Improved Grey Wolf Optimizer with Differential Perturbation for Function Optimization

Qiang Qu, *Member, IAENG*, Hai-hua Wang, and Mei-li Qi

***Abstract*—Grey wolf optimizer (GWO) is a biological-inspired optimization algorithm with the advantages of few parameters, robustness, and easy implementation. It is widely used to solve the function optimization problem. However, GWO can easily fall into local optimum and can suffer from premature convergence. In this study, an improved GWO with differential perturbation, called IGWO, is presented. First, a non-linear reduction strategy is used instead of the linear reduction strategy in GWO to update the convergence factor, which increases the global search capability of IGWO. In addition, a random differential perturbation strategy with strong exploitation capability is embedded in GWO to increase the diversity of the population and ensure the local exploitation capability of IGWO. Finally, IGWO is tested with 16 benchmark functions. The simulation results show that IGWO outperforms PSO, GSA, GWO, ALO, MFO, mGWO, DE-GWO, wd-GWO, and HGWO in terms of convergence accuracy and convergence speed.**

***Index Terms*—grey wolf optimizer, function optimization, differential perturbation, non-linear reduction strategy**

I. Introduction

The practical problems in engineering and other fields are often modeled as problems of function optimization with constraints, and the optimal solutions that meet the functions and constraints can be obtained by global optimization algorithms. The traditional methods for solving global optimization problems are either direct calculation methods or gradient-based methods. For a simple global optimization problem, an analytical solution can be obtained by directly solving the equations of index functions and constraints. For complex global optimization problems where the analytic solution cannot be obtained, one typically uses the first or/and second order gradients of index functions and constraints to guide the search process and obtain the approximate solution. However, it is difficult for traditional optimization algorithms to obtain the optimal solution for a complex global optimization problem, owing to their many local optimums, complex search spaces, and model uncertainty, especially for high-dimensional optimization problems. As a result, in recent years researchers have proposed meta-heuristic algorithms, such as particle swarm optimization (PSO) algorithm [1], ant colony optimization (ACO) algorithm [2], gravitational search algorithm (GSA) [3], moth-flame optimization (MFO) algorithm [4], ant lion optimizer [ALO] [5], a hybrid algorithm based on gravitational search and particle swarm optimization algorithm [6], an improved flower pollination algorithm [7], an improved krill herd algorithm [8], and complementary differential evolution based whale optimization algorithm [9], to solve global optimization problems and obtain better results.

Grey wolf optimizer (GWO) is a novel meta-heuristic algorithm proposed by Mirjalili S. *et al.* in 2014 [10]. Because GWO is simple in principle, and features few parameters, robustness, and easy implementation, it is used to solve global optimal problems. Shakarami M. R. *et al.* used the GWO-based strategy to design the wide area power system stabilizer (WAPSS) and obtained better results than PSO, GA, and DE [11]. For training a q-Gaussian radial basis functional link-nets neural network, Muangkote *et al.* proposed an improved version of GWO [12]. Madadiina A. *et al.* used GWO to adjust the PID controller parameters in DC motors and obtained good results [13].

However, like other meta-heuristic algorithms, GWO also has the disadvantages of poor population diversity, easy stagnation, and difficulty in balancing its exploration and exploitation capabilities. To enhance the performance of GWO, researchers have proposed many improved algorithms. Mittal N. *et al.* studied the possibility of enhancing the GWO exploration process by reducing the value of convergence factor '*a*' using the exponential decay function, and proposed a modified grey wolf optimizer (mGWO) [14]. Gupta S. *et al.* used random walk strategy to enhance the performance of GWO [14]. To boost the efficiency of GWO, the Lévy flight and greedy selection strategies are integrated with the modified hunting phases of GWO in [15]. Inspired by the principle of light refraction in physics, Long W. *et al.* introduced a refraction learning strategy to improve the searchability of GWO and help the population jump out of the local optima [16]. Kumar V. *et al.* presented a modified parameter '$\vec{C}$' strategy to provide a balance between exploration and exploitation of GWO as well as a new random opposition-based learning strategy to assist GWO to jump out of the local optima. The simulation results revealed that the proposed algorithm shows better or at least competitive

Manuscript received August 21, 2021; revised March 21, 2022. This work was supported in part by National Natural Science Foundation of China (Grant No. 71571091 and 71371092).

Qiang Qu is a professor at the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China (phone: 0086-0412-5929760; e-mail: quqiang@ustl.edu.cn).

Hai-hua Wang is a lecturer at the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China (e-mail: Wanghaiha@ustl.edu.cn).

Mei-li Qi is an assistant at the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China (e-mail: Qimeili@ustl.edu.cn).

performance against other compared algorithms on not only global optimization but also engineering design optimization problems [17]. To force GWO to jump out of the stagnation, Wang J. S. *et al.* integrated differential evolution into GWO to update the previous best position of alpha, beta and delta wolves and presented a novel improved grey wolf optimizer (DE-GWO) [18]. Malik M. R. S. *et al.* used the weighted sum of best locations instead of just a simple average location in GWO to propose the weighted grey wolf optimizer (wd-GWO) [19].

To further balance the exploration and exploitation capabilities of GWO, this study proposes an improved grey wolf optimizer (IGWO) based on the nonlinear reduction strategy for convergence factor 'a' and differential perturbation strategy. First, we propose to use a sine function instead of the linear function used in GWO to update convergence factor 'a'. Note that the value of the sine function is always greater than the value of the linear function. The new update strategy can help IGWO to increase the global search capability and prevent IGWO from falling into the local minimum. Then, we embed the differential perturbation operation into GWO to improve the diversity of the population. Finally, a series of simulations are run to illustrate the effectiveness of IGWO.

The rest of the article is organized as follows. In Section II, the principle of GWO is introduced. The motivation and principle of IGWO are presented in Section III. In Section IV, the simulation experiments and results are described in detail. In Section VI, we present our conclusions.

II. Grey Wolf Optimizer

A. Social hierarchy and hunting

Most wolves live in packs, averaging 5 to 12 wolves per pack and with the strict social leadership hierarchy shown in Fig. 1. In the social leadership hierarchy, wolves can be divided into four groups: alpha, beta, delta, and omega. The alpha is the leader and is mostly responsible for the vital decisions about hunting and selecting a place to live. The second category in the social hierarchy is the beta wolf. The beta wolf assists the alpha in delivering messages or organizing other pack activities. The wolves at the bottom of the hierarchy are regarded as omega, and these wolves balance the internal relationship of the wolves and update their positions according to their leaders. The remaining wolves are delta. The delta submit to alpha and beta, but they dominate the omega.

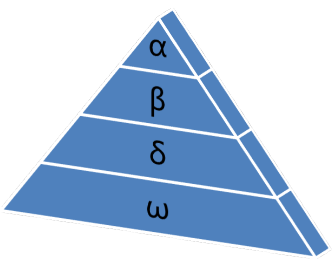


Fig. 1. Hierarchy of grey wolves (dominance decreases from top to bottom)

The mathematical model of the social hierarchy of grey wolves can be built as following: the best solution is considered the alpha (α), the second and third best solutions are respectively assumed to be the beta (β) and delta (δ), and the other solutions are considered omega (ω).

B. Mathematical models of GWO

By imitating the social hierarchy and hunting behavior of grey wolves, GWO solves the globe optimization problem. The hunting process of grey wolves involves searching for the prey, encircling the prey, and attacking the prey.

