Slime Mould Algorithm: A New Method for Stochastic Optimization

Shimin Li[a], Huiling Chen[a], Mingjing Wang[a], Ali Asghar Heidari[b,c], Seyedali Mirjalili[d]

[a] *College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, Zhejiang 325035, China*
simonlishimin@foxmail.com, chenhuiling.jlu@gmail.com, wangmingjing.style@gmail.com
[b] *School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran 1439957131, Iran*
as_heidari@ut.ac.ir, aliasghar68@gmail.com
[c] *Department of Computer Science, School of Computing, National University of Singapore, Singapore 117417, Singapore*
aliasgha@comp.nus.edu.sg, t0917038@u.nus.edu
[d] *4Institute for Integrated and Intelligent Systems, Griffith University, Nathan, QLD 4111, Australia*
ali.mirjalili@gmail.com

Abstract

Slime mould algorithm (SMA) is proposed based on the oscillation mode of slime mould in nature. The proposed SMA has several new features with a unique mathematical model that uses adaptive weights to simulate the process of producing positive and negative feedback of the propagation wave of slime mould based on bio-oscillator and to form the optimal path for connecting food with excellent exploratory ability and exploitation propensity. Source codes of SMA are publicly available at `http://www.alimirjalili.com/SMA.htm` and `https://tinyurl.com/Slime-mould-algorithm`.

Keywords: Slime mould algorithm, Adaptive weight, Engineering design problems, Constrained optimization

1. An Overview of slime mould algorithm

For complete details, please refer to main paper[1] established by [1].

1.1. Approach food

To model the approaching behavior of slime mould as a mathematical equation, the following rule is proposed to imitate the contraction mode:

$$X_{t+1} = \begin{cases} X_b(t) + v_b.\left(W.X_A(t) - X_B(t)\right) & r < p \\ v_c.X_t & r \geq p \end{cases} \tag{1}$$

where v_b is a parameter with a range of $[-a, a]$, v_c decreases linearly from one to zero. t represents the current iteration, X_b represents the individual location with the highest odor concentration currently found, X represents the location of slime mould, X_A and X_B represent two individuals randomly selected from the swarm, W represents the weight of slime mould. The formula of p is as follows:

$$p = \tanh |S(i) - DF| \tag{2}$$

[1]The paper is available at `https://doi.org/10.1016/j.future.2020.03.055` and `https://tinyurl.com/Slime-mould-algorithm`.

where $i \in 1, 2, \ldots, n$, $S(i)$ represents the fitness of X , DF represents the best fitness obtained in all iterations.

The formula of v_b is as follows:

$$v_b = [-a, a] \tag{3}$$

$$a = \text{arctanh}\left(-\left(\frac{t}{max_t}\right) + 1\right) \tag{4}$$

The formula of W is listed as follows:

$$W(SmellIndex(i)) = \begin{cases} 1 + r\log((b_F - S(i))/(b_F - w_F) + 1) & condition \\ 1 - r\log((b_F - S(i))/(b_F - w_F) + 1) & others \end{cases} \tag{5}$$

$$SmellIndex = sort(S) \tag{6}$$

where *condition* indicates that $S(i)$ ranks first half of the population, r denotes the random value in the interval of [0,1], b_F denotes the optimal fitness obtained in the current iterative process, w_F denotes the worst fitness value obtained in the iterative process currently, $SmellIndex$ denotes the sequence of fitness values sorted (ascends in the minimum value problem).

1.2. Wrap food

The mathematical formula for updating the location of slime mould is as follows:

$$X^* = \begin{cases} rand(UB - LB) + LB & rand < z \\ X_b(t) + v_b(WX_A(t) - X_B(t)) & r < p \\ v_c X(t) & r \geq p \end{cases} \tag{7}$$

where LB and UB denote the lower and upper boundaries of the search range, rand and r denote the random value in [0. 1].

1.3. Oscillation

The value of v_b oscillates randomly between $[-a, a]$ and gradually approaches zero as the iterations increase. The value of v_c oscillates between [-1, 1] and tends to zero eventually.

The logic of SMA is shown in Fig. 1 and its Pseudo-code represented in Algorithm 1.

Algorithm 1 Pseudo-code of SMA algorithm

```
Inputs: The population size N and maximum number of iterations max_t
Outputs: The best solution
Initialize the the positions of slime mould X_i (i = 1, 2, ..., n)
while (Stopping condition is not met) do
    Calculate the fitness of all slime mould
    Calculate the W by Eq. (5 )
    for (each search portion (X_i)) do
        Update p, v_b, v_c;
        Update positions by Eq. (7)
Return bestFitness and X_b
```

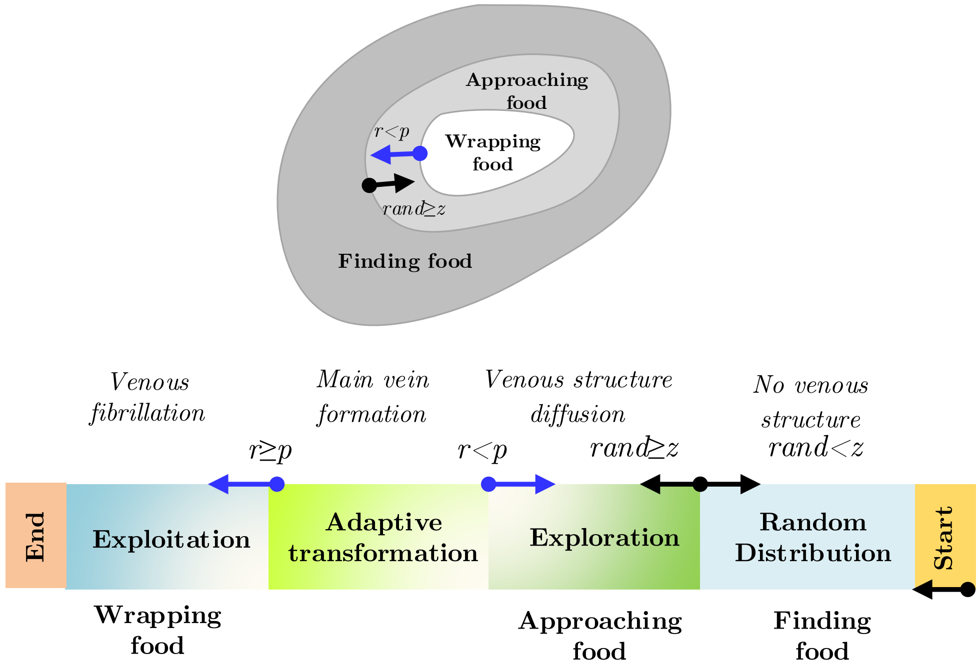

Figure 1: The overall steps of SMA

References

[1] S. Li, H. Chen, M. Wang, A. A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, Future Generation Computer Systems (2020). doi:https://doi.org/10.1016/j.future.2020.03.055.