The Improved Equilibrium Optimization Algorithm with Averaged Candidates

Zhengming Gao¹*, Juan Zhao² and Xuejun Tian¹
¹School of computer engineering, Jingchu university of technology, Jingmen, China,448000
²School of electronics and information engineering, Jingchu university of technology, Jingmen, China,448000
*gaozming@jcut.edu.cn

Abstract. In this paper, we proposed the improvement of the newly raised equilibrium optimization (EO) algorithm by hybridizing the grey wolf optimization (GWO) algorithm. Simulation experiments were carried out and results showed that the hybrid EO algorithm with averaged candidates would perform better than the original one. Considering the better results and the smooth convergence curve on different types of benchmark functions, we recommend to withdraw the concept of equilibrium pool in the construction of the EO algorithm henceforth, and keep the guiding equation relevant to the averaged concentration of four best candidates instead.

1. Introduction

We are facing more and more detailed and complex scientific problems now. Most of the newly raised problems are mathematically described by many parameters and with high dimensionality. Analytical solutions are now not accessible, and henceforth, we need some simulation methods or optimization algorithms to understand the problem fully and find the suitable answers globally. Traditional deterministic algorithms are found difficult to be used and most of them are also easily trapped in local optima [1]. Fortunately, along with the better understanding of nature, we also developed a new series of optimization algorithms called the nature-inspired algorithms.

The nature-inspired algorithms are now classified into four types: evolutionary, physics-based, swarm-based and human-based [2]. The swarm based algorithms introduced the randomness and the inherent behaviour of swarms for hunting, breeding, cooperating and so on into the searching procedure for the best solutions. The individuals would be spread all over the domain with randomness and all of them would carry out the exploration and exploitation along with iterations. In most cases, their positions would be guided by the best one and their own current positions, for instance, the particle swarm optimization (PSO) algorithm [3]. In 2014, the grey wolf optimization (GWO) algorithm [4] was proposed and for the first time, the social hierarchy was also included in construction. The best three candidates were all used to guide and update the positions. Another swarm-based or physics-based algorithm was raised in 2019, we called it the equilibrium optimization (EO) algorithm [5], it introduced four best candidates besides their averaged one and constructs an equilibrium pool. During the iterations, the positions/concentrations of individuals were guided by a random selected representative from the equilibrium pool. It should be mentioned that sometimes the best candidate, or the second best candidate, or the third best candidate would not play the role during the updating.
In this paper, we proposed an improvement for the EO algorithm hybridizing the GWO algorithm and the averaged values of the best candidates would be used guiding and updating the positions/concentrations of individuals.

The rest of this paper is arranged as follows. In section 2, the EO and GWO algorithms would be briefly introduced and simulation experiments would be carried out in section 3. Discussions and conclusions would be made in section 4.

2. The EO Algorithm and its Improvement Hybridizing the GWO Algorithm

Inspiring by the remarkable performance of the GWO algorithm and the solution to the simple well-mixed dynamic mass balance on a control volume, the EO algorithm was raised with a given unit volume $V$, the concentrations of each individual $C$ would be guided and updated with the following equation:

$$C = C_{eq} + (C - C_{eq}) \cdot F + \frac{G}{\lambda V} (1 - F)$$  \hspace{1cm} (1)

Where $C_{eq}$ represents the concentration at an equilibrium state. $F$ and $G$ are the key controlling parameters and $\lambda$ is the random number with the interval of 0 and 1.

2.1. Brief Introduction of the EO Algorithm

The concentrations/positions of each individual would be updated with equation (1) and the key controlling parameters are the exponential parameter $F$ and the generation rate parameter $G$.