Encircling the prey

To establish the mathematical model of encircling behavior, the following formulas are used:

$$\vec{D} = \left|\vec{C}\cdot\vec{X}_p(t) - \vec{X}(t)\right|, \tag{1}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}\cdot\vec{D}, \tag{2}$$

where t indicates the current iteration number, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\vec{X}_p$ indicates the vector of prey position, and $\vec{X}$ is the position vector of the grey wolf.

The coefficient vectors $\vec{A}$ and $\vec{C}$ can be expressed as

$$\vec{A} = 2a\cdot rand_1 - a, \tag{3}$$

$$\vec{C} = 2\cdot rand_2, \tag{4}$$

$$a = 2\cdot\left(1 - t/T_{\max}\right), \tag{5}$$

where $rand_1$ and $rand_2$ are random vectors in [0,1], and $T_{\max}$ indicates the maximum number of iterations. $\vec{A}$ is a random vector in the interval $[-a,+a]$, where a is linearly decreased from 2 to 0 throughout the iterations.

As shown in (2), the search agent will be placed at $\vec{A}\cdot\vec{D}$ away from the prey position at the next time. Because $\vec{A}$ is a random vector in the interval $[-a,+a]$ and $\vec{D}$ is the position deviation between the current search agent and the weighted prey by $\vec{C}$, the search agent may be located at any random position within the hypercube determined by (1) and (2), and around the prey.

The search agents update their positions according to the prey. Since we are unable to determine the previous position of the prey and the leaders (alpha, beta, and delta) have a better understanding of the potential position of the prey, we can estimate the position of the prey according to the positions of the alpha, beta, and delta by (6)-(11):

$$\vec{D}_\alpha = \left|\vec{C}\cdot\vec{X}_\alpha(t) - \vec{X}(t)\right|, \tag{6}$$

$$\vec{D}_\beta = \left|\vec{C}\cdot\vec{X}_\beta(t) - \vec{X}(t)\right|, \tag{7}$$

$$\vec{D}_\delta = \left|\vec{C}\cdot\vec{X}_\delta(t) - \vec{X}(t)\right|, \tag{8}$$

$$\vec{X}_1(t) = \vec{X}_\alpha(t) - \vec{A}_1\cdot\vec{D}_\alpha, \tag{9}$$

$$\vec{X}_2(t) = \vec{X}_\beta(t) - \vec{A}_2\cdot\vec{D}_\beta, \tag{10}$$

$$\vec{X}_3(t) = \vec{X}_\delta(t) - \vec{A}_3 \cdot \vec{D}_\delta \,. \tag{11}$$

The position of the search agent can be calculated by

$$\vec{X}(t+1) = \frac{\vec{X}_1(t) + \vec{X}_2(t) + \vec{X}_3(t)}{3} \,. \tag{12}$$

Attacking the prey

To model the process of attacking the prey, the value of convergence factor a decreases with iterations. When $a < 1$, that is, $\vec{A}$ with random values are in $[-1,1]$, the next position of the search agent can be in any position between its current position and the position of the prey. In other words, the search agent will approach the prey at this time.

Searching for the prey

As attacking the prey, $\vec{A}$ is also related to searching for prey in WOA. When $|\vec{A}| > 1$, the grey wolf stays away from the prey to search for a better solution in the decision space.

In other words, when $|\vec{A}| < 1$, GWO emphasizes exploitation and forces the grey wolves to attack towards the prey, and when $|\vec{A}| > 1$, GWO emphasizes exploration and allows the wolves to search globally.

The vector $\vec{C}$ in GWO is a random vector in the range of [0, 2], which provides the random weight of the prey to randomly emphasize ($\vec{C}$>1) or reduce ($\vec{C}$<1) the effect of the alpha, beta, and delta.

C. *Limitations of GWO*

Obtaining better convergence speed and optimization accuracy requires balancing the global search (exploration) and local search (exploitation) capabilities of GWO. In the classical GWO, exploration and exploitation are guaranteed by convergence factor a or $\vec{A}$. As shown in (5), the value of the convergence factor a is linearly reduced from 2 to 0 throughout the iterations. That is, in the first half of the iterations, $\vec{A}$ tends to meet $|\vec{A}| > 1$, so the GWO has strong global search ability, and in the second half of the iterations, $\vec{A}$ tends to meet $|\vec{A}| < 1$, so the GWO has strong local search ability. Meanwhile, it can be seen from (12) that the search agent only updates its position according to the alpha, beta, and delta individuals, so that the diversity of the population drops rapidly, and the possibility of falling into a local minimum increases, particularly for solving the multi-peak function optimization problem.

III. IMPROVED GREY WOLF OPTIMIZER

To overcome the above shortcomings of GWO, this study proposes two strategies: a nonlinear reduction strategy for convergence factor a and a differential perturbation strategy.

A. *Nonlinear reduction strategy for convergence factor a*

As described above, the value of convergence factor a determines the global search and local search capabilities of GWO. To improve the global search ability of GWO and prevent GWO from falling into the local minimum, this paper proposes a nonlinear reduction strategy for convergence factor a. The nonlinear reduction strategy can be written as (13) and is shown in Fig. 2.

$$\begin{cases} a = 2\left(1 - t/T_{\max}\right) & 0 < t \le T_{\max}/2 \\ a = \sin\left(t\pi/T_{\max}\right) & T_{\max}/2 < t \le T_{\max} \end{cases} \tag{13}$$

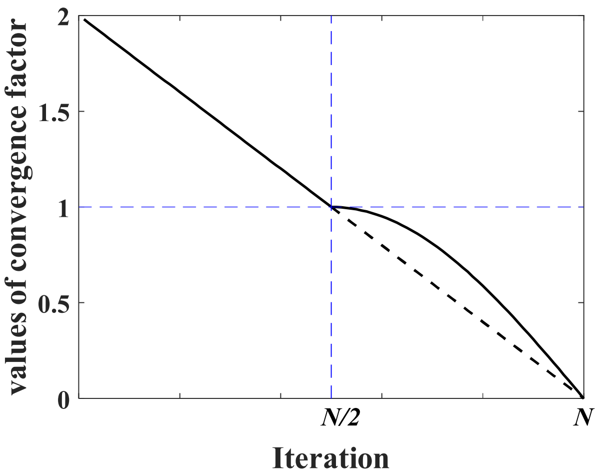


Fig. 2. Shape of convergence factor

As seen in Fig. 2, we use a piecewise function to update the value of convergence factor a. In the first half of the iterations ($0 < t \le T_{\max}/2$), we still use the same linear function as GWO to update convergence factor a. At this stage, a is greater than 1 and IGWO shows strong global search ability. However, in the second half of the iterations ($T_{\max}/2 < t \le T_{\max}$), we use a sine function instead of the linear function applied in GWO. Note that the value of the sine function is always greater than the current value of the linear function. The new update strategy can help IGWO increase the global search capability and prevent IGWO from falling into the local minimum.

B. *Differential perturbation strategy*

In GWO, the grey wolves update their positions by the positions estimated by the alpha, beta, and delta according to (12), and the generated search agents go directly into the next generation, which affects the diversity of the grey wolf population to some extent. To improve the diversity of the grey wolf population and reduce the possibility of falling into the local optimum, we introduce a differential perturbation strategy to generate the disturbed search agent, and then choose the better one between the search agent generated by (12) and the disturbed search agent to enter the next generation.

