a very strict social hierarchy, as shown in Fig. 1.

The first layer is the highest leader of the grey wolves, called α, which is mainly responsible for making decisions about hunting, habitat and so on; the second layer is the grey wolf in the subordinate leader of the grey wolf, called β, which is mainly responsible for assisting leadership management group or other wolf pack activities; the third layer is δ, which is mainly responsible for watching the boundaries of the territory, warning the wolf pack in case of any danger

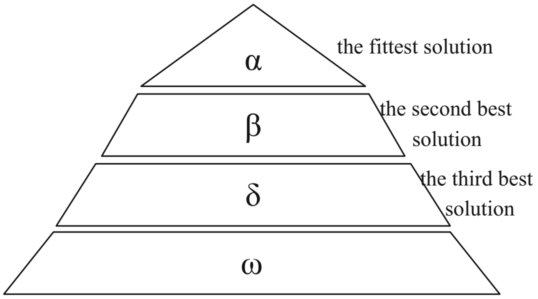


Fig. 1 The leadership hierarchy of wolves

and caring for the weak and wounded grey wolves. The fourth layer is the lowest level grey wolf in the population, called ω, which has to submit to all the other dominant grey wolves. It may seem that the ω wolves are not an important character in the wolf pack, but indispensable for balancing the internal relations of the population.

The leadership hierarchy of wolves plays a crucial role in hunting of prey. Firstly, the grey wolves search and track the prey; secondly, the α grey wolf leads the other wolves to encircle the prey in all directions; thirdly, α grey wolf commands the β and δ wolves to attack the prey. If the prey escapes, the other wolves which are supplied from the rear will continue to attack the prey; finally, grey wolves catch the prey.

2.2 Grey wolf optimization algorithm description

Grey wolf optimization algorithm simulates the leadership hierarchy of wolves and predatory behaviour, and then utilizes the grey wolf abilities, which are search, encirclement, hunting and other activities in the predation process, to achieve the purpose of optimization. Assuming that the number of wolves is N and the search area is d, the position of the ith wolf can be expressed as: $X_i = (X_{i1}, X_{i2}, X_{i3}, \ldots, X_{id})$. In order to mathematical model the social hierarchy of wolves, the fittest solution is considered as the alpha (α) wolf. Consequently, the second- and third-best solutions are named beta (β) and delta (δ) wolves, respectively. The rest of the candidate solutions are assumed to be omega (ω) wolves. In the algorithm, the location of the prey corresponds to the position of the alpha wolf.

The encircling behaviour of grey wolves can be mathematically modelled as follows:

$$D = \left|C \times X_{\mathrm{p}}(t) - X(t)\right| \tag{1}$$

$$X(t+1) = X_{\mathrm{p}}(t) - A \times D \tag{2}$$

where the set t indicates the current iteration, and the set $X_p(t)$ represents the position vector of the prey, the set $X(t)$ is the position vector of a grey wolf, the set C is a control coefficient, which is determined by the following formula:

$$C = 2r_1 \tag{3}$$

where the set r_1 is the random variable in the range of [0, 1]. The set A is convergence factor, which is calculated as follows:

$$A = 2ar_2 - a \tag{4}$$

$$a = 2\left(1 - \frac{t}{T\max}\right) \tag{5}$$

where the set r_2 is the random variable in the range of [0, 1]. The set a is the control coefficient, which linearly decreases from 2 to 0 over the course of iterations, that is (Sahoo and Chandra 2016), $a_{\max} = 2, a_{\min} = 0$.

When the grey wolves catch prey, firstly, the leader wolf α leads the other wolves to surround the prey. Then, the α wolf leads β and δ wolves to capture the prey. In the grey wolves, α, β and δ wolves are the closest to the prey, so the location of the prey can be calculated by their positions. The specific mathematical model is as follows:

$$D_\alpha = |C_1 \times X_\alpha(t) - X(t)| \tag{6}$$

$$D_\beta = \left|C_2 \times X_\beta(t) - X(t)\right| \tag{7}$$

$$D_\delta = |C_3 \times X_\delta(t) - X(t)| \tag{8}$$

$$X_1 = X_\alpha - A_1 \times D_\alpha \tag{9}$$

$$X_2 = X_\beta - A_2 \times D_\beta \tag{10}$$

$$X_3 = X_\delta - A_3 \times D_\delta \tag{11}$$

$$X(t+1) = \frac{X_1 + X_2 + X_3}{3} \tag{12}$$

The distance between $X(t)$ and α, β, ω wolves is calculated by formula (6)–(11), and then, the position of the wolves move to the prey is calculated by formula (12). The flowchart of GWO is given in Fig. 2.

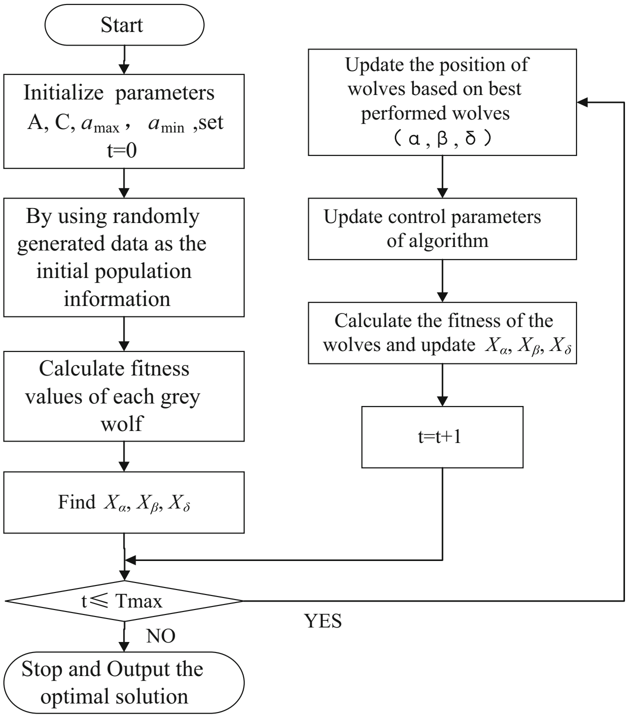

Start
Initialize parameters A, C, $a_{\max}$, $a_{\min}$,set t=0
By using randomly generated data as the initial population information
Calculate fitness values of each grey wolf
Find X_α, X_β, X_δ
t⩽ Tmax
NO
Stop and Output the optimal solution
YES
Update the position of wolves based on best performed wolves (α , β , δ)
Update control parameters of algorithm
Calculate the fitness of the wolves and update X_α, X_β, X_δ
t=t+1

Fig. 2 GWO algorithm optimization process

3 Improved hybrid grey wolf algorithm

3.1 Chaos initialization

GWO algorithm usually solves function optimization problem by using randomly generated data as the initial population information, which will not retain the diversity of the population and lead to the poor optimization result of the algorithm. Therefore, this paper proposes Tent chaotic map to initialize the population.

Chaotic motion has the characteristics of randomness, regularity, and ergodicity. When solving function optimization problems, these features can lead algorithm to escape local optima, so as to maintain the diversity of the population and improve the global search ability. Chaotic maps include Tent maps, Logistic maps and so on. However, different chaotic mappings have different search characteristics. So far logistic map is mostly used in the literature. But it has a higher value rate in [0, 0.1] and [0.9, 1] that leads to inhomogeneous distribution of values. Shan et al. (2005) proves that the Tent map can perform better than the Logistic mapping in traversal homogeneity and generate a more uniform initial value between [0, 1], so as to improve the optimization speed of the algorithm.

