Improved Dung Beetle Optimization Algorithm Integrated with Hybrid Strategies

Yaowei Wang

***Abstract*—The Dung Beetle Optimization algorithm (DBO), as an innovative algorithm, possesses excellent optimization performance and has been widely applied to solve numerous optimization problems. However, it suffers from the imbalance between global and local exploration, which makes it prone to falling into local optimal solutions and often experiencing stagnation during the later stages of iteration. In view of this, this paper proposes an improved algorithm (HSFDBO) integrating multiple improvement strategies. Firstly, HSFDBO adopts an adaptive probability adjustment strategy to selectively choose suitable improvement measures at different iteration stages. Meanwhile, it balances its exploration ability by using cosine adaptive weight and lens imaging reverse learning strategies to avoid local optima. Additionally, through the introduction of Aquila optimization mutation operation, the current optimal individual is perturbed and mutated to effectively prevent stagnation in the later iterations. To verify the effectiveness of the improved algorithm, the optimization ability of HSFDBO is evaluated using the CEC2022 test functions, and the results show that its optimization seeking ability has been significantly improved.**

***Index Terms*—DBO algorithm, HSFDBO algorithm, Multiple strategies, Test functions**

I. INTRODUCTION

Outlined by XUE et al.[1], the DBO algorithm emerges as a fresh and innovative intelligence optimization technique. This algorithm draws inspiration from the natural behaviors of dung beetle groups, determining the individual updates by simulating five behaviors: rolling dung balls, dancing, laying eggs, foraging, and pilfering. The DBO algorithm demonstrates exceptional optimization precision and swift convergence rate. However, the No Free Lunch theorem[2] logically asserts that no singular optimization method can universally address all issues. While excelling in optimization performance, the DBO algorithm encounters difficulties in maintaining a balance between exploring globally and optimizing locally, making it susceptible to locally optimal solutions.

To overcome the limitations of the DBO algorithm, there have been efforts to enhance and optimize it. Zhu et al.[3] proposed an improved dung beetle algorithm (QHDBO). They employed a good point set strategy to enhance the diversity of the initial population, reducing the likelihood of getting trapped in local optima. Balancing global and local exploration in the algorithm is achieved through improvements in the convergence factor and other methods. They also utilized a t-distribution variation strategy based on quantum to expand the exploration space of solutions, facilitating better escape from local optima. Finally, through simulation experiments, they demonstrated the competitive performance of the QHDBO algorithm in terms of convergence speed, optimization capability, and other aspects. Shen et al.[4] proposed a dung beetle optimizer with multiple strategies (MDBO). They employed a dynamic reflection learning strategy to expand the algorithm search space and utilized Levy distribution for boundary handling. Additionally, the MDBO algorithm mitigated the impact of local optima on the algorithm through the incorporation of two crossover operators. Finally, through comparisons with other algorithms on test functions and an unmanned aerial vehicle path planning problem, they demonstrated the effectiveness of MDBO. Li et al.[5] improved the algorithm by integrating strategies such as random reverse learning and spiral foraging. Their focus was on enhancing population diversity, improving algorithm convergence speed, and avoiding issues like getting stuck in local optima. Through simulations and applications to path planning problems, they demonstrated the outstanding search capabilities of the proposed Multi-Strategy and Improved DBO (MSIDBO) algorithm.

Manuscript received March 11, 2025

Yaowei Wang, School of Computer Science and Technology, Tiangong University, Tianjin, China

II. IMPROVED DUNG BEETLE OPTIMIZATION ALGORITHM

In light of previous research findings and by addressing the limitations of the original DBO algorithm, this study innovatively proposes the HSFDBO algorithm. Firstly, introduce adaptive probability adjustment to modify the probability of strategy selection at different stages. Subsequently, cosine adaptive weight and lens imaging reverse learning are employed to help the algorithm balance the global and local search capabilities and overcome the challenge of falling into local optima. Additionally, the Aquila mutation is incorporated to prevent late-stage search stagnation. These improvements to the DBO algorithm advance the theoretical development of algorithmic enhancements.