The differential perturbation can be described as

$$\vec{X}'(t+1) = \vec{X}(t+1) + \vec{\varepsilon} \,, \tag{14}$$

where $\vec{X}$(t+1) is the search agent generated according to (12), $\vec{\varepsilon}$ is the differential perturbation vector, and $\vec{X}'(t+1)$

is the disturbed search agent. After obtaining the disturbed search agent, we use the greedy algorithm to select the better search agent to enter the next generation. That is,

$$\vec{X}(t+1)=\begin{cases}\vec{X}(t+1) & \text{if } f(\vec{X}(t+1))<f(\vec{X}'(t+1))\\ \vec{X}'(t+1) & \text{else}\end{cases}. \quad (15)$$

The perturbation vector $\vec{\varepsilon}$ can be described as

$$\begin{aligned}\vec{\varepsilon}&=(1-\gamma)\, rand_1\left(\vec{X}(t)-\vec{X}_b(t)\right)\\ &+\gamma\, rand_2\left(\vec{X}_{r1}(t)-\vec{X}_{r2}(t)\right)\end{aligned}, \quad (16)$$

$$\gamma=\left(1-t/T_{\max}\right)^2-\gamma_{\min}, \quad (17)$$

where $\vec{X}_b$ is the position vector randomly selected from the positions of alpha, beta, or delta. γ is the perturbation scope. $rand_1$ and $rand_2$ are random vectors in [0,1]. $\vec{X}_{r1}$ and $\vec{X}_{r2}$ are the position vectors of two different wolves randomly selected from population. $T_{\max}$ indicates the maximum number of iterations, and $\gamma_{\min}$ is the minimum value of γ.

Algorithm Improved grey wolf optimizer

```
Initialize the grey wolf population X_i, (i = 1,2,···,N)
Initialize a by (13)
Initialize A and C by (3) and (4)
Calculate the fitness of each search agent
Set X_α =the best search agent
Set X_β =the second-best search agent
Set X_δ =the third-best search agent
while ( t <Maximum number of iterations)
  for each search agent
    Calculate the position of the generated search agent by (12)
    Calculate the perturbation vector ε by (16)
    Calculate the position of disturbed search agent X'(t+1) by
(14)
    Update the position of current search agent by (15)
  end for
  Update parameters a , A , and C
  Calculate the fitness of all search agents
  Update X_α , X_β , X_δ
   t = t+1
end while
return X_α
```

Fig. 3. Framework of IGWO

Note that $\vec{X}_b(t)$ is a random position vector of alpha, beta, or delta, and $(1-\gamma)<1$. The first half of (16) moves the search agent closer to the better solution (alpha, beta, or delta), which helps IGWO improve the convergence speed. The second half of (16) brings the search agent closer to the difference vector $\vec{X}_{r1}(t)-\vec{X}_{r2}(t)$. Because $\vec{X}_{r1}(t)$ and $\vec{X}_{r2}(t)$ are two different individuals randomly selected from the population, the difference vector $\vec{X}_{r1}(t)-\vec{X}_{r2}(t)$ has strong randomness, so this part is conducive to improving the diversity of search agents and preventing IGWO from falling into a local minimum.

In addition, from the basic principle of the meta-heuristic algorithm, it should searche as large a decision space as possible to increase the probability of obtaining a better solution in early iterations, and searche for a nearly optimal solution to improve the convergence accuracy in later iterations. To meet the requirement, the perturbation scope γ should decrease as iterations increase. In this paper, γ is a quadratic function, as shown in (17).

C. *Pseudocode of IGWO algorithm*

The pseudocode of IGWO is shown in Fig. 3.

IV. Simulation Results and Analysis

A. *Test functions*

In the simulation experiments, 16 standard functions were used to verify the performances of IGWO. The 16 benchmark functions are shown in Table I, which contains the search range (Range), dimension (D.) and minimum values of the benchmark functions (F_{min}). The benchmark functions were divided into two groups: the unimodal and multimodal functions. Because they have only one global optimum, the unimodal functions (f_1 - f_8) were used to examine the convergence rate of the meta-heuristic algorithms. The multimodal functions (f_9 - f_{16}) with multiple local optimums were more suitable for evaluating exploration [20]. The experimental platform was MATLAB R2016a, running on a 64-bit Windows 10 computer with an Intel Core i7-3820Qm with 2.5GHz and 8GB RAM. Each simulation experiment was run 30 times.

B. *Simulation results for other meta-heuristic algorithms*

To verify the performance of the improved algorithm, we compared the proposed IGWO with GWO [10], PSO [1], ACO [2], GSA [3], and MFO [4]. We set the same common parameters for all the algorithms for a fair comparison. The dimensions of each function were set to 30. Each simulation was run 30 times, and the maximum number of iterations was 500. The other operating parameters of each algorithm are shown in Table II. The results of simulation are listed in Tables III and IV, and the function convergence curves are shown in Fig. 4. Tables III and IV list the optimum (Opt.), average value (Ave.), and standard deviation (Std.) for each experiment result.

From Table III, we can see that IGWO shows excellent optimization performances for the unimodal functions. It is the best optimizer for functions F1, F2, F3, F4, F6, F7, and F8. Although the convergence accuracy of IGWO is slightly lower than those of GSA, PSO, and ALO on F5, it still obtains relatively ideal results. Meanwhile, as can be seen from Table IV, IGWO is also competitive in optimizing multimodal functions. Except for F1 and F13, the average of best fitness obtained by IGWO is better than that obtained by the other five meta-heuristic algorithms. Meanwhile, IGWO also obtains the theoretical optimum for F9, F11, and F15.

TABLE I
STANDARD BENCHMARK FUNCTIONS

Function	D.	Range	F_{min}
$f_1(\vec{X})=\sum_{i=1}^{D} x_i^2$	30	[-100,100]	0
$f_2(\vec{X})=\sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2$	30	[-100,100]	0
$f_3(\vec{X})=\max_i\left\{\lvert x_i\rvert, 1\le x_i\le D\right\}$	30	[-100,100]	0
$f_4(\vec{X})=\sum_{i=1}^{D}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	30	[-30,30]	0
$f_5(\vec{X})=\left(\lvert x_i+0.5\rvert\right)^2$	30	[-100,100]	0
$f_6(\vec{X})=\sum_{i=1}^{D} i x_i^4+rand[0,1)$	30	[-1.28,1.28]	0
$f_7(\vec{X})=\sum_{i=1}^{D} i x_i^4$	30	[-1.28,1.28]	0
$f_8(\vec{X})=\sum_{i=1}^{D} i x_i^2$	30	[-10,10]	0
$f_9(\vec{X})=\sum_{i=1}^{D}\left[x_i^2-10\cos\left(2\pi x_i\right)+10\right]$	30	[-5.12,5.12]	0
$f_{10}(\vec{X})=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{D}\cos\left(2\pi x_i\right)\right)+20+e$	30	[-32,32]	0
$f_{11}(\vec{X})=\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[-600，600]	0
$f_{12}(\vec{X})=\sum_{i=1}^{D}\lvert x_i\sin\left(x_i\right)+0.1x_i\rvert$	30	[-10,10]	0
$f_{13}(\vec{X})=\sum_{i=1}^{D}\left(x_i-1\right)^2\left[1+\sin^2\left(3\pi x_{i+1}\right)\right]+\sin^2\left(3\pi x_1\right)+\lvert x_D-1\rvert\left[1+\sin^2\left(3\pi x_D\right)\right]$	30	[-10,10]	0
$f_{14}(\vec{X})=1-\cos\left(2\pi\sum_{i=1}^{D}x_i^2\right)+0.1\sqrt{\sum_{i=1}^{D}x_i^2}$	30	[-100,100]	0
$f_{15}(\vec{X})=0.1D-\left(0.1\sum_{i=1}^{D}\cos\left(5\pi x_i\right)-\sum_{i=1}^{D}x_i^2\right)$	30	[-1,1]	0
$f_{16}(\vec{X})=0.5+\frac{\sin^2\left(\sqrt{\sum_{i=1}^{D}x_i^2}\right)-0.5}{\left(1+0.001\cdot\sum_{i=1}^{D}x_i^2\right)^2}$	30	[-100,100]	0