Therefore, this paper proposes Tent chaos initialization, that is, we use Tent chaos to initialize the grey wolves. The mathematical model of Tent chaos map is as follows:

$$x(t+1) = \begin{cases} \frac{x(t)}{u} & 0 \leq x(t) < u \\ \frac{1-x(t)}{1-u} & u \leq x(t) \leq 1 \end{cases} \tag{13}$$

When $u = 1/2$, the Tent map has the most typical form. At this time, the resulting sequence has a uniform distribution and has an approximately uniform distribution density for different parameters.

Thus, the formula for the Tent chaotic map cited in this article is:

$$x_{t+1} = \begin{cases} 2x_t & 0 \leq x_t \leq \frac{1}{2} \\ 2(1-x_t) & \frac{1}{2} < x_t \leq 1 \end{cases} \tag{14}$$

The steps for generating a sequence of Tent chaos maps are as follows:

Step 1 Take the random initial value x_0 to avoid falling into the small cycles points$\{0.2, 0.4, 0.6, 0.8\}$. Mark the array $y(1) = x_0, i = 1, j = 1$;
Step 2 According to formula (14) to produce a set of x sequences. After each iteration, $i = i + 1$;
Step 3 If the number of iterations reaches the maximum, jump Step 4; Else If there is $x_i = \{0, 0.25, 0.5, 0.75\}$ or $x_i = x_{i-k}, k = \{0, 1, 2, 3, 4\}$, replace the initial value of the iteration by the formula $x(i) = y(j+1) = y(j) + c, j = j + 1$; Else go to Step 2;
Step 4 Terminate the operation, save x sequence data.

3.2 Nonlinear control parameter strategy

The GWO algorithm is mainly composed of two steps: the prey positioning and the grey wolf individual's predatory behaviour. According to formula (1), the parameter A plays a very crucial role in balancing the global exploration and local exploitation capability of the GWO algorithm. When $|A| > 1$, the group will expand the search range to find a better candidate solution, which is the global exploration ability of the GWO algorithm. When $|A| < 1$, the group will narrow the search range and perform detailed search in the local area, which is the local exploitation capability of the GWO algorithm. At the same time, we can see from formula (2) that in the process of iteration, the value of A changes continuously with the change of control parameter a. From formula (4), it can be seen that the control parameter a decreases linearly with the increase in the number of iterations. However, the optimization process of GWO algorithm is very complicated. The linear change of parameter a cannot reflect the actual optimization search process of the algorithm. Wei et al. (2016) and Yi-Tung and Erwie (2008) proposed that the control parameter a varies nonlinearly with the number of iterations. And through the standard test function optimization results show that the use of nonlinear change strategy is better than the linear strategy optimization. But they still cannot meet the needs of the algorithm.

Therefore, this paper presents a new nonlinear control parameters, as shown in the following formula:

$$a_1(t) = a_{\text{ini}} - (a_{\text{ini}} - a_{\text{fin}}) \times \left(\frac{t}{T\max}\right)^2 \tag{15}$$

where the set a_{ini} and a_{fin}, respectively, represent the initial value and final value of the control parameter a. The set t is current iteration and the set Tmax is the maximum number of iterations.

In order to verify the validity of the control parameter a proposed in this paper, we compare it with the linear control parameters and the nonlinear control parameters which are proposed in Yang (2013) and Mirjalili et al. (2014), and the formula is as follows:

$$a_2(t) = a_{\text{ini}} - a_{\text{ini}} \times \left(\frac{t}{T\max}\right) \tag{16}$$

$$a_3(t) = a_{\text{ini}} - (a_{\text{ini}} - a_{\text{fin}}) \times \tan\left(\frac{1}{\varepsilon} \times \frac{t}{T\max} \times \pi\right) \tag{17}$$

$$a_4(t) = a_{\text{ini}} - a_{\text{ini}} \times \left(\frac{1}{e-1} \times \left(e^{\frac{t}{T\max}} - 1\right)\right) \tag{18}$$

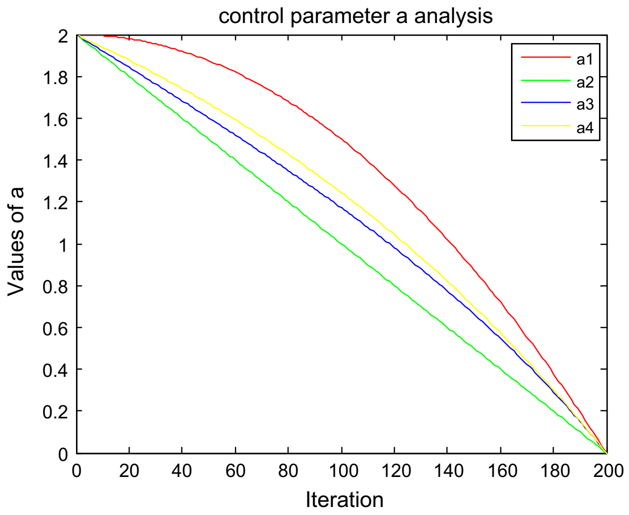


Fig. 3 The change curve of the control parameter *a*

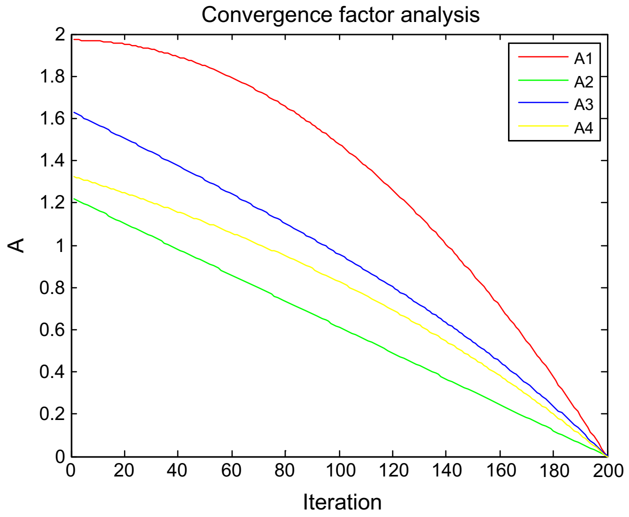


Fig. 4 Convergence factor dynamic curve

where the set ε is a nonlinear adjustment coefficient.

Four kinds of control parameters are simulated as shown in Fig. 3. It can be seen from Fig. 3 that the nonlinear control parameter as proposed in this paper slowly declines at the early stage and falls fast in the later period. So the nonlinearity of the proposed algorithm is better.

Meanwhile, the formula for the convergence factor A is as follows:

$$A = 2a_1 r_2 - a_1 \tag{19}$$

The dynamic curve of the four kinds of convergence factor A is shown in Fig. 4. It can be seen that the decline speed of convergence factor A is slow in the early stage, which can increase the global search ability and avoid the algorithm falling into the local optimum. The decline speed of convergence factor A is quick in the later stage, which can improve local search and speed up algorithm optimization. Therefore, this improvement can further weigh the exploration and exploitation capabilities.

3.3 PSO thought

In the process of location updating, GWO algorithm takes into account only the location information of individual wolves and the optimal solution, second-best solution, and third-best solution location information of the wolf pack, which realizes the exchange of information between individuals and wolf pack. But it ignores the exchange of information between the wolf and its own experience. Therefore, the idea of PSO algorithm is introduced to improve the location updating process.

