An Adaptive Dung Beetle Optimization Algorithm with Golden Sine for Optimizing Numerical Unconstrained Problems

Abstract

The dung beetle optimization (DBO) algorithm is a newly swarm intelligence optimization algorithm inspired by the biological behaviors of dung beetles while it still has disadvantages of easy convergence to the local optimal, slow convergence speed, and poor global search capability. This paper proposes an adaptive dung beetle optimization algorithm with a golden sine algorithm (Gold-SA), denoted as the Gold-SA-based adaptive DBO (GSDBO) algorithm. Firstly, the PWLCM chaotic mapping is introduced to generate population individuals to increase diversity of population and explore more search space. Secondly, the position update formula for the mathematical model of dung beetle ball-rolling behavior without obstacle is replaced by that of Gold-SA, which can accelerate the convergence speed and improve the convergence accuracy. Finally, the adaptive weight coefficients are used to improve the update stage of thief beetles. The strategy can boost and balance the exploration vs exploitation, simultaneously. Furthermore, the GSDBO is proved to be effective by comparing some intelligence optimization algorithms on benchmark functions of different characteristics. The results demonstrate that the GSDBO can improve optimization accuracy and stability.

Keywords: Dung beetle optimization (DBO); PWLCM chaotic mapping; Golden sine algorithm; Adaptive weight coefficients; Unconstrainted optimization

1 Introduction

The swarm intelligence (SI) optimization algorithms inspired by the biological behaviors in nature have been extensive used on top of unconstrainted optimization problems due to their easy implementation and simple framework [1-4]. For example, the dung beetle optimization (DBO) algorithm which is based on the biological behaviors of dung beetles including ball rolling, egg laying, foraging, and stealing [5], tuna swarm optimization (TSO) which imitates the foraging behavior of tuna populations [6], the harris hawk algorithm (HHO) which simulates the harris hawk predation [7], whale optimization algorithm (WOA) which mimics the social behavior of humpback whales [8], grey wolf optimizer (GWO) based on the leadership hierarchy and hunting mechanism of grey wolves [9], and so on. However, slow convergence speed and low convergence accuracy in the early stages of establishment are common drawbacks in SI optimization algorithms which prevent them from being applied well in other fields. Based on these drawbacks, a series of improved SI optimization algorithms by introducing several modification tactics to enhance the optimization performance have been proposed [10-12].

In recent years, The DBO studied in this paper suffers from not only the problems which most SI optimization algorithms have but also poor global search capability and premature convergence to the local optima. Hence, many scholars have put forward some improvement algorithms based on the DBO and successfully applied in different fields [10,13]. Thus, it is meaningful to draw several modification tactics into the DBO to address its problems. This paper proposes an adaptive dung beetle optimization algorithm with a golden sine algorithm (GSDBO), in which PWLCM chaotic mapping, Gold-SA and adaptive parameters are used in population individual initialization stage of the DBO, ball-rolling behavior without obstacle and stealing behavior of dung beetles in the DBO respectively to enhance the convergence performance including accuracy and speed.

The paper is organized as follows: Section 2 provides a brief introduction to the DBO; the PWLCM chaotic mapping, golden sine algorithm, adaptive parameters and how to combine three modification tactics with the DBO are explained at length in Section 3; the experiments that compare the DBO with other SI optimization algorithms on different types of benchmark functions are conducted for evaluating the efficacy of the proposed GSDBO in Section 4; the conclusions of this paper are made in Section 5.

2. The dung beetle optimization(DBO) algorithm

The DBO is a novel SI optimization algorithm inspired by the behaviors of dung beetle including ball-rolling, dancing, foraging, breeding, and stealing. The DBO is designed to handle both unconstrained and constrained optimization problems. The population of the DBO is composed of four subpopulations including ball-rolling dung beetles, breeding dung beetles, small dung beetles, and stealing dung beetles. Four subpopulationsown their unique updating rules which are described as follow at length [10].