TABLE II
PARAMETER SETTINGS OF THE ALGORITHMS

Name of parameter	Parameter values
Population size	$n=30$
Maximum number of iterations	$T_{\max}=500$
Learning factors	$c_1=c_2=2$
Inertia weight	$w=0.9$
Coefficient of gravity	$G_0=100$
Scaling factor	$F=0.5$
Crossover probability	$P_C=0.8$
Evolutionary factor	μ=0.75
Control parameter maximum	$\psi_{\max}=0.9$
Control parameter minimum	$\psi_{\min}=0.1$

The convergence curves of the GWO, PSO, GSA, ALO, MFO, and IGWO algorithms on 16 benchmark functions are shown in Fig. 4. IGWO shows different convergence characteristics in the exploration and exploitation phases.

In the exploration phase, the convergence speed of the two GWO-based meta-heuristic algorithms (GWO and IGWO) are faster than that of PSO, GSA, ALO, MFO, particularly for functions F1, F5, F6, F7, F8, F9, F10, F11, F12, and F15. In the exploitation phase, the convergence accuracy of GWO and IGWO tends to increase as the iteration increases, except for functions F4, F5, F10, F13, F14, and F16.

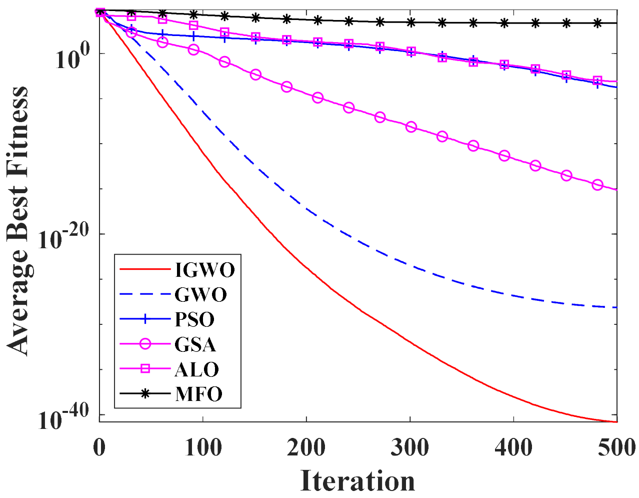
There are two possible reasons for this phenomenon. First, the social hierarchy strategy helps GWO and IWGO to save the best solutions over the course of iterations. Second, the adaptive value of parameter A helps GWO and IWGO to convert between exploration and exploitation seamlessly.

TABLE III
SIMULATION RESULTS FOR FUNCTIONS F1-8

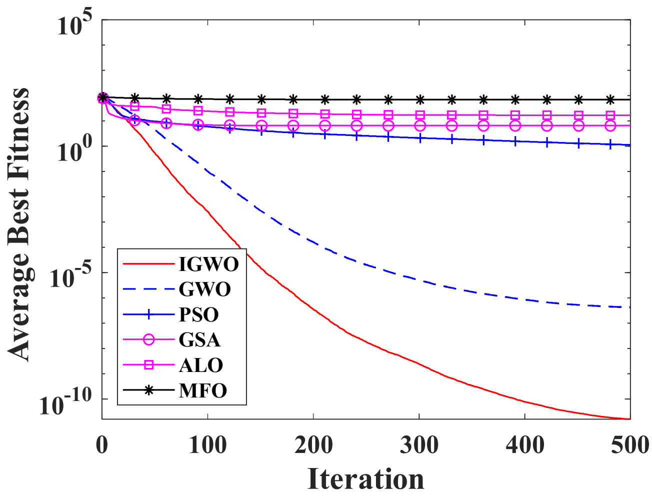
Algorithms		F1	F2	F3	F4	F5	F6	F7	F8
PSO	Ave.	1.3958e-04	72.7208	1.0473	95.7994	1.0411e-04	0.1786	2.3408e-05	0.0023
	Std.	1.4448e-04	23.5901	0.1752	71.9624	1.1643e-04	0.0567	4.1792e-05	0.0049
	Opt.	1.3547e-04	1.4643e+02	0.8525	82.9454	3.0056e-04	0.2807	1.6635e-06	8.3150e-05
GSA	Ave.	2.3483e-16	9.6132e+02	7.4497	56.5141	7.4402e-12	0.0827	5.4248e-32	5.3582e-04
	Std.	1.1245e-16	4.0860e+02	2.1257	59.3262	3.7542e-6	0.0422	4.9356e-32	0.0022
	Opt.	1.3100e-16	9.7564e+02	8.1473	26.7833	1.9861e-12	0.0591	1.5283e-31	2.0025e-15
GWO	Ave.	1.5921e-27	1.8178e-05	7.2412e-07	27.1900	0.0023	0.0020	3.8314e-50	2.1118e-28
	Std.	3.6921e-27	2.7256e-05	8.5566e-07	0.7936	0.0013	0.0012	1.9903e-49	3.9759e-28
	Opt.	1.0428e-27	9.4476e-07	3.1608e-07	27.1293	0.5076	0.0015	2.9970e-51	4.9947e-28
ALO	Ave.	0.0012	4.6588e+03	16.0751	4.4583e+02	1.6732e-04	0.2690	5.4057e-09	3.7387
	Std.	9.4469e-04	1.6609e+03	4.4535	5.4008e+02	1.7523e-04	0.0772	1.3360e-08	3.4423
	Opt.	0.0013	8.3690e+02	17.0836	1.4736e+03	5.4720e-04	0.2869	3.2215e-09	3.6727
MFO	Ave.	3.0039e+03	1.8149e+04	65.9201	1.6623e+04	6.8621e+02	2.5295	0.8951	5.7689e+02
	Std.	5.3482e+03	1.1468e+04	10.5364	3.3972e+04	2.5598e+03	5.1330	2.3731	6.4571e+02
	Opt.	2.6982	2.5429e+04	77.2426	9.1517e+04	1.0102e+04	0.0683	1.7268e-04	7.0068e+02
IGWO	Ave.	7.4656e-40	7.0045e-10	4.0672-11	26.6990	0.0015	0.0011	1.4858e-71	5.4909e-41
	Std.	1.5921-39	1.6393e-09	4.8888e-11	0.6412	0.0011	5.6735e-04	5.0364e-71	8.4216e-41
	Opt.	2.0748e-42	3.0109e-11	6.9014e-12	26.1736	2.6678e-02	9.2180e-04	9.9275e-73	3.9570e-40