A. Adaptive Probability Adjustment

During the rolling phase in the original DBO algorithm, an inequality $\delta < 0.9$ is used to control whether an advancing dung beetle encounters an obstacle. This approach may limit the ability of the DBO algorithm to select policies at different stages of its execution. In the HSFDBO algorithm, a referenced adaptive probability adjustment strategy[6] is employed to determine the updating strategy for rolling dung beetles in different execution stages. This aimed at boosting the algorithm ability in strategy selection. The equation for the adaptive probability p is as follows:

$$p = z - \alpha \times \frac{(Iter_{max} - t)}{Iter_{max}} \quad (1)$$

Where t stands for the present iteration count. z is a constant, set to 0.6 and α is the adaptive probability coefficient, set to 0.1. Using $\delta < p$ to control the individual updating approach selection for rolling dung beetles.

B. Cosine Adaptive Weight

Inspired by literatures[7, 8], this paper introduces a cosine adaptive weighting factor that integrates with the dancing behavior when a dung beetle encounters an obstacle. The cosine adaptive weighting factor affects the algorithm optimization capability at different periods through periodic variations. This method additionally ensures that the algorithm can perform a certain degree of global search in the later stages of execution, preventing premature convergence. The equation for the cosine adaptive weight can be expressed as:

$$\omega = \cos(\frac{\pi \cdot t}{2 \cdot Iter_{max}} + \pi) + 1 \quad (2)$$

C. Lens Imaging Reverse Learning

As an optimization algorithm improvement strategy, reverse learning expands the optimization scope by computing the reverse solution. The DBO algorithm encounters a reduction in population diversity in the later stages of execution, with the dung beetle population converging around the best individual. If the best individual becomes ensnared in a locally the best solution, the dung beetle population will strive to escape from the extreme local regions. This results in a reduced optimization range and decreased optimization accuracy for the algorithm. Inspired by literature[9], this paper employs the lens imaging reverse learning to enhance individual diversity. This strategy strengthens the DBO algorithm capacity to evade local optima. The schematic diagram of lens imaging reverse learning is shown in Fig. 1.

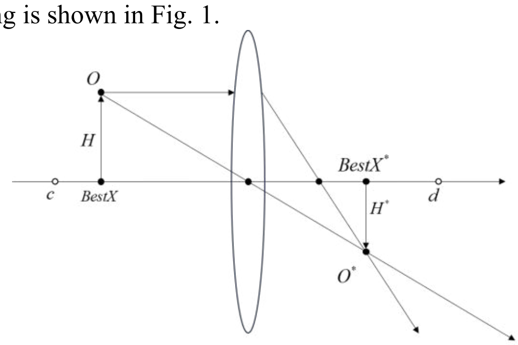


Figure 1: Lens Imaging Reverse Learning

Based on the fundamentals of convex lens imaging, the following equation can be formulated: In a specified search region, the position of the current best individual is $BestX$, $[c,d]$ represents the range of the search region. Placing a convex lens at the origin, assuming there is an object O, where the height of the object is H, and $BestX$ stands for the projected value of the object on the x-axis. The convex lens imaging results in an image O^*, where H^* is the height of the image. At this point, convex lens imaging facilitates obtaining the inverse solution $BestX^*$ for $BestX$. Based on the fundamentals of convex lens imaging, the following equation can be formulated:

$$\frac{(c_j + d_j)/2 - BestX_j}{BestX_j^* - (c_j + d_j)/2} = \frac{H}{H^*} \quad (3)$$

By performing a comprehensive substitution using $H / H^* = \mu$, the above equation can be transformed into:

$$BestX_j^* = \frac{(c_j + d_j)}{2} + \frac{(c_j + d_j)}{2\mu} - \frac{BestX_j}{\mu} \quad (4)$$

This way further enhances the information exchange ability within the dung beetle population, aiding the algorithm in escaping local optima. By adjusting the specific value of μ, the lens imaging reverse learning improvement strategy can achieve better outcomes. In this paper, μ is set to 1.5.