In the PSO algorithm, the current position of the particle is updated by using the best position information of the particle itself and the best position information of the group. This paper combining with PSO algorithm will introduce the optimal location of individual experience into the position updating formula, which enables it to keep its own optimal position information. The new location update formula is as follows:

$$\begin{aligned} X_i(t+1) = {} & c_1 r_1 (w_1 X_1(t) + w_2 X_2(t) + w_3 X_3(t)) \\ & + c_2 r_2 (X_{\text{ibest}} - X_i(t)) \end{aligned} \tag{20}$$

where the set c_1 is a social learning factor, the set c_2 is a cognitive learning factor. They, respectively, represent the influence of the individual optimal value and the group optimal value. The value of c_1 is large which can improve the global search capability; the value of c_2 is large which can improve the local search capability. But if the c_1 is too large, it will result in too many particles remain in the vicinity of the local. If the c_2 is too large, it will cause the particles to reach the local optimum in advance and converge to this value. According to Clerc (2002), this paper selects $c_1 = c_2 = 2.05$. The set r_1 and r_2 are the random variable in the range of [0, 1]. The set X_{ibest} indicates that the grey wolf has experienced the best position. The set w_1, w_2, w_3 are inertia weight coefficients. By adjusting weight ratio of the α, β, δ wolves, the global and local search ability of the algorithm can be balanced dynamically. The specific formula is as follows:

$$w_1 = \frac{|X_1|}{|X_1 + X_2 + X_3|} \tag{21}$$

$$w_2 = \frac{|X_2|}{|X_1 + X_2 + X_3|} \tag{22}$$

$$w_3 = \frac{|X_3|}{|X_1 + X_2 + X_3|} \tag{23}$$

In Eq. (20), the first part is expressed as the mean value of the best prey position which utilizes α, β and δ wolf to search, which expands the search interval to increase the global search ability of the algorithm; the second part is expressed as the effect of the personal historical best position on algorithm search, which preserves the optimal position experienced by the individual.

3.4 PSO_GWO algorithm flow

The specific implementation steps of the improved wolf algorithm are as follows:

Step 1 Set the size of the population to N, dimension d, and initialize the A, C, a values;
Step 2 Generate population individuals by using Tent mapping $\{X_i, i = 1, 2, 3, \ldots, N\}$, then calculate the individual fitness value $\{f_i, i = 1, 2, 3, \ldots, N\}$;
Step 3 Sort the order of fitness values by size, and take the first three fitness values corresponding to the individual as α, β, δ. The corresponding position information is: X_α, X_β, X_δ;
Step 4 Use formula (15) to calculate the nonlinear control parameters a, and then update the value of A and C according to formula (2) and (19);
Step 5 Use formula (20) to update the location of individuals, then recalculate the fitness values and update values of α, β, δ;
Step 6 Judge whether t reaches Tmax value, if reached, the fitness value of α is output, that is, the best solution. Else go to Step 3.

4 Experimental data and simulation analysis

4.1 Benchmark functions

In order to verify the effectiveness of the improved hybrid wolf algorithm, this paper selects 18 benchmark functions (Liu and Yin 2016; Lu et al. 2017) to do simulation, compares with the grey wolf optimization algorithm (GWO) (Mirjalili et al. 2014), the improved grey wolf optimization algorithm (IGWO) (Yao and Wang 2016) and the grey the wolf algorithm based on the third strategy (GWO_3) (Zuo et al. 2017). The specific benchmark functions are shown in Table 1, and Figs. 5, 6 and 7 illustrate the 2D versions of the benchmark functions used.

```
PSO_GWO algorithm pseudo-code
Input: grey wolf population $X$, Size $N$, number of iterations $Tmax$,Control parameter $a$, Swing factor $C$, Learning factor $c_1$, $c_2$
Output: Optimal solution

Algorithm Description
for i=1:N
    for j=1:d
        Use equation (14) to generate Tent chaotic sequence
    end
end // Use Tent to initialize the grey wolf population
for t=1:$Tmax$
    Use equation(15) to update parameter $a$ value
    for i=1:N
        for j=1:d
            $D$=|$CX$p(t)-$X$(t), then according to equation (6)(7)(8) to calculate $D$(α, β, ω)
            According to equation (21)(22)(23) to calculate the coefficient $w$
            According to equation (9)(10)(11)(20) to update the position of the grey wolf
        end
    end
    Calculate fitness values and update α, β, δ
end // the algorithm ends
```

4.2 Experimental parameters

The performances of the proposed algorithm have been evaluated by using 18 commonly used benchmark functions. For all the experimentation, population size is $N = 30$; dimensions are $d = 30$, 50, 100. The maximum number of iterations is $T\max = 500$. In the IGWO algorithm, $\varepsilon = 5, b_1 = 0.5, b_2 = 0.5$. In the PSO_GWO algorithm, $c_1 = c_2 = 2.05$; $a_{\text{ini}} = 2, a_{\text{fin}} = 0$; $r_1 = r_2 = r_3 = r_4 =$ rand [0, 1].

4.3 Simulation analysis

In order to verify the optimal performance of PSO_GWO, we use 18 benchmark functions to test the performance of four algorithms. These benchmark functions are described in Table 1 and classified in Figs. 5, 6 and 7.

First of all, we use 30 wolves as the grey wolf population and separately conduct simulation experiments on 30, 50 and 100 dimension. We separately simulate 30 times on unimodal and multimodal functions and record the average value and standard deviation. The specific results are shown in Table 2.

Fig. 5 2-D versions of unimodal benchmark functions

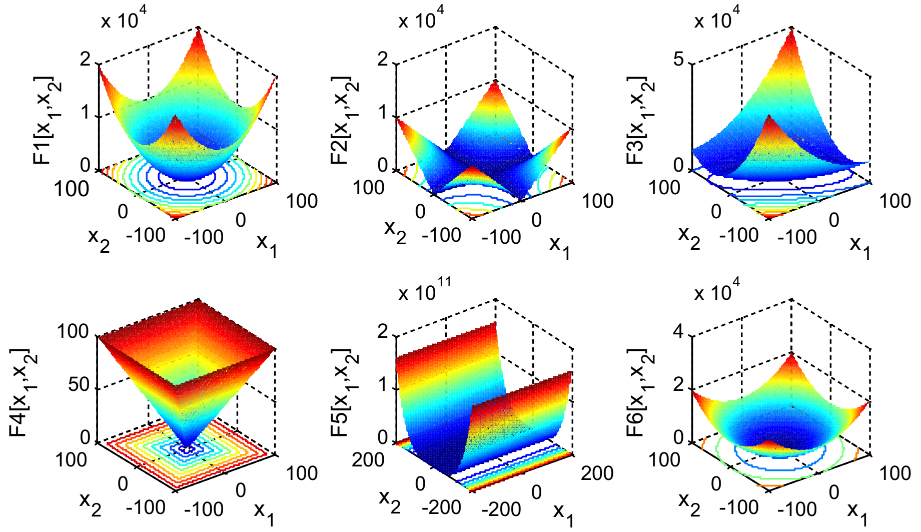
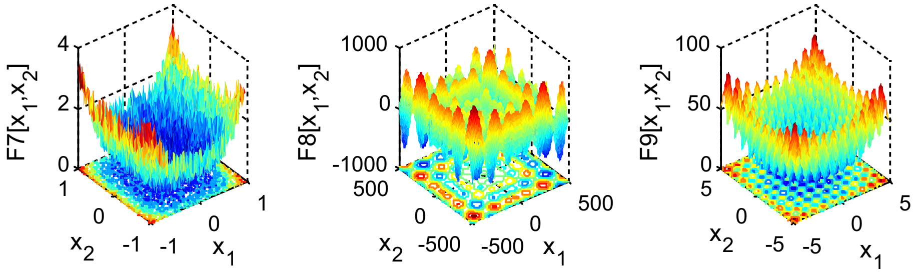


