The Improved Equilibrium Optimization Algorithm with Best Candidates

Juan Zhao¹ and Zhengming Gao²

¹School of electronics and information engineering, Jingchu university of technology, Jingmen, China, 448000
²School of computer engineering; Jingchu university of technology, Jingmen, China, 448000
ajuan323@jcut.edu.cn

Abstract. The best candidates play the important role during the exploration and exploitation of individuals in almost all of the swarm-based algorithms. More best candidates were involved in such procedure of the grey wolf optimization algorithm and the newly raised equilibrium optimization (EO) algorithm we called here. The EO algorithm introduced four best candidates besides their average and constructs an equilibrium pool. However, the best candidates would still perform the guiding role and a novel improvement was introduced. Experiments on some classical benchmark functions were carried out and results show the better performance than the original one. Consequently, the EO algorithm should be improved and focused more on the best candidates furthermore.

1. Introduction

Nowadays, we believed that we were better understanding the nature and therefore, some of the physical, mathematical, even statistical rules are included in solving the problems we faced during our exploring, exploiting and conquering procedures of nature. The nature-inspired algorithms have been popular for almost all of the scientist and engineers in various of research fields. And lots of algorithms for optimization have been raised along with their better performance. Literally, the nature-inspired algorithms could be classified in four types: the physics-based, swarm-based, human-based and evolutionary algorithms [1]. The swarm-based algorithms would involve a swarm of individuals in a random procedure of exploring or exploiting the solution. Consequently, the best candidates would be the final results. Henceforth, most of the swarm-based algorithms were paid attention to how to find the better way updating the positions of the candidates and resulting a fast convergence.

However, most of the swarms have their own social hierarchy and the second, the third, even the fourth best candidates would also play an important role. In 2014, the grey wolf optimization (GWO) algorithm [2] was raised, and for the first time, the best candidates were not only on our eyes. Three of the best candidates were involved in the exploration and exploitation procedure. Although all of the candidates were treated the same in the original GWO algorithm, improvements could be made with the inherent social hierarchy [3]. The better performance of GWO algorithms were noticed and many of the improvements, applications were made hereafter. In 2019, another algorithm we called here the equilibrium optimization algorithm (EOA) was raised and four of the candidates besides another averaged one were introduced [4]. The EOA were experimented and results showed that it would perform better than the rest of swarm-based algorithms. It seems that the best candidates should not be only one. Considering the rest of swarm-based algorithms, it might be true. In this paper, we would
eliminate the equilibrium pool and update the positions/concentrations of the individuals in the swarm with only the best candidates.

The rest of the paper would be arranged as follows: in section 2 we would briefly introduce the EOA and the inspiration. We would carry out some experiments in section 3. Discussions and conclusions would be made in section 4.

2. The EOA and its Improvements with Best Candidates

The EOA was inspired by the solution to the simple well-mixed dynamic mass balance on a control volume. In this algorithm, the positions of individuals were renamed as concentrations, and their guiding equations for each iteration would be as follows:

\[ C = C_{eq} + (C - C_{eq}) \cdot F + \frac{\gamma}{V} (1 - F) \] (1)

Where \( C \) represents the concentration of individuals, \( C_{eq} \) represents the concentration at an equilibrium state. \( F, G \) and \( \lambda \) are all the controlling parameters. \( V \) is the control volume and here \( V=1 \).

2.1. Brief Introduction of the EOA

The guiding equation for updating the positions/concentrations of individuals might be unique seen from equation (1). However, the real magic is the construction of the controlling parameters and the equilibrium pool.

2.1.1. The exponential parameter. The exponential parameter was formulated with the iteration function and the randomness.

\[ F = a_1 \text{sign}(r - 0.5)(e^{-\lambda t} - 1) \] (2)

Where \( a_1 \) is a constant value and it controls the exploration ability. \( a_1 = 2 \). \( t \) is the function of iterations and the maximum allowed iteration times \( \text{maxIter} \):

\[ t = (1 - \frac{it}{\text{maxIter}})^{(a_2 \text{maxIter})} \] (3)

Where \( a_2 \) is another constant value and it controls the exploitation ability. \( a_2 = 2 \).

