The improved slime mould algorithm with Levy flight

Juan ZHAO¹, *, Zheng-Ming GAO²,a, Wu Sun²,b

¹School of electronics and information engineering, Jingchu university of technology, Jingmen, Hubei, China
²School of computer engineering, Jingchu university of technology, Jingmen, Hubei, China
*Corresponding author: ajuan323@163.com
agaozming@jcuc.edu.cn
b213148193@qq.com

Abstract—In this paper, we proposed the Levy flights to replace the random numbers whether it was in Gauss distribution or uniform distribution in the standard slime mould algorithm (SMA). Three kinds of classical benchmark functions such as the unimodal, multimodal benchmark functions and those who have basins which we could see clearly from their three dimensional profiles. Comparisons were made through the average results over 100 Monte Carlo simulation experiments in order to eliminate the influence of randomness. Results showed that the improved SMA with Levy flight might not always perform better. The improved SMA replacing the uniform distributed parameter achieved better performance whereas the improved SMA replacing the Gauss distributed parameter gained nothing in performance, even worse sometimes than the standard SMA.

1. INTRODUCTION
It is very important to formulate the updating equation for individuals during the exploration and construction of new swarm based nature inspired algorithms. The updating equations might be the characteristics for almost all of the swarm based nature inspired algorithms because each of them would have a different one. Traditionally, the best candidates would be always involved in such as the bat algorithm[1], the whale optimization algorithm[2], the historical best positions might also play a role such as in the particle swarm optimization algorithm[3]. Taking consideration of the leading packs in the swarm, more than one best candidates might be also introduced to guide the individuals during the iterations. It was proved and verified in the grey wolf optimization (GWO) algorithm[4], where three best candidates called the alpha wolf, beta wolf, and delta wolf, guided the rest of warms called delta wolves in their exploration and exploitation. Another newly proposed algorithm called the equilibrium optimization algorithm[5] took four best candidates and their averaged one in guiding the updating of positions or concentrations in each iterations.

Besides the choice of various of candidates to construct the new algorithms, many new methods are also introduced to improve the capability of the existed algorithms. Literally speaking, the binary version might be very useful for the real digital problems, such as the binary equilibrium optimization algorithm[6]. Other improvements such as the variable weights in GWO algorithm[4], the chaotic improvements[7], the random walk improvements[8] and so on, are also promising.
In this paper, we introduced Levy flight to a new-born swarm based nature inspired algorithm called the slime mould algorithm (SMA) which was just raised by a chinese student in most recent years.

2. THE ORIGINAL SMA

In the SMA, the updating equation for i-th individual is randomly selected from three parts.

(1) The random selected positions in the searching scope \([LB, UB]\) with proportional number \(z\).

(2) The composed positions of the global best candidate \(x_b\), the random selected candidates \(x_A(t)\) and \(x_B(t)\) in the current \(t\) iteration with proportions \(p\).

(3) The current position with proportion \(1-p\):

\[
 x_i(t+1) = \begin{cases} 
 r_1 \cdot (UB - LB) + LB & r_2 < z \\
 x_b + v_b \cdot [W \cdot x_i(t) - x_B(t)] & r_3 < p \\
 v_c \cdot x_i(t) & p \leq r_3 \leq 1 
\end{cases}
\]

2.1 The randomness

Two kinds of random numbers with Gauss and uniform distributions are involved in the randomness in equation (1). \(r_1, r_2\) and \(r_3\) are three random numbers with Gauss distribution in the interval of 0 and 1. They are generated to control the selection and they are used to balance the exploration, exploitation for individuals in swarms. \(v_b\) and \(v_c\), however, are two random numbers in uniform distribution, they are generated in uniform distribution falling two symmetric domain \([-a, a]\) and \([-b, b]\), where \(a, b\) are relevant to the current iteration and the maximum allowed iteration times \(maxIter\):

\[
 a = \text{atanh} \left(1 - \frac{t}{maxIter}\right) \quad \text{(2)}
\]

\[
 b = 1 - \frac{t}{maxIter} \quad \text{(3)}
\]

2.2 The threshold

There are two threshold in equation (1). The first threshold \(z\) is introduced to increase local exploration when the individuals found less information approaching to the target food. This threshold parameter might be the characteristics of the SMA, it represents a situation that if the individuals gain less information for the global optima, the algorithm would choose some of the individuals to abandon the current exploration and initially carry on the exploration from the beginning. Considering the good capability of swarm based nature inspired algorithms, the values for \(z\) is quite small, \(z=0.03\) as simulation experiments verified.

