Improved grey wolf optimisation algorithms

Zheng-ming Gao1✉, Juan Zhao2, Yu-rong Hu3

1School of Computer Engineering, Jingchu University of Technology, Jingmen, People’s Republic of China
2School of Electronics and Information Engineering, Jingchu University of Technology, Jingmen, People’s Republic of China
3Department of Science and Technology, Jingchu University of Technology, Jingmen, People’s Republic of China
✉E-mail: gaozming@jcut.edu.cn

Abstract: The grey wolf optimisation (GWO) algorithm has fewer numbers of variables and appears quite simple with outstanding capabilities in solving the problems, which are used to describe mathematically what human met in nature. However, it still has its capability to be improved in the convergence ratio, stability, and reduce the errors. And it is also easily trapped in local optimum and converged slowly approaching the end, which is just the same defect appearing in other meta-heuristic algorithms such as the bat algorithm (BA), the particle swarm optimisation (PSO) algorithm, and the genetic algorithm (GA). Lots of improvements have been proposed before. In this paper, we propose an improved GWO algorithm inspired by the PSO algorithm to fasten the convergence ratio and reduce the errors. Empirical work and verifications are carried out; and results show its better performance than the standard GWO algorithm and other well-known meta-heuristic algorithms.

1 Introduction

Lots of optimisation algorithms have been proposed since 1950s, and most of them could be classified theoretically in two kinds: the deterministic algorithms and the stochastic algorithms [1]. Along with the computing hardware developments, the stochastic algorithms are paid more and more attention to. Then, among of the stochastic algorithms, the bionic algorithms have been a hot spot. A lot of bionic algorithms have been proposed such as the genetic algorithm (GA) [2], the simulated annealing (SA) [3], and the ant colony optimisation (ACO) algorithm [4]. Traditionally, the individuals are considered to play a same role in optimisation in most of these algorithms. The intelligence and the social hierarchy are not included. However, an Australian team led by Seyedali Mirjalili in 2014 proposed a new algorithm called the grey wolf optimisation (GWO) algorithm [5]. For the first time, the intelligent structure was introduced. The grey wolves was classified into four groups based on their social hierarchy, the alpha wolf, the beta wolf, the delta wolf, and the rest of them are all called the omega wolves. Consequently, the novel algorithm was proved to be more capable to solve the problems which were used to describe mathematically what the human met in nature. Therefore, the GWO algorithm had been paid great attention to just since its birth.

The GWO algorithm has been proved to be efficient and capable to solve the problems human met. It has been applied in diagnosis of diseases [6], dataset clustering [7], denoising of images [8], feature selection [9], economic load dispatch problems [10], multi-objective problems [11], and image segmentations [12]. Then, a lot of improvements are proposed worldwide to the original GWO algorithm together with the random walk [13], chaos [14], binaries [15], and levy flights [16].

In this paper, we proposed an improved GWO algorithm inspired by the PSO algorithm to fasten the convergence ratio and lower the residual errors. For simplicity, the original GWO algorithm is briefly described in Section 2 and the improved GWO algorithm is described in detail in Section 3. Simulations are carried out; results and conclusions are made in Section 4 and 5, respectively.

2 GWO algorithm

2.1 Brief descriptions

The GWO algorithm was proposed according to the searching and hunting behaviour of the grey wolves. Detailed studies show that the swarms of the grey wolves live with a strict social hierarchy. The strongest grey wolf leads the searching procedure and it always eats at first. It leads some stronger grey wolves in searching and hunting, and others such as the old and young grey wolves, eat at last. Obviously, they do not play a same role in life.

2.2 Mathematical modelling and the GWO algorithm

The original GWO algorithm imitates the searching and hunting behaviour of the grey wolves. Mathematically speaking, the best fitness value is called the alpha wolf, which is the leader of the swarms. The second-best fitness value is called the beta wolf, it ranks next to the alpha in the hierarchy of the pack, and the delta wolf ranks the third. All of other grey wolves are called omega.

For a given D-dimensional problem with the population size N, every individual of the swarms is initialised randomly in an equation as follows:

\[ X_i = (x_i^1, x_i^2, \ldots, x_i^D) \] (1)

where \( i = 1, 2, \ldots, N \). Supposing the current iteration time is \( t \) and positions of the individuals in the next iteration \( t+1 \) will be controlled by two kinds of randomness as follows:

\[ X_{i+1} = \frac{1}{2} \sum_{g=\text{alpha}}^{\text{delta}} (X_i^g - A \cdot |C \cdot X_i^g - X_i|) \] (2)

where \( g \) represents the alpha, beta, and the delta wolf. Then, the controlling parameters \( A \) and \( C \) are governed by:

\[ A = (2r_1 - 1)a \] (3)

\[ C = 2r_2 \] (4)

where \( r_1 \) and \( r_2 \) are two random numbers. \(|a|\) controls the behaviour of searching. If \(|a|>1\), the individuals would run away from the dominant wolves such as the alpha, beta, and the delta. It...
is called a global search. Consequently, if |A|<1, the individuals would approach the dominants and they make local searching.

