The hybridized Harris hawk optimization and slime mould algorithm

Juan ZHAO1*, Zheng-Ming GAO2

1 School of electronics and information engineering, Jingchu University of technology, Jingmen 448000, China

2 School of computer engineering, Jingchu University of technology, Jingmen 448000, China

Corresponding author: ajuan323@jcut.edu.cn

Abstract. Both the Harris hawk optimization (HHO) algorithm proposed in 2016 and the slime mould algorithm proposed recently had complicated disciplines for individuals to update their positions. And both of them were proved to be capable of finding the best solutions for either benchmark functions or real engineering problems. In this paper, we further hybridized the SM and HHO algorithms and allowed the individuals in swarms to take more ways to update their positions. Simulation experiments were carried out and the better performance in either accuracy or convergence rate verified the capability of the hybridization.

1. Introduction

In order to improve the capability of nature inspired algorithms in optimization, the scientists and engineers made efforts on every aspects as they could discover. As for the control equations, only simple addition, subtraction, multiplication, or division were involved, such as the ant colony optimization (ACO) algorithm[1], the bat algorithm[2]. However, arctanh function was currently involved for the slime mould (SM) algorithm[3], sine and cosine functions were involved for the sine-cosine (SC) algorithm[4], and exponential function was involved for the equilibrium optimization (EO) algorithm[5]. As for the candidates involved in updating their positions, only the averaged information were considered in the ACO and bat algorithm, while in the particle swarm optimization (PSO) algorithm[6], the best candidates and the historical trajectories were all involved in the guidance. Furthermore, the grey wolf optimization (GWO) algorithm[7] would take the top three candidates in updating positions of individuals in swarm,. In the EO algorithm[8], four top candidates and their average were even took consideration in the guidance of updating, somehow only the averaged one could also succeed in doing so[9]. Another aspect of constructing the algorithms, might be the updating ways for individuals. In the ACO, PSO, or bat algorithms, all of the candidates would update their positions in the same way, and later, it appeared that the individuals could update their positions in different ways, such as the individuals in the SC algorithm, they are equally split into two groups, while in the SM algorithm, the individuals were grouped in three, and for the Harris hawk optimization (HHO) algorithm[10], five ways were introduced to the individuals.

In this paper, we would focus on the multiple updating discipline which mean that the individuals would have multiple choice to update their positions. Considering individuals in the HHO and SM algorithms having many ways to update their positions, we hybridize them and proposed an improved HHO algorithm with seven ways even further, and verify whether it would be true that the more ways for individuals to update their positions, the better they would perform.
The following sections would be arranged as follows, in Section 2, we would briefly describe the HHO and SM algorithms, and simulations would be carried out in Section 3. In Section 4, we would discuss about the multiple updating discipline and draw the conclusions based on the results.

2. The HHO and SM algorithms and their hybridization

In this section, we would briefly describe the multiple updating discipline involved in both the HHO and SM algorithms, and then proposed the hybridization.

2.1. The HHO algorithm

The HHO algorithm was inspired by the hunting behaviour of Harris hawks for rabbits. When the rabbits have much energy, the Harris hawk would explore the definitional domain \([LB, UB]\) with the following equation:

\[
x_i(t + 1) = \begin{cases} 
    x_r - r_1 |x_r - 2r_2 x_i(t)| & q \geq 0.5 \\
    (x_b - x_m) - r_3 (LB + r_4 (UB - LB)) & q < 0.5
\end{cases}
\]  

(1)

Where, \(x_i(t + 1)\) is the position of the \(i\)-th individual in the next iteration of \(t\), \(x_r\) is the position of a random selected candidate at the current iteration, \(x_b\) and \(x_m\) are the best and averaged positions in/of the swarms. \(r_1, r_2,\) and \(r_3\) are three random numbers in Gauss distribution. \(q\) represents the chance of individuals to follow which one of the two ways, this means that it is also a random number.

The energy of the rabbits embraced, represented by symbol \(E\), would be declined from the maximum value linearly to zero:

\[E = 2E_0 \left(1 - \frac{t}{\text{maxIter}}\right)\]  

(2)

Where \(E_0\) is the initial stage of the energy, which is also fluctuated in the interval of 0 and 1. \(\text{maxIter}\) represents the maximum allowed iteration number, which would be setup at the beginning. When \(|E| < 1\), the Harris hawk would approach to the rabbit with strategies, which would be described in details as follows.

2.1.1. Soft besiege. If \(|E| \geq 0.5\) and \(r \geq 0.5\), the Harris hawk would encircle the rabbits softly and scare rabbits to run, aiming at to make the rabbits exhausted. In this strategy, the Harris hawk would update their positions with the following equation:

\[x_i(t + 1) = x_b - x_i(t) - E |J \cdot x_b - x_i(t)|\]  

(3)

Where, \(J = 2(1 - r_h)\), represents the random jumps towards the rabbits.