Fig. 6 2-D versions of multimodal benchmark functions



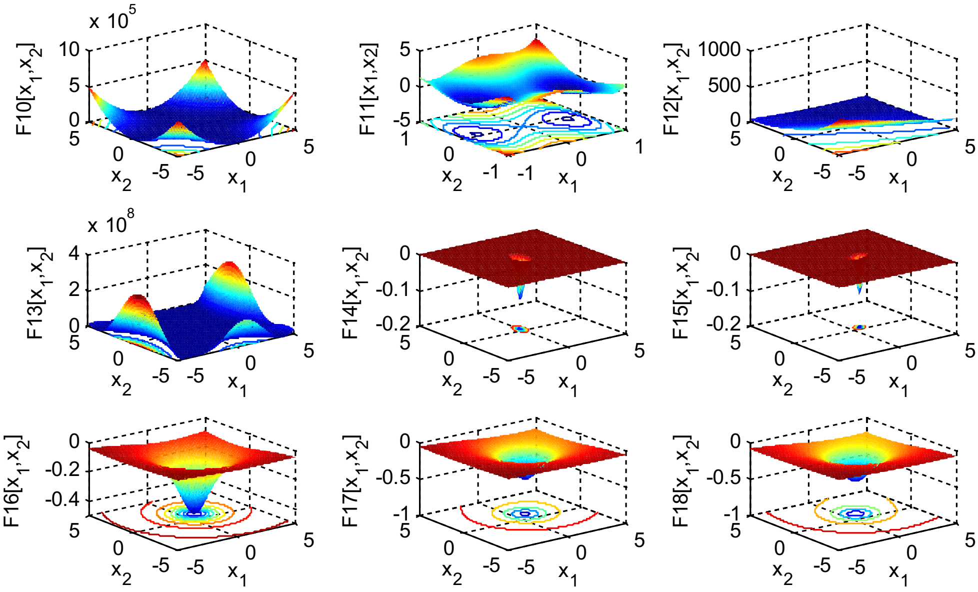


Fig. 7 2-D version of fixed-dimension multimodal benchmark functions

Table 1 Benchmarking function

Function expression	d	Search interval	$f_{\min}$
$f_1(x) = \sum_{i=1}^{30} x_i^2$	30	[−100, 100]	0
$f_2(x) = \sum_{i=1}^{30} \lvert x_i \rvert + \prod_{i=1}^{30} \lvert x_i \rvert$	30	[−10, 10]	0
$f_3(x) = \sum_{i=1}^{30} \left(\sum_{j=1}^{i} x_j \right)^2$	30	[−100, 100]	0
$f_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le 30\}$	30	[−100, 100]	0
$f_5 = \sum_{i=1}^{30} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[−30, 30]	0
$f_6 = \left\{ \left[\sum_{i=1}^{30} \sin^2(x_i) \right] - \exp\left(-\sum_{i=1}^{30} x_i^2 \right) \right\} \cdot \exp\left[-\sum_{i=1}^{30} \sin^2 \sqrt{\lvert x_i \rvert} \right]$	30	[−10, 10]	−1
$f_7 = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{30} x_i^2} \right) - \exp\left(\frac{1}{n} \sum_{i=1}^{30} \cos(2\pi x_i) \right) + 20 + e$	30	[−5.21, 5.21]	0
$f_8 = \sum_{i=1}^{30} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[−32, 32]	0
$f_9 = \frac{1}{4000} \sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	30	[−600, 600]	0
$f_{10} = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[−5, 5]	0.00030
$f_{11} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[−5, 5]	−1.0316
$f_{12} = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	[−5, 5]	0.398
$f_{13} = \left[1 + (x_1 + x_2 + 1)^2 \left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right]$ $\times \left[30 + (2x_1 - 3x_2)^2 \times \left(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2\right)\right]$	2	[−2, 2]	3
$f_{14} = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij} \left(x_j - p_{ij}\right)^2 \right)$	3	[1, 3]	−3.86
$f_{15} = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij} \left(x_j - p_{ij}\right)^2 \right)$	6	[0, 1]	−3.32
$f_{16} = -\sum_{i=1}^{5} \left[(X - a_i)(X - a_i)^{\mathrm{T}} + c_i \right]^{-1}$	4	[0, 10]	−10.1532
$f_{17} = -\sum_{i=1}^{7} \left[(X - a_i)(X - a_i)^{\mathrm{T}} + c_i \right]^{-1}$	4	[0, 10]	−10.4028
$f_{18} = -\sum_{i=1}^{10} \left[(X - a_i)(X - a_i)^{\mathrm{T}} + c_i \right]^{-1}$	4	[0, 10]	−10.5363

We use the mean and standard deviation as reliability criteria. It can be seen from Table 2 that when the dimension is same, the PSO_GWO can obtain better average in benchmark function F1–F9. Especially in the test function F8, the PSO_GWO algorithm can converge to zero. In addition, the standard deviation of the PSO_GWO algorithm is small in 9 benchmark functions experiments, which indicates that the robustness of this algorithm is better. This is because the PSO_GWO algorithm introduces nonlinear control parameters to balance the global exploration and local exploitation capability. However, the control parameters of other algorithms have poor nonlinearity, resulting in not balancing the local and global search ability well and easily fall into local optimum during the search period. Meanwhile, regardless of dimension $d = 30$, 50 or 100, PSO_GWO algorithm can get better optimal solution than the other three algorithms. But the optimization ability of PSO_GWO algorithm decreases slightly with the increase in dimension.

In addition, we also propose the next set of the benchmark functions analysed in this paper, which are the fixed-dimension multimodal benchmark functions. Then we separately simulate 30 times and record the average value and standard deviation. The results are shown in the following table.

Table 2 GWO, GWO_3, IGWO, PSO_GWO algorithm optimization results on unimodal and multimodal functions