2.1.1. The exponential parameter.

The exponential parameter was relevant to the iteration numbers with some constant control numbers:

$$F = a_1 \text{sign}(r - 0.5)(e^{-\lambda t} - 1)$$  \hspace{1cm} (2)

Where $a_1 = 2$ is a constant value and it controls the exploration ability. $t$ is the function of the current iteration number and the maximum allowed iteration times $\text{maxIter}$

$$t = \left(1 - \frac{it}{\text{maxIter}}\right)^{(a_2/\text{maxIter})}$$  \hspace{1cm} (3)

Where $a_2 = 2$ is another constant value and it controls the exploitation ability.

2.1.2. The generation parameter.

The generation parameter was used to control the exploitation rate.

$$G = GCP \cdot (C_{eq} - \lambda C) \cdot F$$  \hspace{1cm} (4)

$$GCP = \begin{cases} 0.5 r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases}$$  \hspace{1cm} (5)

Where $r_1$ and $r_2$ are two random numbers with the interval of 0 and 1. $GP=0.5$ is a constant value balancing the exploration and exploitation ratio.

2.1.3. The equilibrium pool.

The concentration at the equilibrium state $C_{eq}$ is a random selected representative from the equilibrium pool:

$$C_{\text{pool}} = \{C_{eq(0)}, C_{eq(1)}, C_{eq(2)}, C_{eq(3)}, C_{\text{ave}}\}$$  \hspace{1cm} (6)

For a given problem $f(x)$, the fitness value of the candidates in the equilibrium pool is restricted as follows:

$$f(C_{eq(0)}) \leq f(C_{eq(1)}) \leq f(C_{eq(2)}) \leq f(C_{eq(3)}) \leq f(C_{ave})$$  \hspace{1cm} (7)

The averaged candidates $C_{\text{ave}}$ is calculated as follows:
\[ C_{\text{ave}} = \frac{C_{eq(0)} + C_{eq(1)} + C_{eq(2)} + C_{eq(3)}}{4} \]  

(8)

2.2. The Improvement Hybridizing the GWO Algorithm

In the EO algorithm, the concentrations/positions of each individual is updated with equation (1), and the concentration at the equilibrium state is selected randomly from the equilibrium pool constructed by equation (6), (7), and (8). However, we know that the best candidates would always the nearest to the target, their significance should be noticed and involved. For the GWO algorithm, the positions of the individuals are updated by the average positions of the alpha, beta, and delta wolf:

\[ X_{i}^{t+1} = \frac{X_{1}^{t} + X_{2}^{t} + X_{3}^{t}}{3} \]  

(9)

Where \( it \) represents the current iteration, \( X_{1}, X_{2}, \text{and} X_{3} \) are the function of the positions of alpha, beta, and delta wolf. \( X_{i}^{t+1} \) represents the position of individuals in the next iterations. The better performance shows that the equal balancing the positions of the best three wolves could be an outstanding choice. And consequently, we would improve the original EO algorithm hybridizing the concepts of equal balance of GWO algorithm. The pseudo code of the improved EO algorithm was listed in Table 1.

**Table 1.** Pseudo code for the improved EO algorithm with averaged candidates

| Phase            | Description                                                                 |
|------------------|-----------------------------------------------------------------------------|
| Initializing     | Setup the population size \( n \)                                          |
|                  | Setup the dimension \( d \)                                                |
|                  | Setup the values of controlling parameters \( a_1 = 2, a_2 = 2, gp = 0.5, maxIter \) |
|                  | Spread the individuals randomly all over the domain                         |
|                  | While \( it < \text{maxIter} \)                                            |
|                  | For \( i = 1:n \)                                                          |
|                  | Calculate the fitness values                                               |
|                  | If \( f(C_i) < f(C_{eq(0)}) \): Replace \( C_{eq(0)} \), \( f(C_{eq(0)}) \) with \( C_i \) and its fitness value |
|                  | Else if \( f(C_{eq(1)}) \) \leq f(C_i) \leq f(C_{eq(1)}) Replace \( C_{eq(1)} \), \( f(C_{eq(1)}) \) with \( C_i \) and its fitness value |
|                  | Else if \( f(C_{eq(2)}) \) \leq f(C_i) \leq f(C_{eq(2)}) Replace \( C_{eq(2)} \), \( f(C_{eq(2)}) \) with \( C_i \) and its fitness value |
|                  | Else if \( f(C_{eq(3)}) \) \leq f(C_i) \leq f(C_{eq(3)}) Replace \( C_{eq(3)} \), \( f(C_{eq(3)}) \) with \( C_i \) and its fitness value |
|                  | Calculate the average \( C_{ave} \)                                       |
|                  | Accomplish memory saving(if \( it > 1 \))                                  |
|                  | For \( i = 1:n \)                                                          |
|                  | Generate random numbers                                                    |
|                  | Calculate F and G                                                          |
|                  | Updating the concentrations                                                |
| Exploring and exploiting |                                                                              |
| Results          | \( C_{eq(0)} \) and its fitness value \( f(C_{eq(0)}) \) are the final best concentration and value. |