TABLE IV
SIMULATION RESULTS FOR FUNCTIONS F9-16

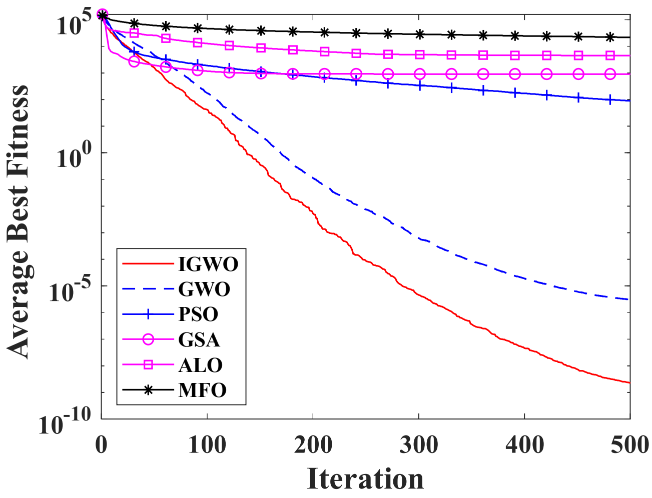
Algorithms		F9	F10	F11	F12	F13	F14	F15	F16
PSO	Ave.	56.3088	0.3112	8.5503e-06	0.0584	0.0623	0.4399	0.4719	0.0758
	Std.	14.3647	0.6181	0.0109	0.1379	0.1818	0.0674	0.2745	0.0092
	Opt.	54.1370	0.0083	7.1805e-06	0.0306	3.9192e-16	0.4999	0.4443	0.0782
GSA	Ave.	26.7975	5.1341e-07	3.0706e-08	4.5135e-04	0.0442	4.1125	1.0838e-14	0.5376
	Std.	7.6666	2.7485e-06	7.2028	0.0022	0.0893	2.7421	1.4998e-14	0.0412
	Opt.	34.8235	6.7165e-09	1.1496e-08	8.1037e-09	0.1102	2.5027	2.2204e-15	0.0482
GWO	Ave.	1.9963	1.0107e-13	0.0031	2.4221e-16	0.3869	0.2032	1.4803e-17	0.0231
	Std.	2.3889	1.3976e-14	0.0066	2.2494e-16	0.8069	0.0450	5.1035e-17	0.0640
	Opt.	5.5914	1.0036e-13	0.0122	6.6019e-17	0.2384	0.3999	0	0.0782
ALO	Ave.	78.4055	5.0644	0.0577	17.8095	0.0579	2.6632	1.7434	0.0293
	Std.	19.3345	3.4041	0.0339	6.4582	0.0589	0.8336	0.4701	0.0402
	Opt.	82.5818	2.8145	0.06186	21.2666	0.0058	0.9999	1.4778	0.4942
MFO	Ave.	1.60374e+02	15.4657	9.9192	55.4070	11.7751	4.9099	0.7344	0.6271
	Std.	46.8957	6.3272	27.5003	16.2548	5.1453	3.3010	0.6102	0.0233
	Opt.	30.8437	19.9549	1.0419	53.5266	0.1876	3.9999	0.5917	0.4825
IGWO	Ave.	0.1083	8.1120e-15	0.0047	2.1899e-24	1.0914	0.1199	0	0.0152
	Std.	0.5934	6.4863e-16	0.0018	2.2331e-24	0.7234	0.0305	0	0.0112
	Opt.	0	7.9936e-15	0	7.1114e-25	0.0713	0.0999	0	0.0032



