Artificial Intelligence Research and Applications (AIra)
https://www.ijaira.org/ Online ISSN: 3108-3889
A. Sabo et al., Vol. 2, No. 2, June, 2026 Research Article

# Levy Flight Dream Optimization Algorithm: A Modified Version of Dream Optimization Algorithm

Aliyu Sabo*, Abel E. Airoboman*, Kazeem Lawal*, Dahiru Dauda*‡, Kenneth Okedu**, Auwal Aliyu*, Usman Isah Yusuf***, Jetta Somtochuku Ezeagba****

* Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna, Nigeria.

** School of Information Technology and Engineering, Melbourne Institute of Technology, 3000, Victoria, Australia.

*** Department of Computer Engineering, Ahmadu Bello University, Zaria-Nigeria.

**** Department of Engineering Management Enugu State University of Technology, Nigeria.

(saboaliyu98@gmail.com, airobomanabel@nda.edu.ng, klawal2011@gmail.com, dawooddahiru@gmail.com, kokedu@academic.mit.edu.au, engr.auwalaliyu@gmail.com, engrusmanisah@gmail.com, jettaezeagba@gmail.com)

‡ Corresponding Author: Dahiru Dauda, Department of Electrical/Electronic Engineering, Nigerian Defence Academy, Kaduna, Nigeria, dawooddahiru@gmail.com.

*Received: 10.02.2026, Revised: 07.03.2026, Accepted: 18.03.2026*

**Abstract-** This manuscript elucidates a novel Modified Dream Optimization Algorithm (mDOA). The foundational framework of the Dream Optimization Algorithm (DOA) is informed by the cognitive phenomena associated with human dreaming. These cognitive mechanisms (memory retention, forgetfulness, supplementation strategies, and dream sharing processes) are systematically encoded as an optimization agent designed to tackle global optimization dilemmas. The DOA is afflicted by the challenge of an imbalance between exploration and exploitation, exhibiting a higher propensity for exploration than for exploitation, which results in an elevated likelihood of becoming ensnared in local optima. The enhancement of mDOA was achieved through the integration of a Levy flight variant to boost the exploitation phase. The efficacy of mDOA is evaluated against six prominent metaheuristics utilizing ten benchmark test functions (Schwefel, Ackley, Michalewicz, Griewank, Pathologic, Rastrigrin, Rosenbrock, Schaffer, Sphere, and Bohachevsky1), it demonstrated 85% enhancement in its convergence towards global optima. From the simulation results obtained, it shows that the mDOA succeeded in attaining the optimal global solution in 7 out of 10 cases, constituting 70.0% of the benchmark functions. Conversely, the 0ther algorithms used achieved 3 out of 10 cases, representing 30.0% of the benchmark functions. These shows an improvement in the mDOA.

**Keywords:** Dream optimization algorithm, exploration, exploitation, levy flight, standard function.

## 1. Introduction

The use of nature-inspire optimization methodologies in recent years has demonstrated efficiency in addressing different types of optimization issues with considerable performance [1]. Optimization comprises a systematic approach for initiating solutions to challenges that are confined by specific limitations through the most effective utilization of available resources. The same outcome is produced by decisive search algorithms if the inceptive conditions remain the same. Even when the initial conditions remain stable, stochastic algorithms generate definite

**Cite this article as:** A. Sabo, A. E. Airoboman, K. Lawal, D. Dauda, K. Okedu, A. Aliyu, U. I. Yusuf, and J. S. Ezeagba, "Levy Flight Dream Optimization Algorithm: A Modified Version of Dream Optimization Algorithm," *Artificial Intelligence Research and Applications (AIra)*, Vol. 2, No. 2, pp. 90–102, June, 2026, https://doi.org/10.20508/hyyn8602.

solutions every time they run because there is irregular in their search procedure [2]. Universal optimization algorithms, analytical intelligence, and contemporary soft computing paradigms extensively depend on metaheuristic approaches that are inspired by natural phenomena. Optimization algorithms come in two varieties: deterministic and stochastic. [3].