3. Simulation Experiments

Representatives of three types of benchmark functions would be used to verify the capability of the improved EO algorithm with averaged candidates (labelled ‘eoa_average’). Comparisons would be
made between the improved algorithm and the original EO algorithm (labelled ‘eoa’). 100 Monte Carlo simulation experiments would be carried out in order to avoid the influence of randomness.

### 3.1. Comparison Experiments on the Unimodal Benchmark Functions

In this experiment, we would optimize Schwefel 2.20 function:

\[
y = \sum_{i=1}^{d} |x_i| \tag{10}
\]

This is a unimodal, scalable function. Its profile is not smooth but regular (seen from Figure 1). The averaged results over 100 Monte Carlos experiments are shown in Figure 2. Obviously, the improved EO algorithm with averaged candidates perform better than the original one.

Furthermore, the convergence curves show that random selection from the equilibrium pool would not find further better values after some tens of iterations, the residual errors for Schwefel 2.20 function under the optimization by the original EO algorithm would not converge at all. However, when the averaged best candidates take the place, the residual errors would directly go far near the global optimum.

![Figure 1. Profile of Schwefel 2.20 function](image)

![Figure 2. Optimization procedure for Schwefel 2.20 function](image)

### 3.2. Comparison Experiments on the Multimodal Benchmark Functions

In this experiment, we would use the improved EO algorithm with averaged candidates to optimize the famous Salomon function:

\[
y = 1 - \cos \left( 2\pi \sqrt{\sum_{i=1}^{d} x_i^2} \right) + \frac{1}{10} \sqrt{\sum_{i=1}^{d} x_i^2} \tag{11}
\]

Salomon function has lots of local optima and peaks (see Figure 3), the individuals would be easily trapped in local optima. With the capability of exploring and exploiting, both the EO algorithm and improved EO algorithm with averaged candidates would optimize them, we can draw such conclusion from Figure 4. However, the improved algorithm would perform much better, both in the residual errors and the smooth curve towards the global optimum.
3.3. Comparison Experiments on the Bottom Flat Like Benchmark Functions

Traditionally, the benchmark functions are classified into unimodal and multimodal types considering the peaks’ number of their profiles. However, we would come across some other situations, for instance, in optimization of Schwefel 2.20 function, we found the final residual errors reach to $10^{-12}$ or so in quantum, the values were far away from the final values for other smooth function such as Sphere function. Schwefel 2.20 function is composed by some plates while Sphere function is quite smooth as a part of a sphere. Therefore, we would carry on further experiments on such kind of benchmark functions who have the basin or flat-like bottom profile near the global optimum. Two famous benchmark functions would be introduced such as Levy function and Trid function:

**Levy function:**

$$y = \sin^2(\pi w_0) + \sum_{i=1}^{d-1} (w_i - 1)^2[1 + 10 \sin^2(\pi w_i + 1)] + (w_d - 1)^2 \cdot (1 + \sin^2(2\pi w_d)) \quad (12)$$

Where $w_i = 1 + (x_i - 1)/4$.