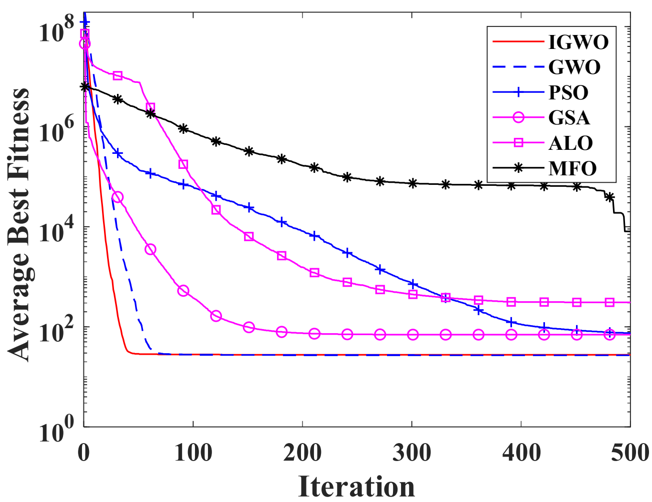
(a) F1



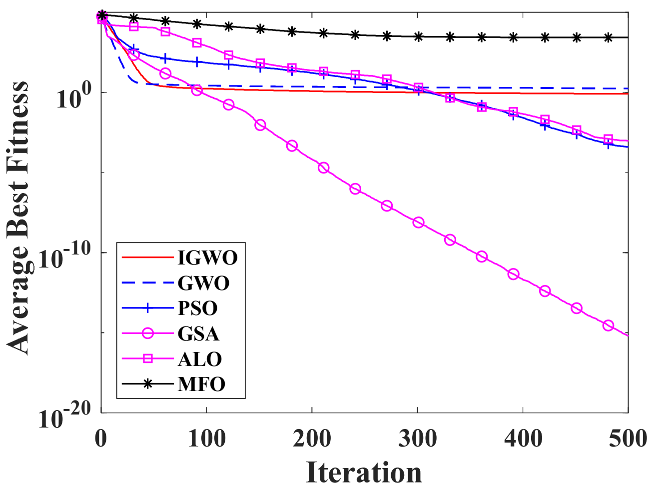
(c) F3



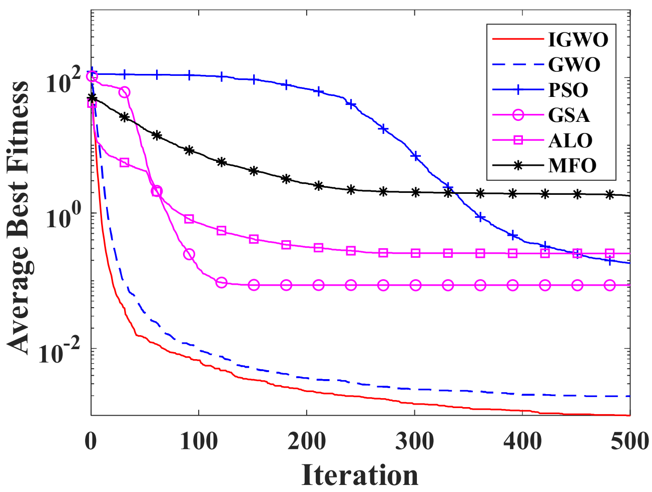
(b) F2



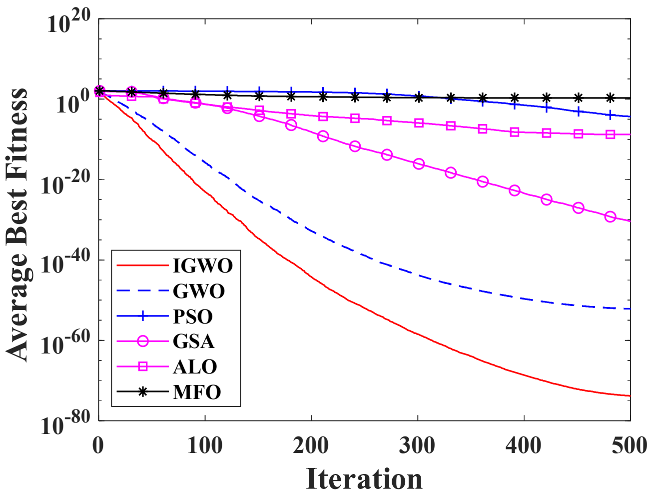
(d) F4



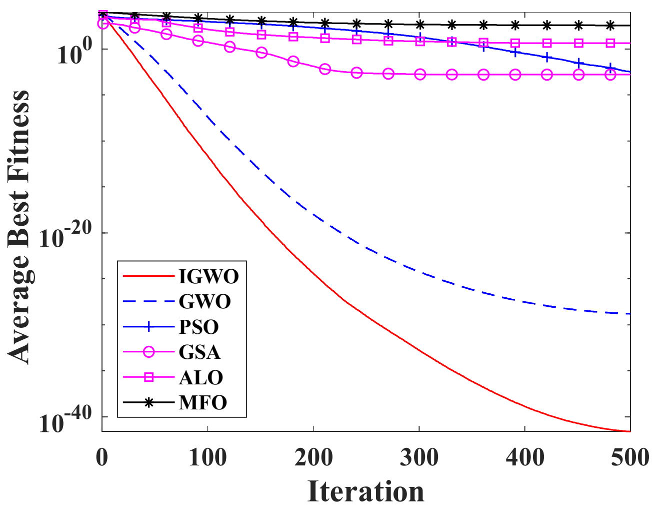
(e) F5



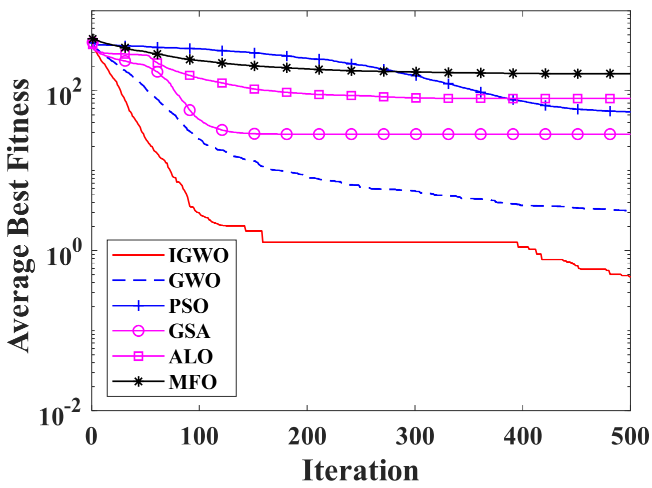
(f) F6



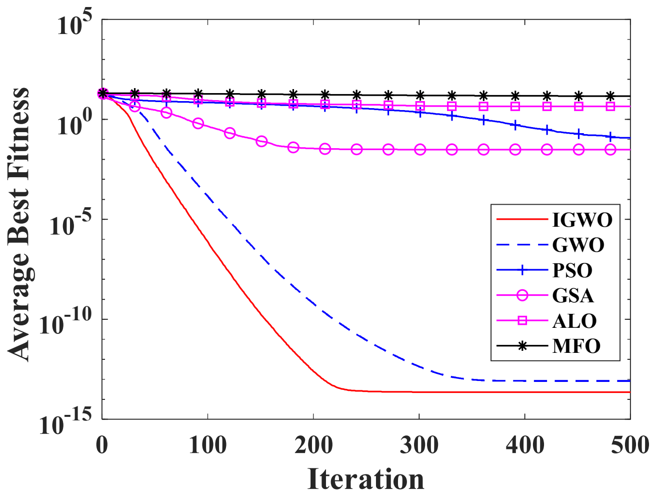
(g) F7



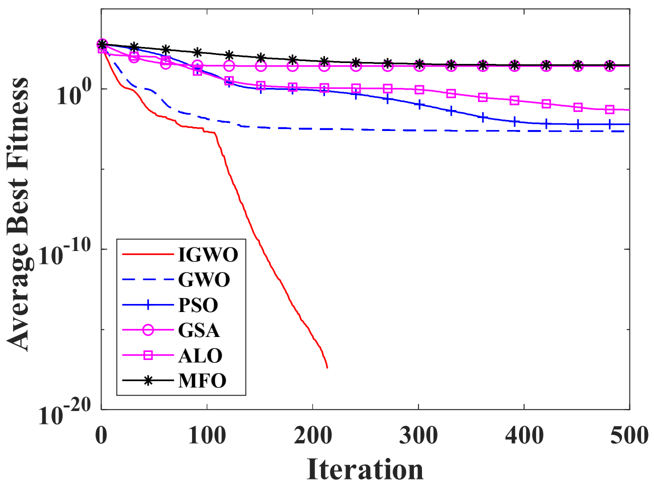
(h) F8



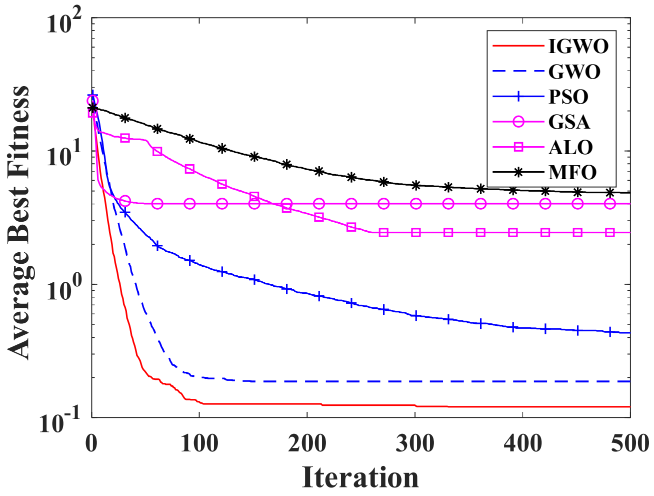
(i) F9



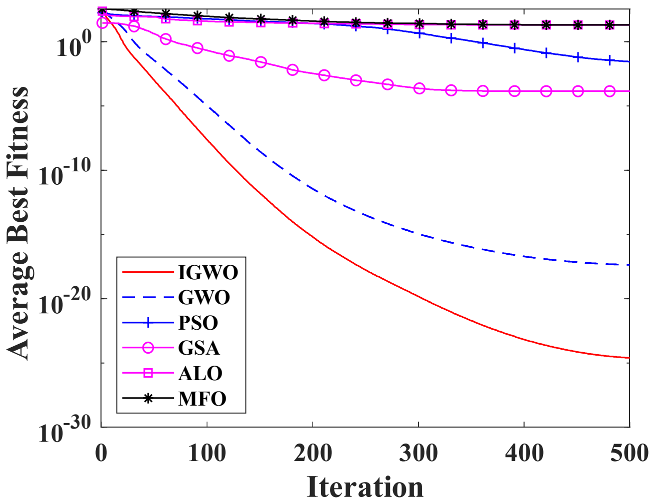
(j) F10



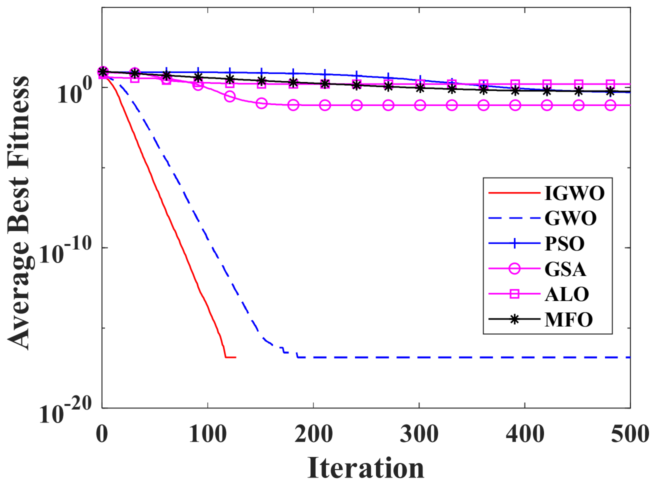
(k) F11



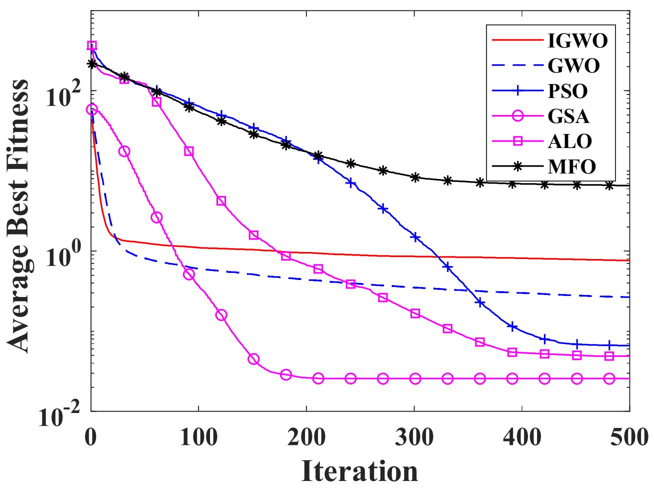
(n) F14



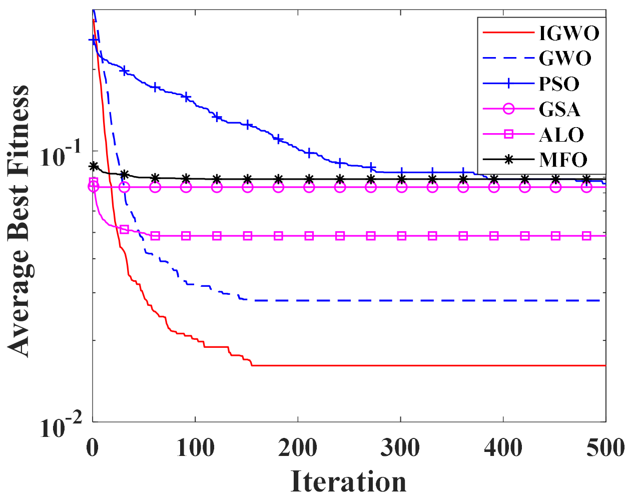
(l) F12



(o) F15



(m) F13



(p) F16

Fig. 4. Convergence curves of functions F1–F16

TABLE V
SIMULATION RESULTS FOR FUNCTIONS F1-8

Algorithms		F1	F2	F3	F4	F5	F6	F7	F8
mGWO	Ave.	5.9349e-34	1.6555e-09	1.2833e-07	27.9672	0.0012	0.0014	1.0113e-51	1.4983e-28
	Std.	3.1468e-34	4.8511e-07	1.6184e-08	0.8623	5.8991e-04	6.5336e-04	4.9154e-51	8.2044e-28
	Opt.	7.3930e-35	6.8313e-10	4.1043e-8	27.1543	0.0026	0.0012	3.3569e-59	2.1743e-33
DE-GWO	Ave.	2.0635e-14	0.2969	0.0214	27.4035	0.0053	0.0007	1.6197e-27	2.9768e-15
	Std.	8.4086e-14	1.0253	0.0507	0.7327	0.0053	0.0039	3.9169e-27	9.9082e-15
	Opt.	1.8219e-16	0.3231	7.6619e-04	27.9672	0.0082	0.0024	7.1920e-28	3.1387e-16
wd-GWO	Ave.	2.6691e-03	4.1108e-04	0.8370	27.3924	0.0287	1.3481	0.9738	2.7801e-12
	Std.	8.8200e-02	9.2331e-03	0.8477	20.4476	0.0287	0.8414	0.7808	1.0275e-02
	Opt.	4.2022e-03	3.8150e-04	0.63905	29.5832	0.0196	0.9690	0.2478	2.3843e-02
HGWO	Ave.	5.4087e-31	1.3703e-06	0.0011	27.3792	0.0540	0.0317	9.5423e-55	9.5962e-31
	Std.	2.6097e-30	6.8558e-06	0.0023	10.8831	0.2209	0.0311	5.1552e-54	3.7939e-31
	Opt.	2.7624e-31	6.6975e-09	0.0091	27.0415	8.8006e-06	0.0012	2.8246e-63	9.5955e-34
IGWO	Ave.	7.4656e-40	7.0045e-10	4.0672-11	26.6990	0.0015	0.0011	1.4858e-71	5.4909e-41
