Research Reports on Computer Science
https://ojs.wiserpub.com/index.php/RRCS

UNIVERSAL WISER PUBLISHER

Research Article

Improved Driving Training-Based Optimization Algorithm Using Levy Flight and Crowding Distance Techniques

Daniel Kwegyir*, Michael Dugbartey Terkper, Francis Boafo Effah, Emmanuel Kwaku Antoh and Stacy Gyamfuah Lumor

Department of Electrical Engineering, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
Email: daniel.kwegyir@knust.edu.gh

Received: 30 January 2024; **Revised:** 8 April 2024; **Accepted:** 24 April 2024

Abstract: This study presents an improved version of the Driving Training-Based Optimization (DTBO) algorithm, the Improved Driving Training-Based Optimization (IDTBO). The work addresses fundamental issues in selecting drivers and learners for the conventional DTBO, which substantially impact the algorithm's accuracy and convergence speed. Two significant improvements are proposed: including the crowding distance technique for more diverse driver and learner selection and incorporating the Levy Flight distribution for better exploration and local optima avoidance. The IDTBO's performance is evaluated using twelve benchmark functions, including unimodal and high-dimensional multimodal optimization functions. The results indicate that IDTBO performs exceptionally well, with more extraordinary exploitation ability on unimodal functions and consistent achievement of the global optima. The proposed IDTBO demonstrated exceptional exploration capabilities on high-dimensional multimodal functions and performed competitively with other algorithms in the literature. From six functions, the IDTBO obtained zero optimal values. Again, the rate of convergence analysis demonstrates that IDTBO finds optimal solutions in fewer iterations, demonstrating its capacity to balance exploration and exploitation. To assess the strength of the IDTBO in solving real-world problems, the improved DTBO is further tested on two practical benchmark engineering problems. The IDTBO again produced a competitive performance against other algorithms in the literature. The study shows that IDTBO is a valuable metaheuristic algorithm that can tackle various real-world optimization problems.

***Keywords*:** optimization, metaheuristics algorithms, driving training-based optimization, crowding distance, levy flight

1. Introduction

Metaheuristic algorithms developed from the inspiration of nature and other phenomena, have proven vital in solving multifaceted engineering optimization problems [1,2]. Their inherent flexibility, coupled with their wide-ranging search strategies, have established them as the preferred solution tools for many optimization challenges [3]. At a high level, these algorithms can be categorized into trajectory-based and population-based techniques. The trajectory-based techniques, encompassing algorithms such as simulated annealing (SA) deals with refining a single solution through the algorithms search space for the entire period of solving the problem [4]. In contrast, population-based techniques such as genetic algorithms (GA) and particle swarm optimization (PSO), begin with the search process with several potential solutions, refining them iteratively until an optimal or near-optimal solution emerges [5,6]. In addition, population-based algorithms are further classified into swarm intelligence-based algorithm (SI) and evolutionary algorithm (EA) [2,7]. Evolutionary algorithms are

DOI: https://doi.org/10.37256/rrcs.3120244384

designed based on the concept of species evolving over time to change their characteristics to improve their adaptability to their niche. Swarm intelligence algorithms mimic the stochastic behaviour of animals that live in groups such as birds, fish, etc. to solve optimization problems [2,4,7]. The SIs are also known for their ability to search wider space within the problem boundary to produce quality solution compared to other types of algorithms [8]. Teaching–Learning-Based Optimization (TLBO) draws inspiration from classroom dynamics, emphasizing interactions between students and the teacher. TLBO operates by updating population members through a process akin to teacher training and information exchange among peers [9]. Grey Wolf Optimization (GWO) is influenced by the social behaviour and hierarchical structure observed in grey wolf packs during hunting. GWO employs four types of wolves—alpha, beta, delta, and omega to replicate the leadership hierarchy, with population members updated according to simulations of the hunt's stages: prey search, encircling, and attack [10]. The Whale Optimization Algorithm (WOA) mimics humpback whales' social behaviour and hunting strategy, precisely their bubble-net hunting technique. WOA updates population members through three phases inspired by the whales' hunting behaviour: prey search, encirclement, and bubble-net foraging [11]. The Tunicate Swarm Algorithm (TSA) is developed by simulating tunicates' jet propulsion and swarm behaviour during navigation and foraging. TSA updates its population through four phases: conflict avoidance, neighbour attraction, convergence toward the best agent, and collective swarm behaviour [12]. On the other hand, EA algorithms begin with a set of solutions and continuously evolve them through generations to achieve a solution for a problem being solved.

Driving training-based optimization (DTBO) is a new optimizer recently introduced by [13]. The DTBO's metaheuristic processes are based on the simulation of human leaning activities during driving training. The DTBO has been assessed on benchmark optimization functions and real-world optimization problems and has been found to be effective when compared to some popular algorithms [13]. The DTBO has been applied to solve pressure vessel design problem to minimize the cost of design. Again, it has been used to optimize the design of the welded beam problem to minimize cost of fabrication. Again, the DTBO method has been applied to solve the partial shading (PS) problem quickly and reliably in maximum power point (MPP) detection for PV systems [14]. The DTBO improved the tracking speed and reduced fluctuations in the power output during the tracking period. In these applications, the DTBO algorithm produced the most effective solution compared to popular algorithms in the literature. Although the DTBO, has proven to provide acceptable results for solving many optimization and engineering problems, it has some limitations which require attention and further research to improve its performance. Also, based on the theory of no free lunch, the classical DTBO cannot be assumed to be the best optimizer hence there is room for further improvement in its performance.

This paper proposes an enhanced version of DTBO utilizing Levy Flight and Crowding Distance theory to improve convergence speed and solution quality, addressing its existing limitations. The proposed variant of the DTBO provides researchers with more accurate solutions for various applications. This work, therefore, contributes to the evolution of metaheuristic optimization techniques in the computer science field.

This paper is organized as follows: The original DTBO is described in Section 2, Section 3 presents the proposed improved driving training optimization (IDTBO). Testing of the IDTBO on benchmark functions and the test parameters used for testing the algorithms are presented in Section 4. Results are presented and analysed in Section 5, Section 6 concludes the paper and Section 7 sets the future direction for the research done.

2. Driving Training-Based Optimization

The driving training-based optimization (DTBO) algorithm was designed and implemented using the teaching methodology in which a student learns how to drive under the guidance of a driving instructor [13]. The DTBO algorithm, which is modelled after driving instructors who instruct their pupils in driving skills investigates and utilizes the solution space to identify the best solution to the optimization problems.

2.1 *Inspiration and Main Idea*

The learner-teacher relationship found in driving schools serves as the basis for DTBO. In the DTBO, learners select from a variety of instructors to learn driving skills. The DTBO algorithm uses this technique to improve the way it explores the solution space, which lays the foundation for the design of the algorithm.

2.2 *Mathematical Model*

In the algorithm, drivers and learners are represented as candidate solutions for the problem being solved. Based on learners acquiring skills from various instructors, candidate solutions iteratively improve their search efficacy, thereby reaching convergence faster to obtain optimal solution. The initial positions of the solution space are initialized using (1).

$$X_{ij} = lb_j + r \cdot \left(ub_j - lb_j \right); \; i = 1, 2, \ldots N, \; j = 1, 2, \ldots m \tag{1}$$

where X is the population of DTBO, X_i is the i^{th} candidate solution, X_{ij} is the value of the j^{th} variable determined by the i^{th} candidate solution, N is the size of the population of DTBO, m is the number of problem variables, r is a random number from the interval [0, 1], and are the lower and upper bounds of the j^{th} problem variable, respectively.

