Improved DBO Algorithm Incorporating Disorienting Behavior and Dynamic Population Strategy for Engineering Problem Solving

Xu Ji, Qiang Qu*, Yu-Long Ren, Jia-Xun Lian, Tian-Ran Jiang

Abstract—**To enhance the balance between global and local search capabilities of Dung Beetle Optimization (DBO) algorithm, this study proposes a multi-strategy hybrid improvement, resulting in Dynamic Population Dung Beetle Optimization (DPDBO) algorithm. Initially, a Piecewise Linear Chaotic Map (PWLCM) is employed for population initialization, thereby improving the uniformity of the initial population distribution and, consequently, enhancing the global search capability of DPDBO. Subsequently, the disorienting behavior of the ball-rolling dung beetle is modeled to expand the exploration of the solution space by integrating the original linear search method with an enhanced spiral search strategy. Additionally, a dynamic population strategy is introduced to adaptively allocate ball-rolling and stealing dung beetles, effectively balancing the exploration and exploitation capabilities of DPDBO. Comparative analysis of DPDBO against six other meta-heuristic algorithms (DBO, Sparrow Search Algorithm (SSA), Harris Hawks Optimization (HHO), Whale Optimization Algorithm (WOA), Genetic Algorithm (GA), and Sine Cosine Algorithm (SCA)) using CEC2017 and CEC2022 test functions demonstrates that the proposed algorithm achieves faster convergence, higher optimization accuracy, and greater robustness. Furthermore, DPDBO is applied to three engineering applications (e.g., sensor coverage optimization, tension spring design, and welded beam design) showcasing its efficacy in solving real-world engineering problems.**

Index Terms—**Dung beetle optimization algorithm; Chaotic mapping; Disorienting behavior; Dynamic population strategy; Engineering problems.**

I. Introduction

OPTIMIZATION problems have been extensively utilized in various fields, including engineering design[1], neural networks[2], image processing[3], job scheduling[4], and path planning[5], utilize optimization methods. The conventional approach to solving optimization problems involves approximating the optimal solution using first-order or second-order derivatives. However, this traditional method is prone to getting trapped in local optima due to its reliance on derivatives, making it challenging to solve complex multi-dimensional problems. Therefore, researchers introduced numerous metaheuristic algorithms to address optimization problems. These algorithms have garnered significant interest since their inception because of their simplicity, flexibility, independence from derivatives, and ability to avoid getting stuck in local optima.

Meta-heuristic algorithms are primarily categorized into two types: natural physical mechanism-based algorithms and swarm intelligence-based algorithms. Natural physical mechanism-based algorithms consist of Genetic Algorithm (GA) [6], Simulated Annealing (SA) algorithm [7], Optical Microscope Algorithm (OMA) [8]. GA replicates the process of biological evolution by producing offspring through crossover and mutation operations, and subsequently selecting the most optimal individuals to form the next generation. On the other hand, SA imitates the physical transformation of a solid material from a high temperature to a low temperature. Additionally, inspired by the concept of magnification in optical microscopes, Cheng and Sholeh introduced OMA.

The swarm intelligence-based algorithms primarily consist of Ant Colony Algorithm (ACA) [9], Particle Swarm Optimization (PSO) [10], Artificial Bee Colony Algorithm (ABC) [11], Grey Wolf Optimizer(GWO) [12], Whale Optimization Algorithm (WOA) [13], Harris Hawks Optimization (HHO) [14], Sparrow Search Algorithm(SSA) [15], Snake Optimizer (SO) [16], Sand Cat Swarm Optimization (SCSO) [17], Spider Wasp Optimizer (SWO) [18]. PSO, as a notable example of swarm intelligence optimization algorithms, was introduced by Kennedy and Eberhart [10]. It emulates the foraging behavior of bird flocks and achieves global search capability through adaptive learning and individual interaction. ACA replicates the behavior of ants releasing pheromones to determine the optimal path or solution. ABC imitates the honey harvesting behavior of a bee colony. GWO achieves optimization by simulating the predation behavior of a grey wolf pack. WOA utilizes the spiral search mechanism to mimic humpback whale-specific search methods. HHO models the predatory characteristics of Harris hawk and incorporates Levy flights to address complex multi-dimensional problems. SSA, introduced in 2020, imitates the behavior of sparrows in the search for food and evasion of natural enemies. SO and SCSO simulate the foraging and mating behaviors of snakes and sand cats, respectively. SWO, by mimicking the hunting, nesting, and mating behaviors of female wasps, employs various unique updating strategies to

Manuscript received June 17, 2024; revised on 21 November 2024.

This work was supported by the National Natural Science Foundation of China (71371092), and the Scientific Research Fund Project of Liaoning Provincial Department of Education (2020LNJC04).

Xu Ji is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (email: 16642298929@163.com).

Qiang Qu is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (*corresponding author to provide email: quqiang@ustl.edu.cn).

Yu-Long Ren is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (email: r18764691056@163.com).

Jia-Xun Lian is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (email: lianjiaxun100@163.com).

Tian-Ran Jiang is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (email: 1134709671@qq.com).

enhance optimization performance. Additionally, there are several other algorithms in this category. with outstanding performance has been suggested in recent years [19-24].

Xue et al. [25] introduced Dung Beetle Optimization (DBO) algorithm, inspired by dung beetle behaviors such as foraging and spawning. DBO has gained popularity due to its minimal adjustable parameters and effective optimization capabilities. However, similar to other swarm intelligence algorithms, DBO is prone to slow convergence and local optima trapping. Consequently, researchers have developed numerous enhanced algorithms. Zhang et al. [26] proposed a producer model to accelerate the convergence rate of DBO. Additionally, a dimension learning strategy was implemented to facilitate information exchange among individuals within the population, thereby improving the algorithm's exploration of solution spaces. Zhu et al. [27] introduced a Quantum Computing and Multi-strategy Fusion-based Dung Beetle Optimization (QHDBO) algorithm. This algorithm utilizes a good point-set strategy to enhance population diversity. The quantities of spawning and foraging dung beetles are dynamically regulated to maintain a balance between global and local search capabilities. The dynamic parameter R is adjusted in a non-linear manner to enhance convergence speed. Furthermore, a t-distribution variation strategy based on quantum computing is employed to perturb the optimal solution and increase the algorithm's ability to escape local optima. Zhang et al. [28] proposed an improved DBO algorithm incorporating cosine inertia weights. A nonlinear factor was integrated into the ball-rolling dung beetle model to manage the impact of light intensity on global and local searches. The cosine inertia weight was embedded in the stealing dung beetle model to enhance the algorithm's search performance. Li et al. [29] utilized Fuch mapping and a reverse-learning strategy to diversify the population during initialization.

While these enhancements to DBO demonstrate satisfactory performance, there are still deficiencies present in DBO. For instance, the initial distribution of the population lacks uniformity, thereby impacting DBO's global search capability to certain degree. Neglecting to consider the disorienting behavior of dung beetles in the absence of light can diminish the diversity of search strategies. The fixed quantities of ball-rolling dung beetles and dung beetles engaged in stealing hinder DBO's capacity to adequately balance exploration and exploitation.

Based on the aforementioned analysis, this paper posits the subsequent enhancement strategies:

To improve the global search capability, the population of DBO algorithm is initialized using a Piece-Wise Linear Chaotic Map (PWLCM).

The spiral factor is applied to replicate the disorienting behavior of ball-rolling dung beetles in the absence of a guiding light source. This combines the straight-line approach of the traditional DBO to create a new search strategy, enhancing DBO's ability to navigate and explore the solution space.

This study proposes a dynamic population strategy that is based on an adaptive factor to balance the capacities for exploration and exploitation.

The remainder of this study is organized as follows: Section II describes the fundamentals of DBO, Section III analyzes DBO and outlines three improvement strategies for Dynamic Population Dung Beetle Optimization (DPDBO). Following this, Section IV conducts simulation experiments, while Section V examines and analyzes the engineering problems. Lastly, Section VI concludes the results of this study.

II. Dung Beetle Optimization Algorithm

DBO categorizes dung beetles into four groups according to their behavior: ball-rolling dung beetles, spawning dung beetles, foraging dung beetles, and stealing dung beetles.

A. Ball-rolling behavior

In nature, the dung beetle transforms dung into a spherical ball and transports it to its nest. This dung-ball serves as both its sustenance and a foundation for reproduction. During the rolling process, the dung beetle relies on sunlight or moonlight to guide the dung-ball, ensuring it maintains a consistent trajectory. The equation that governs the location adjustments involved in this ball-rolling behavior can be articulated as follows:

$$x_i(t+1) = x_i(t) + \alpha k \cdot x_i(t-1) + b \cdot \Delta x_1 \tag{1}$$

where $\Delta x_1 =\mid x_i(t) - \ worst \mid$ represents the fluctuation of light intensity; t represents the current iteration number; $x_i(t)$ represents the positional data of the *i*-th dung beetle at the *t*-th iteration; $k \in (0, 0.2)$ represents the deflection coefficients; *b* represents a constant within the range of (0, 1); α represents a natural coefficient taking values of either 1 or -1, and $worst$ represents the global worst value.

When faced with an obstacle while rolling the dung-ball, the dung beetle exhibits a dance-like behavior to reorient itself and proceed in the desired direction. The equation used to update its position during this behavior is given as follows:

$$x_i(t+1) = x_i(t) + \tan(\theta) \cdot \Delta x_2 \tag{2}$$

where $\Delta x_2 = |x_i(t) - x_i(t-1)|$ and $\theta \in [0, \pi]$. When θ = 0, $\pi/2$, and π, the position of the dung beetle remains unchanged.

In summary, the position update formula for the ball-rolling dung beetle can be derived as follows:

$$\begin{cases} x_i(t+1) = x_i(t) + \alpha \cdot k \cdot x_i(t-1) + b \cdot \Delta x & \delta < p \\ x_i(t+1) = x_i(t) + \tan(\theta) \cdot \Delta x_2 & \delta \geq p \end{cases} \tag{3}$$

where δ represents a random number within the range of [0, 1] and p represents a factor that impact the likelihood of the dung beetle exhibiting dancing behavior. In the original DBO study, p was assigned a value of 0.9.