	Std.	1.5921-39	1.6393e-09	4.8888e-11	0.6412	0.0011	5.6735e-04	5.0364e-71	8.4216e-41
	Opt.	2.0748e-42	3.0109e-11	6.9014e-12	26.1736	0.02.66	9.2180e-04	9.9275e-73	3.9570e-40

TABLE VI
SIMULATION RESULTS FOR FUNCTIONS F9-16

Algorithms		F9	F10	F11	F12	F13	F14	F15	F16
mGWO	Ave.	0.4867	2.3033e-14	3.2169e-04	3.0950e-22	2.8137	0.1365	2.4532e-18	0.0190
	Std.	1.6200	4.3445e-15	0.5018	6.4190e-22	1.1220	0.0490	2.7854e-18	0.0131
	Opt.	0	1.8652e-14	0	1.3422e-21	3.8325	0.4999	0	0.0097
DE-GWO	Ave.	2.3334	2.7110	0.0423	1.6986e-08	10.0323	0.1799	1.9244e-16	0.0244
	Std.	7.0097	7.0300	0.0155	8.6363e-08	1.7855	0.0551	5.6755e-16	0.0140
	Opt.	1.2869e-10	1.5423e-08	2.4869e-14	2.3739e-11	8.5120	0.2999	0	0.0372
wd-GWO	Ave.	2.5765e+02	1.9966	0.2632	4.6027e-02	31.8524	7.5449	1.7733	0.1778
	Std.	39.4685	0.0039	0.9472	1.8697e-02	12.7235	1.4435	0.4910	0.0441
	Opt.	3.0944e+02	19.9640	0.2183	3.2339e-02	20.7585	8.8924	2.0048	0.1785
HGWO	Ave.	8.3213	2.5994e-14	0.5324	0.0663	0.3108	0.0356	1.6543e-20	0.0094
	Std.	13.3464	1.5186e-14	0.4325	0.3634	1.1978	0.0598	7.9876e-21	0.0479
	Opt.	0	3.9968e-14	0	6.3625e-24	1.8480e-05	4.1436e-12	0	0.0980
IGWO	Ave.	0.1083	8.1120e-15	0.0047	2.1899e-24	1.0914	0.1199	0	0.0152
	Std.	0.5934	6.4863e-16	0.0018	2.2331e-24	0.7234	0.0305	0	0.0112
	Opt.	0	7.9936e-15	0	7.1114e-25	0.0713	0.0999	0	0.0032

Furthermore, from Fig. 4, we can also see that the convergence characteristics (convergence speed and convergence accuracy) of IGWO are better than those of GWO, except for function F13. This is mainly because the random differential perturbation strategy with strong randomness increases the diversity of population and improves the local exploitation capability of IGWO. In addition, the non-linear reduction strategy of parameter a also helps IGWO to increase the global search capability.