The randomness involved in (3) and (4) classifies the whole pack of grey wolves in two parts, some of them are performing global search and some of them are in local searching. The number of grey wolves in global searching is expected to be larger at the beginning, and the number of them in local searching is expected to be larger when the searching procedure is approaching the end. This expectation is governed by the controlling parameter and defined as follows:

\[ a = 2(1 - t/M) \]  

(5)

where \( t \) is the cumulative iteration number and \( M \) is the maximum allowed iteration times.

### 3 Improved GWO algorithm

#### 3.1 PSO algorithm

The PSO algorithm is a kind of meta-heuristic algorithms [17]. The PSO algorithm is inspired from the biological swarms of birds. Every bird is considered as a particle and it was found to be flying around a central bird and according to its history trail. Considering the current speed (v), it is a randomness involved combination of the previous speed, the group centre and the history, the mathematical formula appeared as follows:

\[ v_{i+1} = w \cdot v_i + c_1(p_{gb} - x_i) + c_2(p_b - x_i) \]  

(6)

\[ x_{i+1} = x_i + v_{i+1} \]  

(7)

where \( t \) indicates the current iteration time, \( v \) represents the current speed of the swarm individual, \( p_{gb} \) is the best history result, and the \( p_b \) is the global best result of all, \( c_1 \), \( c_2 \) are two random numbers inside the domain [0, 1], and \( w \) is a controlling weight ratio.

#### 3.2 Inspirations

The above(6) and (7) could be rewritten to a single one. Then, in the PSO algorithm, the new positions of individuals as described correspondingly in (2) would appear as follows:

\[ X_{t+1} = X_t + w \cdot v_t + c_1r_3(p_{gb} - X_t) + c_2r_4(g_b - X_t) \]  

(8)

where \( v_t \) is the individual velocity parameter; \( p_{gb} \) is the individuals’ best positions in history; and \( g_b \) is the global best position. \( r_3 \) and \( r_4 \) are two other random numbers; \( c_1 \) and \( c_2 \) are two parameters; \( w \) is the weight balancing the history positions of the individuals.

Although the individuals of the swarms are considered to play a same role in optimisation, they update their new positions based on their history, their individual best and the global best positions.

We know the PSO algorithm has been widely applied in optimisations despite its lower capabilities compared to the GWO algorithm (reference paper [5] in 2014). Obviously, both the historical best position and the overall best candidate contribute to the fast convergence ratio and reduce the errors in the PSO algorithm. On the other hand, from the comparison results as described in reference paper [5], we can draw the conclusion that if the social hierarchy plays its role during the process, the searching procedure could be speed up wildly. That is why the GWO algorithm works better in convergence ratio and the minimum errors than the PSO algorithm. However, the GWO algorithm has not considered the best positions and the overall best candidates in its model. Consequently, the individuals’ current positions and their best positions in history should not be forgotten in improvements of the GWO algorithm.

#### 3.3 Improvements

Inspired by the PSO algorithm, we then rewrite (2) according to (8):

\[ X_{t+1} = b_1r_5X_t + b_2r_6(p_{gb} - X_t) \]

\[ + \frac{1}{\sum_{\alpha \in \text{alpha}} X_{t+1}^\alpha - A \cdot |X_{t+1}^\alpha - X_t|} \]  

(9)

where \( b_1 \) is the inertial weight, it controls the potential capability of the individuals’ current positions; \( b_2 \) is the individual memorising coefficient, it decides the role of the individuals’ history positions. \( r_5 \) and \( r_6 \) are two another random numbers. The improved algorithm can be applied as described in Table 1.

### 4 Simulated experiments

#### 4.1 Benchmark functions

There are 175 kinds of benchmark functions in literature [18]. In this study, we will carry out the simulation experiments based on 18 benchmark functions such as: Sphere (f1), Schwefel 2.22 (f2), Schwefel 1.2 (f3), Schwefel 2.21 (f4), Rosenbrock (f5), Quartic (f6), Summsquare (f7), Rastrigin (f8), Ackley (f9), Griewank (f10), Alpine (f11), Levi (f12), Cosine Mixture (f13), and Zakharov (f14) are unimodal benchmark functions; and f8–f14 are multimodal functions. All of them are going to be searched for the global optima for \( N = 500 \) iteration times.

