The negative mayfly optimization algorithm

Juan ZHAO¹, Zheng-Ming GAO²

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

Abstract. The global best or historical best positions were involved in updating positions of individuals in swarms for almost all of the swarm-based nature-inspired algorithms with exceptions for the averages. However, literal research to the particle swarm optimization (PSO) algorithm had proved that the all of the candidates would also leave the global best and the historical best candidates and such improvement would result in a better performance. Similarly, the negative mayfly optimization (MO) algorithm was proposed based on such conditions. Simulation experiments were carried out and verified that the negative MO algorithm could perform better than the original MO algorithm, especially in optimizing the multimodal benchmark functions.

1. Introduction
At the beginning of nature-inspired algorithms, the scientists and engineers would focus mainly on the behaviour of exploration, exploitation of individuals. The average would be calculated and introduced to guide the overall individuals in swarms of the ant colony optimization (ACO) algorithm [1], the bat algorithm [2], for instance. Better understanding of survivor of the fittest, we introduced the best candidate even some of the top best candidates to guide the updating. For example, in the grey wolf optimization (GWO) [3] algorithm, the individuals would be sorted and the top three grey wolves named the alpha, beta, and delta wolf, would be used to guide the overall swarms in exploration and exploitation. Another example would be the latest equilibrium optimization (EO) algorithm [4], four best candidates including their average construct an equilibrium pool, a random selected candidate from the pool would be used to guide the individuals in updating positions. However, efforts had been made and results showed that if there is only the average involved in updating, the EO algorithm would perform much better [5]. In addition to the averages and the global best candidates, the historical trajectories might also helpful. For the individuals in the particle swarm optimization (PSO) algorithm [6], the global best candidate and the historical trajectories were all take part in the updating. Simulation experiments have proved that the PSO algorithm would perform better, steadier with higher accuracy.

Literal research also proved that the worst candidate could also be involved in updating [7]. In this paper, we introduced this idea to improve the mayfly optimization (MO) algorithm [8], which was just proposed in this year 2020 and proved to be capable in optimization.

The following sections would be arranged as follows: in Section 2, we would describe the MO algorithm and the improved negative MO (NMO) algorithm. Simulation experiments would be carried out in Section 3. Discussions would be made and conclusions would be drawn in Section 4.
2. The MO and NMO algorithms

In this section, we would briefly talked about the mayfly algorithm and the inspiration for the improvement.

Mayfly is a kind of insects. The baby mayflies would live in water for several years and when they got mature, they would decorticate themselves and fly in air. However, the mature mayflies would only survive in air for one to seven days. Consequently, the mature mayflies would be in a rush for reproduction. Observing the male and female mayflies rushing for mates, the MO algorithm was proposed. Developed from the PSO algorithm, the mayflies would also update their positions $p_i(t)$ with velocities $v_i(t)$ for the $i$-th individual in the current iteration $t$:

$$ p_i(t + 1) = p_i(t) + v_i(t) \quad (1) $$

Where, $p_i(t + 1)$ would be the position for the $i$-th individual in the next iteration. Considering the different behaviours of male and female mayflies, their velocities would be updated in different ways.

2.1. Movements of female mayflies

The female mayflies have the duty to reproduce, therefore, they are all in a hurry for mates. Their velocities would be updated according to their mates. That is to say, the velocity of the $i$-th female mayfly would be guided to update according to its fitness value $f[y_i(t)]$ and its mate $f[x_i(t)]$:

$$ v_i(t) = \begin{cases} 
  g \cdot v_i(t) + a_1 e^{-\beta m_f} [x_i(t) - y_i(t)] & f[y_i(t)] > f[x_i(t)] \\
  g \cdot v_i(t) + f \cdot r_1 & f[y_i(t)] \leq f[x_i(t)] 
\end{cases} \quad (2) $$

Where, $g$ and $f$ are weights which would be declined from their maximum to minimum value. $\alpha_i$ and $\beta$ are constants. $r_1$ is the random number in uniform distribution with the interval of -1 and 1. $r_m$ represents the Cartesian distance between the female and male couple, it would be calculated as follows:

$$ \|x_i - y_i\| = \sum_{k=1}^{n} (x_{ik} - y_{jk})^2 \quad (3) $$

2.2. Movements of male mayflies

Male mayflies would always be strong and fly with their own willing. The male mayflies would carry on their own exploration and exploitation in the whole definitional domain $[lb, ub]$ and update their best position $x_g$ and the historical best trajectories $x_{h_i}$. They also have two updating ways with a comparison of their own best fitness values $f[x_i(t)]$ to the historical best trajectories $f[x_{h_{i}}(t)]$:

$$ v_i(t) = \begin{cases} 
  g \cdot v_i(t) + a_2 e^{-\beta h_f} [x_{h_{i}}(t) - x_i(t)] + a_3 e^{-\beta f} [x_g - x_i(t)] & f[x_i(t)] > f[x_{h_{i}}(t)] \\
  g \cdot v_i(t) + d \cdot r_2 & f[x_i(t)] \leq f[x_{h_{i}}(t)] 
\end{cases} \quad (4) $$

