An improved grey wolf optimization algorithm with multiple tunnels for updating

Juan ZHAO1* and Zheng-Ming GAO2
1 School of electronics and information engineering, Jingchu University of technology, Jingmen, Hubei, 448000, China
2 School of computer engineering, Jingchu University of technology, Jingmen, Hubei, 448000, China
*Corresponding author’s e-mail: ajuan323@jcut.edu.cn

Abstract. The grey wolf optimization (GWO) algorithm was proposed in 2014 and after several years of applications, it was used worldwide and all over the subjects which involved computation. Various improvements have been raised to increase the capability of optimization. Based on the best performance of the slime mould (SM) algorithm in optimization, a hybridization of the SM and GWO algorithms was proposed about the updating equations, and the GWO algorithm with multiple tunnels for individuals to update their positions during iterations was revised. Simulation experiments were carried out and comparisons were made between the GWO algorithm with variable weights and our proposed new one. Better performance were confirmed and reported as conclusions.

1. Introduction
The grey wolf optimization (GWO) algorithm, which was raised as the grey wolf optimizer[1] in 2014, proved to be capable of optimization and perform better than other algorithms raised before, such as the particle swarm optimization (PSO) algorithm, the genetic algorithm (GA). The GWO algorithm performed better in optimizing both benchmark functions and real world jobs, such as load frequency control[2], or economic dispatching[3]. Literally speaking, the GWO algorithm was the first nature inspired algorithm that introduced social hierarchy of swarms in nature. In either the PSO algorithm or others that was raised before, the updating equations would only consider the best candidates, or their historical trace sometimes. The structure of GWO algorithm opened the scientists’ mind for developing new algorithms.

Most recently, a new kind of nature inspired algorithm called the slime mould (SM) algorithm was proposed[4]. This algorithm was also inspired by nature, the behaviour and morphological changes of slime mould specifically. The characteristics that should be pointed out, is also the updating equations for individuals in swarms. For the SM algorithm, the individuals in each iteration would be separated into three tunnels and each of them had their own updating equations. The better performance in optimization might be supported by such operations although the program for the original SM algorithm would show series of warning or runtime errors because of the limitation of functions. Consequently, such operations could also be considered for the existed algorithms.

In this paper, we would hybrid the SM algorithm and GWO algorithms and focus mainly on the updating equations. The rest of the paper would be arranged as follows: in section 2, a brief description on the GWO and SM algorithms would be given. The hybridized improvements would be given in
section 3. And simulation experiments would be done with the benchmark functions in section 4. Discussions would be made and conclusions would be drawn in the last section.

2. A brief description of the algorithms

In this section, we would firstly describe the original algorithms and their mathematical relationship.

2.1. The GWO algorithm with variable weights

Theoretically speaking, although the original GWO algorithm introduced the social hierarchy of grey wolf swarms, the updating equations were not strictly following it. Therefore, we proposed an improvements called the GWO algorithm with variable weights\(^5\), and simulations experiments verified that the improvements would perform better with high accuracy, and fast convergence. The updating equations for individuals in swarms would be formulated as follows:

\[ X_i(t + 1) = w_1 X_i + w_2 X_2 + w_3 X_3 \quad (1) \]

Where, \(X_i(t + 1)\) is the position for the \(i\)-th individual in the next iteration \(t\). \(X_1, X_2\) and \(X_3\) are middle variables of the positions of alpha, beta, and delta grey wolf and the current positions of the same individual:

\[ X_1 = X^f - A_1 [C_1 \cdot X^f - X_1], \quad X_2 = X^f - A_2 [C_2 \cdot X^f - X_1], \quad X_3 = X^f - A_3 [C_3 \cdot X^f - X_1] \quad (2) \]

\(A_1, A_2, A_3, C_1, C_2, C_3\) are two middle variables relevant to the random numbers:

\[ A_i = 2 \alpha r_i - \alpha, \quad C_i = 2 r_i \quad (3) \]

