Grey Wolf Optimizer based on Nonlinear Adjustment Control Parameter

LONG Wen [1,2]

[1]Key Laboratory of Economics System Simulation, Guizhou University of Finance & Economics, Guiyang 550025, China

[2]School of Mathematics and Statistics, Guizhou University of Finance & Economics, Guiyang 550025, China

lw6822@163.com

Keywords: Grey wolf optimizer, Control parameter, Nonlinear, Function optimization

Abstract. Grey wolf optimizer (GWO) is a relatively novel stochastic optimization technique which has bee shown to be competitive to other methods. However, the control parameter a of GWO is decreased from 2 to 0 over the course of iterations. Inspired by particle swarm optimization (PSO), a novel nonlinear adjustment strategy of control parameter a is designed to enhance the performance of GWO algorithm. In addition, to enhance the global convergence of GWO algorithm, when generating the initial population, opposition-based learning strategy is employed. Simulation results show that the proposed algorithm is able to provide very competitive results compared to other algorithms.

Introduction

Grey wolf optimizer (GWO) is a population-based optimization technique developed by Mirjalili et al. [1], which mimics the social leadership hierarchy and hunting behavior of grey wolf in nature. GWO algorithm has few parameters and easy to implement, which make it superior than gravitational search algorithm (GSA), particle swarm optimization (PSO), and fast evolutionary programming [1]. As a result, GWO has caused much attention and has been used to deal with a number of practical optimization problems, such as optimal control of DC motor [2], optimal power flow [3], economic load dispatch problem [4], optimal reactive power dispatch problem [5], two-stage assembly flow shop scheduling problem [6], unit commitment problem [7], feature selection [8], and so on.

However, like other stochastic population-based algorithms, such as genetic algorithm (GA) and PSO, as the growth of the search space dimension, GWO also faces up to some problems. For instance, GWO algorithm is easily trapped in the local optimal value and provides a poor conver- gence behavior at exploitation. Therefore, researchers increasingly are paying close attention to the improvement of GWO for overcoming these disadvantages. Zhang and Zhou [9] present an extended GWO algorithm based on Powell local optimization method for global optimization and clustering analysis. Zhu et al. [10] presents a hybrid GWO (HGWO) algorithm with differential evolution (DE) to accelerate the convergence speed of GWO and improve its performance. Saremi et al. [11] propose the use of evolutionary population dynamics (EPD) in the grey wolf optimizer.

In the GWO algorithm, exploration and exploitation are guaranteed by the adaptive values of control parameter a. However, the control parameter a is linearly decreased from 2 to 0 over the course of iterations. Inspired by PSO, this paper designed a nonlinear adjustment strategy of control parameter a in the GWO algorithm. In addition, the opposition-based learning strategy is introduced to initialize the population. The experimental results show that the proposed algorithm not only has higher convergence speed but also can find out the optimal solution compared to the other algorithms.

Grey Wolf Optimizer Algorithm

In 2014, Mirjalili et al. [1] developed a novel population-based optimization technique, GWO, which mimics the social leadership and hunting behavior of grey wolves in nature. Similarly to other population-based algorithms, GWO initials the search process by a population of randomly generated candidate solutions. To formulate the social hierarchy of wolves when designing GWO, the current three best candidate solutions are called α, β, and δ respectively. The rest of the candidate solutions are named as ω and required to encircle α, β, and δ to find better solutions. The encircle process could be formulated as follows [1]:

$$X(t+1) = X_p(t) - A \cdot | C \cdot X_p(t) - X(t) | \tag{1}$$

Where t is the current iteration, $A = 2a \cdot r_1 - a$, $C = 2 \cdot r_2$, X_p is the position vector of the prey, X is the position vector of a grey wolf, r_1 and r_2 are random vectors in [0,1], respectively, a is linearly decreased from 2 to 0 over the course of iterations.

It should be noted that ω is required to update its position with respect to α, β, and δ simultaneously as follows [1]:

$$\begin{aligned} X_1(t) &= X_a(t) - A_1 \cdot | C_1 \cdot X_a(t) - X(t) | \\ X_2(t) &= X_b(t) - A_2 \cdot | C_2 \cdot X_b(t) - X(t) | \\ X_3(t) &= X_d(t) - A_3 \cdot | C_3 \cdot X_d(t) - X(t) | \end{aligned} \tag{2}$$

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

where X_α is the position of α, X_β is the position of β, X_δ is the position of δ, A_1, A_2, A_3 and C_1, C_2, C_3 are all random vectors.