#### 4.2 Empirical study of the parameters

Obviously, the improved GWO algorithm is under the same conditions with all of other meta-heuristic algorithms that the more population involves in the optimisation procedure, the best it performs along with the more complex the algorithm will be. Thus, there is a need to keep the population size be limited in numbers as fewer as possible according to the computing hardware.

Fig. 1 shows the relationship between the population size of the swarms and the best-fitted values after \( M = 500 \) iterations of optimisation. Results show that the population size must be larger than 20 in order to obtain a better optimisation procedure.

Two parameters are not sure at the current state in (9): the inertial weight \( b_1 \) and the individual memorising coefficient \( b_2 \). Apparently, there would be two suitable weights balancing the current positions, the individuals’ best history positions, and positions of the ranked top third wolves, but they might be different for different problems.

---

**Table 1** Pseudo-codes of the improved GWO algorithm

| State       | Description |
|-------------|-------------|
| initialising| all of the parameters |
| optimisation| for \( i \) from 1 to \( N \) do |
| &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n...
To find the best pairs of $b_1$ and $b_2$, an experiment was carried out with $M = 500$ searching iteration times, and the results are shown in Fig. 2.

From Fig. 2, we can see that the inertial weight $b_1$ should be much <1.0 which is the standard weight of the top third best positions. This means that the current positions should not influence the next iteration so much, this also could be seen from Fig. 3 as follows: the individual memorising coefficient $b_2$ has little bad influence where it is large or small. Detailed data show it does its best when it is near 0.5, in Fig. 4. Therefore, the best choice would be $b_1 = 0.05$ and $b_2 = 0.5$.

4.3 Comparisons between the improved GWO and other algorithms

In this experiment, we will carry out the comparisons between the improved GWO algorithm and the original GWO algorithm, together with the PSO algorithm, and the random walk improved GWO (RW-GWO) algorithm [13]. All of the four algorithms will be applied in solving the benchmark functions mentioned in Section 4.1.

One random selected experiment result was plotted as shown in Fig. 5.

In Fig. 5, simple comparisons are made according to the fitted values among the three algorithms. We can see from this figure that the improved GWO algorithm labelled ‘MGWO’ in Fig. 5 will work in better convergence, high accuracy, and better performance. More detailed work was done, as shown in Table 2. 100 Monte Carlo (MC) simulations were carried out applying the improved GWO algorithm, the GWO algorithm, and the PSO algorithm in solving the benchmark functions listed in Section 4.1. Results were statistically analysed and the mean best-fitted values and the standard derivations (st.Dev) were listed in Table 2. The succeeded simulation times was summed up and listed in the right columns of each algorithm for a reason that sometimes the algorithms could not approach the end.

We can then draw the conclusions from Table 2 that:

(i) The improved GWO algorithm solves the benchmark functions for the best at almost all times if there is a best one, except for the Levy function ($f_{12}$).

(ii) Sometimes the improved GWO and the original GWO algorithms could reach to the final best positions, see results in the table of Ratrigin function ($f_8$), Griewank function ($f_{10}$), and the Cosine Mixture function ($f_{13}$).

(iii) Although the improved GWO algorithm is inspired directly from the PSO algorithm, it is not capable to solve the Levy function ($f_{12}$) efficiently as well as the original GWO algorithm. In some cases, it performs even worse than the original GWO algorithm. The possible reason is that the Levy function ($f_{12}$) has similar outlines along the even dimensional directions (see from the outline of two-dimensional sketch [19]).

(iv) Although the algorithms are all capable of solving high-dimensional problems, the larger numbers the dimensions are, the more difficulty to meet the ideal results. Therefore, one more experiment would be further carried out to verify their capabilities of solving high-dimensional problems.

4.4 Capabilities verification on solving high-dimensional problems

For simplicity, we carry out an experiment to find the best-fitted values after $N = 500$ iteration times for Sphere function ($f_1$), the population size is varying from 20 to 1000 in maximum. Results are shown in Fig. 6.
In this paper, we propose an improved GWO algorithm inspired this study. However, the improved GWO algorithm proposed always performs the best.

5 Conclusions

In this paper, we propose an improved GWO algorithm inspired directly by the particle swarm optimisation (PSO) algorithm. Empirical study was carried out and found that the least numbers of swarms’ populations should be >20; and the best choice for the inertial weight \( b_1 \) and the individual memorising coefficient \( b_2 \) would be \( b_1 = 0.05 \) and \( b_2 = 0.5 \).