Another threshold parameter is \(p\), which is introduced to guide the individuals to carry on exploration or exploitation in a given iteration. This value is critically important because it would balance the exploration and exploitation ratio during the iterations. Moreover, the right value could reduce the rate for individuals trapped in local optima. The equation for this parameter is formulated as follows:

\[
 p = \frac{\text{tanh}|S_i - DF|}{|DF|} \quad \text{(4)}
\]

Where \(S_i\) represents the fitness value for \(x_i\) and \(DF\) is the global best fitness value for all candidates found until the current iteration.

2.3 The weights matrix

The weights \(W\) are in matrix for all individuals under every iteration. It would be relevant to the sort in goodness for all candidates by means of the fitness values:

\[
 si = \text{sort}(S) \quad \text{(5)}
\]

And then, the weights matrix would be calculated as follows:

\[
 W_{si(i)} = \begin{cases} 
 1 + r_4 \cdot \log \left(1 + \frac{b_F - S_i}{b_F - w_F}\right) & \text{condition} \\
 1 - r_4 \cdot \log \left(1 + \frac{b_F - S_i}{b_F - w_F}\right) & \text{others} 
\end{cases}
\]

\(r_4\) is another random number in the interval of 0 and 1. \(b_F\) and \(w_F\) are the best and worst fitness values in the current iteration. For conditions, the SMA would choose half of the individuals in the current iteration to carry on with weights larger than 1 and half of them with weights less than 1.
3. THE IMPROVED SMA WITH LEVY FLIGHTS

We have just already mentioned about the random numbers involved in the updating equation of the SMA algorithm. Three random parameters are in Gauss distribution and two random parameters are in uniform distribution.

The random numbers $r_1$ in Gauss distribution is used to spread the individuals all through the domain. $r_2$ and $r_3$ are used to select a path to calculate the values for individuals randomly.

The random numbers in uniform distribution $v_{b}$ and $v_{c}$ are used to weight the compositions in the current path. The uniform distribution means the equal weights for every values inside the scope. The equal importance would lead the individuals to carry on exploration or exploitation in steady. However, experiments had proved and verified that if the exploration and exploitation would carry on some larger steps between several small steps, the individuals in swarms would find the best optima more faster and avoid being trapped in local optima[9]. This kind of job could be under control of Levy flight, one kind of popular random walk formulated as follows:

\[ S = 0.01 \cdot \frac{u}{|v|^\beta} \]  \hspace{1cm} (7)

Where $u$ and $v$ are global mean value and the standard derivation in Gauss distribution with $u = 0.0$ and

\[ \sigma_u = \left[ \frac{\Gamma(1 + \beta) \cdot \sin \left( \frac{\pi \beta}{2} \right)}{\Gamma \left( 1 + \frac{\beta}{2} \right) \cdot \beta \cdot 2^{\frac{\beta + 1}{2}}} \right]^{\frac{1}{\beta}} \]  \hspace{1cm} (8)

Where $\Gamma$ represents the standard gamma function and $\beta = 1.5$ as defaulted values. Figure 1 shows the cumulative position changing regarding 1000 iterations at all times. Obviously, the random steps with levy flights would be more complicated. The levy flight would carry on some long steps between several small steps. In such circumstances, the selected individuals would occasionally carrying on some large exploration after several small explorations.