B. Spawning behavior

During the breeding season, dung beetles transport dung-balls to a secure location for reproduction. This secure location can be determined as follows:

$$\begin{cases} Lb^* = \max(X^* \cdot (1-R), Lb) \\ Ub^* = \min(X^* \cdot (1+R), Ub) \end{cases} \tag{4}$$

where X^* represents the current local optimum position; Lb^* and Ub^* represents the lower and upper bounds of the spawning region, repectively; $R = 1 - t/T_{max}$, T_{max} denotes the maximum number of iterations; while Lb and

Ub represent the lower and upper bounds of the optimization problem, respectively.

Once the spawning area has been identified, the female dung beetle deposits her eggs in the dung-ball (spawning ball). Assuming that each female dung beetle lays only one egg on the hatching ball, the positional adjustment of the hatching ball can be expressed as follows:

$$\begin{aligned} x_i(t+1) = X^* + b_1 \cdot (x_i(t) - Lb^*) \\ + b_2 \cdot (x_i(t) - Ub^*) \end{aligned} \tag{5}$$

where b_1 and b_2 represent two 1-D independent random vectors, while D represents the dimension of the optimization problem.

C. Foraging behavior

When the juvenile dung beetles inside the hatching ball reach maturity, they emerge to search for food, and the area where they search can be defined as follows:

$$\begin{cases} Lb^b = \max(X^b \cdot (1-R), Lb) \\ Ub^b = \min(X^b \cdot (1+R), Ub) \end{cases} \tag{6}$$

where X^b represents the global optimal position; while Lb^b and Ub^b represent the lower and upper boundaries of the spawning area, respectively. The position update of the baby dung beetle during foraging can be described as follows:

$$\begin{aligned} x_i(t+1) = x_i(t) + C_1 \cdot (x_i(t) - Lb^b) \\ + C_2 \cdot (x_i(t) - Ub^b) \end{aligned} \tag{7}$$

where C_1 represents a random variable following a normal distribution, and C_2 represents a random vector from the interval (0, 1).

D. Stealing behavior

Within the dung beetle population, certain individuals engage in the act of stealing dung-balls from their counterparts. The studies of DBO operated under the assumption that the global optimum represents the most favorable food source. Consequently, the dung beetles involved in theft would navigate towards this global optimum. The equation used to update the position in response to this behavior is expressed as follows:

$$x_i(t+1) = X^b + S \cdot g \cdot (|x_i(t) - X^*| + |x_i(t) - X^b|) \tag{8}$$

where X^b represents the global optimum; g represents a 1-D random vector following a normal distribution; and 1-D represents a constant.

In the original DBO, the distribution of four populations of dung beetles is predetermined, with 20% for ball-rolling dung beetles, 20% for dung beetles that spawn balls, 25% for small dung beetles, and 35% for dung beetles that steal dung. Fig. 1 illustrates the distribution of the dung beetle populations.

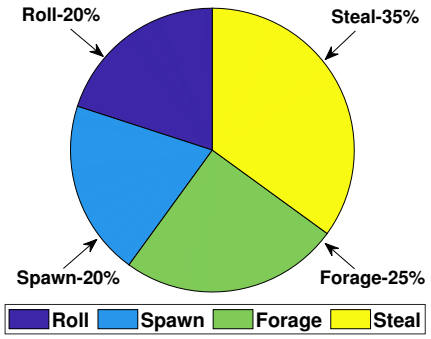


Fig. 1: Distribution map of dung beetle populations

III. Improved Dung Beetle Optimization Algorithm

While DBO algorithm outperforms many other swarm intelligence optimization algorithms, it still exhibits certain limitations. These include uneven initial population distribution, which hampers the global search capability of the algorithm; reliance on a single search method, which can constrain search performance; and a fixed proportion of ball-rolling and stealing dung beetles, which limits the balance between exploration and exploitation. To overcome these challenges, this study proposes three improvement strategies aimed at enhancing the optimization capability of DBO algorithm.

A. Population initialization strategy based on PWLCM

In swarm intelligence optimization algorithms, the quality of the initial population plays a crucial role in enhancing the convergence speed and global search capability of the algorithm [30]. However, DBO method employs a random approach for population initialization, resulting in an uneven distribution of populations and constraining the search capacity of DBO. To address this limitation, chaotic maps have been extensively utilized to enhance the population diversity of swarm intelligence optimization algorithms, leveraging their inherent randomness and ergodicity. Notably, chaotic maps such as Tent [31], Logistic [32], and Sine [33] have been commonly applied in this domain. Among these maps, PWLCM stands out for its superior randomness and ergodicity, attributed to its segmentation mechanism within the mapping space [34]. Therefore, this study utilizes PWLCM for population initialization. PWLCM is mathematically defined by the following equation:

$$f(i+1) = \begin{cases} \frac{f(i)}{p}, 0 \le f(i) < p \\ \frac{f(i)-p}{0.5-p}, p \le f(i) < 0.5 \\ \frac{1-p-f(i)}{0.5-p}, 0.5 \le f(i) < 1-p \\ \frac{1-f(i)}{p}, 1-p \le f(i) < 1 \end{cases} \tag{9}$$

where p represents the control parameter of the chaotic mapping, $p \in [0, 0.5]$.

When the values of p and $f(0)$ are determined, it is possible to generate a random sequence within the interval

(0,1) through iteration. The initialization procedure for the dung beetle population according to PWLCM is outlined as follows:

(1) The control parameters p and $f(0)$ are initialized, where $p \in [0, 0.5]$ and $f(0) \in [0, 1]$.

(2) PWLCM sequence $z_{ij}(i = 1, 2, ..., N; j = 1, 2, ..., D)$ is generated through Eq. (9), where N is the number of populations and D is the dimension of optimization problem.

(3) Transform the resulting disorderly sequence into the solution space by utilizing Eq. (10):

$$x_{ij} = lb_j + (ub_j - lb_j) \cdot z_{ij} \tag{10}$$

where $x_{ij}(i = 1, 2, ..., N; j = 1, 2, ..., D)$ represents the coordinate of the *i*-th dung beetle in the *j*-th dimension; lb_j and ub_j represent the upper and lower bounds of the *j*-th dimension coordinate.

Fig. 2 depicts the initial positions of 30 dung beetle individuals acquired through PWLCM initialization in a 2D space.

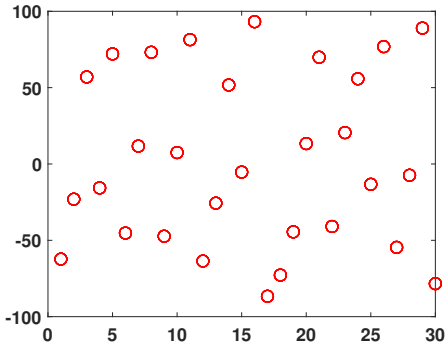


Fig. 2: Populations initialization using PWLCM in 2D space

In Fig. 2, the populations produced by PWLCM exhibit an even distribution in the solution space. This characteristic is advantageous as it enhances the quality of the initial populations.

B. Modeling the Disorienting behavior of dung beetles

In Eq. (3), the ball-rolling dung beetle depends on light source guidance to navigate and performs rectilinear motion. The study does not address the movement patterns of dung beetles when there is no light source to guide them. In natural settings, without light cues, dung beetles struggle to orient themselves accurately, leading to erratic movements such as walking in curved paths or forming elliptical trajectories [35]. The behavior of the ball-rolling dung beetle is defined by Eq. (11):

$$\begin{aligned} x_i(t+1) = \quad & x_i(t) + b \cdot l \cdot \cos(2\pi l) \cdot \mid x_i(t) - worst \mid \\ & + \alpha \cdot k \cdot x_i(t-1) \end{aligned} \tag{11}$$

where $b \cdot l \cdot \cos(2\pi l)$ represents a spiral factor designed to emulate the helix-shaped motion of Disorienting ball-rolling dung beetle. α is utilized to quantify the impact of wind and other environmental variables, typically assigned a value of 1 or -1. The parameter k is set at 0.1.

The spiral factor $b \cdot l \cdot \cos(2\pi l)$ represents an isosceles spiral, with b denoting a constant that determines the pitch of the spiral line. A larger value of b corresponds to a larger pitch. The variable l ranges randomly between -1 and 1. Fig. 3 illustrates 2D trajectories of a disorienting ball-rolling dung beetle for b values of 3, 4, and 5.

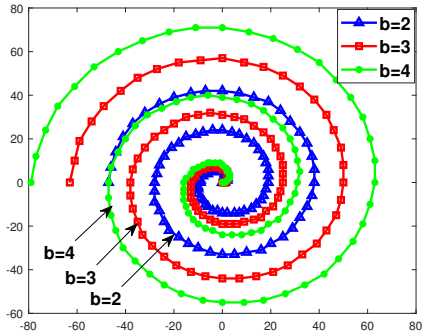


Fig. 3: 2D representation of spiral curve

In Fig. 3, the pitch of the spiral curve widens with an increase in the parameter b. Consequently, the search range expands while the search accuracy diminishes. In this study, a value of 4 was selected for b.

The ball-rolling dung beetle moves in a straight line when guided by a light source, but exhibits disorienting behavior and follows a spiral trajectory in the absence of light. To accurately model this dual behavior, a probability p_1 is introduced, representing the likelihood of selecting either the straight-line motion or the spiral trajectory to update the position of the ball-rolling dung beetle. The corresponding mathematical model is presented as follows:

$$\begin{cases} Eq.(3) & \alpha > p_1 \\ Eq.(11) & \alpha \leq p_1 \end{cases} \tag{12}$$

where α represents a random number within the range of [0, 1]. Given the infrequent occurrence of Disorienting in dung beetles in their natural habitat, the value of p_1 is set at 0.3. The incorporation of the spiral search strategy into the conventional linear search approach enhances the search capabilities of DBO, elevates the probability of discovering the optimal solution, and serves as a deterrent against DBO being trapped in local optimal solutions.