Improved Grey Wolf Optimizer Algorithm

Initial Population by Opposition-based Learning Strategy

Population initialization is a crucial task in GWO because it can affect the quality of the final solution and the convergence speed [12]. If no information about the solution is available, then random initialization is the most commonly used method to generate candidate solutions (initial population), which often makes candidate solutions concentrated in a local area. According to [13], replacing the random initialization with the opposition-based learning population initialization can get better initial candidate solutions and then accelerate convergence speed.

Therefore, this papaer employs opposition-based learning strategy to generate initial population which can be used instead of a pure random initialization. The pseudo code of the opposition-based learning initialization is presented in **Algorithm 1**.

Algorithm 1 Opposition-based learning initialization

Set the population size N .
Random initialization $\{X(N)\}$
For $i = 1$ to N do
    For $j = 1$ to D do
        $x_{i,j} = x_{\min,j} + rand(0,1) \cdot (x_{\max,j} - x_{\min,j})$
    End for
End for
Opposition-based learning initialization $\{OX(N)\}$
For $i = 1$ to N do
    For $j = 1$ to D do
        $ox_{i,j} = x_{\min,j} + x_{\max,j} - x_{i,j}$
    End for
End for
Choose N best solutions from $X(N)$ and $OX(N)$ as initialization population.

Nonlinear Adjustment Strategy of Control Parameter

As we know, population-based stochastic optimization algorithms must have a good balance between exploration and exploitation. In the standard GWO algorithm, exploration and exploitation are guaranteed by the adaptive values of control parameter a. However, the values of control parameter a are linearly decreased from 2 to 0 over the course of iterations. To bring about a balance between the exploration and exploitation characteristics of GWO, inspired by PSO algorithm, we design a novel nonlinearly adjustment strategy of control parameter a as follows:

$$a(t) = a_{initial} \cdot \left(1 - l \cdot \left(\frac{t}{\max\_iter}\right)^2\right) \tag{4}$$

where t is the current number of iterations, $\max\_iter$ is the maximum number of iterations, $a_{initial}$ is the initial value of control parameter a, and l is the nonlinear modulation index. According to equation (4), the values of control parameter a are nonlinearly varying over the course of iterations.

The Proposed Algorithm

Based on the above explanation, the pseudo code of the proposed algorithm (denoted as IGWO) is demonstrated in **Algorithm 2**.

Algorithm 2 The proposed IGWO algorithm

```
Initialize the algorithm parameters: population size N, maximum number of iterations (max_iter), control parameter
a, A, and C
Set t = 0
Initialize the grey population X_i (i = 1,2,..., N) by opposition-based learning strategy
Calculate the fitness values of each individual
X_α = the best individual
X_β = the second best individual
X_δ = the third best individual
While (t < max_iter)
    For each individual
        Update the position of the current individual by equations (2) and (3)
    End for
    Update control parameter a by equation (4), then update parameter A and C
    Calculate the fitness vlaues of all individuals
    Update X_α, X_β, and X_δ
    t = t + 1
End while
Return X_α
```

Simulation Experiment and Comparison

In order to validate the performance of the proposed IGWO algorithm, we use 6 benchmark test functions [1] compared with the other population-based algorithms. These benchmark test functions are listed in Table 1 where *Name* indicates name of the function, *Function* is the equation of the function, $f_{\min}$ is the global optimum, and *Range* is the boundary of the functions search space.