2.1.2. Hard besiege. if \(|E| < 0.5\), and \(r \geq 0.5\), the rabbits have already been exhausted to a low energy, then, the Harris hawk would perform hard besiege and take the surprised pounce.

\[x_i(t + 1) = x_b - E |x_b - x_i(t)|\]  

(4)

2.1.3. Soft besiege with progressive rapid dives. When \(|E| \geq 0.5\) and \(r < 0.5\), the rabbits still have enough energy, so the Harris hawks still carry out soft besiege, however, in a more intelligent one:

\[x_i(t + 1) = \begin{cases} 
    Y & f(Y) < F(x_i(t)) \\
    Z & f(Z) < F(x_i(t))
\end{cases}\]  

(5)

Where,

\[Y = x_b - E |J \cdot x_b - x_i(t)|\]  

(6)

\[Z = Y + S \times LF(D)\]  

(7)

\(D\) is the dimension of the problems, and \(S\) is a random number. \(LF(x)\) represents the Levy flights in \(D\) dimension:

\[LF(x) = 0.01 \times \frac{\mu \times \sigma}{|v|^\frac{1}{\beta}}, \sigma = \left(\frac{\Gamma(1 + \beta) \times \sin(\frac{\pi \beta}{2})}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\frac{\beta-1}{2}}}\right)^{\frac{1}{\beta}}\]  

(8)
2.1.4. Hard besiege with progressive rapid dives. When $|E| < 0.5$, and $r < 0.5$, the rabbits are exhausted and the Harris hawks would carry on hard besiege with intelligence. The updating equations might be the same as equation (5), however, the middle parameter of $Y$ would be changed to be relevant to the averages $x_m$:

$$Y = x_h - E|J \cdot x_h - x_m|$$  \hspace{1cm} (9)

We can see that the Harris hawks are very smart and there would be five operations for them to search and hunt for the rabbits.

2.2. The SM algorithm

In the SM algorithm, the individuals would also have multiple ways to update their positions. Individuals in the SM algorithm would have a small chance of $z$ to be reinitialized same as at the beginning. They normally separated into two groups with a declined proportional number $p$. Parts of them would update their positions according to the best candidates and the weighted distance between two random selected candidates $x_A(t)$, and $x_B(t)$. Parts of them would stick to their own trajectories with randomness:

$$x_i(t + 1) = \begin{cases} 
    r_6(UB - LB) + LB & r_7 < z \\
    v_b \cdot [W \cdot x_A(t) - x_B(t)] & r_9 < p \\
    v_c \cdot x_i(t) & r_8 \geq p
\end{cases}$$  \hspace{1cm} (10)

Where, $r_6$, $r_7$, and $r_8$ are random numbers in Gauss distribution, $W$ is the weight, $v_b$ and $v_c$ are two random numbers in uniform distribution and their boundary limitations are also declined linearly. $p$ is also declined from 1 to zero along with the iterations.

2.3. The hybridization of HHO with SM algorithm

Both the HHO and SM algorithms have multiple updating ways, five ways for the individuals in the HHO algorithm and three ways for the individuals in the SM algorithm. So we will abandon the weighted distance between two random selected candidates and fill it with the intelligent operations carried by Harris hawks, then we might have more ways for individuals to follow:

$$x_i(t + 1) = \begin{cases} 
    r_6(UB - LB) + LB & r_7 < z \\
    v_b \cdot [W \cdot x_A(t) - x_B(t)] & r_9 < p \\
    r_9 \cdot x_i(t) & r_8 \geq p
\end{cases}$$  \hspace{1cm} (11)

$p$ is defined as a function of the current iteration number and the maximum allowed iteration number:

$$p = 1 - t_max/t$$  \hspace{1cm} (12)

And consequently, the individuals would have more ways to update their positions, they have many choices and might improve their performance in searching for the best solutions to problems we want to solve.

3. Simulation experiments

With a hypothesis that the more ways to update the positions, the individuals would perform better in their searching for the best solutions to problems we wanted to solve, simulation experiments would be needed to prove it to be true. We have verified that the unimodal benchmark functions are easy to optimize, either the EO algorithm or its binary version[11], or the GWO algorithm. Therefore, we would not carry on the simulation experiments on unimodal benchmark functions anymore. To reduce the influence of randomness, we still carry on the Monte Carlo methods in experiments and the final results would be their average.

3.1. Simulation experiments on multimodal benchmark functions

When searching for the best solutions to the multimodal benchmark, individuals in swarms would be easily trapped in local optima, which would be found frequently. In this section, we would carry on the simulation experiments with Cosine Mixture function:
\[ f(x) = \frac{d}{10} + \sum_{i=1}^{d} x_i^2 - \frac{1}{10} \sum_{i=1}^{d} \cos(5\pi x_i) \] (13)

Cosine Mixture function has a fluctuated profile over the definitional domain, which could be seen from its three-dimensional profile as shown in Figure 1. The hybridized HHO with SM algorithms would be significantly improved the capability of optimization, which was demonstrated by the results in Figure 2.