Past studies as classified Stochastic algorithms as either heuristic or metaheuristic. The heuristic methodology is limited to a single type of optimization problem, making them problem dependent. The metaheuristic-based search algorithm is problem-independent universal problem solver algorithm that is used to tackle a variety of optimization problems [4]. The metaheuristic search algorithms combine exploitation (intensification) and exploration (diversification). To find the optimal local solutions inside the search space, the algorithm is directed by the exploration process. The exploitation procedure directs the algorithm to search among the generated local optimal for the global optimum solution. In finding a balance between exploration and exploitation, the metaheuristic search algorithm can connect with the global optimum result [5]. Metaheuristic search algorithms draw inspiration from biological systems. Numerous optimization challenges have been solved using these nature-inspired metaheuristic search algorithms [6]. Particles swarm are an illustration of a metaheuristic search algorithm influence by nature. Swarms of birds and fish shoal served as inspiration for swarm optimization [7]. However, classical PSO still has some weaknesses, such as poor local search that may lead to traps in local minimum affecting the convergence performance that results in uncertainties in the outcomes obtained [8].

The firefly algorithm simulates flashing behavior of fireflies [9], while the cat swarm optimization algorithm imitates the hunting and stalking manners of cats towards their target [10], The echolocation behavior of microbats is mimicked by bat algorithms [11], while the foraging behavior of ant colonies is mimicked by ant colony optimization algorithms [12]. The Dream Optimization Algorithm (DOA) was motivated by desire [13], which shares many properties with the optimization process in metaheuristic algorithms, including partial memory retention, forgetting, and logical self-organization. To stabilize exploration and exploitation, DOA incorporate a fundamental memory method, a forgetting and supplementing technique, and a dream-sharing strategy to enhance the amplitude to escape local optima. Exploration and exploitation phases make up the optimization process, which produces good optimization outcomes. According to the literature review, DOA offers potential benefits over other metaheuristic algorithms including Particle Swarm Optimization (PSO) [14], Grey Wolf Optimization (GWO) [15], Sparrow Search Algorithm (SSA) [16], Differentiated Creative Search (DCS) [17], Great Wall Construction Algorithm (GWCA) [18], and Whale Optimization Algorithm (WOA) [19], among others. Strong convergence, progress, stability, adaptability, resilience, and reliability to initial control parameter values are some of these benefits. The dream optimization method still has an imbalance issue despite these many benefits, allying exploration and exploitation because of the continual impact of control settings, optimization hyperspace, and incomplete knowledge. The paper employed levy flying process to produce an improved Dream Optimization Algorithm. The capacity of the mDOA algorithm was estimated to be using ten (10) standard test functions, the outcome of the results was contrast to those of the traditional DOA, White Shark Optimization (WSO), Seagull Optimization Algorithm (SOA), Sunflower Optimization (SFO), Golden Jackal Optimization (GJO) and African Vultures Optimization Algorithm (AVOA). Comparison demonstrated the dominance of the mDOA algorithm over the other algorithm. The contribution of this paper is the modification of DOA using levy flight variant.

The report's remaining sections are assembled as follows: Section 2 introduces the DOA algorithm and its levy flight. In Section 3, the proposed mDOA result is shown. In Section 4, the effectiveness of mDOA is evaluated and compared to the traditional DOA and others algorithms approach.

## 2. Research Design Method

DOA draws from human consciousness during dreams, particularly leveraging features of dreams such as partial memory retention, forgetting certain information, and a natural logical self-organization of elements. This biological inspiration is translated into computational strategies.

### *2.1. Concept of DOA Algorithm*

Optimization algorithm assumptions When integrating the attributes of human aspirations with the principles of optimization algorithms, we delineate the subsequent four postulations:

- Fitness values can be used to assess the quality of dreams.
- The basis of pre-existing memories is intimately linked to the beginning of a dream.
- People add logically self-organized information to partially forgotten memories.
- Memory capacities are rather unpredictable and vary between people or groups.

The dream optimization method is developed based on these fundamental presumptions. These four underlying presumptions are reflected in the procedural architecture, exploration phase, and various approaches used during the algorithm's development phase.

### *2.2. Initialization Phase*

In order to initiate the algorithm's optimisation process, mDOA first creates a random sample inside the search space called the initial sample. The initial sample can be acquired using Equation 1:

$$X_i = X_l + rad \times (X_u - X_l), \quad i = 1,2, \dots \dots \dots . N \qquad (1)$$

where $N$ is the number of individuals, i.e., the sample size. $X_i$ is the $ith$ individual in the sample; $X_u$ and $X_l$ for the lower and upper boundaries of the search space, accordingly, the resulting sample can be shown as follows: $rad$ is a Dim-dimensional vector, where each proportion is a random number between 0 and 1.

$$X = \begin{bmatrix} X_1 \\ X_2 \\ X_N \end{bmatrix} = \begin{bmatrix} X_{11} & \cdots & X_{1,Dim} \\ \vdots & \ddots & \vdots \\ X_{N,1} & \cdots & X_{N,Dim} \end{bmatrix} \quad (2)$$

where $X_{ij}$, denotes the location of the $ith$ individual in the $jth$ dimension, and Dim denotes the dimensionality of the optimization issue.

