The Improved Slime Mould Algorithm with Cosine Controlling Parameters

Zheng-Ming Gao¹, Juan Zhao² and Su-Ruo Li¹

¹School of Computer Engineering, Jingchu University of Technology, Jingmen 448000, China
²School of Electronics and Information Engineering, Jingchu University of Technology, Jingmen 448000, China
Email: gaozming@jcut.edu.cn

Abstract. Along with the fact that the problems we human met are becoming more complicated and complex, and more variables are involved, the new proposed algorithms would embrace more controlling parameters. However, not all of the controlling equations involved initially are satisfactory. Considering the ongoing runtime errors/warnings occurred in the program of a newly raised slime mould algorithm (SMA), we hence propose a replacement by the cosine function, which would produce a similar and slow-change for the controlling parameters. And it also met all of the values in the definitional domain. Simulations experiments were carried out and results proved that the improved SMA with cosine controlling parameters could not only eliminate the ongoing warnings/errors, it could also perform better in optimization.

1. Introduction
When we human achieved more in understanding nature, we still find more to be understood. And furthermore, more relevant factors would be involved in a given problem if we had another glance at it. Consequently, the problems would be more complicated and complex. Such experience had been successfully transferred to the design and proposing of new algorithms. In this paper, we would focus mainly on the nature inspired algorithms, which had already been a hot spot for scientists and engineers majored in computer science, mathematics, and all of others who need computation and optimization.

In the particle swarm optimization (PSO) algorithm [1], only two random numbers \( r_1 \) and \( r_2 \) and two learning parameters \( c_1 \) and \( c_2 \) are involved. In the standard PSO algorithm, there are no controlling variables and all of the individuals in swarms would be governed by a simple specific equation. In the bat algorithm (BA) [2], the frequency of sound \( F_i \), the amplitude \( A_i \) and the ratio \( R_i \) are involved, and furthermore, two another controlling parameters called the loudness deceasing factor \( \alpha \) and the pulse rate decreasing factor \( \rho \) are also introduced to constraint the parameters. In the equilibrium optimization algorithm [3], an exponential parameter \( F \) and the generation parameter \( G \) are involved and the number of their controlling parameters are six [3]. The performance indeed is increasing along with the increasing in numbers of controlling parameters.

Recently, a new kind of swarm-based algorithm called the slime mould algorithm (SMA) [4] was proposed. This time the numbers of controlling parameters had been increased greatly, and the individuals would be separated in two types carrying exploration and exploitation separately [5]. Despite of the numbers of controlling parameters, the SMA embraced the arctanh function to control the parameters for a proper change. However, the embedded constraints in arctanh resulted in a sequence...
of runtime warnings/errors in programming. The arctanh function might not be proper in this algorithm, so in this paper, we would propose a replacement by the cosine function [6].

The rest of this paper would be arranged as follows. A brief review on the SMA would be shown in section 2. And in section 3, we would talk about the cosine controlling parameters and simulation experiments would be carried out in details in section 4. Discussions would be made and conclusions would be drawn in section 5.

2. A Brief Review on the SMA

The SMA was inspired by the behaviour and morphological changes of slime mould. Individuals in the swarms would be separated into three types, some of them would be chosen to be reborn with a proportional number \( z \) and carry on the exploration from the beginning. Some of them would continue their exploration based on their current positions, and the rest of them would be guided towards to the best candidate. These complicated operations would be mathematically described as follows:

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

where \( r_1, r_2 \) and \( r_3 \) are three random numbers in Gauss distribution in the interval of 0 and 1. \([UB, LB]\) is the researching domain for the given problems. \( x_b(t) \) represents the position of the best candidate found by now. And \( t \) represents the current iteration. \( v_b \) and \( v_c \) are two random numbers in uniform distribution. They are controlled by the arctanh function and the maximum allowed iteration numbers \( maxIter \), which would be initialized at the beginning:

\[
a = \arctanh \left( 1 - \frac{t}{maxIter} \right)
\]

\[
b = 1 - \frac{t}{maxIter}
\]