\(\alpha\) is the controlling parameters and \(r_i\) is the random number. The controlling parameter \(\alpha\) is very important and simulation experiments proved that a maximum value of \(\alpha\) would be 1.7 would be a best choice\(^5\), that is to say \(\alpha_m = 1.7\). It would decline linearly from the maximum value to zero and relevant to the maximum allowed iteration times \(\text{maxIter}\), and it would be computed as follows:

\[ \alpha = \alpha_m \left( 1 - \frac{t}{\text{maxIter}} \right) \quad (4) \]

\(w_1, w_2, w_3\) are variable weights and they are calculated as follows:

\[ w_1 = \cos \theta, \quad w_2 = \frac{1}{2} \sin \theta \cdot \cos \phi, \quad w_1 + w_2 + w_3 = 1 \quad (5) \]

Where \(\theta\) and \(\phi\) are two parameters relevant to the current iteration \(t\):

\[ \phi = \frac{1}{2} \arctan(t), \quad \theta = \frac{2}{\pi} \arccos \frac{1}{3} \cdot \arctan(t) \quad (6) \]

2.2. The SM algorithm

The SM algorithm was proposed most recently and the updating equations is:

\[ x_i(t + 1) = \begin{cases} r_1 (UB - LB) + LB & r_2 < z \\ x_b(t) + v_b \cdot \left[ W \cdot x_d(t) - x_b(t) \right] & r_3 < p \\ v_c x_i(t) & r_3 \geq p \end{cases} \quad (7) \]

Where \(r_1\) is a random number in Gauss distribution and parts of the individuals in swarms would be reinitialized all over the definitional domain \([LB, UB]\) again and again during each iterations with a proportional lower value \(z=0.03\). \(x_d(t)\) and \(x_b(t)\) are positions of two random selected candidates from the swarms in each iterations with weights \(W\). \(x_i(t)\) is the position of the best candidates found so far in this iteration. And \(v_b, v_c\) are two other random numbers in uniform distributions. Some of the individuals would approach to the best candidates with a compensation of the weighted distance between two random selected individuals. The proportional number \(p\) is an important control parameter for the SM algorithm, it would balance the exploration and exploitation of individuals. The rest of the candidates would just directly follow their historical exploring procedure.

3. The improved GWO algorithm with multiple tunnels

With a glance at the updating equations for the GWO and SM algorithms, we could easily find that all of the individuals in the GWO algorithm would be guided by their superiors and anyone who get the top three best positions would be rescheduled for alpha, beta, and delta wolves. However, individuals in the
SM algorithm would be classified into three types, some of them would be reinitialized from the beginning, parts of them would stick to their own steps, and some of them would carry on the exploration based on the best candidates and distance between two random selected candidates. Although the social hierarchy of slime mould is not considered, two representatives were also involved even they are random selected. These two algorithms vary from the construction to the performance. However, the ways for updating the positions of individuals in the SM algorithm could also be introduced to the individuals in the GWO algorithm. Holding this principle in mind, we hereby re-formulate the updating equation for individuals in the GWO algorithm as follows:

\[
x_i(t + 1) = \begin{cases} 
  r_1(UB - LB) + LB & r_2 < z \\
  w_1 X_1 + w_2 X_2 + w_3 X_3 & r_3 < p \\
  r_4 \cdot x_i(t) & r_3 \geq p
\end{cases}
\]  

Just for simplicity, only random numbers in Gauss distribution are considered, for example, \(r_1, r_2, r_3\) and \(r_4\). And the controlling parameter \(p\) would just be declined linearly from 1 to 0:

\[
p = 1 - \frac{t}{\text{maxIter}}
\]  

The new improved GWO algorithm with multiple tunnels for updating would be quite the same to the original one, seen Table 1.