Using the previously described cosine adaptive weight and lens imaging reverse learning strategy to replace the dancing behavior of dung beetles in the original DBO algorithm when encountering obstacles. This approach aims to make the entire search process more balanced and enhance the coordination of local and global direct. Simultaneously, this method alleviates the influence of local optima on the original DBO algorithm. The individual updating method for rolling dung beetles encountering obstacles in the HSFDBO algorithm can be represented as:

$$x_i(t+1) = \omega \cdot (\frac{(c_j + d_j)}{2} + \frac{(c_j + d_j)}{2\mu} - \frac{BestX_j}{\mu}) \quad (5)$$

D. Aquila Mutation

In the later stages of iteration, the DBO algorithm faces the issue of insufficient search capability. The DBO algorithm relies on information exchange among individual dung beetles as it explores the solution space. For instance, during the rolling dung beetle stage, the algorithm avoids the current globally worst solution to find the local optimum. In other stages, it relies on the local best value to determine the search range. This information exchange mechanism allows the algorithm to swiftly converge towards a promising solution. However, this information exchange mechanism may cause the DBO algorithm to become stagnant in its subsequent iterations.

Introducing mutation operation into intelligent optimization algorithm facilitates a more extensive exploration of the solution region and enhances solution diversity. The paper proposes a mutation method known as Aquila mutation. Abualigah et al.[10] proposed the Aquila Optimization Algorithm in 2021. The inspiration for Aquila mutation stems from the fourth individual update approach within the Aquila optimization algorithm. This approach is characterized by its accelerated convergence capabilities. In the HSFDBO algorithm, Aquila mutation is applied on the presently global best individual, and then the greedy strategy is used to obtain the optimal solution of this iteration. This mutation method helps extend the search range of DBO algorithm when encountering stagnation and improves how effectively the algorithm optimizes in the subsequent iterations. The Aquila mutation is represented by the following equation:

$$\begin{cases} x_{new}(t) = t^{\frac{2\times rand()-1}{(1-Iter_{\max})^2}} \times X^*_{Best} - c_1 \times levy(d) \\ c_1 = 2\times(1-\frac{t}{Iter_{\max}}) \end{cases} \quad (6)$$

$x_{new}(t)$ represents the mutated individual generated in the tth iteration. X^*_{Best} represents the optimal solution for this iteration, and $levy(d)$ stands for the levy flight coefficient. s and β are set to 0.01 and 1.5, respectively. u and v are random numbers generated within the range of 0 to 1.

E. The Execution Flow of HSFDBO

Fig. 2 illustrates the execution flow of the HSFDBO algorithm.