C. Dynamic population strategy

Dung beetles exhibit communal behavior by releasing pheromones to attract additional companions when food is located, aiding in the search for resources and their transportation to the habitat. However, as the food availability within the habitat grows, an increasing number of dung beetles opt to pilfer resources rather than actively search for them. Consequently, the population of dung beetles engaged in rolling dung balls diminishes with successive iterations, while the population of dung beetles engaged in stealing food increases.

To replicate the behavior of dung beetles, it was hypothesized that the combined population of ball-rolling dung beetles and dung beetles engaged in stealing constitutes 55% of the total population. The quantities of ball-rolling dung beetles and stealing dung beetles fluctuate with each iteration.

The mathematical model is outlined as follows:

$$\begin{cases} p_S = p_{TS} - p_T \\ p_{TS} = pop \cdot 55\% \\ p_T = Round(w) \\ w = w_1 + c - \frac{(w_1 - w_2)}{(0.75 + e^{-2(2t-M)/M})} \end{cases} \tag{13}$$

where p_S represents the number of stealing dung beetles; p_{TS} represents the sum of ball-rolling beetles and stealing dung beetles; and p_T represents the number of ball-rolling dung beetles; w represents the adaptive factor; t represents the number of iterations; M represents the maximum number of iterations, c is the adjustment factor; while w_1 and w_2 represent the initial and final values of the ball-rolling dung beetle.

$$w_1 = p_{TS} \times Q_1, w_2 = p_{TS} \times Q_2 \tag{14}$$

where Q_1 and Q_2 represent the initial and final ratios of the rolling dung beetles, respectively. In the scenario where Q_1 equals 80%, Q_2 equals 20%, and c equals 1.25, the graph depicting the variation curve of the number of ball-rolling dung beetles is illustrated in Fig. 4.

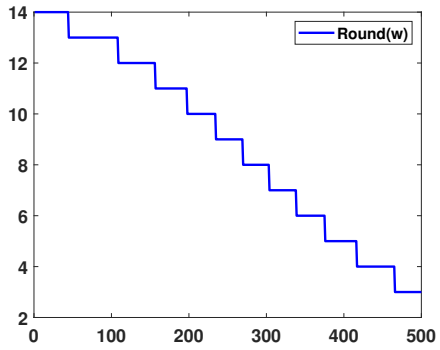


Fig. 4: Convergence curve of algorithm post-rounding

In Fig. 4, the quantity of ball-rolling dung beetles diminishes with successive iterations and stabilizes within a specific range. This range spans approximately $M/(w_1 - w_2)$ iterations. For a more elucidated understanding of the dynamic population strategy, the evolutionary progression of the dung beetle population over 500 iterations is depicted in Fig. 5.

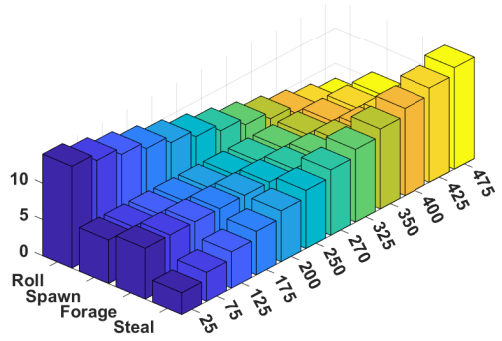


Fig. 5: Population size dynamics across 500 iterations

DBO algorithm balances global and local search performance primarily through the complementary behaviors of ball-rolling and stealing dung beetles. The ball-rolling dung beetles, characterized by their unrestricted range and larger step size, enhance the global search capability of DBO. Conversely, stealing dung beetles focus their search around the global optimum, thereby improving the algorithm's local search performance. By adaptively adjusting the proportions of ball-rolling and stealing dung beetles, DPDBO algorithm effectively enhances global search capability during the early iterations, increasing the likelihood of identifying the global optimum. In later iterations, the algorithm shifts towards a stronger local search focus, thereby improving convergence accuracy and reducing the risk of entrapment in local optima.

Algorithm 1 DPDBO algorithm

Input: Pop, $MaxIter$ and Dim.
Output: Optimal position X_b and its fitness value f_b.

Initialisation of populations by Eq.(10)
while $t \leq T$ max **do**
  Determine the number of ball-rolling dung beetles and stealing dung beetles by Eq.(13)
  if $i =$ ball-rolling dung beetle **then**
    $\alpha = \text{rand}(1), \quad \delta = \text{rand}(1)$
    if $\alpha < 0.7$ **then**
      if $\delta < 0.9$ **then**
        Update the location by Eq.(1).
      else
        Update the location by Eq.(2).
      end if
    else
      Update the location by Eq.(11)
    end if
  end if
  if $i =$ hatching balls **then**
    Update the location by Eq.(4) and Eq.(5)
  end if
  if $i =$ foraging dung beetle **then**
    Update the beetle location by Eq.(6) and Eq.(7).
  end if
  if $i =$ stealing dung beetle **then**
    Update stealing dung beetle location by Eq.(8).
  end if
  $t = t + 1$
end while
return X_b and its fitness f_b

D. complexity analysis

Assuming a population size of N, a maximum of T iterations, a problem dimension of D, and a time complexity of $O(N * D)$ for population initialization. The time complexity for individual fitness calculation and position update is $O(N * T * D)$. The overall time complexity of DBO is $O(N * D + N * T * D)$. The population initialization complexity for DPDBO, as proposed in this study using PWLCM mapping, is $O(N * D)$. Despite potential changes in the number of populations during the iteration process, the total number of populations remains constant, and no new loops are introduced. Consequently, the time

complexity for individual fitness computation and location update in DPDBO is $O(N * T * D)$. Therefore, the time complexity of DPDBO remains $O(N * D + N * T * D)$. In conclusion, DPDBO exhibits a similar order of magnitude as standard DBO and does not compromise time complexity for performance.

IV. Experimental results

CEC2017 [36] and CEC2022 [37] test functions were utilized to assess and compare the performance of DPDBO against other optimization algorithms (DBO [25], SSA [15], HHO [14], WOA [13], GA [6], and Sine Cosine Algorithm (SCA) [22]) in this section. To ensure a fair comparison, all experiments were carried out on MATLAB (2023b) platform on Windows 11, with a fixed population size of 30. To mitigate the impact of randomness, each experiment was independently executed 50 times, with the mean (Ave) and standard deviation (Std) serving as evaluation criteria. Friedman test [38] was employed to analyze the experimental results. Table I outlines the parameters of the seven algorithms.

TABLE I: Compared algorithm parameters

Algorithm	Parameter	Value
DPDBO	k and λ, b, S	0.1, 0.3, 0.5
	w_1, w_2	13, 4
DBO	k and λ, b, S	0.1, 0.3, 0.5
SSA	PD, SD, $R2$	0.2, 0.1, 0.8
HHO	E_0	[-1, 1]
WOA	a	Decreased from 2 to 0
SCA	a	2
GA	Cross, mutation	0.2, 0.1

A. Analysis of the effectiveness of improvement strategies

In this research, three enhanced strategies are suggested. To assess the impact of each strategy, they are integrated with the conventional DBO method, resulting in three upgraded algorithms (DBO1, DBO2, and DBO3). DBO1 incorporates PWLCM strategy into DBO. DBO2 accounts for the disorienting behavior observed in ball-rolling dung beetles, while DBO3 implements the dynamic population strategy.

Six benchmark functions from CEC2005 test functions [39] were selected for analysis. These functions are detailed in Table II, with F1-F3 representing unimodal functions and F4-F6 representing multimodal functions.

Table III illustrates that DBO1, DBO2, DBO3, and DPDBO algorithms exhibit superior convergence accuracy compared to DBO, with DPDBO demonstrating the best performance. These results suggest that each strategy employed in this research contributes to enhancing the balance between exploration and exploitation capabilities. Furthermore, DPDBO proves to be the most effective across the majority of test functions due to its simultaneous utilization of three enhanced strategies.

The convergence curves of DBO, DBO1, DBO2, DBO3, and DPDBO algorithms for the six benchmark functions are illustrated in Fig. 6, Fig. 7. It reveals that DBO1, DBO2, DBO3, and DPDBO algorithms exhibit different convergence patterns during the optimization of the test functions.

TABLE II: CEC2005 benchmark test functions

Type	Function	Mim
Unimodal	$F1(x) = \sum_{i=1}^{n} x_i^2$	0
	$F2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	0
	$F3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	0
Multimodal	$F5(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	0
	$F6(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	
	$F7(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	0

Firstly, DBO1 demonstrates a tendency to exhibit higher initial accuracy and converge rapidly towards the optimal solution (F4 and F5). This can be attributed to PWLCM strategy, which evenly distributes the initial population across the solution space, thereby increasing the probability of discovering the potentially optimal solution. Secondly, the convergence curves of DBO2 consistently lie below those of DBO across all test functions, likely due to the introduction of the disorienting behavior of the ball-rolling dung beetle. This behavior enriches the search method and enhances the likelihood of attaining the optimal solution. Thirdly, DBO3 exhibits a higher convergence speed during later iterations (F1 and F3). This accelerated convergence is possibly a result of the dynamic population strategy, which adaptively adjusts the number of ball-rolling dung beetles and stealing dung beetles to further enhance the algorithm's global and local search capabilities. Fourthly, DPDBO integrates three improvement strategies and showcases the fastest convergence rate and highest accuracy. Furthermore, it illustrates that DPDBO effectively balances the exploration and exploitation aspects of the algorithm, thereby facilitating the discovery of the global optimum.