![Figure 1 Profile of Cosine Mixture function](image1)

![Figure 2 best fitness values versus iterations](image2)

3.2. Simulation experiments on benchmark functions with valleys
In this section, Bent Cigar function would be involved in this simulation experiments:

\[ f(x) = x_1^2 + 10^6 \sum_{i=2}^{d} x_i^2 \] (14)

Bent Cigar function might be difficult to optimize because the global optimum is located at the valley, as shown in Figure 3. When individuals are coming to the optima, they could get barely nothing along the valley. However, this time the improved HHO algorithm would also perform very better than the standard HHO algorithm, as shown in Figure 4.

![Figure 3 Profile of Bent Cigar function](image3)

![Figure 4 best fitness values versus iterations](image4)

3.3. Simulation experiments on non-symmetric benchmark functions
The benchmark functions differed in modality, separability, dimensionality, or even scalability\(^{[12]}\). Simulation experiments showed that most of them could be optimized, even they have basins or valleys or plates in their profiles. However, researches had proved that if the global optima were not at the
Origin, we called this characteristic symmetry, the algorithms would perform worse, even failed to do so\cite{13}. The global optima of most benchmark functions are located at the Origin, and the definitional domain is symmetric in numbers, for example, [-100, 100]. Their profiles would be axial-symmetric or mirror symmetric. Most of the algorithms would solve them very easy, especially if parts of the individuals would stick to their trajectories as in equations (10) and (11). So, what about the non-symmetric benchmark functions, like Schwefel 2.26:

$$f(x) = \max (|x_1 + 2x_2 - 7|, |2x_1 + x_2 - 5|)$$ (15)

The global optimum of Schwefel 2.26 function is (1, 3), as shown in Figure 5. Although Schwefel 2.26 function is quite simple, both of the algorithms perform bad jobs. However, the improved algorithm still act better than the original one, as shown in Figure 6.

![Figure 5 Profile of Schwefel 2.26 function(d=2)](image1)

![Figure 6 best fitness values versus iterations](image2)

### 4. Discussions and conclusions

The scientists and engineers are proposing many kinds of algorithms, and most of the nature inspired swarm-based algorithms have some similarity. In this paper, we focused on the multiple updating discipline for individuals in swarms.

Traditionally, the swarms would update their positions in a same way, however, it seemed that if the individuals have multiple choices, they would perform better in approaching to the global optima and gain better results. We can easily draw the conclusion as the ACO, PSO, SC, SM, HHO algorithms go along the roads.

Therefore, we proposed a hybridization of the HHO and SM algorithms, which have the most ways for individuals to update their positions. Simulation experiments were done to verify such hypothesis. The final results proved that it was true. If the individuals have proper multiple ways for updating their positions, they indeed perform well in optimization. Embracing the historical trajectories of their own, the individuals would perform better if the global optima are locating at the Origin, simulation experiments on multimodal benchmark functions even who have valleys in their profiles support such conclusion, we can also find it right with mathematics. Further experiments on non-symmetric benchmark functions also support the hypothesis, although the overall results were not promising.

Consequently, we can draw the conclusion that if we construct more ways for individuals to follow in updating their positions properly, they would find the global optima faster, and the algorithms would perform better than before.

### 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 and 2020YFYB033, 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 and PY202003.

References
[1] Tülin Inkaya, Sinan Kayalgil, Nur Evin Özdemirel. Ant Colony Optimization based clustering methodology[J]. Applied Soft Computing, 2015, 28 (0): 301-311. HTTP://DX.DOI.ORG/10.1016/J.ASOC.2014.11.060
[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] Shimin Li, Huiling Chen, Mingjing Wang, et al. Slime mould algorithm: A new method for stochastic optimization[J]. Future Generation Computer Systems, 2020, 111: 300-323. HTTPS://DOI.ORG/10.1016/J.FUTURE.2020.03.055
[4] Seyedali Mirjalili. SCA: A Sine Cosine Algorithm for solving optimization problems[J]. Knowl. Based Syst., 2016, 96: 120-133.
[5] 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
[6] J. J. Lake, A. E. Duwel, R. N. Candler. Particle Swarm Optimization for Design of Slotted MEMS Resonators With Low Thermoeelastic Dissipation[J]. Microelectromechanical Systems, Journal of, 2014, 23 (2): 364-371. 10.1109/JMEMS.2013.2275999
[7] 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
[8] Zheng-Ming GAO, Juan ZHAO, Su-Ruo Li, et al. The Improved Equilibrium Optimization Algorithm: 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), 24-26 April 2020, 2020[C]. 26-30. 10.1109/AEMCSE.50948.2020.00013
[9] 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
[10] Ali Asghar Heidari, Seyedali Mirjalili, Hosseam Faris, et al. Harris hawks optimization: Algorithm and applications[J]. Future Generation Computer Systems, 2019. HTTPS://DOI.ORG/10.1016/J.FUTURE.2019.02.028
[11] 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. 10.1145/3383972.3384064
[12] 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.
[13] 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