Table 1. Benchmark test functions

Name	*Function*	*Range*	$f_{\min}$
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	[-100,100]	0
Schwefel's 2.22	$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,10]	0
Schwefel's 2.21	$f_3(x) = \max_i \{\lvert x_i \rvert, 1 \le x_i \le n\}$	[-100,100]	0
Rastrigin	$f_4(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2p\, x_i) + 10]$	[-5.12,5.12]	0
Ackley	$f_5(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2p\, x_i)\right) + 20 + e$	[-32,32]	0
Griewank	$f_6(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600,600]	0

The set of experiments tested on 6 unconstrained optimization functions are executed to compare the performance of IGWO algorithm with that of GWO algorithm and HGWO [10] algorithm. For a fair comparison among three algorithms, they are tested using the same setting of the parameters, that is, and the population size is set 30, the maximum number of iterations is set to 500, and the dimension is set to 30 for all test functions. All results reported are obtained based on 30 independent runs. We adopted the best, the mean, the worst, the standard deviation of fitness as the criterion of experimental validation. The statistical results are reported in Table 2. Meantime, for the sake of reliability, the results of HGWO algorithm reported in [10] are used in Table 2 directly. For clarity, the results of the best algorithms are marked in boldface.

Table 2. Experimental results comparison of IGWO, GWO and HGWO on 6 test functions

Function	Algorithm	Best values	Mean values	Worst values	St.dev
f_1	GWO	3.15E-029	3.58E-027	9.87E-027	2.93E-027
	HGWO	2.92E-034	1.12E-032	8.95E-032	2.32E-032
	IGWO	**2.21E-052**	**1.05E-050**	**2.35E-049**	**7.41E-050**
f_2	GWO	3.12E-017	1.10E-016	2.52E-016	5.53E-017
	HGWO	1.65E-020	9.33E-020	3.60E-019	6.92E-020
	IGWO	**8.50E-031**	**7.83E-030**	**3.49E-029**	**1.03E-029**
f_3	GWO	5.18E-008	7.17E-007	3.11E-006	8.71E-007
	HGWO	5.81E-009	4.17E-008	2.39E-007	4.56E-008
	IGWO	**3.98E-015**	**1.32E-013**	**7.16E-013**	**2.41E-013**
f_4	GWO	5.68E-014	4.7798	18.8698	6.3709
	HGWO	**0**	2.27E-001	4.7666	9.20E-001
	IGWO	**0**	**0**	**0**	**0**
f_5	GWO	7.90E-014	1.00E-013	1.15E-013	2.02E-014
	HGWO	3.64E-014	4.27E-014	5.06E-014	4.37E-015
	IGWO	**7.99E-015**	**1.15E-014**	**1.51E-014**	**2.99E-015**
f_6	GWO	**0**	2.21E-003	4.36E-002	6.99E-003
	HGWO	**0**	1.37E-003	3.12E-002	5.82E-002
	IGWO	**0**	**0**	**0**	**0**

As can be seen from Table 2, compare with GWO algorithm, IGWO can find better results on 5 test problems (f_1, f_2, f_3, f_4, and f_5). For function f_6, two algorithms obtained similar “best” values. However, IGWO found better “mean”, “worst”, and “st.dev” values. With respect to HGWO algorithm, IGWO is able to obtain better results on 4 test problems (f_1, f_2, f_3, and f_5). For test functions f_4 and f_6, HGWO and IGWO found similar “best” results. In contrast, IGWO found better “mean”, “worst”, and “st.dev” results. The above experimental results reveal that IGWO has the increasing advantage over the other compared algorithms for complex high-dimensional global optimization problems.

Figure 1 illustrates the convergence curves of fitness values with respect to the number of iterations for the 6 test functions with d=30. It can be observed from Figure 1 that the proposed IGWO algorithm is faster than GWO algorithm on all the test functions.