TABLE III: Performance comparison of different optimization strategies

Algorithm		DBO	DBO1	DBO2	DBO3	DPDBO
F1	mean	8.73E-101	6.88E-112	2.32E-118	1.64E-124	**1.21E-127**
	std	4.78E-100	3.74E-111	1.26E-117	8.98E-124	**5.92E-127**
F2	mean	6.72E-52	9.91E-55	4.37E-60	2.17E-60	**8.52E-63**
	std	3.67E-51	5.41E-54	1.66E-59	1.19E-59	**4.63E-62**
F3	mean	3.29E-49	4.49E-61	7.23E-70	1.83E-87	**6.59E-103**
	std	1.80E-48	2.46E-60	3.96E-69	1.00E-86	**3.61E-102**
F4	mean	-8.34E+03	-1.26E+04	-1.04E+04	-1.11E+04	**-1.26E+04**
	std	1.70E+03	3.83E+00	1.82E+03	1.85E+03	**7.43E-02**
F5	mean	7.30E-01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	2.89E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
F6	mean	4.82E-03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	std	2.00E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**

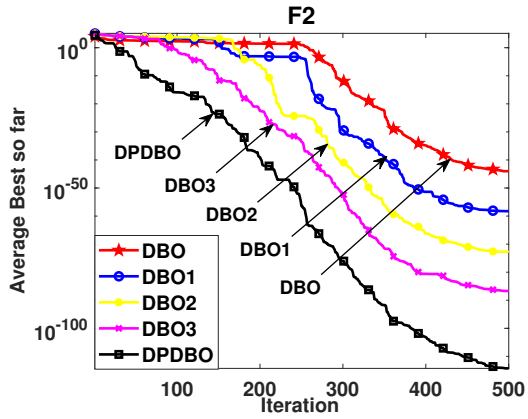


Fig. 6: Distribution map of dung beetle populations

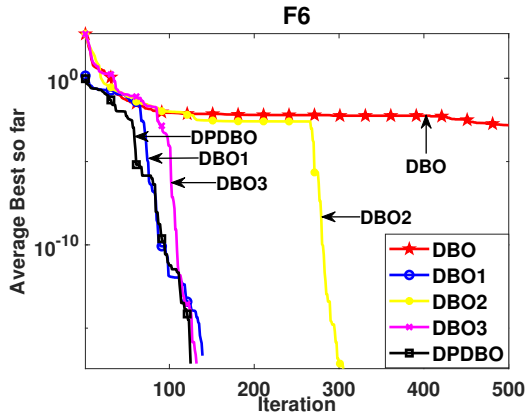


Fig. 7: Distribution map of dung beetle populations

TABLE IV: CEC2017 benchmark test functions

Func	No.	Dim	Range	Min
Unimodal	1	10	[-100,100]	100
Function	3	10	[-100,100]	300
Multimodal	4	10	[-100,100]	400
Functions	5	10	[-100,100]	500
	6	10	[-100,100]	600
	7	10	[-100,100]	700
	8	10	[-100,100]	800
	9	10	[-100,100]	900
	10	10	[-100,100]	1000
Hybrid	11	10	[-100,100]	1100
Functions	12	10	[-100,100]	1200
	13	10	[-100,100]	1300
	14	10	[-100,100]	1400
	15	10	[-100,100]	1500
	16	10	[-100,100]	1600
	17	10	[-100,100]	1700
	18	10	[-100,100]	1800
	19	10	[-100,100]	1900
	20	10	[-100,100]	2000
Composition	21	10	[-100,100]	2100
Functions	22	10	[-100,100]	2200
	23	10	[-100,100]	2300
	24	10	[-100,100]	2400
	25	10	[-100,100]	2500
	26	10	[-100,100]	2600
	27	10	[-100,100]	2700
	28	10	[-100,100]	2800
	29	10	[-100,100]	2900
	30	10	[-100,100]	3000

B. *Results and Analysis of CEC2017 test functions*

To evaluate the efficacy of DPDBO introduced in this study in comparison to other meta-heuristic algorithms (such as DBO, SSA, HHO, WOA, GA, and SCA), simulation experiments were conducted using CEC2017 test functions.

1) Introduction of CEC2017 test functions: Introduction of CEC2017 test functions: CEC2017 test functions consist of a variety of function types. Specifically, F1-F3 are unimodal functions, while F4-F10 are multimodal functions. In this study, F2 is excluded due to its limited testing capability [25]. Additionally, F11-F20 are hybrid functions, and F21-F30 are composite functions. Further details regarding these functions are outlined in Table IV. The benchmark functions are standardized to have dimensions of 10, and the number of iterations is fixed at 1000.

2) Comparison of Test Results: The statistical outcomes for DPDBO, DBO, SSA, HHO, WOA, GA, and SCA after 1000 iterations and 50 independent runs against CEC2017 test function are presented in Table V. It illustrates the mean and standard deviation (std) for each algorithm, with bolded values highlighting the superior results. Additionally, Table V includes a summary at the bottom indicating the number of victories (+), draws (=), and failures (-) for each algorithm. Further discussion on the analysis of these results follows:

The analysis of Table V reveals that the average performance of DPDBO across 21 test functions surpasses that of other algorithms. Additionally, the standard deviation of DPDBO is notably lower than that of the other algorithms across most test functions, indicating superior optimization capabilities and stability. The following analyses are derived from Table V:

The results pertaining to the unimodal functions F1 and F3 indicate that DPDBO exhibits a superior convergence rate. Furthermore, DPDBO demonstrates optimal performance in relation to both mean and standard deviation across these functions. In the case of the multimodal functions F4-F10, DPDBO surpasses other algorithms in performance, with the exception of F4. DPDBO algorithm demonstrates superior performance compared to other algorithms when applied to function F10. For hybrid functions, a competitive relationship is observed between DPDBO and SSA. While DPDBO exhibits slightly weaker performance than SSA on functions F13, F15, F18, and F19, it surpasses other algorithms in overall performance. Experimental results on composition functions (F21-F29) indicate that DPDBO is highly competitive, outperforming other algorithms in all functions except for F25 and F28, where it marginally underperforms compared to HHO and SCA. Notably, DPDBO shows a significant advantage on function F30, emphasizing its effectiveness in solving composite problems.

3) Convergence curves comparison: The convergence curves of various algorithms including DPDBO, DBO, SSA, HHO, WOA, GA, and SCA for CEC2017 test functions are depicted in Fig. 8. DPDBO algorithm demonstrates superior performance compared to the other algorithms across most functions in CEC2017, achieving the highest level of accuracy. Notably, the convergence of DPDBO shows an accelerated trend with increasing iterations. This behavior can be attributed to the utilization of chaotic map initialization in DPDBO, which aids in exploring promising regions of the search space during the initial iterations. Additionally, the