### 2.3. Exploration Phase

The formulas and particular update mechanism are as follows:

#### 2.3.1. Memory strategy

First, group q members can reset their formation by recalling the formation position of their group's best member before dreaming, according to the foundational memory technique. position to that of the best member of the group:

$$XX_i^{t+1} = X_{bestq}^t \quad (3)$$

where $X_i^{t+1}$ denotes the $ith$ person at iteration $t+1$ and $X_{bestq}^t$ denotes the best individual in group $q$ at iteration $t$.

#### 2.3.2. Forgetting and supplementation strategy

The forgetting and supplementing approach combines local and global search capabilities. This strategy, which is like the memory technique, enables people to self-organize and forget the position information in the forgetting dimensions. The following is the revised formula:

$$X_{ij}^{t+1} = X_{bestqj}^t + \left(X_{ij} + rand \times \left(X_{uj} - X_{lj}\right)\right) \times \frac{1}{2} \times \left(\cos\left(\pi \times \frac{t + T_{max} - T_d}{T_{max}}\right) + 1\right), \quad j = K_1, K_2 \ldots\ldots\ldots K_{k_q} \quad (4)$$

where $X_{ij}^{t+1}$ 1; tthe position of the $ith$ individual in the $jth$ dimension at iteration $t+1$: the $X_{bestqj}^t$ indicate a position of the best location in group $q$ in the $jth$ dimension at iteration t; $X_{lj}$ and $X_{uj}$ represent the minimum and maximum bounds of the search space in the $jth$ dimension, respectively; $rand$ is a random number between 0 and 1;t is the current iteration number, $T_{max}$ is the maximum number of iterations, and $T_d$ is the maximum number of iterations during the exploration phase.

#### 2.3.3. Dream-sharing strategy

In mDOA, the dream-sharing technique improves the capacity to break out of local optima. People can randomly gather position information from others in the forgetting dimensions using this method, which functions in agreement with the forgetting and augmenting methods and follows the memory strategy. The following is the update formula:

$$X_{ij}^{t+1} = \begin{cases} X_{mj,}^{t+1} & m \le i \\ X_{mj,}^{t} & i < m \le N \end{cases} \quad j = K_1, K_2, \ldots\ldots\ldots K_{k_q} \quad (5)$$

where $X_{ij}^{t+1}$, is the location of the ith person in the jth dimension at iteration t+1; m is an arbitrarily chosen natural number from the range [1, N] for each dimension update.

### 2.4. Exploitation Phase

The Levy Flight (LF) was used to enhance the exploitation phase; grouping is no longer done throughout the stage of development (iteration count from $T_d$ to $T_{max}$).

The population shows the best dream from the past iterations of the entire population or the best individual from the previous iterations before each dreaming session. Each person's whereabouts in the forgetting dimensions is then updated.

The number of forgetting dimensions $k_r$ is the same for every member of the population. The locations in the $k_r$ forgetting dimensions—designated as $K_1, K_2, \ldots\ldots\ldots K_{k_r}$ are updated once they are randomly chosen from the D dimensions.

The improving method is like Equation (2) and (3), with the update formula as follows.

#### 2.4.1. Memory strategy

$$X_i^{t+1} = X_{best}^t \quad (6)$$

where $X_i^{t+1}$ is the $ith$ individual at iteration $t+1$, and $X_{best}^t$ indicate the best person of the sample at a given iteration $t$.

#### 2.4.2. Forgetting and supplementation strategy

$$X_{ij}^{t+1} = X_{bestj}^t + \left(X_{ij} + rand \times \left(X_{uj} - X_{lj}\right)\right) \times \frac{1}{2} \times \left(\cos\left(\pi \times \frac{t}{T_{max}}\right) + 1\right), \quad j = K_1, K_2 \ldots\ldots\ldots K_{k_r} \quad (7)$$

Where $X_{ij}^{t+1}$ is the position of the $ith$ individual in the $jth$ dimension at iteration $t+1$, $X_{bestj}^t$ denotes the position of the best person of the entire population in the $jth$ dimension at iteration $t$; $X_{lj}$ and $X_{uj}$ are the minimum and maximum bounds of the search space in the $jth$ dimension, respectively; $rand$ is a arbitra number between 0 and 1; $t$ is the current iteration number, and $T_{max}$ is the maximum number of iterations for the algorithm.