![Figure 1 Levy flights steps sketch](image)

In this paper, we would propose the replacements for $r_1$ and $v_c$ parameters in Gauss and uniform distributions respectively by the Levy flight, the guide equation would then be changed as follows:

\[
x_{i}(t+1) = \begin{cases} 
 Levy(D) \cdot (UB - LB) + LB & r_2 < z \\
 x_{b} + v_{b} \cdot \left[ W \cdot x_{c}(t) - x_{b}(t) \right] & r_3 < p \\
 v_{c} \cdot x_{c}(t) & p \leq r_3 \leq 1 
\end{cases} \hspace{1cm} (9)
\]

\[
x_{i}(t+1) = \begin{cases} 
 r_1 \cdot (UB - LB) + LB & r_2 < z \\
 x_{b} + v_{b} \cdot \left[ W \cdot x_{c}(t) - x_{b}(t) \right] & r_3 < p \\
 Levy(D) \cdot x_{c}(t) & p \leq r_3 \leq 1
\end{cases} \hspace{1cm} (10)
\]
4. SIMULATION EXPERIMENTS
In this section, we would carry on some simulation experiments on the improved SMA with Levy flights with some classical benchmark functions. To verify the performance of the improved algorithm, we made comparisons with the standard SMA. In addition, in order to avoid the influence of randomness in both the improved and standard SMA, we would carry on 100 Monte Carlo simulation experiments and the final result would be computed as their average.

The benchmark functions are a series of functions derived from the nature or proposed to evaluate the performance or capabilities of optimization algorithms. Some characteristics would influence the performance ultimately, such as the modality, dimensionality, scalability, valleys or basins and so on[10]. Therefore, three types of benchmark functions would be involved in this paper.

4.1 Experiments on unimodal benchmark functions
The unimodal benchmark functions are almost all easily to optimize because they are all have one global optimum. The best global optimum would be easily found for even the scalable functions with high dimensionality. In this simulation experiments, we would introduce a benchmark function with fixed dimensionality: the Bartels Conn function:

\[
f(x) = |x_1^2 + x_2^2 + x_1x_2| + |\sin(x_1)| + |\cos(x_2)| - 1
\]  

Bartels Conn function is a symmetric, non-differentiable, non-separable, non-scalable unimodal benchmark functions with fixed two dimensionality. The domain for every parameter is not limited and the global optimum is located at \(x^* = (0, 0)\) and it is \(f(x^*) = 0.0\).

The final results averaged over 100 Monte Carlo simulation experiments are shown in Figure 2. We can clearly see that: (1) Not all of the improvements with Levy flights would result in better performance; (2) The improvements replacing the random number in uniform distribution \(v_c\) by Levy flights would gain better performance; (3) The improvements replacing the random number in Gauss distribution \(r_1\) by Levy flights result in a slight reduction in performing optimization; (4) The slight reduction in performance demonstrates that slight difference between the randomness in Gauss distribution and Levy flight, even though there are big differences between them.

4.2 Experiments on multi-modal benchmark functions
The multi-modal benchmark functions would have multiple local optima whether they have one or more global optima. Therefore, the individuals would be easily trapped in local optima and thus result in worse performance, even failure in optimization.

In this sub-section, we would also introduce a fixed-dimensional benchmark functions which would be multi-modal, Adjiman function:

![Figure 2 Optimized results for Bartels Conn function](image-url)
Adjiman function is not symmetric with fixed domain [-2, 2]. It was continuous, differentiable, non-separable, non-scalable, multi-modal benchmark function. It has one global minimum optimum and it was not located at the Origin, the global optimum is located at \(x^* = (2, 0.10578)\) and \(f(x^*) = 0\), the final results were shown in Figure 3.

\[
f(x) = 2.02181 - \cos(x_1) \sin(x_2) - \frac{x_1}{x_2^2 + 1}
\]

(12)

Figure 3 Optimized results for Adjiman function

Simulation experiments on a representative for multi-modal benchmark functions showed that: (1) Not all of the improvements would perform better in optimization, which was similar to the results concluded from Figure 2; (2) The improvements replacing the random number in uniform distribution \(v_c\) by Levy flights would gain better performance; (3) The improvements replacing the random number in Gauss distribution \(r_1\) by Levy flights result in a critical reduction in performing optimization; (4) The critical reduction in performance demonstrates that although there are big differences between the randomness in Gauss distribution and Levy flight, the final results are not obtained as expected.

4.3 Experiments on benchmark functions with valleys

Besides the multi-modal benchmark functions, there are another kind of benchmark functions difficult to optimize even they are unimodal. The benchmark functions with valleys or basins are also challenging for optimization algorithms because when the individuals in swarms coming and approaching to the global optima which were inside the valleys or basins, they can derive rarely information towards the global optima.