The matrix \( W \) would be quite a little complicated. It would weight half of the top individuals to be larger than 1 and half of them to be smaller than 1, and the increased values are based on their best and worst fitness values \( bF, wF \):

\[
w_{si}(i) = \begin{cases} 
  1 + r_4 \cdot \log \left( 1 + \frac{bF - S(i)}{bF - wF} \right) & \text{condition} \\
  1 - r_4 \cdot \log \left( 1 + \frac{bF - S(i)}{bF - wF} \right) & \text{others}
\end{cases}
\]

where \( S(i) \) is the fitness value of \( i^{th} \) individual.

Another controlling parameter \( p \) is mathematically computed as follows:

\[
p = \tanh |S(i) - DF|
\]

3. Defects of Arctanh Function and the Replacements Cosine Function

The SMA would separate its individuals into three groups and carrying on the exploration and exploitation for better performance. The SMA performed better in optimization than other swarm-based algorithms even their improvements, claimed after detailed simulation experiments by the authors.

The arctanh function is the reverse function of tanh function, the mathematical expression for arctanh function is:

\[
y = \frac{1}{2} \ln \left( \frac{1 + x}{1 - x} \right)
\]
The embedded natural logarithm function and the fraction constraints the definitional domain of this function to be an open one: (-1, 1). However, in equation (2), the values of iteration time $t$ would be continuous integers from 0 to $\text{maxIter}$. And consequently, the real definitional domain would be [-1, 1] and errors would occur at the beginning and the end.

Furthermore, as the weights of the current positions or distances between two random selected candidates, $v_c$ and $v_b$ should not change abruptly. Therefore, we recommend the cosine function to replace the $\text{arctanh}$ function as follows:

$$a = 1 + \cos \left( \frac{t}{\text{maxIter}} \cdot \pi \right)$$

(7)

4. Simulation Experiments

Obviously, not all of the improvements would result in better performance, the bad improvements should be ignored. The improvements must be verified before applications. Fortunately, the mathematicians have raised a lot of test functions called the benchmark functions. They would be used to test the capability of algorithms in finding the best solutions. The capability of the improved algorithms could also be verified with such benchmark functions. Some of the benchmark functions are quite simple and easy to optimize, some of them not. The difference between them might rely on the characteristics such as multimodality, dimensionality, separability, scalability [7] and so on. We found that most of the benchmark functions difficulty to optimize would be highly multimodal, or have basins/valleys in their profiles.

Furthermore, the randomness is involved in most algorithms, and therefore, the final results would fluctuate with randomness. The normal methods to reduce the influence of randomness is Monte Carlo. Therefore, we would carry on 100 Monte Carlo simulation experiments for each problem and the final results would be the overall averages.

4.1. Experiments on Unimodal Benchmark Functions

Some of the benchmark functions would have only one local optimum, which could also be called the global optimum. This kind of benchmark functions are called the unimodal benchmark functions. Discarding some of the unimodal benchmark functions who have basins/valleys in their profiles, all of the unimodal benchmark functions are easy to optimize. In this experiment, we would introduce a little complex unimodal benchmark function as a representative: Ackley 1 function:

$$f(x) = -20e^{-0.02\sum_{i=1}^{d} \frac{x_i^2}{d}} - e^{-\sum_{i=1}^{d} \cos(2\pi x_i)} + 20 + e$$

(8)

The global/local optimum of Ackley 1 is located at the Origin $x^* = (0, 0, \ldots, 0)$ and $f(x^*) = 0$. Its three dimensional profile is shown in figure 1 and the Monte Carlo results are shown in figure 2.

We can see directly from figure 2 that the proposed improvements would perform better. The final results would be obtained after several rounds of iterations. And the improved SMA with cosine function would converge faster than the original one.