TABLE V: CEC2017 benchmark test functions results

Algorithm		DPDBO	DBO	SSA	HHO	WOA	GA	SCA
F1	Ave	**4.461E+03**	1.101E+04	4.573E+03	2.570E+05	1.235E+05	3.562E+05	5.369E+08
	Std	**3.565E+03**	2.361E+04	4.073E+03	1.458E+05	3.161E+05	7.420E+05	2.346E+08
F3	Ave	**3.000E+02**	**3.000E+02**	**3.000E+02**	3.011E+02	5.872E+02	5.403E+04	9.786E+02
	Std	**4.845E-14**	3.507E-07	7.764E-14	3.567E-01	3.762E+02	2.169E+049	2.993E+02
F4	Ave	4.082E+02	4.206E+02	**4.029E+02**	4.150E+02	4.155E+02	4.616E+02	4.348E+02
	Std	1.601E+01	2.802E+01	**1.014E+01**	2.701E+01	3.148E+01	5.669E+01	1.473E+01
F5	Ave	**5.287E+02**	5.329E+02	5.353E+02	5.445E+02	5.496E+02	5.798E+02	5.422E+02
	Std	1.065E+01	1.198E+01	1.189E+01	1.722E+01	2.235E+01	1.842E+01	**6.870E+00**
F6	Ave	**6.063E+02**	6.087E+02	6.073E+02	6.254E+02	6.338E+02	6.552E+02	6.150E+02
	Std	5.042E+00	5.953E+00	5.047E+00	1.449E+01	1.333E+01	1.615E+01	**3.137E+00**
F7	Ave	**7.443E+02**	7.451E+02	7.717E+02	7.788E+02	7.785E+02	7.812E+02	7.675E+02
	Std	1.321E+01	1.465E+01	2.677E+01	1.951E+01	2.605E+01	2.097E+01	**7.162E+00**
F8	Ave	**8.236E+02**	8.288E+02	8.299E+02	8.302E+02	8.406E+02	8.677E+02	8.379E+02
	Std	**4.932E+00**	1.064E+01	1.108E+01	7.133E+00	1.615E+01	1.514E+01	6.977E+00
F9	Ave	**9.401E+02**	9.602E+02	1.084E+03	1.326E+03	1.292E+03	9.758E+02	9.941E+02
	Std	**5.010E+01**	9.284E+01	2.508E+02	2.351E+02	2.535E+02	1.207E+02	5.288E+01
F10	Ave	**1.443E+03**	1.754E+03	1.905E+03	1.967E+03	2.063E+03	1.805E+03	2.197E+03
	Std	**1.484E+02**	3.126E+02	2.937E+02	2.538E+02	3.102E+02	2.974E+02	2.115E+02
F11	Ave	**1.143E+03**	1.175E+03	1.178E+03	1.161E+03	1.223E+03	6.815E+03	1.188E+03
	Std	**3.201E+01**	6.635E+01	5.566E+01	5.567E+01	7.552E+01	7.978E+03	3.834E+01
F12	Ave	**1.010E+04**	1.952E+06	1.353E+04	1.745E+06	4.530E+06	4.965E+06	6.377E+06
	Std	**1.574E+04**	3.156E+06	3.487E+04	1.819E+06	4.541E+06	5.432E+06	5.046E+06
F13	Ave	9.191E+03	9.576E+03	**6.944E+03**	1.601E+04	1.986E+04	2.201E+04	1.842E+04
	Std	9.194E+03	9.577E+03	**6.941E+03**	1.609E+04	1.983E+04	2.201E+04	1.846E+04
F14	Ave	**1.475E+03**	1.506E+03	1.491E+03	1.516E+03	1.590E+03	7.914E+03	1.577E+03
	Std	2.937E+01	5.003E+01	3.651E+01	**2.788E+01**	3.604E+02	7.166E+03	5.211E+01
F15	Ave	1.688E+03	1.823E+03	**1.577E+03**	2.490E+03	3.582E+03	1.024E+04	2.001E+03
	Std	9.283E+01	3.678E+02	**5.985E+01**	8.443E+02	1.801E+03	8.406E+03	7.135E+02
F16	Ave	1.763E+03	1.752E+03	1.808E+03	1.867E+03	1.802E+03	1.920E+03	**1.693E+03**
	Std	1.112E+02	9.412E+01	1.635E+02	1.429E+02	1.200E+02	1.396E+02	**6.244E+01**
F17	Ave	**1.750E+03**	1.752E+03	1.782E+03	1.778E+03	1.782E+03	1.790E+03	1.775E+03
	Std	2.122E+01	**2.061E+01**	6.915E+01	2.590E+01	4.075E+01	5.235E+01	2.911E+01
F18	Ave	7.932E+03	1.747E+04	**3.930E+03**	1.947E+04	2.252E+04	1.577E+04	6.211E+04
	Std	7.784E+03	1.535E+04	**4.960E+03**	1.281E+04	1.312E+04	1.216E+04	3.823E+04
F19	Ave	2.025E+03	2.155E+03	**1.976E+03**	7.112E+03	2.990E+04	9.173E+03	2.316E+03
	Std	2.855E+02	3.939E+02	**6.101E+01**	5.511E+03	5.496E+04	7.348E+03	3.193E+02
F20	Ave	**2.049E+03**	2.083E+03	2.095E+03	2.156E+03	2.142E+03	2.184E+03	2.089E+03
	Std	3.932E+01	4.530E+01	5.385E+01	6.902E+01	6.126E+01	7.488E+01	**1.514E+01**
F21	Ave	**2.201E+03**	2.205E+03	2.326E+03	2.299E+03	2.324E+03	2.371E+03	2.226E+03
	Std	**1.822E+00**	1.998E+00	4.264E+01	7.260E+01	6.122E+01	4.025E+01	4.001E+01
F22	Ave	**2.298E+03**	2.314E+03	2.303E+03	2.362E+03	2.346E+03	2.492E+03	2.351E+03
	Std	1.845E+01	**9.000E+00**	3.065E+02	2.748E+02	1.672E+02	3.295E+02	3.121E+01
F23	Ave	**2.620E+03**	2.641E+03	2.636E+03	2.667E+03	2.652E+03	2.705E+03	2.655E+03
	Std	1.329E+01	1.573E+01	1.752E+01	2.917E+01	2.414E+01	1.781E+01	**6.215E+00**
F24	Ave	**2.555E+03**	2.602E+03	2.746E+03	2.809E+03	2.784E+03	2.866E+03	2.748E+03
	Std	7.751E+01	1.198E+02	8.433E+01	1.036E+02	**2.443E+01**	5.297E+01	9.765E+01
F25	Ave	2.923E+03	2.942E+03	2.948E+03	2.920E+03	**2.949E+03**	3.013E+03	2.956E+03
	Std	2.102E+01	6.554E+01	2.316E+01	6.538E+01	4.793E+01	6.303E+01	**1.887E+01**
F26	Ave	**2.924E+03**	3.049E+03	3.372E+03	3.403E+03	3.365E+03	3.532E+03	3.055E+03
	Std	1.633E+02	1.212E+02	5.406E+02	5.559E+02	5.421E+02	5.320E+02	**5.691E+01**
F27	Ave	**3.095E+03**	3.103E+03	3.112E+03	3.154E+03	3.147E+03	3.230E+03	3.101E+03
	Std	7.366E+00	6.363E+00	2.802E+01	5.122E+01	4.579E+01	4.438E+01	**1.286E+00**
F28	Ave	3.396E+03	3.307E+03	3.403E+03	3.372E+03	3.411E+03	3.455E+03	**3.243E+03**
	Std	6.799E+01	1.392E+02	1.334E+02	1.577E+02	1.908E+02	1.831E+02	**1.473E+01**
F29	Ave	**3.168E+03**	3.206E+03	3.271E+03	3.282E+03	3.346E+03	3.324E+03	3.219E+03
	Std	4.432E+01	4.676E+01	9.678E+01	7.049E+01	9.233E+01	7.695E+01	**2.257E+01**
F30	Ave	**3,962E+05**	5.371E+05	4.806E+05	1.063E+06	3.966E+05	2.212E+06	4.287E+05
	Std	**3.533E+05**	5.292E+05	6.505E+05	1.558E+06	5.721E+05	2.970E+06	3.073E+05
+/=/-		21/1/7	0/1/28	5/1/23	1/0/28	0/0/29	0/0/29	2/0/27
Friedman Rank		**1.4138**	2.7241	3.2586	4.6897	5.2069	6.4483	4.2586

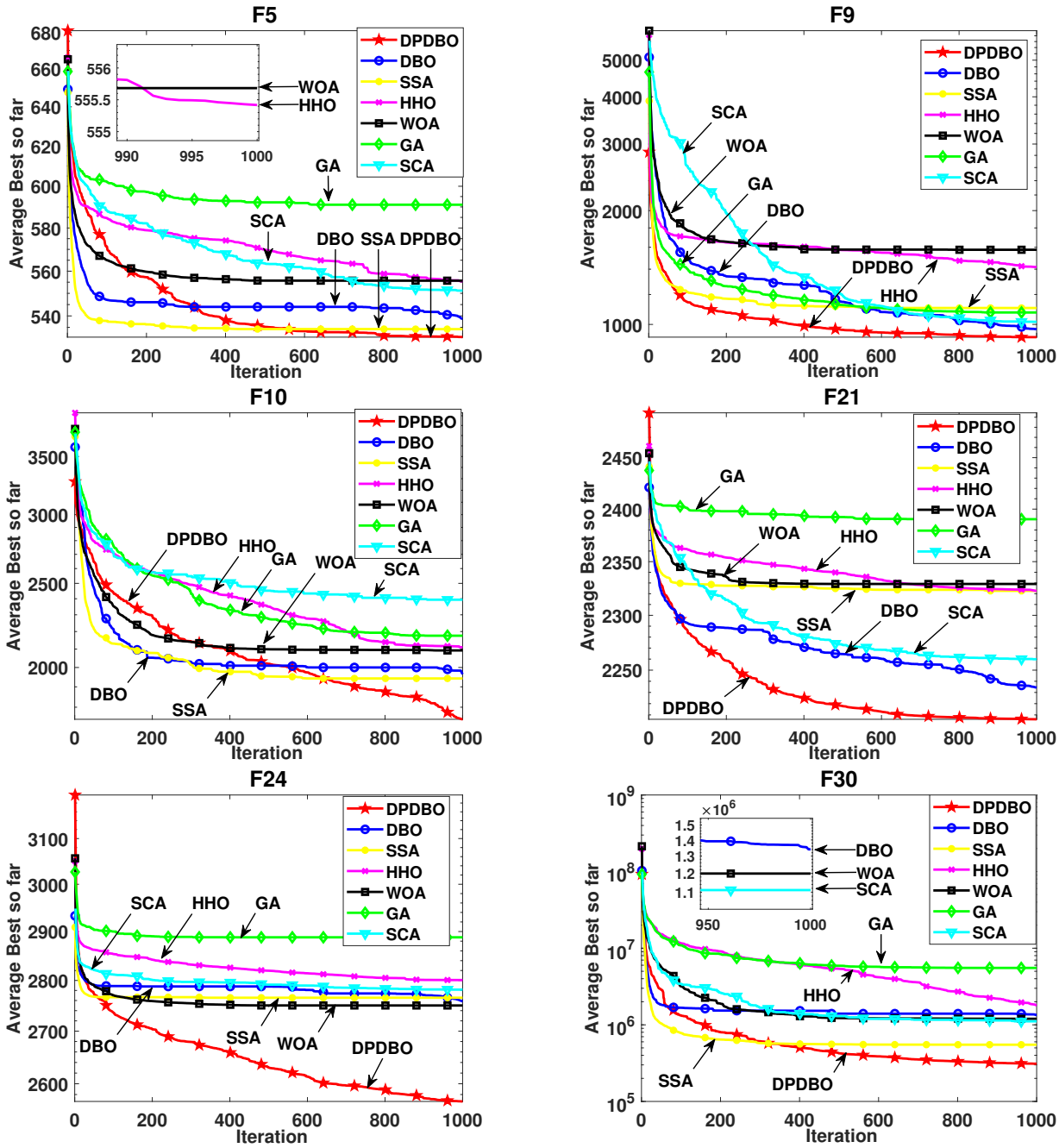


Fig. 8: Comparison of convergence curves for CEC2017 benchmark test functions

dynamic population and hybrid search strategies (spiral and linear search) employed by DPDBO contribute to maintaining search capabilities in the latter stages of the iterations. This superiority is particularly evident in functions F5, F10, F24, and F30.

C. Results and Analysis of the CEC2022 test functions

In this section is to assess the superiority of DPDBO over DBO, SSA, HHO, WOA, GA, and SCA by utilizing CEC2022 test functions.

1) Introduction of CEC2022 test functions: This section provides an overview of CEC2022 test functions, which are characterized by a dimension of 20 and a total of 500 iterations. CEC2022 test functions encompass various types, including unimodal function (F1), elementary functions (F2-F5), hybrid functions (F6-F8), and combined functions (F9-F12). Further details can be found in Table VII.

2) Comparison of Test Results: The results from CEC2022 evaluation are outlined in Table VI. The highlighted data signify the superior results, while the tally of victories (+), ties (=), and shortcomings (-) for each algorithm is displayed at the table's conclusion.