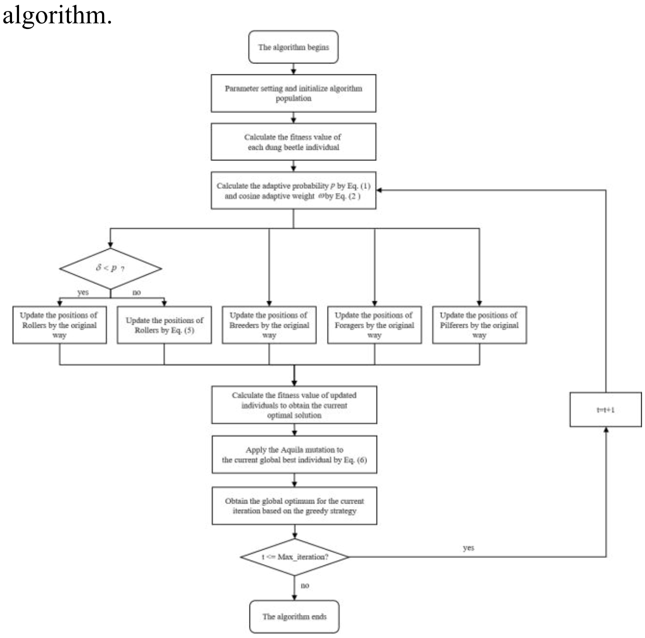


Figure 2: The execution flow of the HSFDBO algorithm

III. Experiment Results and Discussion

In order to validate the optimization capability of the proposed HSFDBO algorithm, CEC2022 test functions are employed for verification. CEC2022 test functions constitute a novel set of test functions, all designed for solving minimization optimization problems. In this paper, five famous intelligent optimization algorithms are compared with the HSFDBO algorithm. The five comparative algorithms are the Subtraction-Average-Based Optimizer (SABO)[11], Whale Optimization Algorithm (WOA)[12], Zebra Optimization Algorithm (ZOA)[13], Chernobyl Disaster Optimization Algorithm(CDO)[14] and DBO Algorithm. The other parameter settings for the compared algorithms are detailed in Table 1.

Table 1: Algorithm parameter settings

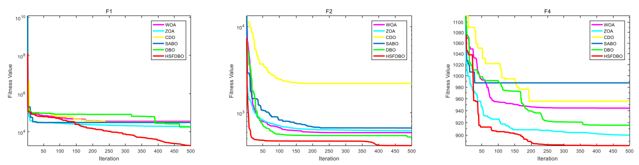
Algorithm	Population Size	Iterations	Parameters
SABO	30	500	-
WOA	30	500	a decreases linearly from 2 to 0
ZOA	30	500	Ps = rand R = 0.1
CDO	30	500	s_γ = Rand (1,300000) s_β = Rand (1,270000) s_α = Rand (1,16000) r = Rand (0,1)
DBO	30	500	k = 0.1, d = 0.3
HSFDBO	30	500	k = 0.1, d = 0.3

In this experiment, the comparison algorithms and experimental parameter settings remain consistent with the above experiments. The dimensions of all test functions are set to 20. For each of the six test algorithms, experiments are conducted 20 times. The performance of the algorithms in optimization is evaluated based on the mean and standard deviation of each experiment. Table 2 presents the performance results of all algorithms on the CEC2022 test functions. Based on the experimental results, it is evident that in most of the CEC2022 test functions, HSFDBO algorithm performs better in terms of average optimal values compared to other algorithms. In functions F1, F2, F6, F9, F10 and F12, HSFDBO algorithm maintains stability while achieving thorough optimization. HSFDBO algorithm demonstrates the best comprehensive performance in functions F3, F5 and F11. For function F8, HSFDBO algorithm exhibits outstanding optimization capability, ranking second only to CDO. Although ZOA outperforms HSFDBO algorithm in functions F4 and F7, the differences between the two are not significant. Across all test functions, the optimization capability of HSFDBO algorithm is notably superior to the original DBO algorithm, providing ample evidence for the effectiveness of the improvement strategy.