Function	*d*	GWO		GWO_3		IGWO		PSO_GWO	
		Average	SD	Average	SD	Average	SD	Average	SD
F1	30	8.33e−27	3.04e−26	1.42e−37	4.93e−37	6.65e−42	2.05e−41	4.83e−119	1.61e−119
	50	9.22e−29	4.12e−28	1.66e−36	7.14e−36	2.26e−40	1.01e−39	1.39-118	4.99e−118
	100	3.24e−30	1.33e−29	1.89e−36	8.43e−36	9.04e−44	2.53e−43	9.61e−117	3.81e−116
F2	30	3.60e−17	1.47e−16	1.80e−18	5.99e−18	8.27e−21	3.69e−20	1.34e−59	4.18e−59
	50	7.08e−16	2.47e−15	1.63e−19	5.52e−19	1.55e−20	6.55e−20	1.53e−57	6.12e−57
	100	5.28e−15	1.82e−14	7.06e−19	1.77e−18	1.53e−20	6.77e−20	7.97e−55	3.46e−54
F3	30	3.23e−25	1.45e−24	2.94e−38	1.31e−37	1.52e−42	6.81e−42	1.22e−113	4.86e−113
	50	3.04e−28	1.36e−27	2.92e−36	1.29e−35	9.54e−39	2.72e−38	3.49e−115	1.56e−114
	100	7.31e−25	3.27e−24	3.87e−38	1.24e−37	4.03e−38	1.80e−37	3.96e−120	1.66e−119
F4	30	5.77e−17	1.34e−16	2.59e−19	1.13e−18	3.61e−18	1.62e−17	6.03e−61	2.31e−60
	50	2.87e−17	1.16e−16	8.83e−20	3.67e−19	1.60e−19	5.12e−19	2.96e−58	1.31e−57
	100	1.58e−15	5.90e−15	1.39e−18	5.91e−18	1.14e−16	4.86e−16	3.10e−55	1.38e−54
F5	30	2.89e+01	5.46e−02	2.87e+01	2.10e−02	2.87e+01	1.00e−02	2.69e+01	1.10e−02
	50	4.89e+01	6.26e−02	4.98e+01	2.13e−02	4.89e+01	1.30e−02	4.78e+01	1.08e−02
	100	9.90e+01	4.76e−02	9.98e+01	2.18e−02	9.87e+01	1.08e−02	9.69e+01	0.89e−02
F6	30	-1.03163	0.0589	-1.0095	0.0246	0.9667	0.0089	-1.0120	0.0081
	50	-0.9925	0.0561	-0.8828	0.0175	-1.0314	0.0112	-1.0289	0.0012
	100	-0.8752	0.0631	-0.5340	0.0119	-1.0314	0.0109	-1.0306	0.0013
F7	30	1.06e−13	2.1e−13	3.28e−14	1.95e−14	3.95e−15	1.82e−15	8.18e−16	1.72e−15
	50	1.11e−13	4.76e−13	7.90e−14	1.71e−15	1.13e−14	2.59e−14	8.81e−16	1.82e−15
	100	1.174-10	6.71e−10	9.33e−14	2.17e−14	7.55e−14	1.60e−14	4.45e−15	1.90e−15
F8	30	1.02e−16	2.34e−15	7.96e−13	6.13e−13	0	0	0	0
	50	5.68e−15	6.78e−14	1.21e−12	8.91e−11	0	0	0	0
	100	1.14e−13	2.46e−13	1.71e−14	5.64e−13	0	0	0	0
F9	30	0.0160	0.0220	0.0071	0.0078	0.0046	0.0951	0.0009	0.0003
	50	0.028	0.0315	0.009	0.0084	0.00513	0.08	0.0012	0.0018
	100	0.076	0.021	0.013	0.06	0.0093	0.016	0.0084	0.0069

Table 3 Comparison between the methods: average and standard deviation with the fixed-dimension multimodal benchmark functions

Function	GWO		GWO_3		IGWO		PSO_GWO	
	Average	SD	Average	SD	Average	SD	Average	SD
F10	5.10e−04	0.0138	6.92e−04	0.0121	4.04e−04	0.0171	6.53e−04	0.0230
F11	− 1.0316	0.0389	− 1.0316	0.0446	− 1.0316	0.0498	− 1.0316	0.0273
F12	0.3979	1.5123	0.3983	1.7130	0.3979	2.7290	0.3979	0.8128
F13	3.0000	6.4140	3.0001	8.9019	3.0019	11.5876	3.0000	7.8205
F14	− 3.7930	1.3842	− 3.8621	1.3719	− 3.6766	1.3144	− 3.6977	1.2907
F15	− 2.3698	0.5705	− 2.3840	0.4409	− 2.7175	0.6213	− 3.3220	0.3825
F16	− 6.0748	0.8532	− 7.6025	0.8037	− 10.0112	0.8116	− 10.1522	0.6882
F17	− 8.7936	1.0331	− 8.6852	1.2324	− 9.8151	1.1807	− 10.4036	0.6020
F18	− 8.3991	0.8492	− 9.9167	1.3469	− 10.5026	0.8874	− 10.5332	1.1117

Table 4 Time-consuming results of benchmark functions

Function	GWO		GWO_3		IGWO		PSO_GWO	
	TIC and TOC	CPU time	TIC and TOC	CPU time	TIC and TOC	CPU time	TIC and TOC	CPU time
F1	1.01422	0.057501	1.01221	0.03708	1.02737	0.05980	1.001474	0..00585
F2	1.01557	0.026003	1.01012	0.01836	1.02655	0.03173	1.002120	0.0028
F3	1.01893	0.003075	1.01511	0.002928	1.01980	0.003292	1.001928	0.03123
F4	1.01609	0.00017	1.01495	0.000146	1.01950	0.000183	1.017832	0.0001973
F5	1.01165	0.02381	1.01055	0.017316	1.01702	0.02424	1.001568	0.001925
F6	1.01872	0.07418	1.02085	0.05620	1.02180	0.082636	1.0191	0.06902
F7	1.01764	0.072664	1.01558	0.05460	1.01792	0.07614	1.001786	0.074928
F8	1.00981	0.046516	1.00977	0.0448	1.00992	0.047552	1.009825	0.0486
F9	1.01738	0.001691	1.01735	0.0011	1.02192	0.0017	1.01461	0.0023
F10	1.02741	0.069128	1.02734	0.06910	1.02824	0.07292	1.02798	0.072643
F11	1.01587	0.001028	1.0149	0.001018	1.01792	0.00127	1.01623	0.0014
F12	1.01625	0.084368	1.01682	0.08304	1.01830	0.073102	1.01535	0.015934
F13	1.04029	0.03480	1.03233	0.010608	1.04088	0.034196	1.04050	0.0199
F14	1.00764	0.051404	1.00601	0.02636	1.00989	0.01324	1.008218	0.04925
F15	1.0159	0.042028	1.01484	0.04396	1.01683	0.045132	1.016263	0.01681
F16	1.017	0.01780	1.01488	0.01692	1.01931	0.0214	1.0178	0.0003
F17	1.00567	0.07276	1.00490	0.06836	1.00601	0.00728	1.00597	0.02182
F18	1.00501	0.021992	1.00421	0.02132	1.00511	0.003060	1.00480	0.0078250

Table 3 shows a comparison among GWO, GWO_3, IGWO, PSO_GWO four algorithms based on the average and standard deviation. It can be known from the above table data that the algorithm presented in this paper is not enough different with the simulation results of the other three algorithms.

Simultaneously, this paper calculates the algorithm's time statistics from both CPU and TIC/TOC. The CPU reflects the time it takes to complete the process when the CPU is working at full speed, while the TIC/TOC is used to calculate the time the program is running. The calculation results are shown in Table 4.

The accuracy of the PSO_GWO algorithm can be verified by starting and ending time of algorithm (TIC and TOC), and CPU time. These experimental data are shown in Table 4. As it can be seen from Table 4, the PSO_GWO algorithm has lower computational complexity than IGWO algorithm. But comparing with the GWO algorithm, the PSO_GWO algorithm has a large computational complexity and has a long running time. This is because the PSO_GWO algorithm is based on the basic GWO algorithm to introduce PSO algorithm ideas, thereby increasing the computational complexity of the algorithm. Therefore, the PSO_GWO algorithm improves convergence accuracy at the expense of computational complexity. Figure 5 shows the eight test function optimization curve of the GWO, GWO_3, IGWO, PSO_GWO algorithm.

As can be seen from Figs. 8, 9 and 10, the search speeds of the four algorithms are roughly the same during the initial search, but as the number of iterations increases, the PSO_GWO algorithm proposed in this paper continues to search until the search for the optimal solution. However, the other three algorithms reach the stagnation state in advance, resulting in poor search results. Therefore, the proposed algorithm has better convergence performance on benchmark function. This is because the other three algorithms did not consider the impact of the individual experience of the grey wolf during the search process. However, the algorithm proposed in this paper uses its own experience to delay the algorithm into a local optimum during the predation process.

To sum up, all simulation results indicate the improved hybrid grey wolf optimization algorithm is very helpful in improving the efficiency of the GWO in the terms of result quality.