C. Simulation results for the other improved grey wolf optimizer

To further show the effectiveness of the IGWO proposed in this paper, we run the simulation using mGWO [14], DE-GWO [18], wd-GWO [19], and HGWO [21] based on the above 16 benchmark functions. For fairness, we set the same common control parameters for all algorithms. The population size N was set to 30, and the maximum number of iterations was set to 500. The dimensions of each function were set to 30. Each algorithm was run 30 times independently. Tables V and VI show the simulation results, where the Ave. and Std. represent the average and standard deviation of the best fitness obtained so far, respectively, and Opt. represents the optimum of the best fitness.

It can be seen from Tables V and VI that IGWO is also competitive when compared with mGWO, DE-GWO, wd-GWO, and HGWO. The average best fitness obtained by IGWO is also better than the other four algorithms for functions F1, F2, F3, F4, F7, F8, F9, F10, F14, and F15. IGWO has similar results for functions F5, F6, and F16 on the average and the best fitness.

V. CONCLUSION

As a novel meta-heuristic algorithm, GWO has increasingly attracted the attention of researchers. However, it has two shortcomings: slow convergence and easily falling into local optimum. So, this study introduces an improved nonlinear reduction strategy for convergence factor a to help GWO balance exploitation and exploration. Then, a differential perturbation strategy is introduced to enhance the exploitation of GWO. Finally, the proposed GWO is tested with 16 benchmark functions. The simulation results show that the proposed GWO is better in terms of convergence speed and accuracy than other meta-heuristic algorithms, such as GWO, PSO, GSA, ALO, MFO, mGWO, DE-GWO, wd-GWO, and HGWO.

REFERENCES

[1] L. Wang, H. Geng, P. Liu, K. Lu, J. Kolodziej, R. Ranjan, and A. Y. Zomaya, “Particle swarm optimization-based dictionary learning for remote sensing big data,” *Knowledge-based Systems*, no. 79, pp. 43-50, 2015.

[2] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28-39, 2006.

[3] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, “GSA: a gravitational search algorithm,” *Information Sciences*, vol. 179, no. 13, pp. 2232- 2248, 2009.

[4] S. Mirjalili, “Moth-flame optimization algorithm: a novel nature inspired heuristic paradigm,” *Knowledge-based Systems*, no. 89, pp. 228-249, 2015.

[5] S. Mirjalili, “The ant lion optimizer,” *Advances in Engineering Software*, no. 83, pp. 80-98, 2015.

[6] J. S. Wang, and J. D. Song, “A hybrid algorithm based on gravitational search and particle swarm optimization algorithm to solve function optimization problems”, *Engineering Letters*, vol.25, no.1, pp.22-29, 2017.

[7] X. X. Ma and J. S. Wang, “An improved flower pollination algorithm to solve function optimization problem,” *IAENG International Journal of Computer Science*, vol.45, no.3, pp. 364-370, 2018.

[8] Y. F. Sun, J. S. Wang, and X. X. Ma, “An improved krill herd algorithm based on meme grouping strategy of shuffled frog leaping algorithm,” *IAENG International Journal of Computer Science*, vol. 46, no.2, 156-162, 2019.

[9] Q. Qu, Y. H. Huang, X. L. Wang, and X. B. Chen, "Complementary differential evolution-based whale optimization algorithm for function optimization," *IAENG International Journal of Computer Science*, vol. 47, no.4, pp. 805-815, 2020.

[10] S. Mirjalil, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, no. 69, pp. 46-61, 2014.

[11] M. R. Shakarami, and I. F. Davoudkhani, “Wide-area power system stabilizer design based on grey wolf optimization algorithm considering the time delay,” *Electric Power Systems Research*, no. 133, pp. 149-159, 2016.

[12] N. Muangkote, K. Sunat, and S. Chiewchanwattana, “An improved grey wolf optimizer for training q-Gaussian radial basis functional-link nets,” *2014 International Computer Science and Engineering Conference*, Khon Kaen, pp. 209-214, 2014.

[13] A. Madadi, and M.M. Motlagh, “Optimal control of DC motor using grey wolf optimizer algorithm”, *Technical Journal of Engineering and Applied Science*, vol. 4, no. 4, pp. 373-379, 2014.

[14] N. Mittal, U. Singh, and B. S. Sohi, “Modified grey wolf optimizer for global engineering optimization,” *Applied Computational Intelligence and Soft Computing*, vol. 2016, pp. 1-17, 2016.

[15] A. A. Heidari, and P. Pahlavani, “An efficient modified grey wolf optimizer with Levy flight for optimization tasks,” *Applied Soft Computing*, vol. 60, pp. 115–134, 2017.

[16] W. Long, T. Wu, S. Cai, X. Liang, J. Jiao, and M. Xu, “A novel grey wolf optimizer algorithm with refraction learning,” *IEEE Access*, vol. 7, pp. 57805-57819, 2019.

[17] V. Kumar, and D. Kumar, “An astrophysics-inspired grey wolf algorithm for numerical optimization and its application to engineering design problems,” *Advances in Engineering Software*, vol. 112, pp. 231-154, 2017.

[18] J. S. Wang, and S. X. Li, “An improved grey wolf optimizer based on differential evolution and elimination mechanism,” *Scientific Reports*, vol. 9, no. 1, pp. 1-21, 2019

[19] M. R. S. Malik, E. R. Mohideen, and L. Ali, “Weighted distance grey wolf optimizer for global optimization problems,” *2015 IEEE International Conference on Computational Intelligence and Computing Research,* Madurai, 2015.

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

[21] Z. M. Gao, J. Zhao, Y. Yang, and X. J. Tian, “The hybrid grey wolf optimization-slime mould algorithm,” *Journal of Physics: Conference Series*, vol. 1617, no. 1, 2020.

Qiang Qu (M’20) received the M.Sc. degree in control science and engineering from University of Science and Technology Liaoning, China in 2002, and the Ph.D. degree in signal and information processing from Dalian University of Technology, China, in 2010. He is currently a professor and Master's supervisor in School of Electronic and Information Engineering, University of Science and Technology Liaoning. His main research interests include modeling of complex industry process, intelligent control, and swarm intelligence.

Hai-hua Wang received the M.Sc. degree in control science and engineering from University of Science and Technology Liaoning, China in 2001. She is a Lecturer in School of Electronic and Information Engineering, University of Science and Technology Liaoning. Her main research interests include swarm intelligence, and intelligent control.

Mei-li Qi received the M.Sc. degree in control science and engineering from University of Science and Technology Liaoning, China in 2021. Her main research interests include swarm intelligence, and intelligent control.