The main goal of metaheuristic algorithm is to update existing solutions to better optimize a given objective function, The DTBO algorithm advances its candidate solutions through three strategic phases outlined below.

2.3 *Exploration*

The exploration stage of the DTBO involves the algorithm systematically searching through the solution space to identify potential solutions. The aim is to navigate beyond local optima, which are suboptimal points, and explore different regions of the solution space. This process aids the algorithms to uncover a diverse set of candidate solutions, improving the chances of finding the global optimum. The exploration processes of the DTBO are outlined below.

2.3.1 Training by the Driving Instructor

In this initial phase, learner drivers select their driving instructors based on sets of criteria. This selection is pivotal as it steers the population members i.e., candidate solutions towards diverse regions in the search space, enhancing global search and aiding in the discovery of the optimal solution. The mathematical model for this phase of the DTBO is given in (2). The new position for each member is calculated using (2). The current position X_{ij} is updated if the fitness value of the new position $X_{i,j}^{P1}$ improves the value of the objective function according to (3).

$$X_{i,j}^{P1} = \begin{cases} X_{i,j} + r \cdot \left(DI_{ki,j} - I \cdot X_{i,j} \right), \; F_{DIki}; \\ X_{i,j} + r \cdot \left(X_{i,j} - DI_{ki,j} \right), \, otherwise \end{cases} \tag{2}$$

$$X_i = \begin{cases} X_i^{P1}, \; F_i^{P1} < F_i; \\ X_i, \; otherwise \end{cases} \tag{3}$$

where X_i^{P1} is the new calculated status for the i^{th} candidate solution based on the first phase of the DTBO, $X_{i,j}^{P1}$ is its j^{th} dimension, F_i^{P1} is its objective function value, I is a number randomly selected from the set {1, 2}, r is a random number in the interval [0, 1], where k_i is randomly selected from the set of driving instructors, represents a randomly selected driving instructor to train the i^{th} member, $DI_{ki,j}$ is its j^{th} dimension, and F_{DIki} is its objective function value.

2.3.2 Patterning of the Instructor's Skills by the Learner Driver

Subsequently, candidate solutions are updated according to (4) by integrating the driving skills and strategies learned from the instructors. The current position X_{ij} is updated if the fitness value of the new position $X_{i,j}^{P1}$ improves the value of the objective function according to (5). This mimicry step propels DTBO members to new positions within the search space, further bolstering the exploration capabilities of the algorithm.

$$X_{i,j}^{P2} = P \cdot X_{i,j} + \left(1 - P \right) \cdot DI_{ki,j} \tag{4}$$

$$X_i = \begin{cases} X_i^{P2}, \; F_i^{P2} < F_i; \\ X_i, \; otherwise \end{cases} \tag{5}$$

where X_i^{P2} is the new calculated status for the i^{th} candidate solution based on the second phase of DTBO, $X_{i,j}^{P2}$ is its j^{th} dimension, F_i^{P2} is its objective function value, and P is the patterning index given by (6).

$$P = 0.01 + 0.9\left(1 - \frac{t}{T}\right) \tag{6}$$

where t and T are the iteration count and total number of iterations, respectively.

2.4 *Exploitation*

The exploitation phase of the DTBO uses information identified during the exploration phase to enhance the algorithm's search efficiency and converge towards optimal solutions. By using information from the exploration phase, the DTBO can iteratively refine and improve candidate solutions, contributing to the overall effectiveness of the algorithm in solving complex optimization problems.

2.4.1 Personal Practice by the Learner Driver

The final phase allows the learner drivers to practice their newly acquired skills, honing them to better adapt to optimal driving conditions. This stage determines the DTBO's ability of utilizing the local search space by allowing each participant to have an improved position through a localized search. In this phase, each member of the population is first assigned a random position using (7). If the fitness value of the new location $X_{i,j}^{P1}$ is better than the previous position's fitness values X_{ij} the position is updated according to (8).

$$X_i^{P3} = X_{i,j} + (1 - 2r) \cdot R \cdot \left(1 - \frac{t}{T}\right) \tag{7}$$

$$X_i = \begin{cases} X_i^{P3}, & F_i^{P3} < F_i; \\ X_i, & otherwise \end{cases} \tag{8}$$

In the DTBO algorithm, the objective function value is used to determine the suitability of each potential solution in the search space. The objective value of candidate solution is compared at the end of each iteration, and the solution with the best objective function value—referred to as X(best)—is chosen as the best member of the population. The top-performing member of the population is carried over to the next iteration together with other members that produce values that improve the optimization of the objective function of the problem being solved. The best candidate is found by repeating this selection procedure through each iteration.

3. Proposed Improved Driving Training-Based Optimization (IDTBO)

The choice of learners and drivers in the DTBO metaheuristic process can significantly impact the algorithm's accuracy. In the DTBO algorithm, drivers are the members used to produce new solutions in each iteration. The new solutions that are produced are known as learners. If the drivers are poorly chosen, the DTBO algorithm's chances of finding optimal solutions are minimal. This is because drivers are accustomed to coming up with fresh ideas, and if the drivers lack quality, the new ideas will also be subpar. Furthermore, if the learners are well chosen, there is a higher chance that the DTBO algorithm will converge to a reasonable solution within the search space. This is because the learners update the population of solutions, and if the learners are subpar, there is less chance that the population of solutions will get better. To deal with these issues, the algorithm's metaheuristic process should include operators such as diversity, learner novelty, and the selection of high-fit solutions to guarantee the efficient selection of drivers and learners. Including diversity will guarantee that the DTBO algorithm searches the solution space well to find the optimal solution with the quickest rate of convergence. The selection of drivers (N_{DI}) in the original DTBO algorithm is based on (9), where N, t, and T are the population size, iteration count, and total number of iterations.

$$N_{DI} = 0.1 \times N \times \left(1 - \frac{t}{T}\right) \tag{9}$$

In (9), the number of drivers selected is always equal to or less than 10 percent of the total population. This presents a challenge to the algorithm in achieving quality solutions and getting stuck in local optima since only few fit members of the solution are selected as drivers and many candidate members are left as learners. To deal with this fundamental weakness in the DTBO, a crowding distance technique for selecting drivers and learners is introduced. Crowding distance is a measure of how similar a solution is to its neighbours [15,16]. In crowding distance selection, the drivers and learners are selected such that the average crowding distance of the selected solutions is maximized. The technique helps to ensure that the drivers and learners are diverse. It also helps to prevent the DTBO algorithm from getting stuck in a local optimum. The theory of crowing distance is applied to the DTBO in the three steps below.

Step 1: For each objective being solved, sort all members of the population from the best to worst solution. Select the objective value of the best and worst solution; f_{max} and f_{min} respectively as boundaries.

Step 2: Calculate the crowding distance of each member i within the boundaries according to (10).

$$d_i = \frac{f_{i+1} - f_{i-1}}{f_{max} - f_{min}} \tag{10}$$

In (10), f_{i+1} and f_{i-1} are the objective values of the immediate neighbours of the individual i in the sorted list for the objective function.