In this experiment, we will introduce a continuous, differentiable, separable, scalable unimodal benchmark functions, Brown function:

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

(13)

Brown function is a symmetric function, it has no constrains for every parameter. Its global optimum is located at Origin \(x^* = (0, 0, \cdots, 0)\) and \(f(x^*) = 0\).

When the individuals are approaching to the global optimum, all of the fitness values are nearly the same to the global optimum, consequently, the individuals can gain nearly nothing towards the right directions, such situations usually increase the difficulty dramatically and sometimes cause the optimization algorithms fail to perform their jobs.

The final results are shown in Figure 4. Similarly, we can draw the conclusions that: (1) Not all of the improvements with Levy flights would result in better performance; (2) The improvements replacing the random number in uniform distribution \(v_c\) by Levy flights would gain better performance; (3) The improvements replacing the random number in Gauss distribution \(r_1\) by Levy flights result in a slight
reduction in performing optimization; (4) The slight reduction in performance demonstrates that slight difference between the randomness in Gauss distribution and Levy flight, even though there are big differences between them.

5. DISCUSSIONS AND CONCLUSIONS
Each of the compositions in the updating equations for individuals in swarms would be weighted normally with random numbers in Gauss distribution. However, the SMA introduce two random parameters in uniform distribution. The updating equations for the newly proposed SMA might be quite efficient because some of the individuals would be reborn in some circumstances. However, the random weights selected whether in Gauss distribution or uniform distribution are all accumulatively steady. Considering the large steps after several small steps, Levy flight would result in good performance during the exploration and exploitation for individuals.

In this paper, we improved the SMA with Levy flights. And Levy flights were introduced to replace the random weight in Gauss distribution \( r_1 \) and the random weight in uniform distribution \( v_c \). Three kinds of simulation experiments were carried out and results were averaged over 100 Monte Carlo simulations in order to eliminate the influence of randomness. Results showed that: Only the improved SMA with Levy flights replacing the uniform distributed random \( v_c \) by Levy flights resulted in better performance, whether the benchmark functions were unimodal, multimodal, or unique with basins in their profiles. The improved SMA replacing the random number in Gauss distribution \( r_1 \) would not achieve better performance, even fail sometimes.

Consequently, the improved SMA with Levy flights should focus mainly on the uniform distributed parameters.

ACKNOWLEDGMENT
The authors would like to thank the supports of 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;
[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.
REFERENCES
1] 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.
2] Seyedali Mirjalili, Andrew Lewis. The Whale Optimization Algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.
3] J. J. Lake, A. E. Duwel, R. N. Candler. Particle Swarm Optimization for Design of Slotted MEMS Resonators With Low Thermoelastic Dissipation[J]. Microelectromechanical Systems, Journal of, 2014, 23 (2): 364-371.
4] Zheng-Ming Gao, Juan Zhao. An Improved Grey Wolf Optimization Algorithm with Variable Weights[J]. Computational Intelligence and Neuroscience, 2019, 2019: 2981282.
5] Afshin Faramarzi, Mohammad Heidarinejad, Brent Stephens, et al. Equilibrium optimizer: A novel optimization algorithm[J]. Knowledge-Based Systems, 2019: 105190.
6] Zheng-Ming Gao, Juan Zhao, Su-Ruo Li. The Binary Equilibrium Optimization Algorithm with Sigmoid Transfer Functions: Proceedings of the 2020 12th International Conference on Machine Learning and Computing, Shenzhen, China, [C]. Association for Computing Machinery, 2020. 193–197.
7] Zheng-ming GAO, Su-Ruo LI, Juan ZHAO, et al. The improved Equilibrium Optimization Algorithm with Tent Map: 2020 IEEE 5th International Conference on Computer and Communication Systems (ICCCS), Shanghai, China, 2020[C]. IEEE, 343-347.
8] Shubham Gupta, Kusum Deep. A novel Random Walk Grey Wolf Optimizer[J]. Swarm and Evolutionary Computation, 2019, 44: 101-112.
9] Jiann-Horng Lin, Chao-Wei Chou, Chorng-Horng Yang, et al. A Chaotic Levy Flight Bat Algorithm for Parameter Estimation in Nonlinear Dynamic Biological Systems[J]. Journal of Computer and Information Technology, Summer 2012, 2 (2): 56-73.