2.1.2. The generation parameter. The generation parameter represents the generation rate. It was introduced to improving the probability of exploitation.

\[ G = G_0 \cdot F \] (4)

\[ G_0 = GCP \cdot (C_{eq} - \lambda C) \] (5)

\[ GCP = \begin{cases} 0.5r_1 & r_2 \geq GP \\ 0 & r_2 < GP \end{cases} \] (6)

Where \( r_1 \) and \( r_2 \) are two random numbers with the interval of 0 and 1. \( GP \) is another constant value controlling the exploration and exploitation, \( GP=0.5 \) is the balanced choice.

2.1.3. The equilibrium pool. Inspired by the better performance of GWO algorithms, four best candidates were involved in the EOA besides their averaged one. For a given optimization problem \( f(x) \), the fitness values of the candidates would be satisfied with the following equation:

\[ f(C_{eq(0)}) \leq f(C_{eq(1)}) \leq f(C_{eq(2)}) \leq f(C_{eq(3)}) \] (7)

The equilibrium pool includes four of the candidates and ordered with equation (7). It also includes the averaged one \( C_{ave} \):

\[ C_{ave} = \frac{C_{eq(0)} + C_{eq(1)} + C_{eq(2)} + C_{eq(3)}}{4} \] (8)
2.2. The Improvement with Best Candidates

The EOA introduced four best candidates besides their averaged one to construct the equilibrium pool. It randomly selects one representative from the pool during the exploration and exploitation with equation (1). The four best candidates and their averaged one would be equally chosen to update the positions/concentrations of individuals in each iteration. However, the representatives from the equilibrium pool would be \(1/5\) times to be the best one, \(1/5\) for the second best and \(3/5\) for the rest. These mean that the best candidate can only play its role for twenty percent ratio. And most of the times, the worse best candidates take part in the guiding. Obviously, it would result in a lower convergence rate and might be not acceptable. Therefore, the equilibrium pool could be abandoned and the best candidate \(C_b\) could be directly used to guide the updating. And then, the guiding equation (1) could be transformed to be:

\[
C = C_b + (C - C_b) \cdot F + \frac{G}{\lambda V} (1 - F) \tag{9}
\]

3. Simulation Experiments

In this section, we would carry out simulation experiments following the classical operations as usual. Trying to eliminate the influence of the randomness in the algorithm, we would carry out 100 Monte Carlo simulations and the results would be averaged. Comparisons would be made between the original EOA (labelled ‘eoa’) and the improvements (labelled ‘eoa_best’).

Three types of benchmark functions would be introduced in these experiments. Some representatives would be chosen for simplicity.

3.1. Comparison Experiments on the Unimodal Benchmark Functions

The unimodal benchmark functions are those who have only one local or global optimum. Causing the unimodal benchmark functions only have one peaks, the individuals would not fall into local optima, therefore, these types of functions are easy to be optimized. We hereby involve two representatives of the unimodal benchmark functions, Sum square function and Sum of different powers function. Their formulation and attributes are listed in Table 1.

| Name                                | Formula                           | Domain[\(lb, ub\)] | Scalable |
|-------------------------------------|-----------------------------------|--------------------|----------|
| Sum square function                 | \(y = \sum_{i=1}^{d} [(i + 1)x_i^2]\) | [-10,10]           | Yes      |
| Sum of different powers function    | \(y = \sum_{i=1}^{d} |x_i|^{i+1}\)                      | [-10,10]           | Yes      |

Considering the simplicity of the unimodal benchmark functions and the 100 percent ratio involved for the best candidates and the 20 percent ratio involved, the improved EOA would be expected to be converge faster and the results should be better. Simulation experiments were carried out and 100 Monte Carlo averaged results were shown in Figure 1 and Figure 2. Obviously, the improved EOA with best candidates works very better than the original one. And furthermore, the improved EOA would result in a smoother convergence curve than that of the original one.
Figure 1. Optimization procedure for Sum square function

Figure 2. Optimization procedure for Sum of different powers function

3.2. Comparison Experiments on the Multimodal Benchmark Functions

Results on the unimodal benchmark functions are really exciting, but most of the benchmark functions are multimodal, they would have many peaks or local optima with one global optimum. Consequently, the individuals would fall into the local optima and be trapped. Therefore, the multimodal benchmark functions are difficult to optimized but they are the reasons and the targets. Similarly, we also take two representatives of the benchmark functions into the experiments: Ackley function and Rastrigin function. We could see from Figure 3 and Figure 4 that they are all complicated and have many local peaks. Their attributes were listed in Table 2.