2.1 The mathematical model of dung beetle ball-rolling behavior

It should be noted that whether there are obstacles during the ball-rolling process will make dung beetles behave differently. When the dung beetlesmove forwards for search based on the navigation of sunwithout obstacle, the ball-rolling dung beetles update their position information based on the following equation:

$$x_i^{g+1} = x_i^g + a \times k \times x_i^{g-1} + b \times \left|x_i^g - x_{worst}^g\right| \#(1)$$

where g represents the number of the current iterations, x_i^g represents the position information of i^{th} dung beetle in the population at the g^{th} iteration, $k \in (0,0.2]$ representsan invariant quantityindicating the flexure coefficient, b represents a fixed parameter belonging to $(0,1)$,a represents a natural coefficientassigned 1 or -1 which indicates no deviation and deviation from the original direction respectively, x_{worst}^g represents the global worst positionat the g^{th} iteration, $\left|x_i^g - x_{worst}^g\right|$ is used to simulate the changes in light intensity.

When there are obstacles preventing dung beetles from processing, they need to adjust rolling direction by dancing based on a tangent function which is only considered to belong to $[0, \pi]$. At this point, the ball-rolling dung beetles update their position information based on the following equation:

$$x_i^{g+1} = x_i^g + \tan(\theta) \left|x_i^g - x_i^{g-1}\right| \#(2)$$

where $\left|x_i^g - x_i^{g-1}\right|$ represents the distance between i^{th} dung beetle at g^{th} iteration and at$(g-1)^{th}$ iteration. When θ takes value of $0, \frac{\pi}{2}$, or π,the position of dung beetle will not be updated.

2.2 The mathematical model of dung beetle breeding behavior

Inspired by the behavior that female dung beetlesattach importance tochoosing a suitable place for laying their eggsto provide a safe environment for their offspring, a frontier option strategy proposed to simulate the area where the eggs are produced can be described as follows:

$$\begin{cases} X^{Lb^*} = max\{X^* \times (1-R), X^{Lb}\} \\ X^{Ub^*} = min\{X^* \times (1+R), X^{Ub}\} \end{cases} \#(3)$$

where X^* represents the current local optimal position, X^{Lb^*} and X^{Ub^*} represent lower and upper limits of the spawning region, X^{Lb} and X^{Ub} representthe lower and upper bounds of optimization problem, $R = 1 - g/G$, and G represents the maximum number of iterations.

It should be noted that each female dung beetle generates only one egg in each iteration in the DBO. The position information of the female dung beetle laying eggs is dynamic for the reason that the boundary range is dynamically adjusted while the iteration progressing which can be described as follows:

$$x_i^{g+1} = X^* + b_1 \times \left(x_i^g - X^{Lb^*}\right) + b_2 \times \left(x_i^g - X^{Ub^*}\right) \#(4)$$

where x_i^g representsthe location information of the i^{th} brood ball at the g^{th} iteration, b_1 and b_2 represent two random and independent vectors with the size of $1 \times D$ each, and D represents the number of variables in the optimization problem.

2.3 The mathematical model of dung beetle foraging behavior

Mature dung beetles which emerge from ground for searching food are designated as small dung beetles. As the number of iterations increases, the optimal foraging area for small dung beetles is dynamically adjusted by the following formula:

$$\begin{cases} X^{Lb^b} = max\{X^b \times (1-R), X^{Lb}\} \\ X^{Ub^b} = min\{X^b \times (1+R), X^{Ub}\} \end{cases} \#(5)$$

where X^b represents the global best position, X^{Lb^b} and X^{Ub^b} represent the lower and upper bounds of the optimal foraging region. The formula for updating the position information of small dung beetles can be written as follows:

$$x_i^{g+1} = x_i^g + C_1 \times \left(x_i^g - X^{Lb^b}\right) + C_2 \times \left(x_i^g - X^{Ub^b}\right) \#(6)$$

where x_i^g represents the position of the i^{th} small dung beetles at the g^{th} iteration, C_1 is a random number with standard normal distribution, and C_2 represents a random vector within the range $(0,1)$ with the size of $1 \times D$.

2.4 The mathematical model of dung beetle stealing behavior

The position update equationfor the thieves which steal dung balls from other dung beetles can be defined as follows:

$$x_i^{g+1} = X^b + S \times t \times \left(\left| x_i^g - X^* \right| + \left| x_i^g - X^b \right| \right) \#(7)$$

where x_i^g represents the location information of the i^{th} thief at the g^{th} iteration, t represents a random vector obeying normal distribution with the size of $1 \times D$, and S represent a fixed parameter.