Step 3: Select the top half members of the population with the highest crowding distance as drivers and other half as learners in the next iteration of the DTBO. The value of k should be set depending on the problem being solved and the desired degree of diversity. A higher value of k will result in a more diverse population, while a lower value of k will result in a less diverse population.

Furthermore, by initializing the algorithm solution using the Levy Flight distribution, the DTBO's performance is further improved. In the classical DTBO, random distributions as defined by (1) are commonly used to initialize the population. This restricts the scope of the search space and could produce less-than-ideal results. However, the algorithm can search outside the immediate search space of initial solutions and potentially find better solutions by using Levy flight to initialize the solution, which generates huge leaps in the search space [17,18,19]. Furthermore, Levy Flight keeps the algorithm from prematurely converging to poor solutions by avoiding local optima. In the modification, (11) is used to generate an initial random solution. Next, using (12), the levy flight is used to scale down the solution.

$$x_{ij} = lb_j + \left(ub_j - lb_j\right) \times rand\left(A, D\right) \tag{11}$$

$$X_{ij} = x_{ij} + rand\left(A, D\right) \times levy\left(A\right) \tag{12}$$

In (11) and (12), A and D are the number of search agents and the dimension of the problem being solved respectively. The steps involved in applying the Levy Flight distribution are outlined below.

Step 1: Calculate the step size for the levy distribution using (13). This is to ensure that the step size is appropriate for the dimensionality of the problem.

$$step\ size = \frac{1}{\sqrt{D}} \tag{13}$$

Step 2: Generate a random number from Cauchy distribution as Cauchy number. In this work, the standard probability distribution function (PDF) with location parameter 0 and scale parameter 1 defined according to (14) is used.

$$f\left(x\right) = \frac{1}{n\left(1 + x^2\right)} \tag{14}$$

Step 3: Calculate the levy number using (15).

$$levy\ number = step\ size \times Cauchy\ number \tag{15}$$

The pseudocode for implementing the proposed improved driving training-based optimization algorithm (IDTBO) is shown in Algorithm 1.

Algorithm 1. Implementation of IDTBO

Start DTBO

Input: The optimization problem information.
Adjust N and T.
Initialize a random solution of the DTBO using (11).
Calculate the step size of the levy distribution using (13)
Generate random number from Cauchy distribution as Cauchy number using (14)
Calculate the Levy number using (15) and use it to scale down the random solution generated using (12)
Evaluate the objective function using the new scaled random solution.
For t = 1 to T
    For i = 1 to N
    Phase 1: Training by the driving instructor (exploration).
        Determine the driving instructor matrix based on the crowding distance method.
        Select a driving instructor at random from the matrix DI.
        Calculate the new position for the i^{th} DTBO member using (2).
        Update the position of the i^{th} DTBO member using (3).
    Phase 2: Learner driver patterning from instructor skills (exploration).
        Calculate the patterning index P using (6).
        Calculate a new position of the i^{th} DTBO member using (4).
        Update the position of the i^{th} DTBO member using (3).
    Phase 3: Personal practice (exploitation)
        Calculate the new position for the i^{th} DTBO member using (7).
        Update the position of the i^{th} DTBO member using (8).
    End.
    Update the best-found candidate solution.
End.
Output: The best candidate solution obtained by DTBO.

End DTBO

4. Testing of Proposed Improved Driving Training-Based Optimization

The proposed improved driving training-based optimization algorithm with Levy Flight and crowding distance (IDTBO) is tested on twelve benchmark functions selected from IEEE CEC2017 benchmark functions. The details of the functions are given in Table 1 [20]. Functions 1–6 are unimodal functions and functions F8–F12 are high dimensional (HD) multimodal functions. The IDTBO is compared against the original DTBO and six (6) popular algorithms in the literature such as Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), Particle Swarm Optimization (PSO), Teaching-Learning Based Optimization (TLBO), Tunicate Swarm Algorithm (TSA) and Wale Optimization Algorithm (WOA). The parameters of these algorithms are shown in Table 2.

Table 1. Benchmark Functions

Function	Search Range	Optimal Value	Dimensions
F1	[-100, 100]	0	30
F2	[-10, 10]	0	30
F3	[-100, 100]	0	30
F4	[-100, 100]	0	30
F5	[-100, 100]	0	30
F6	[-1.28, 1.28]	0	30
F7	[-500, 500]	-419.9829 × Dim	30
F8	[-5.12, 5.12]	0	30
F9	[-32, 32]	0	30
F10	[-600, 600]	0	30
F11	[-50, 50]	0	30
F12	[-50, 50]	0	30

Table 2. Values to Control Parameters of Competitor Algorithms [21]

Algorithm	Parameter	Value
GA	Type	Real coded
	Mutation	Gaussian (Probability = 0.05)
	Crossover	Whole arithmetic (Probability = 0.8)
	Selection	Roulette wheel (Proportionate)
GTO	A	0.4
	H	0.2
GWO	Convergence parameter (a)	a: Linear reduction from 2 to 0.
PSO	Velocity limit	10% of dimension range
	Topology	Fully connected
	Inertia weight	Linear reduction from 0.9 to 0.1.
	Cognitive and social constant	$(C_1, C_2) = (2, 2)$
TLBO	random number	rand is a random number from interval [0, 1].
	T_F: teaching factor	T_F = round [(1 + *rand*)]
TSA	c1, c2, c3	random numbers lie in the interval [0, 1].
	P_{min}	1
	P_{max}	4
WOA	l is a random number in [−1, 1]	
	r is a random vector in [0, 1]	
	Convergence parameter (a)	a: Linear reduction from 2 to 0.

The comparison is done based on the optimum value, mean absolute error (MAE) denoted by (16) [22,23] and convergence speed denoted by the number of iterations to obtain convergence and rank.

$$\text{MAE} = \frac{1}{S}\sum_{i=1}^{S}\left|X_{oi} - X_i\right| \tag{16}$$

In (16), S is the number of cost samples, X_{oi} is the benchmark value of the test function and X_i is the computed optimum value.

To evaluate the performance of the optimization algorithms, each of the competing algorithms, as well as the proposed IDTBO is run for 1000 iterations in 20 independent runs on the objective functions. All algorithms were run on the same computer, an Intel ® Core™ i7-7500U with CPU @ 2.70 GHz 2.90GHz and 12GB RAM using MATLAB simulation software. The mean values are calculated based on (16).

5. Results and Analysis

This study comprehensively evaluates the performance of the proposed Improved Driving Training Optimization Algorithm (IDTBO) against the original DTBO and six established algorithms from the literature: Genetic Algorithm (GA), Grey Wolf Optimizer (GWO), Particle Swarm Optimization (PSO), Teaching-Learning-Based Optimization (TLBO), Tunicate Swarm Algorithm (TSA), and Whale Optimization Algorithm (WOA). The algorithms are compared based on their ability to find optimal solutions and achieve convergence across twelve diverse test functions categorized into unimodal and high-dimensional multimodal landscapes.

Two key metrics are used for comparison:

- **Optimum values**: The best solution achieved by each algorithm on each function, indicating its ability to locate the global optimum.
- **Mean convergence values**: The average value across multiple runs where the algorithm reaches its stopping criterion, showcasing its convergence speed and stability.

5.1 *Evaluation of Exploitation Ability on Unimodal Functions*

Six unimodal benchmark functions (F1–F6) were used to assess the algorithms' performance in terms of their exploitation potential. These functions are intended to evaluate how well the algorithm finds the global optimal value of zero. The optimum values and mean absolute errors are shown in Table 3 and Table 4 respectively.