5 Conclusion

In this paper, we proposed an improved grey wolf algorithm. As described in Sect. 3, it first used Tent chaotic map to initialize grey wolf population, then used nonlinear control parameters to balance the local and global search capabilities of the algorithm and introduced PSO algorithm thought into the position update formula. And in Sect. 4, a series of experiments on the 18 benchmark test functions are exe-

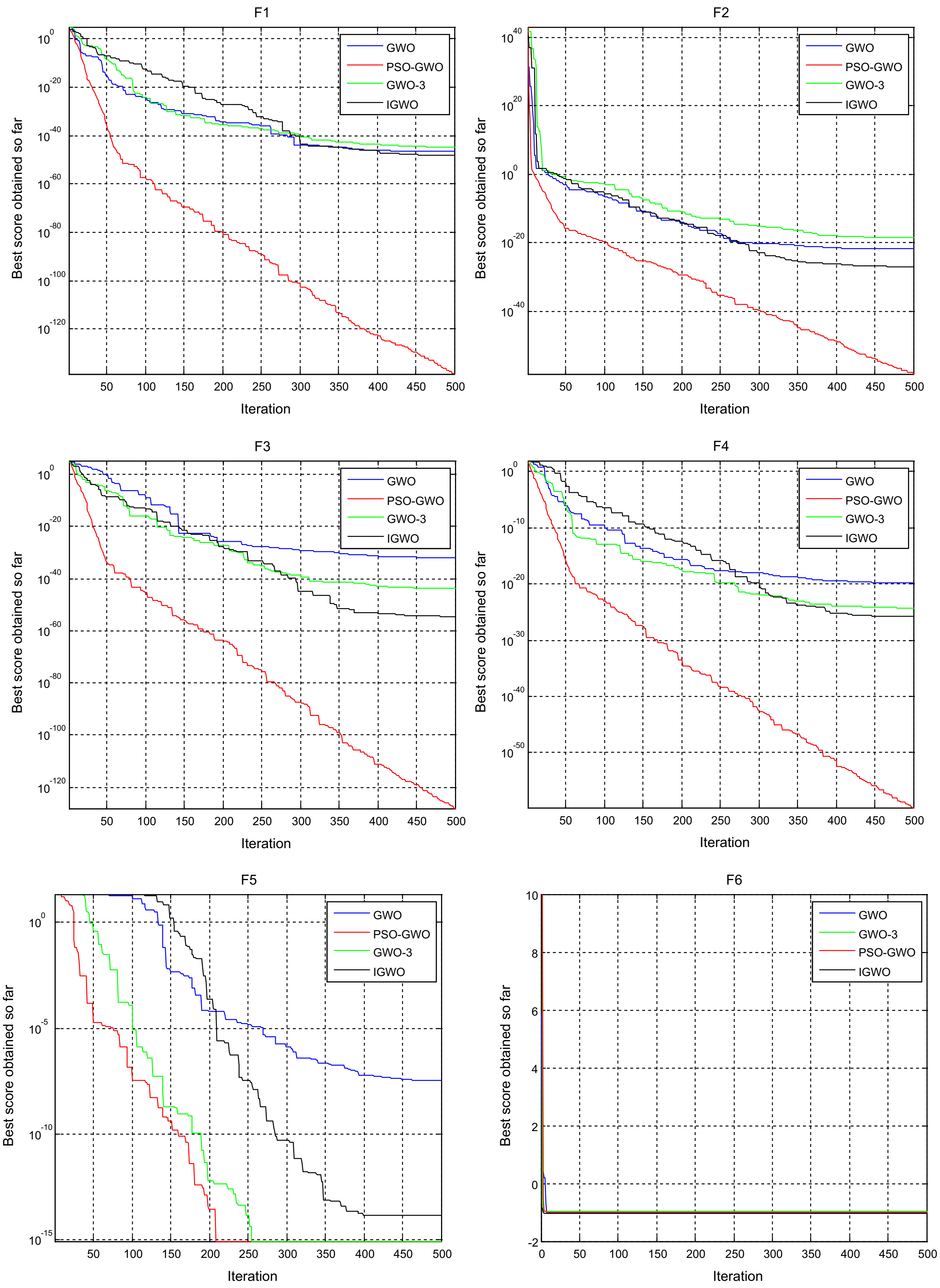


Fig. 8 Convergence curve of GWO, GWO_3, IGWO, and PSO_GWO on unimodal functions

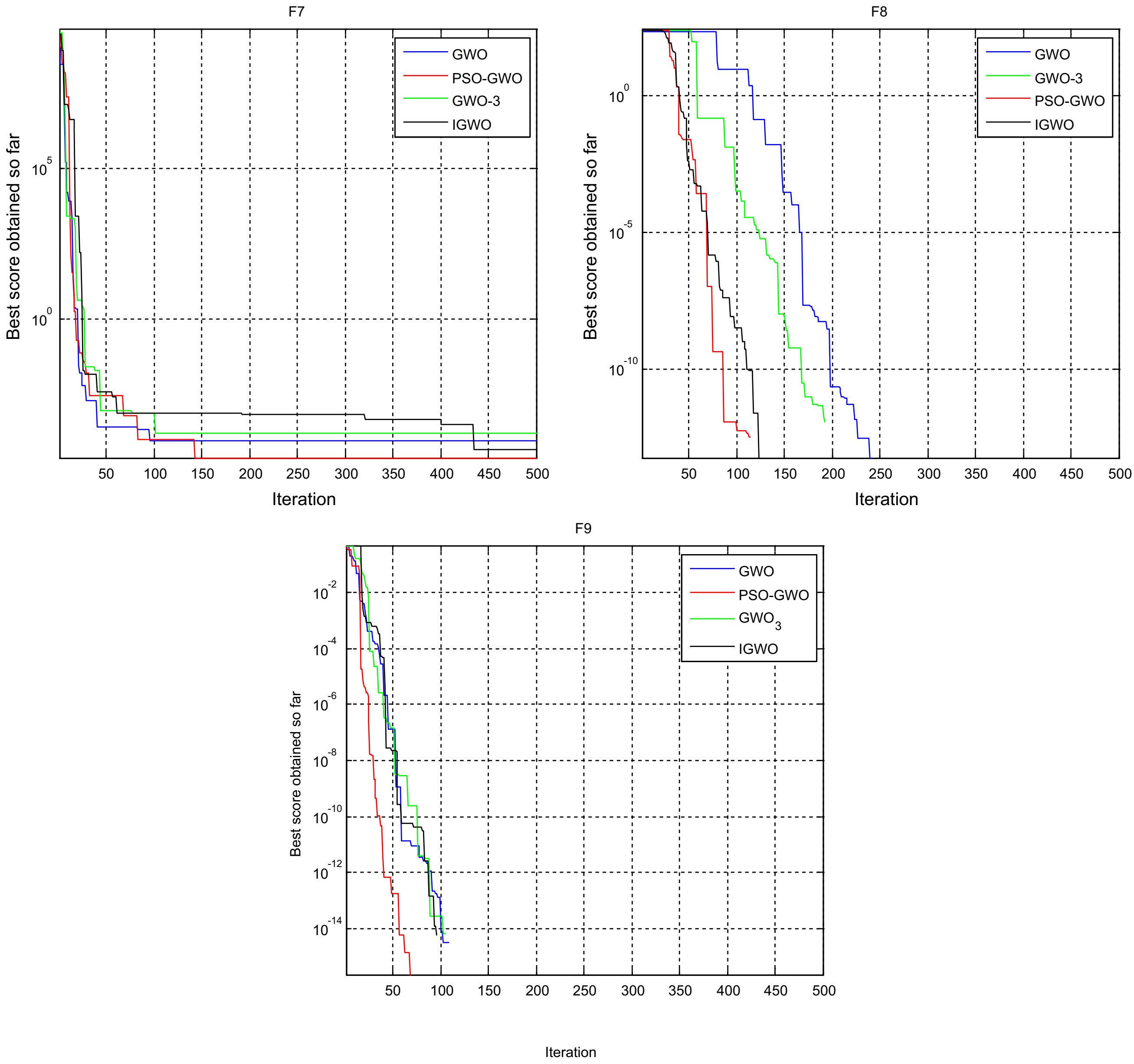


Fig. 9 Convergence curve of GWO, GWO_3, IGWO, and PSO_GWO on multimodal functions

cuted to verify the effectiveness of the improved hybrid wolf algorithm. The experimental results show that the proposed algorithm is superior to other algorithms in the search capability. And it is evident that the proposed algorithm can improve the performance of GWO algorithm in terms of result quality and better robustness.