3. Adaptive dung beetle optimization algorithm with golden sine

The limitations including weak ability of global search and premature convergence to the local optimal prevent the DBO from being applied on complex optimization problems. Hence, an adaptive dung beetle optimization algorithm with golden sine (GSDBO) has been proposed to overcome the existing deficiency. The GSDBO algorithm mainly includes three modification tactics: PWLCM chaotic mapping, Gold-SA and adaptive parameters.

3.1 PWLCM Chaotic Mapping

In DBO algorithm. initial population come from random initialization which can cause rapid declines in population diversity and overconvergence in the subsequent iteration process.It has been confirmed that chaos modelshave significant improvement ondiversity of population in SI optimization algorithms due to their randomness and ergodicity. The basic method of some common chaos models including Kent [14], Tent [15], Logistic [16], and Circle [17] are that they involve their correspondingmapping chaotic sequences into individual search spaces. Simplicity and ergodicity are two significantcharacteristics which must be considered when selecting a suitable chaotic mapping to generate a random sequence. Hence, this paper choosesPWLCM Chaotic Mapping whose equation can be written as follows:

$$x_{i+1} = \begin{cases} \frac{x_i}{\eta}, & 0 \le x_i < \eta \\ \frac{x_i - \eta}{0.5 - \eta}, & \eta \le x_i < 0.5 \\ \frac{1 - \eta - x_i}{0.5 - \eta}, & 0.5 \le x_i < 1 - \eta \\ \frac{1 - x_i}{\eta}, & 1 - \eta \le x_i < 1 \end{cases} \#(8)$$

With the control parameter $\eta \in (0,0.5)$, the system $x_i \in (0,1)$ is in a chaotic state. The initial population individuals in dung beetle optimization algorithm based on the random sequence obtained byPWLCM Chaotic Mappingimprove diversity of population. The parameter η in this paper is set to 0.4. To verify the superiority ofPWLCM Chaotic Mapping, Logistic mapping and Circle mapping are used for comparison when setting the number of iterations to 5000 and the same initial value x_0.Fig. 1 presents the frequency distribution histograms of the three different chaotic mappings and fig. 2 presents the population distribution scatter map of the three different chaotic mappings.It can be seen from above two figs that initial population individuals based on PWLCM Chaotic Mapping have higher degree of dispersion and fewer individuals on the boundary and overlapping individuals than other two chaotic mappings which can ensure population diversity and reduce the attraction of local optima.

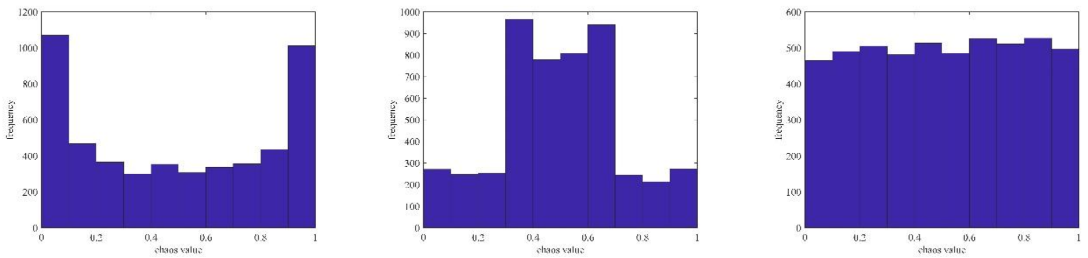


Fig.1. Chaotic Mapping Histogram. (a) Logistic mapping; (b) Circle mapping; (c) PWLCM mapping.

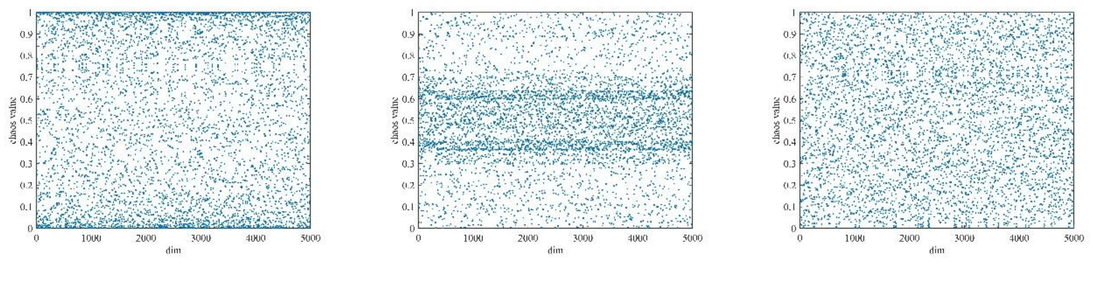


Fig.2. Chaotic Mapping Scatter map: (a) Logistic mapping; (b) Circle mapping; (c) PWLCM mapping.