Table 3. Optimum Values

Function	GA	GWO	PSO	TLBO	TSA	WOA	IDTBO	DTBO
F1	3.71E+02	3.53E-60	1.18E-79	5.32E-179	2.39E-49	2.97E-175	**0.00E+00**	**0.00E+00**
F2	1.50E-02	1.01E-34	1.61E-09	2.99E-89	2.38E-28	4.96E-106	**0.00E+00**	**0.00E+00**
F3	3.18E+03	1.16E-17	1.06E-03	2.28E-38	9.93E-17	2.33E+04	**8.34E-206**	6.65E-112
F4	2.83E+01	1.11E-14	7.06E-03	1.45E-71	6.46E-03	9.13E+01	**0.00E+00**	**0.00E+00**
F5	2.16E+03	7.55E-01	**0.00E+00**	**0.00E+00**	4.06E+00	5.62E-02	**0.00E+00**	**0.00E+00**
F6	4.04E-02	1.25E-03	5.56E-03	5.32E-04	6.86E-03	9.09E-04	**3.29E-06**	3.14E-05

Table 4. Mean of Convergence Values

Function	GA	GWO	PSO	TLBO	TSA	WOA	IDTBO	DTBO
F1	9.09E+02	3.14E+02	1.42E+02	1.17E+02	6.78E+02	4.30E+02	**7.46E-05**	7.38E+01
F2	6.95E+05	5.78E+09	9.88E+02	1.24E+04	1.50E+08	8.61E+08	**1.60E-04**	1.58E+09
F3	4.54E+03	1.85E+03	4.97E+02	5.09E+02	1.65E+03	6.46E+04	**8.90E+01**	2.34E+02
F4	3.01E+01	1.54E+00	2.13E+00	4.59E-01	4.01E+00	9.13E+01	**1.10E-01**	1.26E-01
F5	2.34E+03	2.97E+02	1.40E+02	8.89E+01	3.90E+02	3.43E+02	**7.11E+01**	7.31E+01
F6	2.87E-01	5.30E-01	8.43E-02	**5.12E-02**	7.18E-01	6.05E-01	9.34E-02	8.95E-02

From table 3, for functions F1, F2, F4, and F5, IDTBO achieved four zero optimal values, demonstrating remarkable performance. Even though IDTBO was unable to achieve the optimal value of zero for functions F3 and F6, it was still able to get the lowest values when compared to other algorithms, demonstrating its superior exploitation capacity.

Table 4 further supports this, showing that IDTBO obtained the lowest mean values for functions (F1–F5).

Table 5 shows the ranking of algorithms according to the mean absolute error (MAE) of their optimum values in Table 3 to measure the accuracy of exploitation. IDTBO and DTBO ranked first demonstrating their accuracy in finding the optimum values. These findings support the competitiveness of IDTBO in solving the benchmark functions by highlighting its advantage over the other metaheuristic algorithms in exploitation accuracy.

Table 5. Ranking of Algorithms based on MAE for Unimodal Functions

Algorithm	MAE	Rank
GA	820.6421	7
GWO	0.1081	5
PSO	0.0020	4
TLBO	0.0001	3
TSA	0.5821	6
WOA	3336.6086	8
IDTBO	**0.0000**	**1**
DTBO	**0.0000**	**1**

5.2 *Evaluation of Exploration Ability on High-Dimensional Multimodal Functions*

Six high-dimensional multimodal functions (F7–F12) were used to evaluate the algorithms' ability to solve complex benchmark functions with several local optima. These functions present an exploration issue, requiring algorithms to properly strike a balance between exploitation and exploration to prevent premature convergence. The best values that each algorithm was able to achieve on these functions are listed in Table 6.

Table 6. Optimum Values

Function	GA	GWO	PSO	TLBO	TSA	WOA	IDTBO	DTBO
F7	-4.82E+03	-6.59E+03	-4.71E+03	-7.73E+03	-5.30E+03	-1.26E+04	**-1.26E+04**	**-1.26E+04**
F8	2.99E+01	7.88E+00	2.59E+01	5.60E-04	1.45E+02	0.00E+00	**0.00E+00**	**0.00E+00**
F9	1.18E+01	1.51E-14	9.31E-01	7.99E-15	1.51E-14	4.44E-15	**4.44E-16**	**4.44E-16**
F10	1.69E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.53E-02	**0.00E+00**	**0.00E+00**	**0.00E+00**
F11	1.71E+01	4.79E-02	1.71E-32	2.78E-19	1.38E+00	4.40E-03	**1.57E-32**	**1.57E-32**
F12	1.76E+04	6.98E-01	1.20E-31	1.10E-21	2.22E+00	5.01E-01	**1.35E-32**	**1.35E-32**

From table 6, the IDTBO and DTBO together achieved zero optimal values in two functions (F8 and F10). Notably, when compared to other algorithms in the literature, IDTBO and DTBO achieved the best values for all functions. This cumulative result showcases the exceptional exploration capabilities of IDTBO and DTBO.

Table 7 gives the mean value of the thousand optimum values produced for 1000 iterations. Again, the IDTBO produced minimum errors in two functions: F7 and F10.

Table 7. Mean of Convergence Values

Function	GA	GWO	PSO	TLBO	TSA	WOA	IDTBO	DTBO
F7	-4.60E+03	-4.44E+03	-4.62E+03	-6.59E+03	-5.16E+03	-1.25E+04	**-1.25E+04**	**-1.25E+04**
F8	3.84E+01	3.07E+01	3.62E+01	2.33E+01	1.87E+02	1.08E+01	8.40E-01	**7.77E-01**
F9	1.20E+01	3.81E-01	1.47E+00	1.89E-01	1.64E+00	3.54E-01	5.05E-02	**4.51E-02**
F10	1.91E+01	2.84E+00	1.53E+00	1.10E+00	3.69E+00	3.00E+00	**6.25E-01**	6.62E-01
F11	3.65E+05	2.46E+06	1.63E+05	**1.44E+05**	2.81E+06	1.48E+06	6.13E+05	5.77E+05
F12	1.95E+06	3.14E+06	**2.76E+05**	5.76E+05	3.60E+06	4.11E+06	8.59E+05	1.10E+06

Table 8 presents the rankings of the algorithms based on the MAE of the optimum values in Table 6. The proposed IDTBO ranked first, with DTBO showing its superior ability in solving high-dimensional problems.

Table 8. Ranking of Algorithms based on MAE for High Dimensional Multimodal Functions

Algorithm	MAE	Rank
GA	3630.5068	8
GWO	859.8482	6
PSO	1130.5423	7
TLBO	695.8865	4
TSA	1063.8890	5
WOA	4.4039	3
IDTBO	**4.2858**	**1**
DTBO	**4.2858**	**1**

5.3 *Convergence Curves*

The convergence curves visually represent how a metaheuristic algorithm progresses towards optimal solutions over time. Analyzing these curves provides valuable information about the algorithm’s effectiveness and efficiency. The figures 1–12 show the convergence curves of all the algorithms employed on the 12 benchmark functions. The algorithm with the best convergence is expected to reach the optimum value with minimum number of iterations. The IDTBO produced the best convergence curves in five functions: F3, F6, F7, F8, and F9. In the other functions, it performed competitively with the DTBO as well. This demonstrates how well IDTBO balances exploration and exploitation in comparison to other algorithms, highlighting the algorithm's potency and efficiency in both searching and exploiting the search space.