4.2. Experiments on Multimodal Benchmark Functions

Some of the benchmark functions would have many local optima, consequently, the individuals would be trapped in local optima and the global optima might be difficult to find. Therefore, most of the multimodal benchmark functions are quite a little difficult to optimize. Discarding those who have basins/valleys in their profiles, we would mainly focus on the modality, which is also an important characteristic of benchmark functions. A representative for multi-modal benchmark functions is Pathological function.

$$f(x) = \sum_{i=1}^{d} \left( 0.5 + \frac{\sin^2 \left( \sqrt{100x_i^2 + x_{i+1}^2} - 0.5 \right)}{1 + 0.001(x_i^2 - 2x_i x_{i+1} + x_{i+1}^2)^2} \right)$$

(9)
Pathological function is a continuous, differentiable, non-separable, scalable, and multimodal function, it is highly multimodal with many local optima, see from figure 3. The global optimum is located at the Origin $x^* = (0,0,...,0)$ and $f(x^*) = 0$. The averaged results of 100 Monte Carlo simulation experiments are shown in figure 4.

We can easily draw a similar conclusion that the improved SMA with cosine controlling parameters would also perform better than the original one, and the improved SMA would also converge faster.

### 4.3. Experiments on Benchmark Functions with Valleys

Whether unimodal or multi-modal, the benchmark functions would be difficult to optimize if they have basins or valleys in their profiles. The basins or valleys we here talked about are the basin or valley-like profile with the global optima, which mean that the global or local optima are inside a basin or a valley. In such conditions, the individuals in swarms would gain less information, even nothing towards the global optima, and therefore, the algorithms would fail. In this experiment, we would introduce a representative with a unique plus sign profile called Schwefel 2.20 function:
\[ f(x) = \sum_{i=1}^{d} |x_i| + \prod_{i=1}^{d} |x_i| \]  

(10)

Schwefel 2.22 function is a continuous, differentiable, non-separable, scalable, unimodal benchmark function. It is a symmetric function and it has no constraints for every parameter. The global optimum is located at the origin \( x^* = (0,0,\ldots,0) \) and \( f(x^*) = 0 \). The three dimensional profile appears as a cross, seen from figure 5. The final results for this experiment are shown in figure 6.

Glancing at figure 6, we can find that the improved SMA with cosine function would approach to the global optimum steadier than the original one.

5. Discussions and Conclusions

As we present in this paper, the controlling parameters are very important in helping the algorithm to obtain better performance. However, not all of the controlling equations are satisfactory, there exists a need to find a more suitable one and gain better performance.

In this paper, we focused on the controlling parameter with arctanh function, which had an open definitional domain and would result in warnings/errors in program. We proposed a replacement by cosine function and construct a controlling equation like equation (7). Simulation experiments were carried out and furthermore, we carried out 100 Monte Carlo simulation experiments and the averaged results would significantly reduce the influence of randomness involved in algorithms.

Results showed that the improved SMA with cosine controlling parameters would gain steadier results, converge faster, and perform better than that done by the original one. And most important of all, it would not lead to the warnings/errors in program, which would be very important for engineers applying the SMA in optimizing real engineering problems.

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; 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.
References

[1] Kennedy J and Eberhard R 1995 Particle swarm optimization Proceedings of ICNN’95 - International Conference on Neural Networks pp 1942-1948.

[2] Zhao J and Gao Z M 2014 The bat algorithm and its parameters The 4th International Conference on Electronic, Communications and Networks (CECNet2014) (Boca Raton: CRC Press) pp 1323-1326.

[3] Zhao J and Gao Z M 2020 A random walk equilibrium optimization algorithm 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE) (Shenzhen, China) pp 22-25.

[4] Shimin L, Huiling C, Mingjing W, et al. 2020 Slime mould algorithm: A new method for stochastic optimization Future Generation Computer Systems 111 300-323.

[5] Zhao J and Gao Z M 2020 Simulation research on the binary equilibrium optimization algorithm Proceedings of the 2020 12th International Conference on Machine Learning and Computing (Association for Computing Machinery) pp 140-144.

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

[7] Gao Z M and Zhao J 2019 An improved grey wolf optimization algorithm with variable weights Computational Intelligence and Neuroscience 2019 2981282.