3.2 Golden Sine Algorithm

In the DBO algorithm, equation (1) which is used to update ball-rolling dung beetle position without obstacle has poor capability of local search. Hence, this paper introduces population individuals update formula of golden sine algorithminto the GSDBO algorithm because it has strong ability of global search and meanwhile the golden partition coefficient is introduced to enhance the capability of local search [18]. The formula can be written as follows:

$$x_i^{g+1} = x_i^g \times |\sin(R_1)| - R_2 \times \sin(R_1) \times \left|m_1 \times P_i^g - m_2 \times x_i^g\right| \#(9)$$

where R_1 represents a random number belonging to $[0,2\pi]$ which determines the movement distance of i^{th} individual in the next iteration, R_2 represents a random number belonging to $[0,\pi]$ which determines the movement direction of i^{th} individual in the next iteration, m_1 and m_2 represent golden partition coefficients which are used to reduce the search space and guide the current individual to the global optimal. The coefficients m_1 and m_2 can be calculated by the equations as follows:

$$m_1 = h_1 \times \tau + h_2 \times (1 - \tau)$$

$$m_2 = h_1 \times (1 - \tau) + h_2 \times \tau \#(10)$$

$$\tau = \left(\sqrt{5} - 1\right)/2$$

where a and b represent initial golden values, and τ represents the golden ratio.

3.3 Adaptive weight coefficients

In order to address the drawbacks of the equation (7) which is easy to fall into local optima, the paper newly proposed a set of adaptive weight coefficients is introduced to extend the search space and keep a balance between global exploration and local exploitation. The formula can be rewritten as follows:

$$x_i^{g+1} = \varphi_1 \times X^b + \varphi_2 \times S \times t \times \left(\left|x_i^g - X^*\right| + \left|x_i^g - X^b\right|\right) \#(11)$$

$$\varphi_1 = 1 - \frac{g}{G}$$
$$\#(12)$$
$$\varphi_2 = \frac{g}{G}$$

3.4 The Pseudo Code of GSDBO

By introducing the above three modification tactics in the DBO, the pseudo code of GSDBO can be presented in Algorithm 1.

4. Numerical experiment and analysis

In this section, a series of experiments on some benchmark functions have been conducted to evaluate the efficacy of the proposed GSDBO algorithm.

4.1 Comparison algorithm and experimental parameters settings

Some other SI optimization algorithms including GWO [9], HHO [7], and DBO [5] are selected to compare with GSDBO algorithm under the same condition where the size of population was $N = 30$ and the number of maximum iterations were $G = 500$. The parameter settings of different algorithms are presented in Table 1according to their respective paper. The experiment was conducted on Windows 10 operating system, 64-bit OS, Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz, 96GB. The simulation software was Matlab 2018a.

Table 1. The parameters setting of the compared algorithms.

Algorithm	Parameter	Setting
WOA	b	1
GWO	a	Uniformly lowered from 2 to 0
HHO	Interval of E_0	$[-1,1]$
DBO	k,λ,b, and S	0.1, 0.1, 0.3, and 0.5
GSDBO	h_1,h_2	$-\pi$,and π

Algorithm1 The Pseudo Code of the GSDBO algorithm.

Input: The size of population *N*, the number of maximum iterations *G*, the objective function *f*, the problem bounds Lb and Ub, the problem dimension D.

/* Initialization */
Initialize population $i = 1,2,\cdots,N$ and define relevant parameter of algorithm.
Calculate the fitness of every individual and obtain the optimal solution X^b.
/* **Main loop** */
while $(g \leq G)$ **do**
  for $i = 1:N$
/* **The ball-rolling behavior** */
if $i ==$ ball-rolling dung beetle
      if $\delta_1 < 0.9$
        Update the position according to Eq. (9)
      else
        Update the position according to Eq. (2)
      end if
    end if
/* **The breeding behavior** */
    if $i ==$ brood ball
        Update the position according to Eq. (4)
    end if
/* **The foraging behavior** */
    if $i ==$ small dung beetle
        Update the position according to Eq. (6)
    end if
/* **The stealing behavior** */
    if $i ==$ thief
        Update the position according to Eq. (11)
    end if
  end for
  if the newly generated position is better than previous position
    Accept the new position.
  else
    Accept the previous position.
  end if
$g = g + 1;$
end while

Output: Optimal position X^b and its corresponding fitness value.