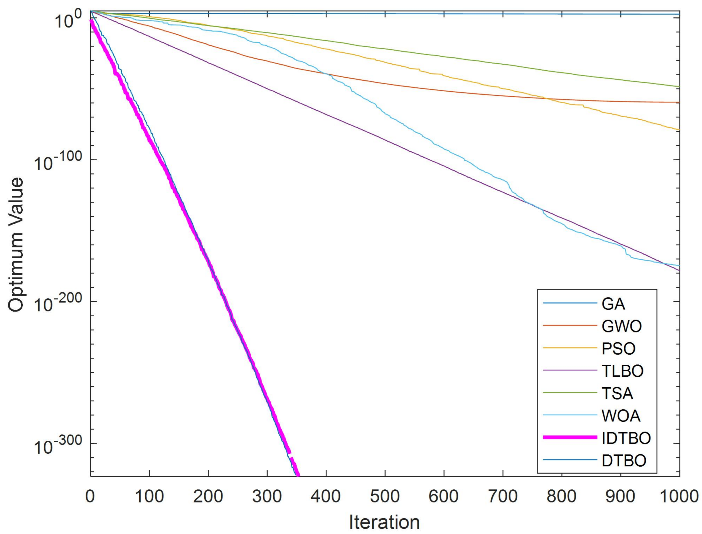

Figure 1. Convergence curve for F1

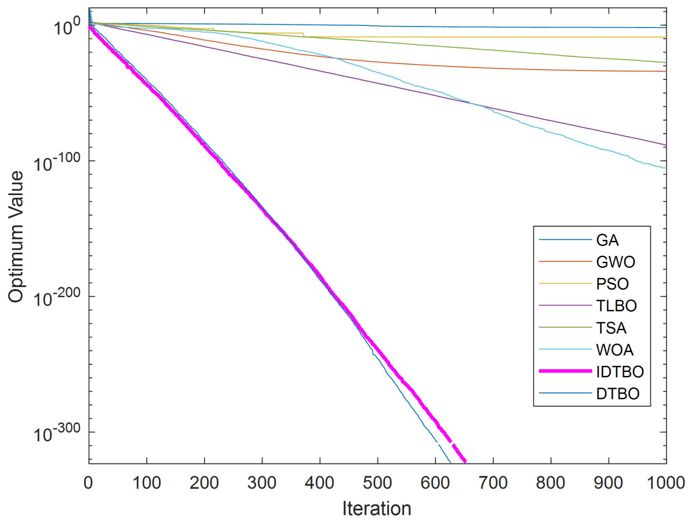

Figure 2. Convergence curve for F2

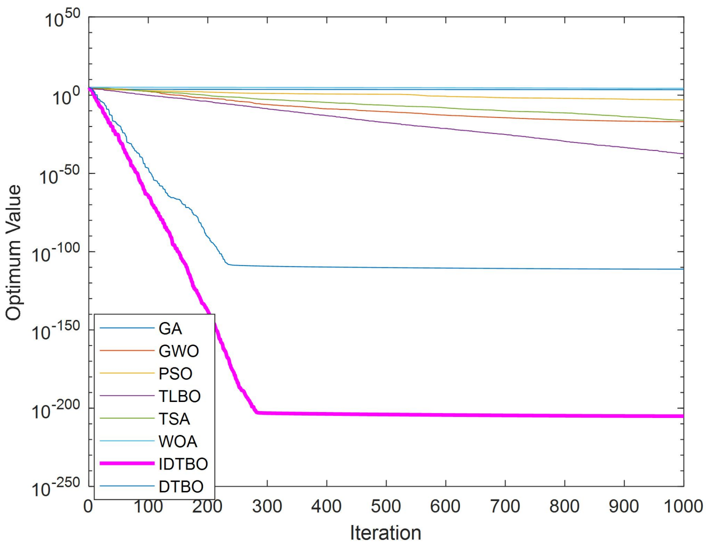

Figure 3. Convergence curve for F3

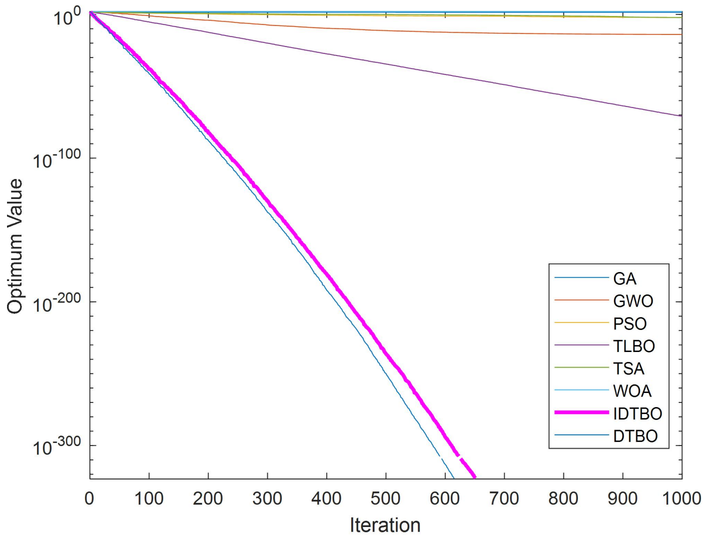

Figure 4. Convergence curve for F4

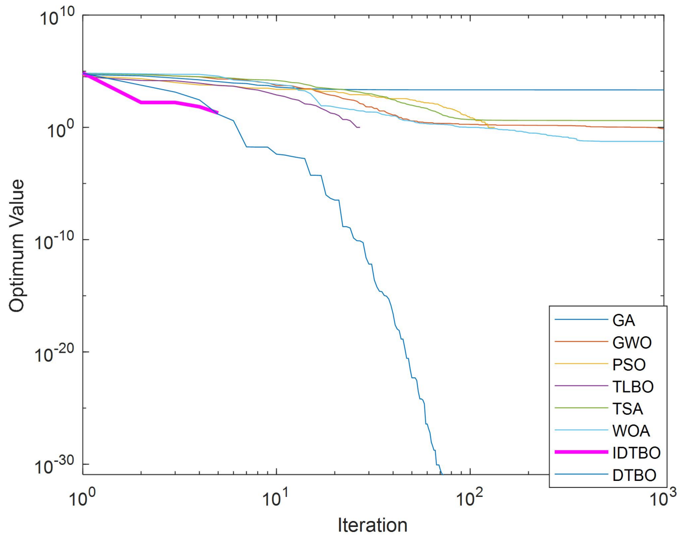

Figure 5. Convergence curve for F5

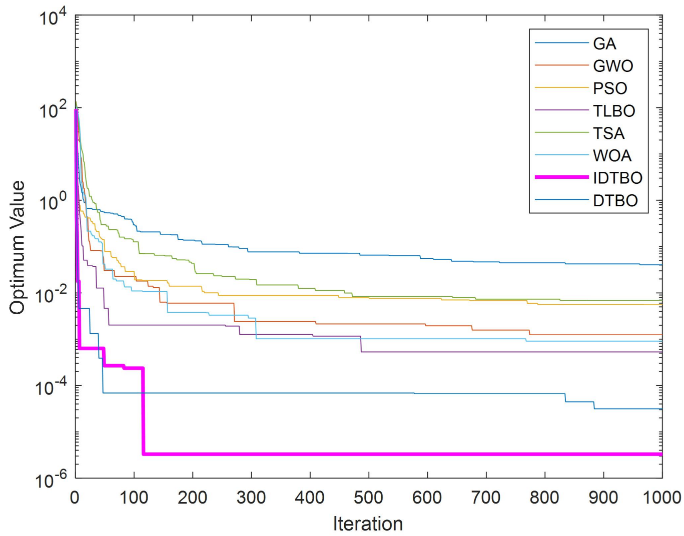

Figure 6. Convergence curve for F6

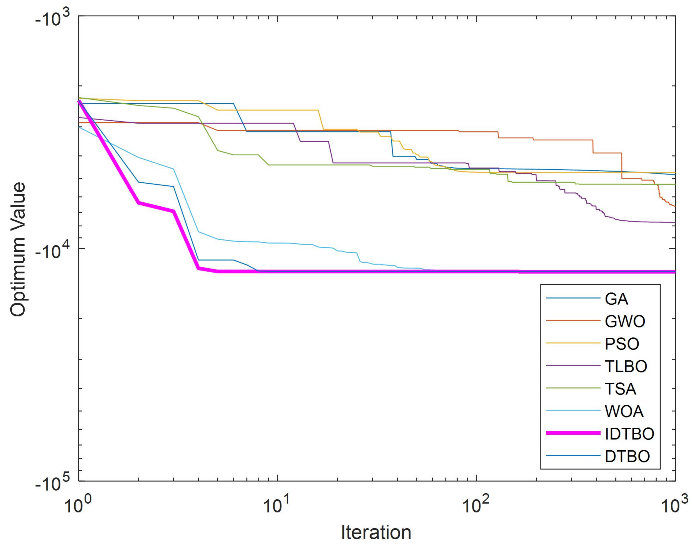

Figure 7. Convergence curve for F7

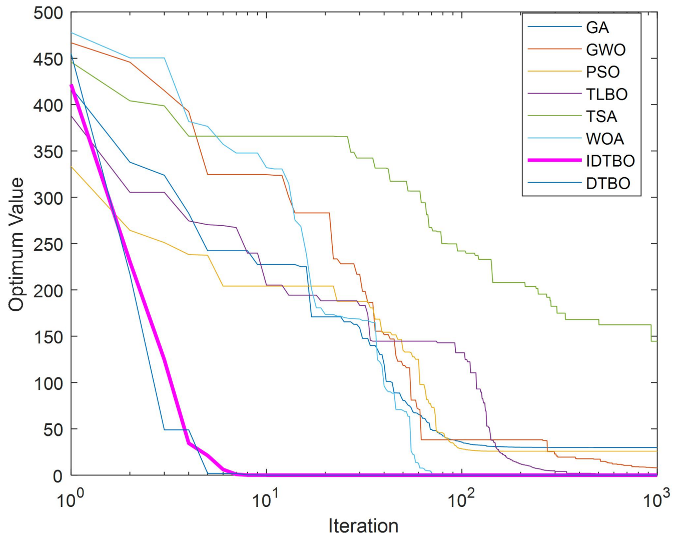

Figure 8. Convergence curve for F8

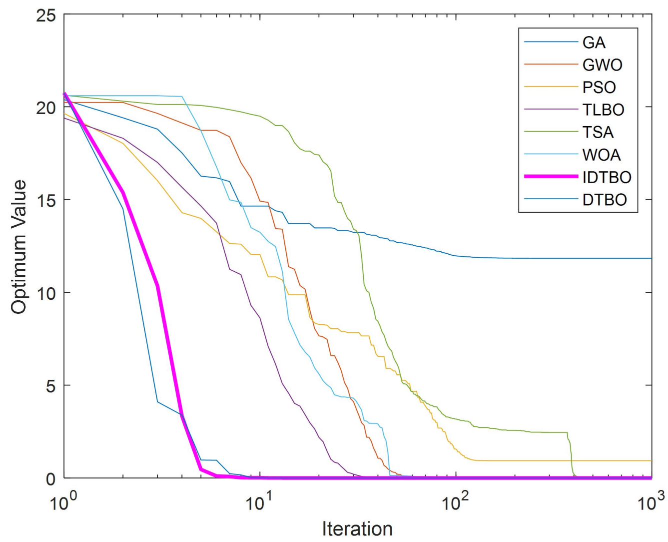

Figure 9. Convergence curve for F9

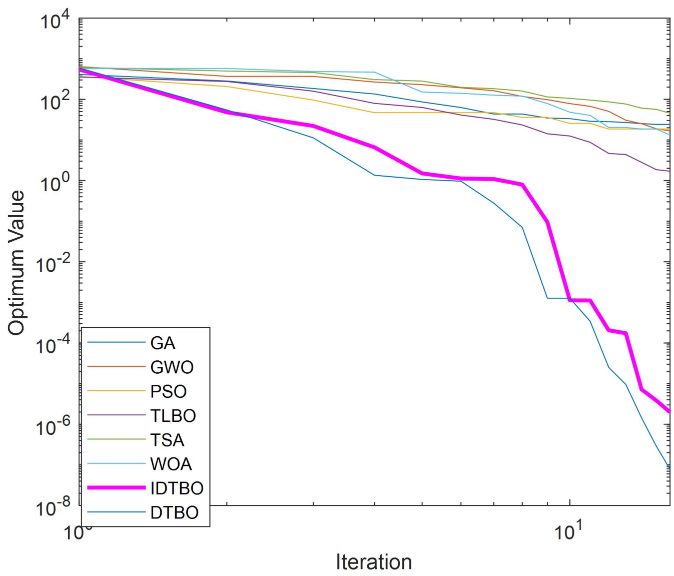

Figure 10. Convergence curve for F10

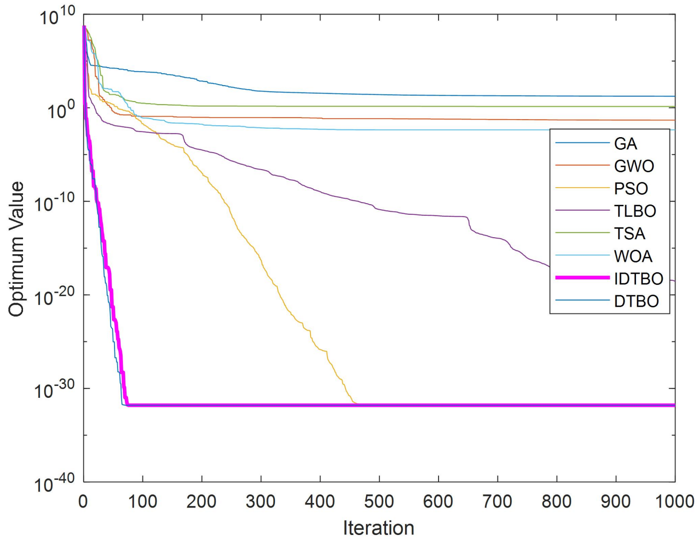

Figure 11. Convergence curve for F11

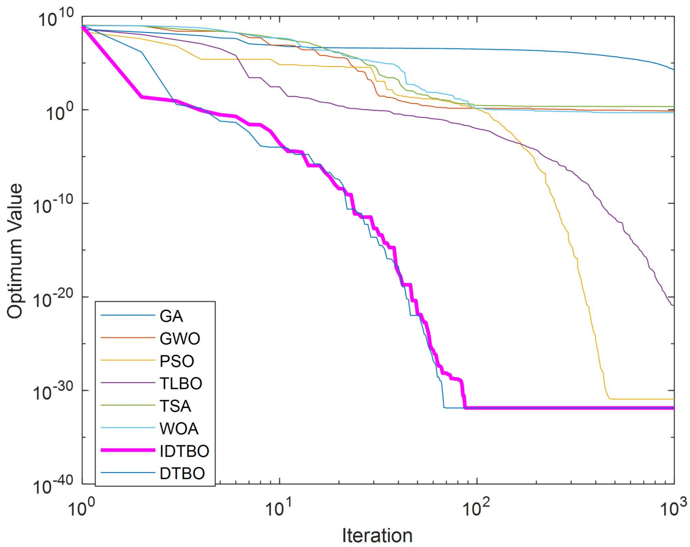

Figure 12. Convergence curve for F12

5.4 *Testing on Real World Engineering Problems*

The IDTBO is further tested on real-world engineering problems to access its effectiveness in solving practical problems. The pressure vessel, cantilever beam and string design problems are used for the testing.

5.4.1 Pressure Vessel Design Problem

The pressure vessel is a continuous optimization problem with the objective to find the most optimal combination of parameters of shell (T_s) and head thickness (T_h), inner radius (R) and length of cylindrical section (L) that ensures the vessel's structural integrity, safety, and functionality can meet stress and material strength constraints, while also minimizing material usage and associated manufacturing costs. The representation of the pressure design problem is shown in figure 13. The description of the optimization problem is given below.

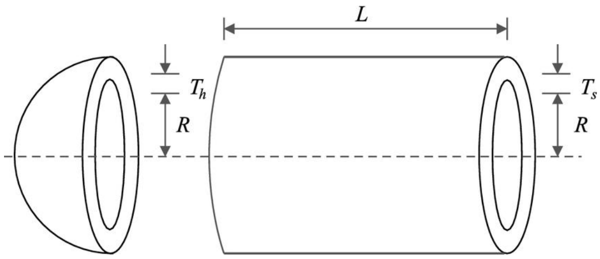

Figure 13. Representation of pressure vessel design

Cost function

$$f_{cost}(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Constraints functions

$$g_1(x) = -x_1 + 0.0193x_3 \le 0$$