The experiments conducted in this section once again highlight the efficacy of DPDBO algorithm. In general, DBO algorithm exhibits subpar performance on CEC2022 test functions, displaying a search accuracy that is inferior to that of HHO, WOA, GA, and SCA algorithms, and surpassing only the SSA algorithm. However, DPDBO algorithm integrates three enhancement strategies to attain optimal performance across all functions, except for F1 where it demonstrates slightly lower performance compared to SCA

TABLE VI: CEC2022 benchmark test functions results

Algorithm		DPDBO	DBO	SSA	HHO	WOA	GA	SCA
F1	Ave	3.162E+04	3.954E+04	6.844E+04	2.404E+04	3.937E+04	9.199E+04	**2.233E+04**
	Std	9.861E+03	9.142E+03	2.305E+04	7.492E+03	1.677E+04	2.499E+04	**5.624E+03**
F2	Ave	**4.232E+02**	8.734E+02	9.445E+02	4.583E+02	4.721E+02	5.034E+02	4.812E+02
	Std	**2.012E+01**	3.195E+02	5.644E+02	7.785E+01	1.035E+02	4.463E+01	2.948E+01
F3	Ave	**6.113E+02**	6.385E+02	6.488E+02	6.412E+02	6.377E+02	6.634E+02	6.230E+02
	Std	**8.041E+00**	8.762E+00	1.865E+01	9.589E+00	1.193E+01	1.505E+01	4.117E+00
F4	Ave	**8.261E+02**	8.398E+02	8.633E+02	8.284E+02	8.357E+02	8.699E+02	8.441E+02
	Std	**6.443E+00**	8.498E+00	1.475E+01	7.331E+00	1.216E+01	1.485E+01	7.114E+00
F5	Ave	**1.015E+03**	1.337E+03	1.704E+03	1.432E+03	1.536E+03	1.299E+03	1.084E+03
	Std	**8.836E+01**	2.122E+02	2.355E+02	1.931E+02	2.913E+02	3.464E+02	1.190E+02
F6	Ave	**2.364E+03**	4.292E+06	1.988E+07	6.013E+03	8.022E+03	6.184E+03	4.665E+06
	Std	**7.921E+02**	1.082E+07	3.964E+07	4.285E+03	8.237E+03	4.792E+03	4.233E+06
F7	Ave	**2.036E+03**	2.071E+03	2.128E+03	2.082E+03	2.085E+03	2.124E+03	2.068E+03
	Std	9.372E+00	2.315E+01	4.035E+01	3.387E+01	3.111E+01	3.932E+01	**1.204E+01**
F8	Ave	**2.231E+03**	2.235E+03	2.278E+03	2.243E+03	2.246E+03	2.299E+03	2.242E+03
	Std	**4.166E+00**	6.803E+00	4.711E+01	1.324E+01	1.255E+01	5.478E+01	3.481E+00
F9	Ave	**2.534E+03**	2.702E+03	2.680E+03	2.611E+03	2.602E+03	2.765E+03	2.587E+03
	Std	**1.311E+01**	5.443E+01	7.826E+01	4.166E+01	6.000E+01	5.612E+01	1.777E+01
F10	Ave	**2.581E+03**	2.564E+03	2.805E+03	2.678E+03	2.791E+03	2.765E+03	2.531E+03
	Std	6.188E+01	7.875E+01	3.811E+02	**1.873E+02**	4.377E+02	3.902E+02	5.695E+01
F11	Ave	**2.696E+03**	3.944E+03	3.268E+03	2.802E+03	2.831E+03	3.385E+03	2.787E+03
	Std	**1.122E+01**	3.334E+02	4.876E+02	1.223E+02	1.688E+02	4.702E+02	9.594E+01
F12	Ave	**2.878E+03**	2.903E+03	2.911E+03	2.945E+03	2.907E+03	3.023E+03	2.889E+03
	Std	2.406E+00	2.377E+01	4.041E+01	6.588E+01	3.673E+01	6.744E+01	**1.640E+01**
+/=/-		11/0/1	0/0/12	0/0/12	0/0/12	0/0/12	0/0/12	1/0/11
Friedman Rank		**1.1667**	4.6667	6.4167	2.75	3.8333	6.25	2.9167

TABLE VII: CEC2022 benchmark test functions

Func	No.	Dim	Range	Min
Unimodal Functions	1	20	[-100,100]	300
Basic Functions	2	20	[-100,100]	400
	3	20	[-100,100]	600
	4	20	[-100,100]	800
	5	20	[-100,100]	900
Hybrid Functions	6	20	[-100,100]	1800
	7	20	[-100,100]	2000
	8	20	[-100,100]	2200
Composition Functions	9	20	[-100,100]	2300
	10	20	[-100,100]	2400
	11	20	[-100,100]	2600
	12	20	[-100,100]	2700

algorithm.

3) Convergence curves comparison: Fig. 9 presents the convergence curves of DPDBO, DBO, SSA, HHO, WOA, GA, and SCA algorithms for 20D CEC2022 test functions. It provides insights into the convergence speed, accuracy, and optimization process of each algorithm, allowing for a clearer understanding of their performance.

Therefore, this study presents the following analyses: In comparison to alternative algorithms, DPDBO demonstrates superior ability in balancing exploration and exploitation behaviors, thereby mitigating the risk of the algorithm getting trapped in a local optimum. Initially, the convergence rate of DPDBO aligns closely with that of other algorithms; however, as iterations progress, DPDBO exhibits a notably accelerated convergence rate, consistently achieving optimal results. Notably, on F6, and F10, while other algorithms tend to converge slowly or cease convergence after 250 iterations, potentially leading to local optima, DPDBO continues to explore and ultimately secures superior values.

DPDBO can achieve optimal results for several reasons. Firstly, the utilization of PWLCM strategy during population initialization results in a more even distribution of initial individuals, thereby enhancing the chances of identifying the optimal solution. Secondly, the introduction of disorienting behavior for ball-rolling dung beetles enhances the search process, thereby increasing the probability of result the optimal solution to certain extent. Lastly, the adaptive application of the dynamic population strategy adjusts the quantity of ball-rolling dung beetles and stealing dung beetles, thereby promoting a balance between the exploitation and exploration aspects of DBO.

V. Engineering application problems

A. Sensor coverage issues

This section explores the deployment of wireless sensor nodes within a 2D rectangular area to maximize wireless network coverage. The sensor model utilized is Boolean perception model, which operates under the premise that a monitored point is sensed if it falls within the perception radius of a sensor node. Therefore, the likelihood of a monitored point $X_{m,n}$ being detected by sensor node p^i can be defined as follows:

$$c_{X_{m,n}}(p^i) == \begin{cases} 1, & disp\left(X_{m,n}, p^i\right) \leq R \\ 0, & disp\left(X_{m,n}, p^i\right) > R \end{cases} \tag{15}$$

where $X_{m,n}$ represents the monitored point located at (x_m, y_n), p^i represents the coordinates of the i-th sensor node ($i = 1, 2, \cdots, N$); N represents the number of sensor nodes, $disp\left(X_{m,n}, p^i\right) = \sqrt{\left(x_m - p_x^i\right)^2 + \left(y_n - p_y^i\right)^2}$ represents the distance from the monitored point $X_{m,n}$ to the sensor node; p^i, p_x^i and p_y^i represent the horizontal and vertical coordinates of the i-th sensor node, respectively. Assuming that there are N nodes in the monitored area

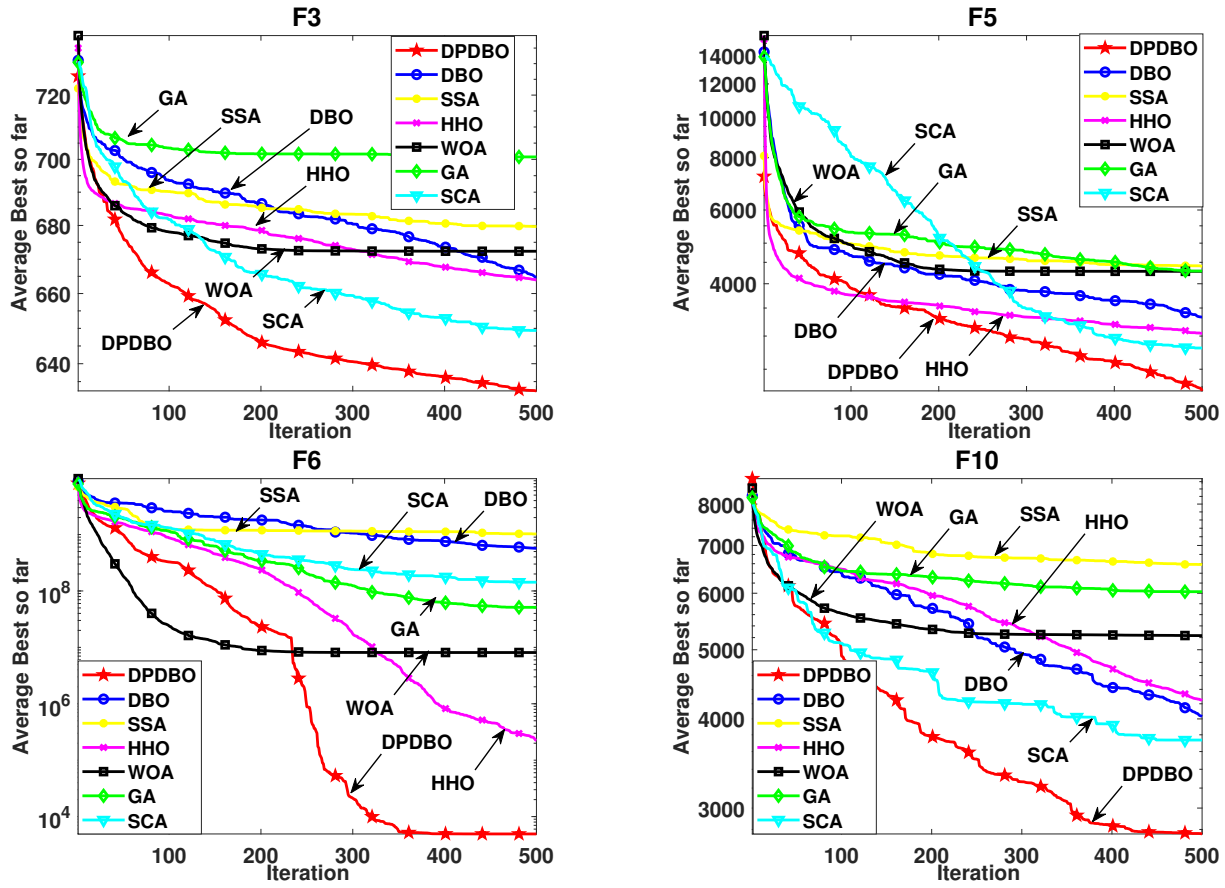


Fig. 9: Comparison of convergence curves for CEC2022 benchmark test functions

with the coordinates of $(p^1, p^2, \cdots, p^N)$, the joint perception probability of all sensor nodes to the monitored point $X_{m,n}$ can be defined as follows:

$$c_{X_{m,n}}(A) = 1 - \prod_{i=1}^{N} \left(1 - c_{X_{m,n}}\left(p^i\right)\right) \tag{16}$$

where A represents all sensors within the monitored region, the monitored area is assumed to take the form of a rectangle with an area of LW m^2, L and W denote the length and width of the rectangle, respectively. For the ease of computation, the rectangle is partitioned into LW grids of uniform area, with the monitored point positioned at the center of each grid. The coverage area and coverage rate of the sensor network can be determined using Eq. (17) and (18) as follows:

$$C(A) = \sum_{n=1}^{W} \sum_{m=1}^{L} \left(c_{X_{m,n}}(A)\right) \tag{17}$$

$$C_r(A) = C(A)/LW = \sum_{n=1}^{W} \sum_{m=1}^{L} \left(c_{X_{m,n}}(A)\right)/LW \tag{18}$$

The objective of this section is to achieve the highest coverage rate through the selection of sensor node coordinates (A). Therefore, the optimization problem can be defined as:

$$f(A) = \max_{A} \left[C_r(A)\right] \tag{19}$$

where A represents the positions of N sensor nodes in the monitoring.