4.2 Benchmark Functions

The 10 benchmark functions including unimodal function ($f_1 \sim f_4$) which only one single global optimum solution utilized to test the speed and exactness of convergence of the algorithm and multimodal benchmark functions ($f_5 \sim f_6$) which contain a global optimum solution and several local optimum solutions used to gauge the performance of the algorithm to overstep the local optimum are selected to validate capacity for optimization of the GSDBO. The detail information of benchmark functions are presented in Table2.

4.3 Experimental results and discussion

In order to reduce the impact of randomness, it is necessary to repeat the proposed GSDBO algorithms and some other compared algorithms for 30 times independently. Three different statistical tools including best-seeking optimum (Best), the mean value (Mean), and the standard deviation (Std) are selected as performance indicators.

Table 2. Detail information of benchmark functions

Function name	Function	Dim	Range	f_{min}
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100,100]$	0

Schwefel 1.2	$f_2 = \sum_{i=1}^{n}\left(\sum_{j=1}^{i} x_j\right)^2$	30	$[-100,100]$	0
Cjgar	$f_3(x) = x_1^2 + 10^6 \sum_{i=2}^{n} x_i^2$	30	$[-100,100]$	0
Zakharov	$f_4(x) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} ix_i\right)^2 + \left(\sum_{i=1}^{n} ix_i\right)^4$	30	$[-5,10]$	0
Rastrigin	$f_5(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	30	$[-5.12,5.12]$	0
Apline	$f_6(x) = \sum_{i=1}^{n}\lvert x_i \sin(x_i) + 0.1x_i\rvert$	30	$[-10,10]$	0

4.3 Experimental results and discussion

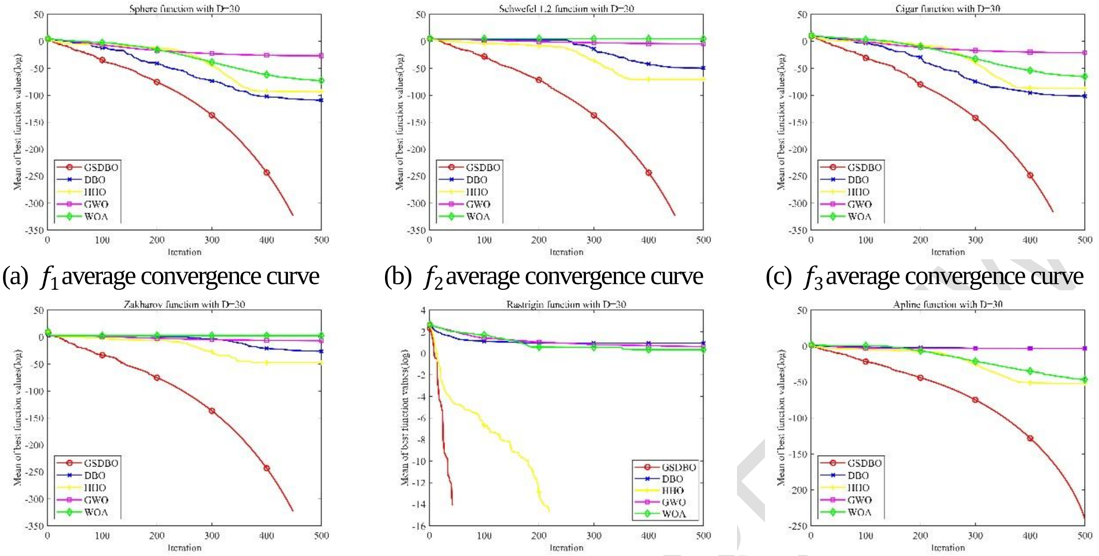
It should be noted that the proposed GSDBO and several compared algorithms need to be repeated for 30 times independently for the purpose of reducing the influence of randomness. This section selects three different statistical tools which include best-seeking optimum (Best), the mean value (Mean), and the standard deviation (Std)as the performance indicators.