Where, $a_2$ and $a_3$ are two other constants. $r_p$ and $r_f$ represent the Cartesian distance between the current individual and the historical best trajectory, the global best candidate respectively. $d$ represents the dance ratio around the current position and $r_2$ is another random number in uniform distribution with interval of -1 and 1.

2.3. Mating and mutating

After the velocities and positions were updated, all of the individuals would be sorted again and the top half best would be renamed as the male mayflies and another half the female mayflies. And then, they would get married and give birth to offspring with fixed number 2. The positions for the two offspring would be calculated as follows:

$$ \text{offspring1} = L \cdot \text{male} + (1 - L) \cdot \text{female} \quad (5) $$

$$ \text{offspring2} = L \cdot \text{female} + (1 - L) \cdot \text{male} \quad (6) $$

A fixed number of offspring in swarms would be also mutated. The mutated individuals and the offspring would further be sorted and half of the top best individuals would be selected as the male mayflies, and another best half would be named as female mayflies.
After some rounds of iterations, the global best candidates might come to the global optima and thus find the solutions.

2.4. The negative improved MO algorithm
All of the individuals might approach to the historical best trajectories and the global best candidates. This is the reasonable thought for the structure of updating equations. However, the updating equations could also be interpreted with an opposite meaning: all of the individuals would run away from their own worst historical trajectories and the global worst positions. According to this law, we could reprogram the codes, and the updating equations would be the same as in equations (2) and (4). The optimized procedure for this proposed negative improve MO algorithm could be seen from its pseudo-code as shown in Table 1.

| Stages   | Descriptions                                          |
|----------|-------------------------------------------------------|
| Initializing | Setup the populations of male, female, and mutants, $n_{male}, n_{female}, n_{mutant}$  |
|           | Setup the dimension of the problems $d$               |
|           | Setup the stopping criterion maximum allowed iteration number: $maxIter$ |
|           | Initialize the control variables $f_l, d, g$          |
|           | Initialize the positions of mayflies                 |
|           | Assign the historical best candidates                |
|           | Find the global best candidates at the beginning      |
| Searching | For it from 1 to $maxIter$:                           |
|           | Update the female mayflies                           |
|           | Update the male mayflies with the worst candidates and the historical worst trajectories |
|           | Reselect the male and female mayflies                |
|           | Give birth to offspring                              |
|           | Choose $n_{mutant}$ offspring to mutate               |
|           | Regroup the offspring and mutated individuals together|
|           | Reselect the male and female mayflies                |
|           | Update the control parameters                        |
| Results   | The global best candidates                           |

We can see from Table 1 that the NMO algorithm only varied from the MO algorithm with selection of the best or worst candidates.

3. Simulation experiments
In this section, we would carry on several kinds of experiments to verify the NMO performance and make comparisons with the original MO algorithm.

Benchmark functions would be introduced to be optimized to find their best solutions. According to the literat researches, most of the benchmark functions would be different from various kinds of characteristics such as the modality, separability, scalability, or dimensionality\cite{9}. Moreover, the symmetry would play an import role because most of the algorithms would be failed to find solutions for non-symmetric benchmark functions\cite{10}.

Considering the fluctuation of results for each simulation experiments, we would introduce the Monte Carlo method to reduce the influence of randomness. The results for each experiments would be averaged to reduce the fluctuation.

3.1. Simulation experiments on unimodal benchmark functions
Most of the unimodal benchmark functions were easy to optimize, we would introduce Schwefel 2.21 function to this kind of simulations:

$$f(x) = \sum_{i=1}^{d} x_i^2$$ (7)
Schwefel 2.21 function is unimodal and its global optimum is located at the Origin, as shown in Figure 1. The optimized procedure and the residual errors would be shown in Figure 2. Obviously, the negative improved version labelled ‘nmoa’ worked worse than the original version as labelled ‘moa’ in the figures.