$$g_2(x) = -x_2 + 0.00954x_3 \le 0$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0$$

$$g_4(x) = x_4 - 240 \le 0$$

Variable regions

$$0 \le x_{1,2} \le 99$$

$$10 \le x_{3,4} \le 200$$

From table 9, the best solution found the improved driving training-based optimization (IDTBO) is better than the DTBO and four other algorithms: WOA, TSA, GWO and GTO. White shark optimization (WOA) and POA produced the best optimum value. Again, the average value obtained by the IDTBO is better than three algorithms: WOA, GTO and DTBO.

Table 9. Results for pressure vessel design

Algorithms	Parameters				Optimum Value	Mean Value
	A	B	C	D		
WSO	0.778169	0.383036	40.319619	200.00	5880.671	6322.652
WOA	0.852402	0.444594	41.942057	178.58	6379.904	7203.060
TSA	0.838759	0.418964	43.414757	162.19	6047.267	6251.416
TLBO	0.778175	0.383041	40.319970	200.00	5880.685	5957.752
POA	0.778169	0.383036	40.319619	200.00	5880.671	6060.958
GWO	0.779065	0.383947	40.359008	199.51	5885.841	6269.511
GTO	0.894653	0.451509	46.295637	131.67	6183.904	8856.756
DTBO	0.996357	0.490441	51.624195	85.96	6362.939	6648.102
IDTBO	0.778847	0.383455	40.354705	199.51	5882.090	6603.937

5.4.2 String Design Problem

The objective function of this problem is to minimize the weight of a tension/compression spring subject to constraints on shear stress, surge frequency and minimum deflection. The representation of the string design problem is shown in figure 14. The design variables are the mean coil diameter $D = x_1$ the wire diameter $d = x_2$ and the number of active coils $N = x_3$.

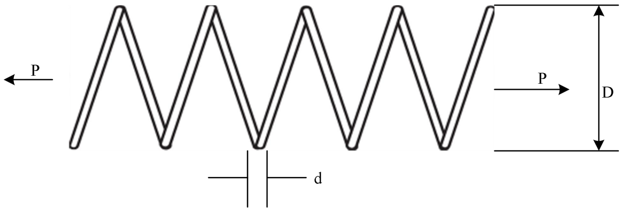

Figure 14. Representation of string design

Cost Function

$$f_{cost}\left(x\right)=\left(x_3+2\right)x_2x_1^2$$

Constraint functions

$$g_1\left(x\right)=1-\frac{x_2^3x_3}{71785x_1^4}\le 0$$

$$g_2\left(x\right)=\frac{4x_2^2-x_1x_2}{12566\left(x_2x_1^3-x_1^4\right)}+\frac{1}{5108x_1^2}-1\le 0$$

$$g_3\left(x\right)=1-\frac{140.45x_1}{x_2^2x_3}\le 0$$

$$g_4\left(x\right)=\frac{x_1+x_2}{1.5}-1\le 0$$

Variable Regions

$$0.05\le x_1\le 2$$

$$0.25\le x_2\le 1.3$$