Table 3. The experimental results of unimodal and multimodal benchmark functions

Function	Indicator	GSDBO	DBO	HHO	GWO	WOA
f_1	Best	0.00E+00	1.41E-156	2.98E-111	7.24E-30	7.56E-89
	Mean	0.00E+00	7.11E-110(+)	4.86E-94(+)	1.58E-27(+)	1.59E-73(+)
	Std	0.00E+00	3.48E-109	2.54E-93	3.43E-27	6.93E-73
f_2	Best	0.00E+00	7.99E-149	3.76E-102	1.16E-08	2.16E+04
	Mean	0.00E+00	2.28E-50(+)	2.02E-71(+)	2.27E-05(+)	4.85E+04(+)
	Std	0.00E+00	1.25E-49	1.10E-70	7.28E-05	1.31E+04
f_3	Best	0.00E+00	5.56E-173	1.82E-104	1.80E-23	7.91E-79
	Mean	0.00E+00	2.14E-102(+)	3.42E-88(+)	1.10E-21(+)	5.49E-66(+)
	Std	0.00E+00	1.17E-101	1.57E-87	2.12E-21	1.69E-65
f_4	Best	0.00E+00	3.45E-97	1.35E-87	6.81E-10	3.37E+02
	Mean	0.00E+00	6.84E-27(+)	8.25E-48(+)	1.69E-07(+)	4.96E+02(+)
	Std	0.00E+00	3.74E-26	4.23E-47	3.65E-07	9.07E+01
f_5	Best	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Mean	0.00E+00	8.56E+00(=)	0.00E+00(=)	4.19E+00(+)	2.17E+00(=)
	Std	0.00E+00	3.26E+01	0.00E+00	4.97E+00	1.19E+01
f_6	Best	0.00E+00	3.79E-80	8.22E-61	6.36E-17	1.79E-58
	Mean	0.00E+00	4.77E-04(+)	1.92E-52(+)	4.51E-04(+)	2.53E-47(+)
	Std	0.00E+00	1.53E-03	6.07E-52	6.22E-04	1.39E-46
+/-/=			5/0/1	5/0/1	6/0/0	5/0/1

The three performance indicators for the GSDBO and its comparative algorithms are presented in Table 3. As shown in Table 3, the GSDBO demonstrates significant improvementfor all six test functions. It can be seen that only the GSDBP proposed in the paper can find the theoretical optimal value 0 and the value of 'Best' and 'Std' are 0 for the unimodal benchmark functions f_1~f_4 which indicate that the GSDBO possesses excellentrobustness and stability. For multimodal benchmark functions f_5, the performance of GSDBO and HHOare comparable and they canachieve theoretical optimal value 0 while remaining algorithms only occasionally realize. For multimodal benchmark functions f_6, the GSDBO achieves 100% of the optimal search effect which is significantly superior to other compared optimization algorithms. In view of the reason that the GSDBO proposed in the paper is an improvement based on the DBO, it is necessary to conduct the performance comparison between the GSDBO and DBO. According to the experimental results in Table 3, the GSDBO has improved the convergence accuracy and stability of the DBO.

In addition, results of Wilcoxon sign-rank test with a significant level α=0.05 are presented in Table 3 to find out whether the proposed GSDBO has a significant performance difference compared with other four algorithms. The symbol '+' indicates the GSDBO is superior to the compared algorithms; the symbol '-' indicates the GSDBO is inferior to the compared algorithms and the symbol '=' indicates the GSDBO is similar to the compared algorithms.The last row of the Table 3 displays the total number in the form of "+/-/=". It can be concluded that the convergence performance of GSDBO is superior to that of the comparative algorithms.



(a) f_1 average convergence curve (b) f_2 average convergence curve (c) f_3 average convergence curve

(d) f_4 average convergence curve (e) f_5 average convergence curve (f) f_6 average convergence curve

Fig.3. Average convergence curve of the benchmark function

Fig. 3 where the horizontal axis denotes the number of iterations and the vertical axis represents the average fitness values expressed in logarithmic form with the base of 10 to better display the trend of convergence presents convergence curves for the GSDBO and comparative algorithms. As shown in Fig. 3, proposed GSDBO exhibits faster convergence rate and higher convergence accuracy than the compared algorithms. For unimodal benchmark functions, the convergence curves of the GSDBO present a near-linear descent to the theoretical optimal values indicating that the GSDBO can determine the region immediately where the global optimal solution may exist and move towards it. For multimodal benchmark functions, the GSDBO converges with a sharp decline in a straight line to obtain theoretical optimal values which suggest that the proposed GSDBO can jump out of the local optima effectively. In general, the three modification tactics adopted to enhance the convergence performance of the DBO are useful.