Experiments on solving the benchmark functions were carried out and the more capabilities of the improved GWO algorithm than that of the original GWO algorithm and that of the PSO algorithm were found. Efforts also show that the improved GWO algorithm was also capable for high-dimensional problems.
Acknowledgments
The authors would like to thank for the guiding of the Research Project of Hubei Provincial Department of Education with grant No. B2018241 and supports of the scientific foundation of the research centre for development of Chinese agricultural valley with grant No. 2018zgng07, 2018zgng08, and the Startup Foundation for Talents of Jingchu University of Technology (No. QDB201605).

References
[1] Mirjalili, S.: ‘The ant lion optimizer’, Adv. Eng. Softw., 2015, 83, pp. 80–98
[2] Holland, J.H.: ‘Adaptation in natural and artificial systems’ (University of Michigan Press, Ann Arbor, 1975), pp. 20–35
[3] Metropolis, N., Rosenblath, A.W., Rosenbluth, M.N., et al.: ‘Equation of state calculations by fast computing machines’, J. Chem. Phys., 1953, 21, (6), p. 1087
[4] Dorigo, M., Birattari, M.: ‘Ant colony optimization’, IEEE Comput. Intell. Mag., 2006, 1, (11), pp. 28–39
[5] Mirjalili, S., Mirjalili, S.M., Lewis, A.: ‘Grey wolf optimizer’, Adv. Eng. Softw., 2014, 69, pp. 46–61
[6] Sharma, P., Sundaram, S., Sharma, M., et al.: ‘Diagnosis of parkinson’s disease using modified grey wolf optimization’, Cogn. Syst. Res., 2019, 54, pp. 100–115
[7] Tripathi, A.K., Sharma, K., Bala, M.: ‘A novel clustering method using enhanced grey wolf optimizer and mapreduce’, Big Data Res., 2018, 14, pp. 93–100
[8] Martin, B., Marot, J., Bourennane, S.: ‘Mixed grey wolf optimizer for the joint denoising and unmixing of multispectral images’, Appl. Soft Comput., 2019, 74, pp. 385–410
[9] Tu, Q., Chen, X., Liu, X.: ‘Multi-strategy ensemble grey wolf optimizer and its application to feature selection’, Appl. Soft Comput., 2019, 76, pp. 16–30
[10] Singh, D., Dhillon, J.S.: ‘Ameliorated grey wolf optimization for economic load dispatch problem’, Energy, 2019, 169, pp. 398–419
[11] Qin, H., Fan, P., Tang, H., et al.: ‘An effective hybrid discrete grey wolf optimizer for the casting production scheduling problem with multi-objective and multi-constraint’, Comput. Ind. Eng., 2019, 128, pp. 458–476
[12] Khairuzzaman, A.K.M., Chaudhury, S.: ‘Multilevel thresholding using grey wolf optimizer for image segmentation’, Expert Syst. Appl., 2017, 86, pp. 64–76
[13] Gupta, S., Deep, K.: ‘A novel random walk grey wolf optimizer’, Swarm. Evol. Comput., 2019, 44, pp. 101–112
[14] Kohli, M., Arora, S.: ‘Chaotic grey wolf optimization algorithm for constrained optimization problems’, J. Comput. Des. Eng., 2018, 5, pp. 458–472
[15] Reddy, S., Panwar K. L.K., Panigrahi, B.K., et al.: ‘Binary grey wolf optimizer models for profit based unit commitment of price-taking GENCO in electricity market’, Swarm. Evol. Comput., 2019, 44, pp. 957–971
[16] Heidari, A.A.; Pahlavani, P.: ‘An efficient modified grey wolf optimizer with levy flight for optimization tasks’, Appl. Soft Comput., 2017, 60, pp. 115–134
[17] Kennedy, J., Eberhart, R.: ‘Particle swarm optimization’. Proc. of ICNN’95 –Int. Conf. on Neural Networks, 1995, vol. 1944, pp. 1942–1948
[18] Jamil, M., Yang, X.-S.: ‘A literature survey of benchmark functions for global optimization problems’, Int. J. Math. Model. Numer. Optim., 2013, 4, pp. 150–194
[19] Laguna, M., Marti, R.: ‘Experimental testing of advanced scatter search designs for global optimization of multimodal functions’. 2002. Available at http://www.av.es/mlarti/paper/docs/global1.pdf, 2003.6

Fig. 6 Best-fitted values with different dimensions for $f_1$