To validate the effectiveness of DPDBO in addressing the wireless sensor network deployment problem, simulations were conducted across three scenarios involving 30, 35, and 40 nodes. The remaining parameters for the sensor nodes are detailed in Table VIII.

TABLE VIII: Parameter settings for optimization experiments

Parameters	Takes values
Regional Scope	100m*100m
Number of nodes	20,25,30
Perception radius	12m
Communication Radius	20m

DPDBO was executed independently 30 times. Table IX outlines the coverage (best and average) for 20 nodes, 25 nodes, and 30 nodes. The results in Table IX indicate that DPDBO consistently outperforms other algorithms regardless of the wireless sensor network size. The optimal and average coverage rates are approximately 4% and 8% higher than those achieved by DBO, respectively.

Fig. 10 depicts 2D node distribution plots for DPDBO algorithm, and DBO algorithm after 200 iterations, with 20 nodes deployed. DPDBO achieves higher node uilization. For instance, in DBO scheme, sensor nodes tend to cluster, significantly reducing overall coverage. In contrast, DPDBO scheme exhibits a more dispersed node arrangement, minimizing overlap and thereby increasing the effective coverage area. Additionally, DBO algorithm, at times, arranges multi-

TABLE IX: Sensor coverage results across different node configurations

Algorithm		DPDBO	DBO	SSA	HHO	WOA	GA	SCA
20Nodes	mean	**0.80872**	0.73702	0.73216	0.74275	0.71484	0.66816	0.66094
	best	**0.8376**	0.79	0.7916	0.7748	0.7512	0.706	0.696
25Nodes	mean	**0.86789**	0.79542	0.78837	0.79024	0.76982	0.72071	0.70662
	best	**0.888**	0.8452	0.836	0.8204	0.8108	0.7516	0.7564
30Nodes	mean	**0.87014**	0.80278	0.79208	0.79916	0.77792	0.71809	0.70363
	best	**0.8956**	0.8492	0.844	0.8284	0.8132	0.7556	0.7312

TABLE X: Results for tension/compression spring design optimization

Algorithm	DPDBO	DBO	SSA	HHO	WOA	GA	SCA
Best	**0.012665**	0.012769	0.012799	0.012732	0.012765	0.013195	0.012789
Ave	**0.013111**	0.014094	0.014799	0.01335	.013872	0.016625	0.013235
x1	0.0506705	0.05	0.0530632	0.053625	0.0515924	0.05	0.0519108
x2	0.332704	0.316827	0.25	0.4051	0.354396	0.310463	0.361104
x3	12.8493	14.1209	13.0087	8.9293	11.4264	15	11.1426

TABLE XI: Optimization results for welded beam design

Algorithm	DPDBO	DBO	SSA	HHO	WOA	GA	SCA
Best	**1.670411**	1.703598	1.717704	1.773721	1.724646	1.934308	1.740212
Ave	**1.715**	1.7974	2.622	2.0644	2.9001	2.2724	1.8419
x1	0.19867	0.17943	0.18882	0.20238	0.20175	0.125	0.19017
x2	3.341	3.7836	3.7975	3.8452	3.6107	5.55105	3.738
x3	9.1925	9.1774	9.1735	9.0162	9.1112	10	9.1687
x4	0.19884	0.19982	0.19965	0.20666	0.20237	0.195463	0.20333

ple nodes along the boundary, leading to inefficient coverage and suboptimal solutions. However, the DPDBO algorithm avoids such inefficiencies, further validating its enhanced global search ability and capacity to achieve more optimal solutions.

B. Tension/compression spring design

The primary objective of this issue is to reduce the weight of the tension or compression spring [40]. The optimization procedure must adhere to various constraints, including shear stress, resonance frequency, and minimal deflection.

There are three variables that impact the weight of the spring: the diameter of the wire (x_1), the average of the coil diameters (x_2), and the number of active coils (x_3). The problem is defined and described as follows:

Minimize :

$$f(\bar{x}) = x_1^2 x_2 (2 + x_3)$$

subject to :

$$g_1(\bar{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0,$$

$$g_2(\bar{x}) = \frac{4x_2^2 - x_1 x_2}{12566(x_2 x_1^3 - 4)} + \frac{1}{5108 x_1^2} - 1 \leq 0,$$

$$g_3(\bar{x}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0,$$

$$g_4(\bar{x}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0,$$

with bounds :

$$0.05 \leq x_1 \leq 2.00$$

$$0.25 \leq x_2 \leq 1.30$$

$$2.00 \leq x_3 \leq 15.0$$

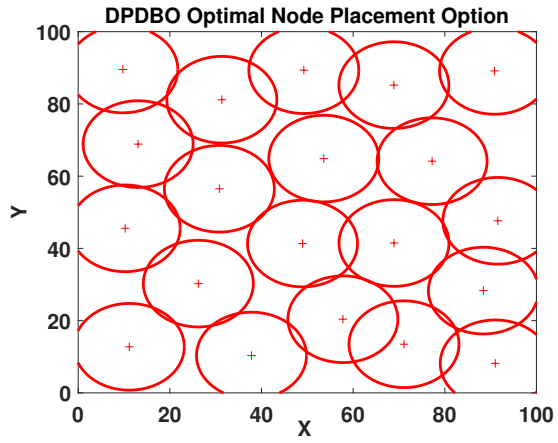
The experimental results of DPDBO and other comparative algorithms in the design of welded beams are summarized in Table X and Fig. 11. The results indicate that DPDBO consistently outperforms the other algorithms in both average and optimal values across 30 runs, and it converges to the optimal value near the early stage, showing a faster convergence speed. DPDBO identifies the optimal solution for the problem, demonstrating its robust global search capability in addressing real-world challenges.

C. Welded beam design

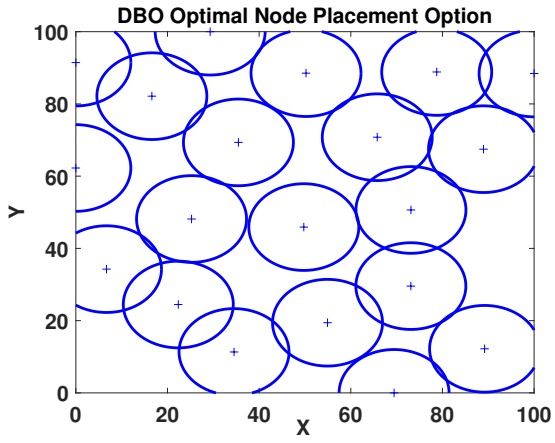
The objective of the welded beam design problem is to minimize construction expenses [40]. This optimization challenge encompasses five limitations: shear stress (δ), beam bending stress (τ), beam end deflection (σ), and bar buckling load (P_c). It involves four variables: weld thickness (x_1), length of the attached bar segment (x_2), bar height (x_3), and bar thickness (x_4). The optimization problem can be expressed as follows:

Minimize :

$$f(\bar{x}) = 0.04811 x_3 x_4 (x_2 + 14) + 1.10471 x_1^2 x_2$$

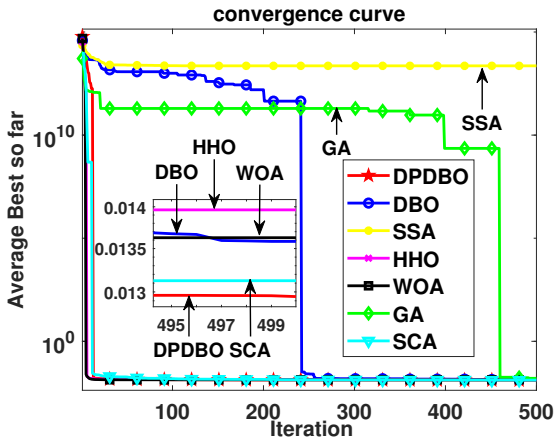


(a)DPDBO

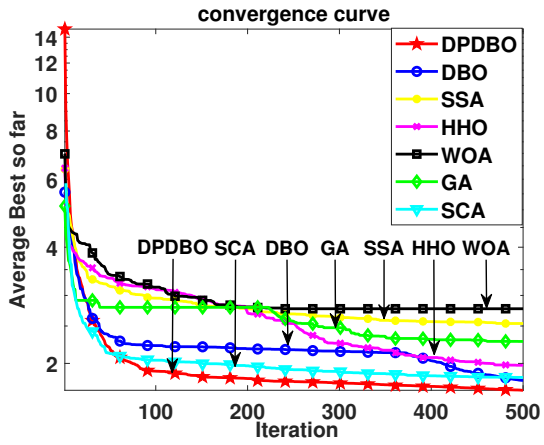


(b)DBO

Fig. 10: Node arrangement simulation diagram



(a)Spring design optimization



(b)Welded beam design

Fig. 11: convergence plot

subject to :