**Trid function:**

$$y = \sum_{i=1}^{d} (x_i - 1)^2 - \sum_{i=2}^{d} x_i x_{i-1} + \frac{d(d + 4)(d - 1)}{6} \quad (13)$$

Trid function is unimodal but the profile has a basin near the global optimum, see Figure 5. On the other hand, Levy function is multimodal and the profile has a plate tunnel near the global optimum (see Figure 6).

![Figure 3. Profile of Arckley function](image1)

![Figure 4. Optimization procedure for Salomon function](image2)

![Figure 5. Profile of Levy function](image3)

![Figure 6. Profile of Trid function](image4)
These kinds of benchmark functions are difficult to be optimized. The averaged results over 100 Monte Carlo experiments are shown in Figure 7 and Figure 8. Obviously, the final values are not satisfactory. Both of them might fail in optimizing Levy and Trid function. However, basically speaking, the improved EO algorithm with averaged candidates perform a little better than the original one.

4. Discussions and Conclusions
Considering the large absence ratio for the best candidates in the original EO algorithm, we improved the original EO algorithm by hybridizing the GWO algorithm. Positions/concentrations of the best four candidates were averaged and the positions/concentrations would be updated under the guiding of the averaged best candidates. Simulation experiments were carried out on benchmark functions, and 100 Monte Carlo simulations were averaged to reduce the influence of randomness. Results on most of the unimodal or multimodal benchmark functions are promising and the less values and more smooth convergence curve demonstrated that the averaged candidates could improve the capabilities indeed. However, when both of the improved and original algorithms were applied in optimizing representatives of the bottom flat-like benchmark functions such as Levy and Trid functions, they would not perform well. Both of them failed to do so, although the improved EO algorithm still perform better and the original one.

We could draw the conclusions from this study that: a) Although the improved EO algorithm with averaged candidates and the original one performed good in optimizing benchmark functions, sometimes they would fail if the profile is not smooth or has flat-like bottom around the global optimum. Further efforts should be made to find the next better one. b) The equilibrium pool is recommended to be removed during applications, they would slow the convergence rate and sometimes would result in failure. c) The guiding equation for the EO algorithm might be changed, the concentration/position at the equilibrium state might be replaced by the averaged best candidates, and results on most of the smooth unimodal or multimodal benchmark functions would confirm their capabilities.

5. Acknowledgments
The authors would like to thank the support of the following projects:
1) The second batch of scientific research teams of Jingchu university of technology with grant number TD202001.
2) Hubei Provincial Natural Science Foundation with grant number 2019CFB661;
3) The key research and development project of Jingmen with grant number 2019YFZD009;
4) The research project of Hubei Provincial Department of Education with grant number B2019213;
5) The general Excellent Students Work Funding Project of Hubei Provincial Colleges with grant number 2019XGJPB3013;
6) The cultivatable science foundations of Jingchu university of technology with grant number PY201903;

6. References
[1] Zheng-Ming Gao, Juan Zhao. An Improved Grey Wolf Optimization Algorithm with Variable Weights Computational Intelligence and Neuroscience[J]. Computational Intelligence and Neuroscience, 2019, 2019 (Article ID 2981282): 13.
[2] Seyedali Mirjalili, Andrew Lewis. The Whale Optimization Algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.
[3] J. Kennedy, R. Eberhart. Particle swarm optimization: Proceedings of ICNN'95 - International Conference on Neural Networks, 27 Nov.-1 Dec. 1995, 1995[C]. 4: 1942-1948 vol.4.
[4] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis. Grey Wolf Optimizer[J]. Advances in Engineering Software, 2014, 69 (0): 46-61.
[5] Afshin Faramarzi, Mohammad Heidarinejad, Brent Stephens, et al. Equilibrium optimizer: A novel optimization algorithm[J]. Knowledge-Based Systems, 2020, 191: 105190.