5. Conclusion

To address the existing problems of the DBO including the poor capability to conduct global search and escape the local optimum, this paper introduces PWLCM chaotic mapping, Gold-SA and adaptive weight coefficients into the DBO to form a new algorithm called adaptive dung beetle optimization algorithm with a golden sine algorithm (GSDBO). The effectiveness of the proposed algorithm is verified by comparing it with other four comparative algorithms on six different type of benchmark functions. The evaluating indicators used in the paper include best-seeking optimum, the mean value, the standard deviation and Wilcoxon sign-rank test. The experiment results show that the proposed algorithm has best convergence performance and enhances the capability of jumping out of the local optima. In future research work, the convergence performance of the GSDBO could be further improved and we can consider combingthe GSDBO with BPNN to perform structural damage identification, image enhancement, and so on.

Reference

[1] Wang X, Hu H, Liang Y, et al. On the mathematical models and applications of swarm intelligent optimization algorithms[J]. Archives of Computational Methods in Engineering, 2022, 29(6): 3815-3842.

[2] Tang J, Liu G, Pan Q. A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends[J]. IEEE/CAA Journal of AutomaticaSinica, 2021, 8(10): 1627-1643.

[3] Phan H D, Ellis K, Barca J C, et al. A survey of dynamic parameter setting methods for nature-inspired swarm

intelligence algorithms[J]. Neural computing and applications, 2020, 32(2): 567-588.

[4] Dhawan S, Gupta R, Rana A, et al. Various swarm optimization algorithms: Review, challenges, and opportunities[J]. Soft Computing for Intelligent Systems: Proceedings of ICSCIS 2020, 2021: 291-301.

[5] Xue J, Shen B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization[J]. The Journal of Supercomputing, 2023, 79(7): 7305-7336.

[6] Xie L, Han T, Zhou H, et al. Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization[J]. Computational intelligence and Neuroscience, 2021, 2021: 1-22.

[7] Heidari A A, Mirjalili S, Faris H, et al. Harris hawks optimization: Algorithm and applications[J]. Future generation computer systems, 2019, 97: 849-872.

[8] Mirjalili S, Lewis A. The whale optimization algorithm[J]. Advances in engineering software, 2016, 95: 51-67.

[9] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in engineering software, 2014, 69: 46-61.

[10] Shen Q, Zhang D, Xie M, et al. Multi-Strategy Enhanced Dung Beetle Optimizer and Its Application in Three-Dimensional UAV Path Planning[J]. Symmetry, 2023, 15(7): 1432.

[11] Yu X, Luo W, Rao R V. Multi-strategy Jaya algorithm for industrial optimization tasks[J]. Journal of Intelligent & Fuzzy Systems, 2022, 43(4): 4379-4393.

[12] Liu Q, Li N, Jia H, et al. A hybrid arithmetic optimization and golden sine algorithm for solving industrial engineering design problems[J]. Mathematics, 2022, 10(9): 1567.

[13] Zhu F, Li G, Tang H, et al. Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems[J]. Expert Systems with Applications, 2024, 236: 121219.

[14] Wang Y, Chen S, Wang Y. Chaos Encryption Algorithm Based on Kent Map** and AES Combination[C]//2018 International Conference on Network, Communication, Computer Engineering (NCCE 2018). Atlantis Press, 2018: 588-592.

[15] Tu N, Fan Z. IMODBO for Optimal Dynamic Reconfiguration in Active Distribution Networks[J]. Processes, 2023, 11(6): 1827.

[16] Jian X, Weng Z. A logistic chaotic JAYA algorithm for parameters identification of photovoltaic cell and module models[J]. Optik, 2020, 203: 164041.

[17] Wang W, Tian J. An Improved Nonlinear Tuna Swarm Optimization Algorithm Based on Circle Chaos Map and Levy Flight Operator[J]. Electronics, 2022, 11(22): 3678.

[18] Tanyildizi E, Demir G. Golden sine algorithm: A novel math-inspired algorithm[J]. Advances in Electrical and Computer Engineering, 2017, 17(2): 71-78.