$$2\le x_3\le 15$$

From table 10, the IDTBO exhibited a competitive performance compared to the other algorithms. All algorithms obtained an optimum value of 0.013, however IDTBO, GTO, TSA and WOA produced the second-best average value of 97040.67. This shows the competitive nature of the proposed improved driving training-based optimization.

Table 10. Results for string design problem

Algorithms	Parameters			Optimum Value	Mean Value
	a	b	c		
WSO	0.051711	0.357253	11.257640	0.013	291121.97
WOA	0.055356	0.451513	7.322753	0.013	97040.67
TSA	0.053098	0.391397	9.547392	0.013	97040.67
TLBO	0.052196	0.369025	10.602337	0.013	68324.05
POA	0.051689	0.356727	11.288403	0.013	169116.24
GWO	0.050207	0.322065	13.657523	0.013	194081.32
GTO	0.056545	0.483588	6.507328	0.013	97040.67
DTBO	0.056860	0.494339	6.211652	0.013	194081.32
IDTBO	0.057434	0.511338	5.842718	0.013	97040.67

6. Conclusion

An enhanced version of the driving training-based optimization (DTBO) algorithm has been proposed. The study addresses the problem of the algorithm getting stacked in local optima due to the selection of leaners and drivers in the metaheuristic processes. Two critical enhancing operators were introduced in the classical DTBO.

First, drivers and learners are chosen using the crowding distance technique to enhance the algorithm's diversity. Again, to enhance exploration and aid the DTBO in escaping local optima entrapment, the proposed IDTBO's initialization phase incorporates the Levy Flight distribution. These were introduced to improve the algorithm's convergence speed, diversity, and accuracy. To assess the effectiveness of the proposed IDTBO, it is tested on twelve benchmark functions, including high-dimensional multimodal and unimodal functions from the literature. Again, the effectiveness of the IDTBO in solving real-world problems is assessed with pressure vessel and string design benchmark engineering problems. The evaluation encompasses exploitation and exploration capabilities, which are crucial for metaheuristic algorithms.