$$
\begin{aligned}
&g_1(\bar{x}) = x_1 - x_4 \leq 0,\\
&g_2(\bar{x}) = \delta(\bar{x}) - \delta_{max} \leq 0,\\
&g_3(\bar{x}) = P \leq P_c(\bar{x})\\
&g_4(\bar{x}) = \tau_{max} \geq \tau(\bar{x}),\\
&g_5(\bar{x}) = \sigma(\bar{x}) - \sigma_{max} \leq 0,
\end{aligned}
$$

where :

$$
\tau = \sqrt{\tau'^2 + \tau''^2 + 2\tau'\tau''\frac{x_2}{2R}}, \tau'' = \frac{RM}{J}, \tau' = \frac{P}{\sqrt{2}x_2x_1},
$$

$$
M = p\left(\frac{x_2}{2} + L\right), R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},
$$

$$
J = 2\left(\left(\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2\right)\sqrt{2}x_1x_2\right),
$$

$$
\sigma(\bar{x}) = \frac{6PL}{x_4x_3^2}, \delta(\bar{x}) = \frac{6PL^3}{Ex_3^2X_4}, L = 14\text{in}, P = 6000\text{lb},
$$

$$
P_c(\bar{x}) = \frac{4.013Ex_3x_4^3}{6L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right), E = 30.10^6\text{psi}.
$$

with bounds :

$$
0.1 \leq x_3, x_2 \leq 10
$$

$$
0.1 \leq x_4 \leq 2
$$

$$
0.125 \leq x_1 \leq 2
$$

The experimental results of the DPDBO algorithm and other contrast algorithms on the design welded beam are shown in Fig. 11. DPDBO and the other algorithms were executed separately 30 times. The results, including optimal values, average values, and optimal design solutions, are presented in Table XI. The experimental results indicate that DPDBO consistently achieved the highest rankings for both average and optimal values across all 30 runs, thereby demonstrating the method's efficacy in addressing real-world challenges.

VI. Conclusion

This study presents three significant improvements to DBO algorithm. First, the integration of PWLCM for population initialization ensures a more uniform distribution of the initial population, thereby reducing the likelihood of the algorithm becoming trapped in local optima. Second, unlike the classical DBO, the proposed approach incorporates the disorienting behavior of ball-rolling dung beetles in the

absence of a light source, resulting in a hybrid search strategy that combines linear and spiral trajectories. This enhancement enriches DBO's search methods, increases the probability of identifying the global optimum, and improves convergence speed. Third, the dynamic population strategy adaptively allocates individuals as either ball-rolling or stealing dung beetles, guided by a nonlinear factor. During the initial iterations, a higher proportion of ball-rolling dung beetles enhances global search capability, while in later iterations, the increasing number of stealing dung beetles ensures a balanced exploration and exploitation process. The performance of DPDBO algorithm was rigorously evaluated using CEC2017 and CEC2022 benchmark test functions. Comparative results indicate that DPDBO exhibits superior global search capability and robustness relative to other metaheuristic algorithms, including DBO, SSA, HHO, WOA, GA, and SCA. Furthermore, DPDBO was applied to two common engineering problems (e.g., tension spring design and welded beam design) as well as a more complex sensor coverage optimization problem. The results demonstrate that DPDBO effectively addresses these engineering challenges, confirming its practical applicability and enhanced performance.

REFERENCES

[1] Z. Zhao, Z. Li, T. An, and S. Wei, "Improved dandelion optimizer based on elite pool and mirror optimization strategy," Engineering Letters, vol. 32, no. 2, pp. 325–339, 2024.

[2] T. Liu, Z. Yuan, L. Wu, and B. Badami, "An optimal brain tumor detection by convolutional neural network and enhanced sparrow search algorithm," Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, vol. 235, no. 4, pp. 459–469, 2021.

[3] M. Ikterina and D. Ertiningsih, "Comparison results of medical image segmentation with genetic algorithm and particle swarm optimization," IAENG International Journal of Applied Mathematics, vol. 54, no. 4, pp. 753–759, 2024.

[4] M. I. Khaleel, "Efficient job scheduling paradigm based on hybrid sparrow search algorithm and differential evolution optimization for heterogeneous cloud computing platforms," Internet of Things, vol. 22, p. 100697, 2023.

[5] Z. Xu and X. Zhang, "An improve grey wolf optimizer algorithm for traveling salesman problems," IAENG International Journal of Computer Science, vol. 51, no. 6, pp. 602–612, 2024.

[6] J. H. Holland, "Genetic algorithms and the optimal allocation of trials," SIAM Journal on Computing, vol. 2, no. 2, pp. 88–105, 1973.

[7] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," Science, vol. 220, no. 4598, pp. 671–680, 1983.

[8] M.-Y. Cheng and M. N. Sholeh, "Optical microscope algorithm: A new metaheuristic inspired by microscope magnification for solving engineering optimization problems," Knowledge-Based Systems, vol. 279, p. 110939, 2023.

[9] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," IEEE Computational Intelligence Magazine, vol. 1, no. 4, pp. 28–39, 2006.

[10] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95-International Conference on Neural Networks, vol. 4, pp. 1942–1948, ieee, 1995.

[11] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm," Journal of Global Optimization, vol. 39, pp. 459–471, 2007.

[12] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," Advances in Engineering Software, vol. 69, pp. 46–61, 2014.

[13] S. Mirjalili and A. Lewis, "The whale optimization algorithm," Advances in Engineering Software, vol. 95, pp. 51–67, 2016.

[14] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," Future Generation Computer Systems, vol. 97, pp. 849–872, 2019.

[15] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," Systems Science & Control Engineering, vol. 8, no. 1, pp. 22–34, 2020.

[16] F. A. Hashim and A. G. Hussien, "Snake optimizer: A novel meta-heuristic optimization algorithm," Knowledge-Based Systems, vol. 242, p. 108320, 2022.

[17] A. Seyyedabbasi and F. Kiani, "Sand cat swarm optimization: A nature-inspired algorithm to solve global optimization problems," Engineering with Computers, vol. 39, no. 4, pp. 2627–2651, 2023.

[18] M. Abdel-Basset, R. Mohamed, M. Jameel, and M. Abouhawwash, "Spider wasp optimizer: a novel meta-heuristic optimization algorithm," Artificial Intelligence Review, vol. 56, no. 10, pp. 11675–11738, 2023.

[19] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," Knowledge-Based Systems, vol. 89, pp. 228–249, 2015.

[20] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm," Computers & Structures, vol. 169, pp. 1–12, 2016.

[21] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, "African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems," Computers & Industrial Engineering, vol. 158, p. 107408, 2021.

[22] S. Mirjalili, "Sca: a sine cosine algorithm for solving optimization problems," Knowledge-Based Systems, vol. 96, pp. 120–133, 2016.

[23] P. D. Kusuma, "Best couple algorithm: A new metaheuristic with two types of equal size swarm splits," IAENG International Journal of Applied Mathematics, vol. 54, no. 8, pp. 1615–1623, 2024.

[24] P. D. Kusuma, "Group better-worse algorithm: A superior swarm-based metaheuristic embedded with jump search," IAENG International Journal of Applied Mathematics, vol. 54, no. 4, pp. 614–622, 2024.

[25] 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.

[26] R. Zhang and Y. Zhu, "Predicting the mechanical properties of heat-treated woods using optimization-algorithm-based bpnn," Forests, vol. 14, no. 5, p. 935, 2023.

[27] 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.

[28] H. Zhang, Y. Liu, and H. Chao, "Density peak clustering based on improved dung beetle optimization and mahalanobis metric," Journal of Intelligent & Fuzzy Systems, vol. 45, no. 4, pp. 6179–6191, 2023.

[29] Y. Li, K. Sun, Q. Yao, and L. Wang, "A dual-optimization wind speed forecasting model based on deep learning and improved dung beetle optimization algorithm," Energy, vol. 286, p. 129604, 2024.

[30] J. O. Agushaka and A. E. Ezugwu, "Initialisation approaches for population-based metaheuristic algorithms: a comprehensive review," Applied Sciences, vol. 12, no. 2, p. 896, 2022.

[31] G. Ling, Z. Wang, Y. Shi, J. Wang, Y. Lu, and L. Li, "Membrane fouling prediction based on tent-ssa-bp," Membranes, vol. 12, no. 7, p. 691, 2022.

[32] F. B. Demir, T. Tuncer, and A. F. Kocamaz, "A chaotic optimization method based on logistic-sine map for numerical function optimization," Neural Computing and Applications, vol. 32, no. 5, pp. 14227–14239, 2020.

[33] A. Mansouri and X. Wang, "A novel one-dimensional sine powered chaotic map and its application in a new image encryption scheme," Information Sciences, vol. 520, no. 3, pp. 46–62, 2020.

[34] A. Hasheminejad and M. Rostami, "A novel bit level multiphase algorithm for image encryption based on pwlcm chaotic map," Optik, vol. 184, no. 9, pp. 205–213, 2019.

[35] M. Dacke, E. Baird, M. Byrne, C. H. Scholtz, and E. J. Warrant, "Dung beetles use the milky way for orientation," Current Biology, vol. 23, no. 4, pp. 298–300, 2013.

[36] N. H. Awad, M. Z. Ali, and P. N. Suganthan, "Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving cec2017 benchmark problems," 2017 IEEE Congress on Evolutionary Computation (CEC), vol. 23, no. 3, pp. 372–379, IEEE, 2017.

[37] R. Biedrzycki, J. Arabas, and E. Warchulski, "A version of nl-shade-rsp algorithm with midpoint for cec 2022 single objective bound constrained problems," 2022 IEEE Congress on Evolutionary Computation (CEC), vol. 43, no. 6, pp. 1–8, IEEE, 2022.

[38] D. W. Zimmerman and B. D. Zumbo, "Relative power of the wilcoxon test, the friedman test, and repeated-measures anova on ranks," The Journal of Experimental Education, vol. 62, no. 1, pp. 75–86, 1993.

[39] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization," KanGAL Report, vol. 2005005, no. 5, p. 2005, 2005.

[40] A. Kumar, G. Wu, M. Z. Ali, Q. Luo, R. Mallipeddi, P. N. Suganthan, and S. Das, "A benchmark-suite of real-world constrained multi-objective optimization problems and some baseline results," Swarm and Evolutionary Computation, vol. 67, no. 7, p. 100961, 2021.