Table 2. Representatives for multimodal benchmark functions

| Name           | Formula                                                                 | Domain[lb,ub] | Scalable |
|----------------|------------------------------------------------------------------------|--------------|----------|
| Ackley function| \[y = 20 - 20e^{-0.02 \sum_{i=1}^{d} x_i^2} - \sum_{i=1}^{d} e^{0.06 \cos(2\pi x_i)} + e^{1}\] | [-10,10]     | Yes      |
| Rastrigin function | \[y = 10d + \sum_{i=1}^{d} \left[ x_i^2 - 10\cos(2\pi x_i) \right]\] | [-5.12,5.12] | Yes      |

Although the multimodal benchmark functions are very complicated. However, we are also confident that the improved EOA would perform better job based on the experience that EOA could
optimize such problems. The final averaged results over 100 Monte Carlo simulation experiments are shown in Figure 5 and Figure 6. We can also draw the conclusion that the improved EOA with best candidates perform better than the original one. And the convergent curve was also smoother than that of the original EOA causing the continuous updating with the best candidates.

![Figure 5. Optimization procedure for Arckley function](image1)

![Figure 6. Optimization procedure for Rastrigin function](image2)

3.3. Comparison Experiments on the Bottom Flat Like Benchmark Functions

Whether unimodal or multimodal, the benchmark functions with flat like bottoms are all difficult to be optimized. The bottom flat like benchmark functions we used here refer to those benchmark functions who have basins around the global optimum. When the individuals are exploring on the basins, they barely gain information about the directions towards the global optimum. And therefore, the bottom flat-like benchmark functions are really challenging for the optimization algorithms. We here introduce two representatives of them: Zakharrov function and Rosenbrock function. Both of them have one flat-like bottom around the global optimum, seen from Figure 7 and Figure 8. Their attribute are listed in Table 3.

![Figure 7. Profile of Zakharrov function](image3)

![Figure 8. Profile of Rosenbrock function](image4)
Table 3. Representatives for bottom flat-like benchmark functions

| Name           | Formula                                                                 | Domain [lb, ub] | Scalable |
|----------------|-------------------------------------------------------------------------|-----------------|----------|
| Zakharrov function | $y = \sum_{i=1}^{d} x_i^2 + \left( \sum_{i=1}^{d} 0.5(i + 1)x_i \right)^2$ | [-5, 10]       | Yes      |
| Rosenbrock function | $y = \sum_{i=1}^{d} \left[ (1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2 \right]$ | [-100, 100]    | Yes      |

The final averaged results over 100 Monte Carlo simulations are shown in Figure 9 and Figure 10. Obviously, Zakharrov function has a quarter of the basin and it could be optimized directly and smoothly. However, Rosenbrock function has a flat-like tunnel and the final results are very bad. However, the improved EOA with best candidates would perform better than the original EOA even for Rosenbrock function.

4. Discussions and Conclusions
From the above three types of experiments, we can see that: a) Either the improved EOA or original one could solve all of the problems, representative of bottom flat-like Rosenbrock function failed to be optimized, although the improved EOA with best candidates performed a little better. b) The mechanism that randomly selecting one representative from the equilibrium pool for the original EOA would result in the fluctuation of the convergence curve along with iterations. Fixing the representative to be the best candidates could avoid such situation and result in a smoother curve and less residual errors. c) It was really not a best choice to construct the equilibrium pool, the best candidates would be a best choice for such exploration and exploitation mechanism.

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

6. References
[1] Seyedali Mirjalili, Andrew Lewis. The Whale Optimization Algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.
[2] Seyedali Mirjalili, Seyed Mohammad Mirjalili, Andrew Lewis. Grey Wolf Optimizer[J]. Advances in Engineering Software, 2014, 69 (0): 46-61.
[3] 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): 1-13.
[4] Afshin Faramarzi, Mohammad Heidarinejad, Brent Stephens, et al. Equilibrium optimizer: A novel optimization algorithm[J]. Knowledge-Based Systems, 2020, 191: 105190.