**Table 1. Pseudo code of the improved GWO algorithm proposed**

| Description | Pseudo code |
|-------------|-------------|
| Problems description | The dimension of the given problems  
The limitations of the given problems  
The Population size  
The controlling parameter  
The stop criterion (maximum iteration times or admissible errors) |
| Initialization | The positions and scores of \(a, \beta\) and \(\delta\)  
the initial positions of other grey wolves |
| Searching | While not the stop-criterion  
calculate the new fitness function  
update the positions  
limit the scope of positions  
refresh \(a, \beta\) and \(\delta\)  
calculate the weights  
update the positions  
update the stop criterion |
| Results | The position of \(a\) wolf and its score |

4. Simulation experiments

Every algorithm must be tested and verified their capabilities literally. The benchmark functions would be used traditionally and the performance of algorithms could be evaluated, or compared. The benchmark functions are a series of functions formulated by scientists and engineers to describe the problems we met during our procedure in exploring, exploiting or conquering nature. Based on the dimensionality, complexity, or scalability and other characteristics, the benchmark functions could be classified into several types \[6\]. Generally, those who have only one global or local optima would be easily to optimize and find the best positions, this kind of benchmark functions are called the unimodal one. On the contrary, if there were lots of local optima with fixed numbers of global optima, then the individuals would be trapped in local optima, this kind of benchmark functions were called the multimodal one. The multimodal benchmark functions would be more difficult to optimize than the unimodal benchmark functions. However, we still found another types of benchmark functions that were also difficult to optimize. If there were basins or valleys in the profiles of benchmark functions and the
global optima are just located in the basins or valleys, this kind of benchmark functions are also quite
difficult to optimize because the individuals could gain rare information, even nothing towards to the
global optima. Therefore, we would carry on three kinds of simulation experiments, on the unimodal,
multimodal benchmark functions and those who have basins or valleys in their profiles.

Furthermore, we also found that the outputs would vary because of the randomness involved. To
reduce the influence of randomness, we would carry on 100 times and their results would be averaged.
With averages over 100 Monte Carlo simulations, the final outputs would be more smooth and steadier.

4.1. Simulations experiments on unimodal benchmark functions

Normally the unimodal benchmark functions are easy to optimize, they have only one global or local
optima and here we discard those who have basins or valleys, and focusing on such kind of benchmark
functions having smooth and round profiles. In this experiment, Schwefel 2.20 function would be
involved:

\[ f(x) = \sum_{i=1}^{d} |x_i| \]  

(10)

Schwefel 2.20 function is unique because of the absolute function involved, its three-dimensional
profile is shown in Figure 1, the profiles might be separated into four slope plates. Nevertheless, it was
also easy to optimize, the best fitness values along with iterations were shown in Figure 2.

Figure 1 Profile of Schwefel 2.20 function(d=2)  Figure 2 best fitness values versus iterations

Obviously, the unimodal benchmark functions would be quite easy to optimize, and the averaged
results over 100 Monte Carlo simulations showed a clear and smooth declination.

4.2. Simulations experiments on multimodal benchmark functions

Being easily trapped in local optima, most of the algorithms would be a little difficult to optimize the
multimodal benchmark functions. A very famous multimodal benchmark functions called Salomon
function is introduced here:

\[ f(x) = 1 - \cos \left( 2\pi \sqrt[3]{\sum_{i=1}^{d} x_i^2} \right) + 0.1 \sqrt[3]{\sum_{i=1}^{d} x_i^2} \]  

(11)

There are lots of peaks called local optima in Salomon profiles, seen Figure 3. However, the GWO
algorithm with variable weights, and the improved GWO with multiple tunnels for updating were all
capable of optimization, see Figure 4.

The results over 100 Monte Carlo simulations are also smooth and steady, and the improved GWO
algorithm proposed in this paper also performed better.
4.3. Simulations experiments on benchmark functions with basins
When the individuals are coming towards to the global optima, they would derive barely nothing because all of the results were so similar. For example, Brown function:

$$f(x) = \sum_{i=1}^{d-1} \left( x_i^2 \right)^{(i+1)} + \left( x_i^2 \right)^{(i+1)}$$

\[(12)\]

Seen from Figure 5, we can clearly find a smooth basin or plate in its three dimensional profile. However, two algorithms involved in this paper were all capable of optimization, and Figure 6 showed the results.