In terms of exploitation ability on unimodal functions, IDTBO demonstrates remarkable performance by achieving optimal values, particularly zero optimal values for several functions. Even when not achieving absolute optimality, IDTBO consistently outperforms other algorithms, showcasing its superior exploitation capacity. Additionally, IDTBO ranks first in accuracy, as reflected in the mean absolute error (MAE) analysis.

Furthermore, in evaluating exploration ability on high-dimensional multimodal functions, IDTBO excels by achieving zero optimal values in multiple functions, showcasing its exceptional exploration capabilities. Again, IDTBO ranks first in accuracy, underscoring its efficacy in solving high-dimensional problems.

The convergence curves further illustrate IDTBO's effectiveness and efficiency, demonstrating its ability to balance exploration and exploitation across various benchmark functions. IDTBO produces superior convergence curves in several functions, highlighting its potency and efficiency in searching and exploiting the search space.

Finally, real-world applications validate IDTBO's competitiveness, as demonstrated in the pressure vessel and string design problems. IDTBO consistently outperforms other algorithms in these scenarios, further reinforcing its potential in practical optimization tasks.

The findings underscore IDTBO's competitiveness and effectiveness across diverse optimization challenges, positioning it as a promising algorithm for real-world applications in various domains.

7. Recommendation

This research opens opportunities for further developments and applications of IDTBO in solving complex and diverse optimization tasks. Applying the algorithm to real-world optimization problems across various domains is recommended to assess the practicality, robustness, and effectiveness of IDTBO in solving real world problems, as well as complex and practical optimization challenges.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

References

[1] Nassef, A.M.; Abdelkareem, M.A.; Maghrabie, H.M.; Baroutaji, A. Review of Metaheuristic Optimization Algorithms for Power Systems Problems. *Sustainability* **2023**, *15*, 9434, https://doi.org/10.3390/su15129434.

[2] Blum, C.; Chiong, R.; Clerc, M.; Dejong, K.; Michalewicz, Z.; Neri, F. Evolutionary Optimization. In *Variants of Evolutionary Algorithms for Real-World Applications*, 1st ed.; Springer Berlin: Heidelberg, Germany, 2014; pp.1–29. https://doi.org/10.1007/978-3-642-23424-8.

[3] Kwegyir, D.; Frimpong, E. A.; Opoku D. Modified Local Leader Phase Spider Monkey Optimization Algorithm. *ADRRI J. Eng. Technol.* **2021**, *5*, 1–18.

[4] Kar, A.K. Bio inspired computing – A review of algorithms and scope of applications. *Expert Syst. Appl.* **2016**, *59*, 20–32, https://doi.org/10.1016/j.eswa.2016.04.018.

[5] Sangaiah, A.K.; Sheng, M.; Zhang, Z. Computational Intelligence for Multimedia Big Data on the Cloud with Engineering Applications; Elsevier: Amsterdam, Netherlands, 2018.

[6] Beheshti, Z.; Mariyam, S.S. A Review of Population-based Meta-Heuristic Algorithm. *Int. J. Adv. Soft Comput. Appl.* **2013**, *5*, 1–35.
[7] Eberhart, R.C.; Shi, Y.; Kennedy, J. Swarm Intelligence. Morgan Kaufmann Publishers: Cambridge, MA, USA, 2001; pp. 512.
[8] Adetunji, K.E.; Hofsajer, I.W.; Abu-Mahfouz, A.M.; Cheng, L. A Review of Metaheuristic Techniques for Optimal Integration of Electrical Units in Distribution Networks. *IEEE Access* **2020**, *9*, 5046–5068, https://doi.org/10.1109/access.2020.3048438.
[9] Rao, R.V.; Savsani, V.J.; Vakharia, D.P. Teaching–learning-based optimization: A novel method for const rained mechanical design optimization problems. *Comput. Aided Des.* **2011**, *43*, 303–315, https://doi.org/1 0.1016/j.cad.2010.12.015.
[10] Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61, https://doi. org/10.1016/j.advengsoft.2013.12.007.
[11] Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, 95, 51–67. https://doi.o rg /10.1016/j.advengsoft.2016.01.008.
[12] Kaur, S.; Awasthi, L.K.; Sangal, A.; Dhiman, G. Tunicate Swarm Algorithm: A new bio-inspired based m etaheuristic paradigm for global optimization. *Eng. Appl. Artif. Intell.* **2020**, *90*, 103541, https://doi.org/10. 1016/j.engappai.2020.103541.
[13] Dehghani, M.; Trojovská, E.; Trojovský, P. A new human-based metaheuristic algorithm for solving optimization problems on the base of simulation of driving training process. *Sci. Rep.* **2022**, *12*, 1–21, https://doi.org/10.1038/s41598-022-14225-7.
[14] Rehman, H.; Sajid, I.; Sarwar, A.; Tariq, M.; Bakhsh, F.I.; Ahmad, S.; Mahmoud, H.A.; Aziz, A. Driving t raining-based optimization (DTBO) for global maximum power point tracking for a photovoltaic system u nder partial shading condition. *IET Renew. Power Gener.* **2023**, *17*, 2542–2562, https://doi.org/10.1049/rp g2.12768.
[15] Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197, https://doi.org/10.1109/4235.996017.
[16] Bechikh, S.; Kessentini, M.; Said, L.B.; Ghédira, K. Preference incorporation in evolutionary multiobjectiv e optimization: A survey of the state-of-the-art. *Adv. Comput.* **2015**, *98*, 141–207, https://doi.org/10.1016/b s.adcom.2015.03.001.
[17] Gao, Y.; Zhang, H.; Duan, Y.; Zhang, H. A novel hybrid PSO based on levy flight and wavelet mutation for global optimization. *PLOS ONE* **2023**, *18*, e0279572, https://doi.org/10.1371/journal.pone.0279572.
[18] Kamaruzaman, A.F.; Zain, A.M.; Yusuf, S.M.; Udin, A. Levy Flight Algorithm for Optimization Problems - A Literature Review. *Appl. Mech. Mater.* **2013**, *421*, 496–501, https://doi.org/10.4028/www.scientific.ne t/amm.421.496.
[19] Chawla, M.; Duhan, M. Levy Flights in Metaheuristics Optimization Algorithms – A Review. *Appl. Artif. Intell.* **2018**, *32*, 802–821, https://doi.org/10.1080/08839514.2018.1508807.
[20] Optimization Test Functions and Datasets. Available online: https://www.sfu.ca/~ssurjano/optimization.ht ml (accessed on 24 April 2024).
[21] Trojovský, P.; Dehghani, M. Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications. *Sensors* **2022**, *22*, 855, https://doi.org/10.3390/s22030855.
[22] Swami, V.; Kumar, S.; Jain, S. An Improved Spider Monkey Optimization Algorithm. In Soft Computing: Theories and Applications. Advances in Intelligent Systems and Computing, Springer: Singapore, 2018, https://doi.org/10.1007/978-981-10-5687-1_7.
[23] Singh, P.R.; Moussa, D.; Shengwu, X.; Singh, B.P. Improved Spider Monkey Optimization Algorithm to train MLP for data classification. *3c Tecnología: glosas de innovación aplicadas a la pyme* **2019**, *8*, 142–165, https://doi.org/10.17993/3ctecno.2019.specialissue2.142-165.