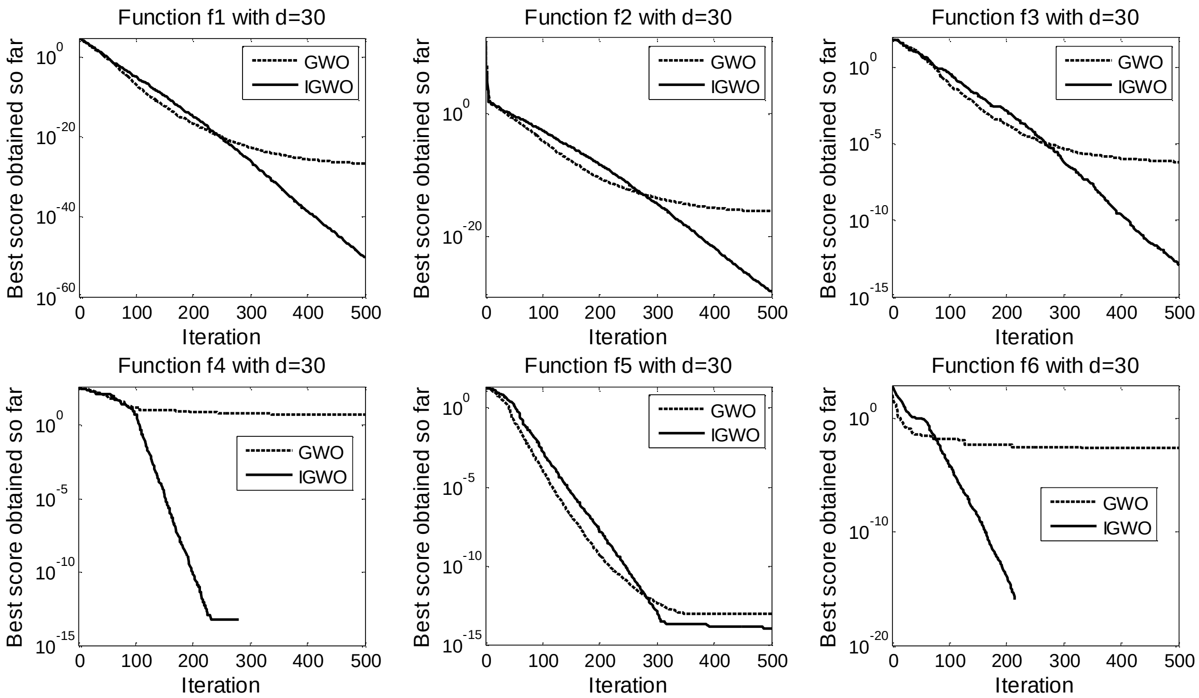

Figure 1 Convergence curve of IGWO and GWO for 6 test functions.

Conclusions

Grey wolf optimizer algorithm has been recently proposed as a novel population-based method inspired by the social leadership hierarchy and hunting behavior of grey wolves in nature, and it is has so far been successfully applied in a variety of fields. This paper proposed an improved GWO (IGWO) algorithm for global optimization problems. 6 benchmark test functions were employed in order to verify the performance of the proposed IGWO algorithm. The results show that the proposed IGWO was able to provide highly competitive results compared to standard GWO algorithm and HGWO algorithm.

Acknowledgements

This research was financially supported by the National Natural Science Foundation of China (61463009), by the Science and Technology Foundation of Guizhou Province ([2016]1022), and by the Key Program of Chinese Ministry of Commerce (2016SWBZD13).

References

[1] S. Mirjalili, S.M. Mirjalili and A Lewis: Adv. Eng. Softw. Forum Vol. 69 (2014), p. 46

[2] A. Madadi, M. M. Motlagh: Tech. J. Eng. Appl. Sci. 4 (2014), p. 373

[3] A. A. El-Fergany, H. M. Hasanien: Elec. Power Compon. Syst. 43 (2015), p. 1548

[4] V. K. Kamboj, S. K. Bath and J. S. Dhillon: Neural Comput. Applic. 27 (2015), p. 1

[5] M. H. Sulaiman, Z. Mustaffa, M. R. Mohamed and O. Aliman: Appl. Soft Comput. 32 (2015), p. 286

[6] G. M. Komaki, V. Kayvanfar: J. Comput. Sci. 8 (2015), p. 109

[7] V. K. Kamboj: Neural Comput. Applic. 27 (2016), p. 1643

[8] E. Emary, H. M. Zawbaa and A. E. Hassanien: Neuroomput. 172 (2016), p. 371

[9] S. Zhang, Y. Q, Zhou: Discr. Dyn. Nature Society. ID 481360 (2015), p. 1

[10] A. Zhu, C. Xu, Z. Li, J. Wu and Z. Liu: J. Syst. Eng. Elect. 26 (2015), p. 317

[11] S. Saremi, S. Z. Mirjalili and S. M. Mirjalili: Neural Comput. Applic. 26 (2015), p. 1257

[12] W. Gao, S. Liu and L. Huang: J. Comput. Appl. Math. 236 (2012), p. 2741

[13] S. Rahnamayan, H. R. Tizhoosh and M. M. A. Salama: IEEE Trans. Evol. Comput. 12 (2008), p. 64