5. Discussions and conclusions
In this paper, we proposed the improved GWO algorithm with multiple tunnels for updating inspired by the SM algorithm. Researches showed that the SM algorithm performed quite better jobs than any other algorithms emerged before. The structure of the updating equation is unique and such ideas would also carry on to the improvements of the GWO algorithm.

Simulation experiments were carried out and three kinds of benchmark functions are tested. The final results over 100 Monte Carlo experiments showed that the improved algorithm would perform much better in such circumstances, compared all of the three kinds of simulation experiments.
better than the original one, which was also an improvement for the standard GWO algorithm. We could get the information that either of them is capable of optimization, regardless of the unimodal, multimodal, or having basins in their profiles.

Results showed that all of the fitness values would be declined almost linearly from the beginning. However, the results of our proposed algorithm would decline more and more faster after detailed comparison.

We can hereby conclude that: 1) The GWO algorithm could be used very easily and it is quite capable of optimization. 2) Discard the defect that the GWO algorithm could only solve the problems which the global optima are located at the Origin\(^7\), the proposed improvement would allocate more individuals to insist on their trajectories and consequently, if the global optima are the Origin, the individuals would step towards to them faster than the original way. 3) Increasing the population size in swarms, and increasing the number of choice, the proposed GWO algorithm with multiple tunnels for updating result in better performance than the compared GWO algorithm with variable weights.

**Acknowledgments**

The authors would like to thank the supports of the following projects: the second batch of scientific research team of Jingchu University of technology with grant number TD202001, the general Excellent Students Work Funding Project of Hubei Provincial Colleges with grant number 2019XGJPB3013, the key research and development project of Jingmen with grant number 2019YFZD009 and 2020YFYB033, Hubei Provincial Natural Science Foundation with grant number 2019CFB661, the research project of Hubei Provincial Department of Education with grant number B2019213, the cultivatable science foundations of Jingchu university of technology with grant number PY201903 and PY202003.

**References**

[1] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis. Grey Wolf Optimizer[J]. Advances in Engineering Software, 2014, 69 (0): 46-61. HTTP://DX.DOI.ORG/10.1016/J.ADVENGSOFT.2013.12.007

[2] Dipayan Guha, Provas Kumar Roy, Subrata Banerjee. Load frequency control of interconnected power system using grey wolf optimization[J]. Swarm and Evolutionary Computation. HTTP://DX.DOI.ORG/10.1016/J.SWEVO.2015.10.004

[3] T. Jayabarathi, T. Raghunathan, B. R. Adarsh, et al. Economic dispatch using hybrid grey wolf optimizer[J]. Energy, 2016, 111: 630-641. HTTPS://DOI.ORG/10.1016/J.ENERGY.2016.05.105

[4] Shimin Li, Huiling Chen, Mingjiing Wang, et al. Slime mould algorithm: A new method for stochastic optimization[J]. Future Generation Computer Systems, 2020, 111: 300-323. HTTPS://DOI.ORG/10.1016/J.FUTURE.2020.03.055

[5] Zheng-Ming Gao, Juan Zhao. An Improved Grey Wolf Optimization Algorithm with Variable Weights[J]. Computational Intelligence and Neuroscience, 2019, 2019: 2981282. 10.1155/2019/2981282

[6] Momin Jamil, Xin-She Yang. A literature survey of benchmark functions for global optimization problems[J]. Int. Journal of Mathematical Modelling and Numerical Optimisation, 2013, 4 (2): 150-194.

[7] Peifeng Niu, Songpeng Niu, Nan liu, et al. The defect of the Grey Wolf optimization algorithm and its verification method[J]. Knowledge-Based Systems, 2019, 171: 37-43. HTTPS://DOI.ORG/10.1016/J.KNOSYS.2019.01.018