In this work, we justify the problems that were used for testing benchmark test functions. Although the PSO_GWO algorithm improves result quality, the computational complexity is increased. So in the future, we will conduct research on the issue of reducing the computational complexity. At the same time, we will also apply the improved algorithm to solve wireless sensor networks coverage problem.

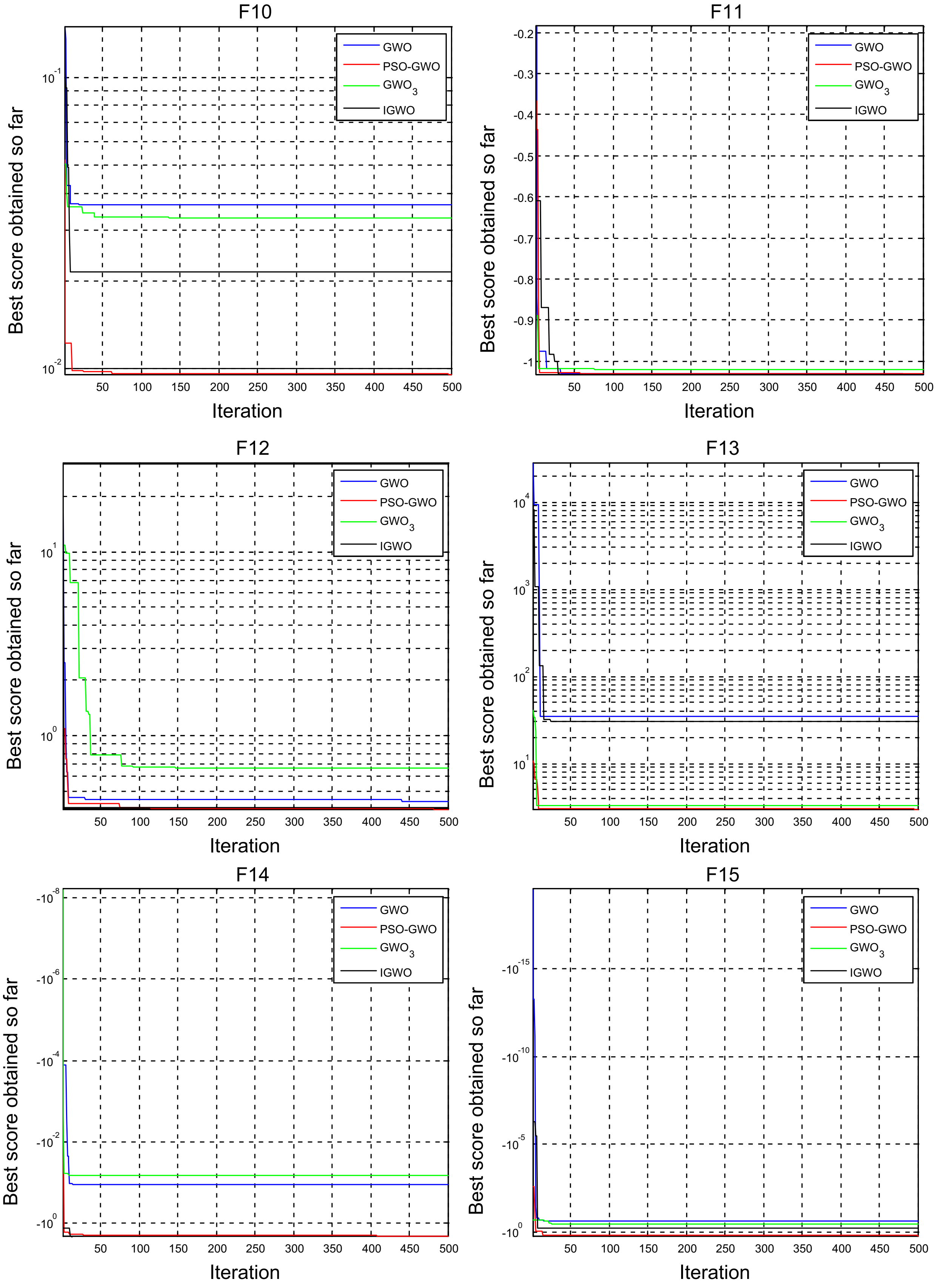


Fig. 10 Convergence curve of PSO, GWO, IGWO and PSO_GWO variants on fixed-dimension multimodal functions