Table 2: CEC2022 benchmark functions test results

		WOA	ZOA	CDO	SABO	DBO	HSFDBO
F1	mean	3.3676E+04	1.7026E+04	3.1336E+04	3.1192E+04	3.4873E+04	**4.5586E+03**
	std	1.1151E+04	4.8388E+03	4.4110E+03	5.8918E+03	1.3464E+04	**3.1547E+03**
F2	mean	6.5870E+02	6.3731E+02	2.1043E+03	6.9348E+02	5.2800E+02	**4.4194E+02**
	std	6.1010E+01	6.4785E+01	5.2631E+01	1.0154E+02	7.4010E+01	**2.8150E+01**
F3	mean	6.6718E+02	6.4700E+02	6.6730E+02	6.4819E+02	6.3799E+02	**6.3132E+02**
	std	1.0030E+01	6.8216E+00	**6.1612E+00**	1.6304E+01	1.1814E+01	1.5725E+01
F4	mean	9.4671E+02	**8.6422E+02**	9.5423E+02	9.5739E+02	9.1415E+02	8.9114E+02
	std	3.0612E+01	**1.2711E+01**	1.5856E+01	1.6266E+01	3.3582E+01	1.7300E+01
F5	mean	4.6947E+03	1.9059E+03	3.5619E+03	2.5772E+03	2.1980E+03	**1.8127E+03**
	std	1.9326E+03	**2.5355E+02**	2.8272E+02	7.0973E+02	5.6106E+02	3.9263E+02
F6	mean	5.0562E+06	7.2405E+06	5.3698E+09	2.2266E+07	1.9496E+06	**8.0222E+03**
	std	3.3232E+06	1.7671E+07	1.2806E+09	2.5132E+07	6.2870E+06	**7.0616E+03**
F7	mean	2.2501E+03	**2.1196E+03**	2.3574E+03	2.1938E+03	2.1712E+03	2.1391E+03
	std	7.5589E+01	4.6408E+01	**4.2479E+01**	4.7699E+01	5.2159E+01	5.5763E+01
F8	mean	2.2972E+03	2.3330E+03	**2.2564E+03**	2.3446E+03	2.3178E+03	2.2652E+03
	std	5.3031E+01	1.2754E+02	**9.9342E+00**	6.1211E+01	7.2724E+01	5.2820E+01
F9	mean	2.5955E+03	2.6393E+03	3.4826E+03	2.7171E+03	2.5015E+03	**2.4666E+03**
	std	6.1166E+01	7.7668E+01	6.1664E+01	6.6028E+01	2.1360E+01	**5.7219E-01**
F10	mean	4.7997E+03	3.3701E+03	6.1812E+03	6.2945E+03	3.6751E+03	**3.1319E+03**
	std	1.6346E+03	1.0534E+03	9.2782E+02	1.4582E+03	1.1914E+03	**6.9743E+02**
F11	mean	4.1965E+03	4.7871E+03	8.5489E+03	5.4919E+03	2.9513E+03	**2.9500E+03**
	std	4.7114E+02	9.3667E+02	**4.6821E+01**	9.0195E+02	1.8266E+02	5.1307E+01
F12	mean	3.1439E+03	3.4569E+03	3.6334E+03	3.0902E+03	3.0636E+03	**2.8970E+03**
	std	1.3871E+02	1.3533E+02	1.9222E+02	5.1104E+01	6.8055E+01	**3.7102E+00**

Fig. 3 shows the fluctuation curves of the optimal fitness values of six experimental algorithms on some CEC2022 test functions. In this crucial test, the convergence curve of HSFDBO trends rapidly and smoothly, and it can stably reach the ideal fitness value range in a shorter period of time, greatly improving the optimization efficiency. This highlights the significant advantages of HSFDBO in the optimization scenarios of complex functions.



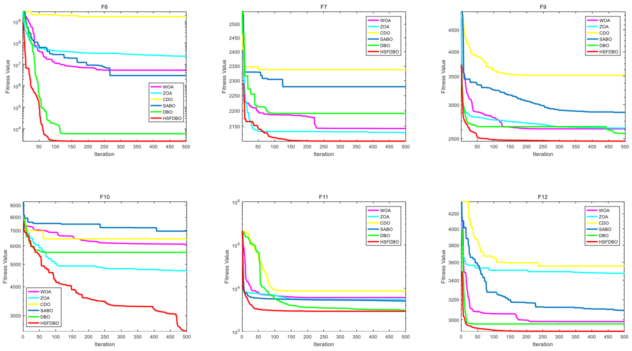


Figure 3: Convergence curves of some test functions

IV. CONCLUSION

The paper introduces the HSFDBO algorithm, an improvement upon the DBO algorithm. The HSFDBO algorithm improves the strategy selection ability of the original algorithm in different stages by incorporating an adaptive probability adjustment strategy. By utilizing the cosine adaptive weight and lens imaging reverse learning strategy, it balances the search capability of the algorithm, effectively avoiding local optima. Finally, inspired by the Aquila Optimizer, the algorithm introduces Aquila mutation in the later stages of execution, preventing the algorithm from stagnating in search and further enhancing its optimization capability. In order to assess the HSFDBO algorithm, this paper employs CEC2022 test functions as evaluation criteria. The experimental data indicates that the HSFDBO algorithm possesses faster convergence rate and superior convergence precision compared to various intelligence optimization algorithms.