Griewank function:

\[ f(x) = \sum_{i=1}^{d} \frac{x_i^2}{4000} - \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1 \]  

(8)

Griewank function is highly multimodal, as shown in Figure 3. However, results as shown in Figure 4 showed that the NMO would perform better than the MO algorithm.

3.3. Simulation experiments on non-symmetric multimodal benchmark functions
Deflected Corrugated Spring function is a continuous, differentiable, separable, non-scalable, and multimodal function:
\[ f(x) = 1 + \frac{1}{10} \sum_{i=1}^{d} (x_i - 5)^2 - \cos \left[ \sqrt{\sum_{i=1}^{d} (x_i - 5)^2} \right] \] (9)

Deflected Corrugated Spring function is also highly multimodal and it is also non-symmetric, as shown in Figure 5. The global optimum for Deflected Corrugated Spring function locates at point \( x^* = (5, 5) \), and \( f(x^*) = 0 \). The Monte Carlo simulation experiments’ results were shown in Figure 6. Obviously, the NMO would work still work better than the MO algorithm.

4. Conclusion

In this paper, we proposed a negative mayfly optimization algorithm. In this algorithm, the male mayflies would update their velocities according to the worst candidates together with their worst trajectories. Unlike the normal positive interpretation, in the NMO algorithm, the male mayflies would run away from their worst trajectories and the global worst candidates.

Simulation experiments showed that both the NMO and MO algorithms would work well in optimizing both the unimodal or multimodal benchmark functions, even for the non-symmetric one. However, for the unimodal benchmark functions, the MO would work better than the NMO. Because of the simple profiles, all of the individuals would perform well and thus, the worst candidates would follow the steps of the best candidate. And consequently, the NMO algorithm would update their positions in a lower ratio than that of the MO algorithm.

While for the multimodal benchmark functions, the best candidates or the best historical trajectories would sometimes fall into the local optima. These time the NMO algorithm would allow the individuals run away from the local optima and consequently, the NMO algorithm would decrease the probability being trapped in local optima. Therefore, the NMO would work better than the MO algorithm.

We can hereby draw the conclusions that the NMO algorithm would work better for the multimodal benchmark functions, either symmetric or non-symmetric. However, this improvement would fail in optimizing the unimodal benchmark functions.

Acknowledgments

The authors would like to thank the supports for the following projects:

[1] The second batch of scientific research team of Jingchu University of technology with grant number TD202001.
[2] The general Excellent Students Work Funding Project of Hubei Provincial Colleges with grant number 2019XGJPB3013.

[3] The key research and development project of Jingmen with grant number 2019YFZD009 and 2020YFYB033.

[4] Hubei Provincial Natural Science Foundation with grant number 2019CFB661.

[5] The research project of Hubei Provincial Department of Education with grant number B2019213.

[6] The cultivatable science foundations of Jingchu University of technology with grant number PY201903 and PY202003.

References

[1] M. Dorigo, M. Birattari. Ant colony optimization[J]. Computational Intelligence Magazine, IEEE, 2006.11, 1 (4): 28-39.

[2] Juan Zhao, Zheng Ming Gao. The bat algorithm and its parameters: The 4th International Conference on Electronic, Communications and Networks (CECNet2014) Boca Raton: CRC Press, [2014], 2015[C]. Boca Raton: CRC Press, [2014], IV: 1323-1326. 10.1201/B18592-237

[3] 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

[4] Afshin Faramarzi, Mohammad Heidarinejad, Brent Stephens, et al. Equilibrium optimizer: A novel optimization algorithm[J]. Knowledge-Based Systems, 2019: 105190. HTTPS://DOI.ORG/10.1016/J.KNOSYS.2019.105190

[5] Zhengming Gao, Juan Zhao, Xuejun Tian. The Improved Equilibrium Optimization Algorithm with Averaged Candidates[J]. Journal of Physics: Conference Series, 2020, 1575: 012105. 10.1088/1742-6596/1575/1/012105

[6] 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. 10.1109/ICNN.1995.488968

[7] Yang Chunming, D. Simon. A new particle swarm optimization technique: 18th International Conference on Systems Engineering (ICSEng'05), 16-18 Aug. 2005, 2005[C]. 164-169. 10.1109/ICSENG.2005.9

[8] Konstantinos Zervoudakis, Stelios Tsafarakis. A mayfly optimization algorithm[J]. Computers & Industrial Engineering, 2020, 145: 106559. HTTPS://DOI.ORG/10.1016/J.CIE.2020.106559

[9] 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.

[10] 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