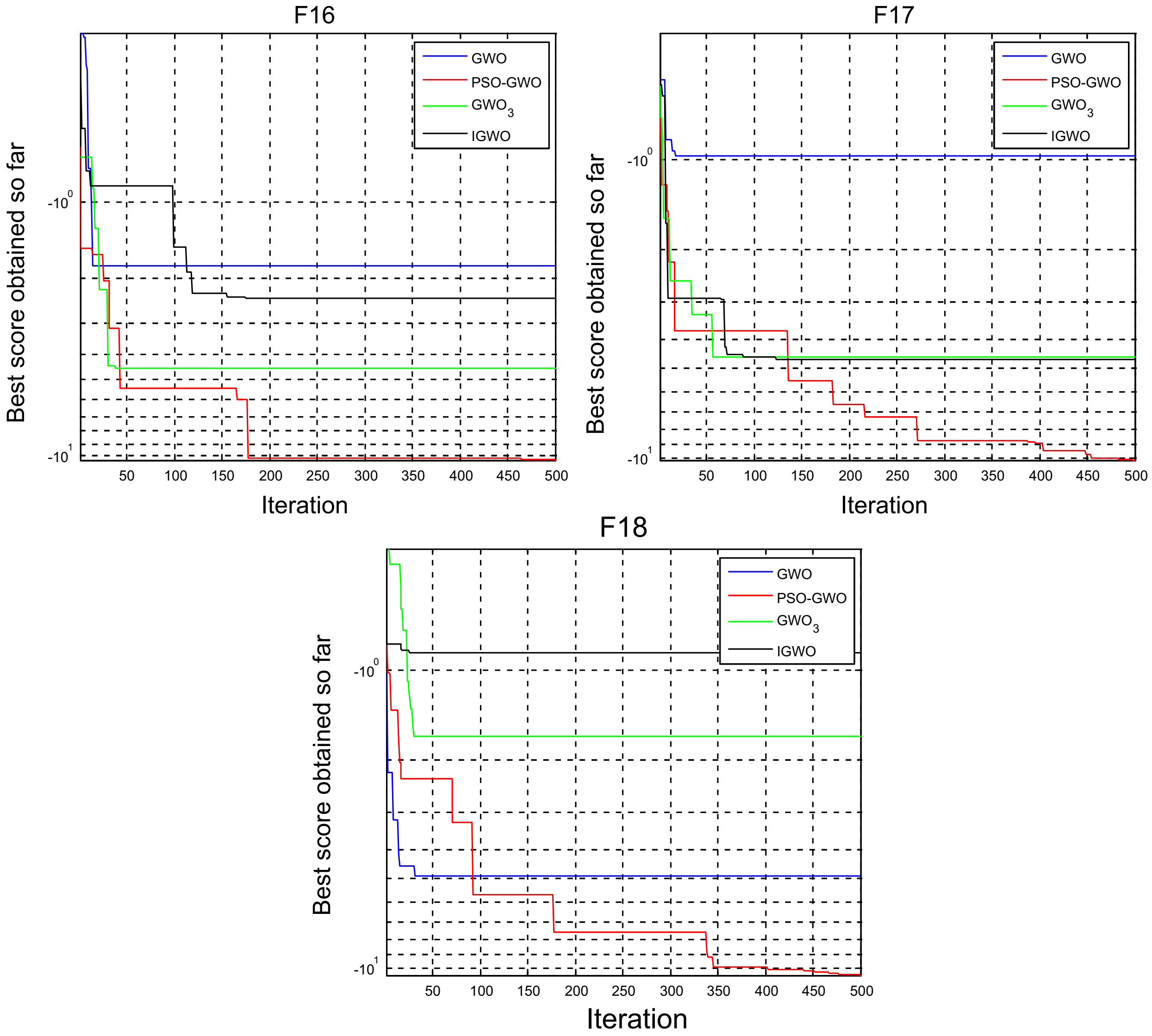


Fig. 10 continued

Acknowledgements The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper. Besides, this work is supported by the National Natural Science Foundation of China (No. 51277023), National Natural Science Foundation Youth Science Foundation Project (No. 61501107), "13th Five-Year" Scientific Research Planning Project of Jilin Province Department of Education (No. JJKH20180439KJ).

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

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

References

Bian XQ, Zhang L, Du ZM et al (2018) Prediction of sulfur solubility in supercritical sour gases using grey wolf optimizer-based support vector machine. J Mol Liq 261(1):431–438

Chen Z, Zhou S, Luo J (2017) A robust ant colony optimization for continuous functions. Expert Syst Appl 81:309–320

Clerc M (2002) The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: Proceedings of the 1999 congress on evolutionary computation, CEC 99, vol 3. IEEE

Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17(12):4831–4845

Guo Z, Liu R, Gong C et al (2017) Study on Improvement of grey wolf algorithm. Appl Res Comput 34(12):3603–3606

Jitkongchuen D (2016) A hybrid differential evolution with grey wolf optimizer for continuous global optimization. In: International conference on information technology and electrical engineering. IEEE, pp 51–54

Khairuzzaman AKM, Chaudhury S (2017) Multilevel thresholding using grey wolf optimizer for image segmentation. Expert Syst Appl 86(15):64–76

Kohli M, Arora S (2017) Chaotic grey wolf optimization algorithm for constrained optimization problems. J Comput Des Eng

Liu T, Yin S (2016) An improved particle swarm optimization algorithm used for BP neural network and multimedia course-ware evaluation. Multimed Tools Appl 76(9):11961–11974

Long W, Wu TB (2017) Improved grey wolf optimization algorithm coordinating the ability of exploration and exploitation. Control Decis 32(10):1–8

Long W, Cai SH, Jiao JJ et al (2016) Hybrid grey wolf optimization algorithm for high-dimensional optimization. Control Decis 31(11):1991–1997

Lu C, Gao L, Li X et al (2017) A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry. Eng Appl Artif Intell 57(C):61–79

Meng X, Liu Y, Gao X et al (2014) A new bio-inspired algorithm: chicken swarm optimization. In: Advances in swarm intelligence. Springer, pp 86–94

Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69(3):46–61

Mirjalili S, Saremi S, Mirjalili SM et al (2016) Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. Expert Syst Appl 47:106–119

Mittal N, Singh U, Sohi BS (2016) Modified grey wolf optimizer for global engineering optimization. Appl Comput Intell Soft Comput 2016(4598):1–16

Nekouie N, Yaghoobi M (2016) A new method in multimodal optimization based on firefly algorithm. Artif Intell Rev 46(2):267–287

Nuaekaew K, Artrit P, Pholdee N et al (2017) Optimal reactive power dispatch problem using a two-archive multi-objective grey wolf optimizer. Expert Syst Appl 87(30):79–89

O'Neil M, Woolfe F, Rokhlin V (2010) An algorithm for the rapid evaluation of special function transforms. Appl Comput Harmon Anal 28(2):203–226

Pan W-T (2012) A new fruit fly optimization algorithm: taking the financial distress model as an example. Knowl Based Syst 26:69–74

Quiniou ML, Mandel P, Monier L (2014) Optimization of drinking water and sewer hydraulic management: coupling of a genetic algorithm and two network hydraulic tools. Procedia Eng 89:710–718

Rakhshani H, Rahati A (2017) Snap-drift cuckoo search: a novel cuckoo search optimization algorithm. Appl Soft Comput 52:771–794

Reihanian M, Asadullahpour SR, Hajarpour S et al (2011) Application of neural network and genetic algorithm to powder metallurgy of pure iron. Mater Des 32(6):3183–3188

Sahoo A, Chandra S (2016) Multi-objective grey wolf optimizer for improved cervix lesion classification. Appl Soft Comput 52:64–80

Saremi S, Mirjalili SZ, Mirjalili SM (2015) Evolutionary population dynamics and grey wolf optimizer. Neural Comput Appl 26(5):1257–1263

Shan L, Qiang H, Li J et al (2005) Chaotic optimization algorithm based on Tent map. Control Decis 20(2):179–182

Singh N, Singh SB (2017) Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. J Appl Math 2017(1–4):15

Tawhid MA, Ali AF (2017) A Hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function. Memet Comput 9(4):1–13

Wei Z, Zhao H, Li M et al (2016) A grey wolf optimization algorithm based on nonlinear adjustment strategy of control parameter. J Air Force Eng Univ (Nat Sci Ed) 17(3):68–72

Xian S, Zhang J, Xiao Y et al (2017) A novel fuzzy time series forecasting method based on the improved artificial fish swarm optimization algorithm. Soft Comput 10:1–11

Yang XS (2013) Flower pollination algorithm for global optimization. In: International conference on unconventional computation and natural computation. Springer, pp 240–249

Yao P, Wang HL (2016) Three-dimensional path planning for UAV based on improved interfered fluid dynamical system and grey wolf optimizer. Control Decis 31(04):701–708

Yi-Tung K, Erwie Z (2008) A hybrid genetic algorithm and particle swarm optimization for multimodal functions. Appl Soft Comput 8(2):849–857

Zheng YJ, Wang Y, Ling HF et al (2017) Integrated civilian-military pre-positioning of emergency supplies: a multiobjective optimization approach. Appl Soft Comput 58:732–741

Zhu A, Xu C, Li Z et al (2015) Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC. J Syst Eng Electron 26(2):317–328

Zou S, Fan Y, Tang Y et al (2016) Optimized algorithm of sensor node deployment for intelligent agricultural monitoring. Comput Electron Agric 127:76–86

Zuo J, Zhang C, Xiao Y et al (2017) Multi-machine PSS parameter optimal tuning based on grey wolf optimizer algorithm. Power Syst Technol 41(09):2987–2995