Despite this, HSFDBO still has certain limitations. Firstly, while HSFDBO algorithm possesses relatively balanced search capabilities, it does not consistently achieve optimal performance across all test functions. In future work, our team will further explore the application of the HSFDBO algorithm in continuous problems, striving to improve its operational efficiency in solving high-dimensional complex problems. Furthermore, considering the broader application of HSFDBO, such as optimal path problems, we will further explore the practical application capabilities of the HSFDBO algorithm.

REFERENCES

[1] J. Xue and B. Shen, "Dung beetle optimizer: A new meta-heuristic algorithm for global optimization," *The Journal of Supercomputing,* vol. 79, no. 7, pp. 7305-7336, 2023, doi: 10.1007/s11227-022-04959-6.
[2] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation,* vol. 1, no. 1, pp. 67-82, 1997, doi: 10.1109/4235.585893.
[3] F. Zhu, G. Li, H. Tang, Y. Li, X. Lv, and X. Wang, "Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems," *Expert Systems with Applications,* vol. 236, p. 121219, 2024, doi: 10.1016/j.eswa.2023.121219.
[4] Q. Shen, D. Zhang, M. Xie, and Q. He, "Multi-Strategy Enhanced Dung Beetle Optimizer and Its Application in Three-Dimensional UAV Path Planning," *Symmetry,* vol. 15, no. 7, p. 1432, 2023, doi: 10.3390/sym15071432.
[5] L. Li *et al.*, "Enhancing Swarm Intelligence for Obstacle Avoidance with Multi-Strategy and Improved Dung Beetle Optimization Algorithm in Mobile Robot Navigation," *Electronics,* vol. 12, no. 21, p. 4462, 2023, doi: 10.3390/electronics12214462.
[6] W. Gao, S. Liu, Z. Xiao, and J. Yu, "Butterfly optimization algorithm based on Cauchy variation and adaptive weight," *Computer Engineering and Applications,* vol. 56, no. 15, pp. 43-50, 2020, doi: 10.3778/j.issn.1002-8331.1907-0048.
[7] Y. Chun and X. Hua, "Improved sine cosine algorithm for optimization problems based on self-adaptive weight and social strategy," *IEEE Access,* 2023, doi: 10.21203/rs.3.rs-2754846/v1.
[8] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Applied Soft Computing,* vol. 11, no. 4, pp. 3658-3670, 2011, doi: 10.1016/j.asoc.2011.01.037.
[9] C. Ouyang, D. Zhu, and Y. Qiu, "Lens learning sparrow search algorithm," *Mathematical Problems in Engineering,* vol. 2021, pp. 1-17, 2021, doi: 10.1155/2021/9935090.
[10] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. Al-Qaness, and A. H. Gandomi, "Aquila optimizer: a novel meta-heuristic optimization algorithm," *Computers & Industrial Engineering,* vol. 157, p. 107250, 2021, doi: 10.1016/j.cie.2021.107250.
[11] P. Trojovský and M. Dehghani, "Subtraction-average-based optimizer: A new swarm-inspired metaheuristic algorithm for solving optimization problems," *Biomimetics,* vol. 8, no. 2, p. 149, 2023.
[12] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software,* vol. 95, pp. 51-67, 2016, doi: 10.1016/j.advengsoft.2016.01.008.
[13] E. Trojovská, M. Dehghani, and P. Trojovský, "Zebra optimization algorithm: A new bio-inspired optimization algorithm for solving optimization algorithm," *IEEE Access,* vol. 10, pp. 49445-49473, 2022, doi: 10.1109/ACCESS.2022.3172789.
[14] H. A. Shehadeh, "Chernobyl disaster optimizer (CDO): a novel meta-heuristic method for global optimization," *Neural Computing and Applications,* vol. 35, no. 15, pp. 10733-10749, 2023, doi: 10.1007/s00521-023-08261-1.