In a similar vein, Eq. (6) demonstrates that in dimensions other than $K_1, K_2 \ldots\ldots\ldots K_{k_r}$ individuals are able to preserve the exact placements in these dimensions while dreaming by recalling the position information of the best position in the population during prior iterations.

Eq. (7) demonstrates that in dimensions $K_1, K_2 \ldots K_{k_r}$, people forget the location of the population's finest person during the prior.

### 2.5. Levy Flight Algorithms

A Levy flight algorithm represents a specific variant of random walk distinguished by step lengths that adhere to a Levy distribution, which is characterized by a power-law tail, resulting in infrequent long jumps interspersed with shorter movements. This distinctive pattern is employed in numerous optimization algorithms, especially within the realm of mathematics, to augment both exploration and exploitation capabilities. The functionalities of the Levy flight are operationalized as:

$$Le'vy(n,d) = \frac{u}{|v|^{1/\beta}}, \qquad u \sim N(0,\sigma^2), \qquad v \sim N(0,1) \quad (8)$$

where $\beta = 1.5$ and $\sigma$ is computed as:

$$\left(\frac{\Gamma(1+\beta).\sin(\pi\beta/2)}{\Gamma\left(\frac{1+\beta}{2}\right).\beta.2^{\frac{\beta-1}{2}}}\right)^{1/\beta} \quad (9)$$

In the foundational dream optimization algorithm, the methodology demonstrates a robust capacity for exploitation; however, it is susceptible to becoming entrenched in local optima. To mitigate this challenge, the paper incorporates a Lévy flight strategy into the exploitation phase to improve the convergence rate.

Table 1 and 2 show the pseudocode for both the standard Dream Optimization Algorithm and the modified version, denoted as DOA-LF.

**Table 1.** Standard DOA pseudo-code.

Algorithm 1 pseudo-code of DOA

Input : Population size ($N_{max}$) the lower limit of variables ($X_l$), the upper limit of variables ($X_u$), size of problem ($D_{im}$), the current number of iteration ($t$), the number of iteration as a demaracation ($T_d$), the maximum number of iteration ($T_{max}$), forgetting dimensions of each group and of exploitation ($k_1, k_2, k_3, k_4\ k_5, k_r$)

Output: the best solution $X_{best}$ and the minimum fitness $Fitness_{best}$.

Generate an initial population $X\ of\ N$ individuals using Eqs. (2)and (3)
Check the bounds of the solutions
Evaluate the fitness of the solution
Detect the best solution $X_{best}$ and the minimum fitness $Fitness_{best}$
Define the current iteration $t = 1$
while $t < T_d\ do$
Update the best solution $X_{best}$ and the minimum fitness $Fitness_{min}$
**for** $q = 1 : 5\ do$
Update the best solution $X_{best}$ and the minimum fitness $Fitness_{minq}$
Update $k_q$ using Eq. (10)
Update $X_i^{i+1}$ using Eq. (4)
$\left(K_1, K_2, \dots \dots K_{k_q}\right) = rad\ perm\left(k_q, N\right)$
$for\ i = (((q-1)/(5 \times N)) + 1) : (q/5 \times N)\ do$
$if\ rad < u\ then$
Update $x_{i,j}$ using Eq. (5)
Check the bound of$x_{i,j}$
**else**
Update $x_{i,j}$ using Eq. (6)
  **end if**
**end for**
**end for**
Update the current number of iteration $t\ by\ t = t + 1$
**end while**
while $t < T_d\ and\ t < T_{max} do$
Update $k_r$ using Eq. (11)
Update $X_i^{t+1}$ using Eq. (7)
$\left(K_1, K_2, \dots \dots K_{k_r}\right) = rad\ perm(k_r, N)$
**for** $i = 1 : N\ do$
Update $x_{i,j}$ using Eq. (8)
Check the bound of $x_{i,j}$
**end for**
Update the current number of iteration $t\ by\ t = t + 1$
**end while**

**Table 2.** DOA with Levy Flight (DAO-LF) pseudo-code.

| Algorithm 1 Pseudo-code of DOA with Levy Flight Enhancement |
|---|
| Input: population size, iterations T, bounds lb, ub, dimension D, objective function fobj |
| Initialise: |
| $x \leftarrow$ random popupation in $[lb, ub]$ |
| $fbestd[m] \leftarrow \infty, sbestd[m] \leftarrow empty, for\ m = 1 \ldots 5$ |
| $fbest[m] \leftarrow \infty, sbestd[m] \leftarrow empty, for\ m = 1, \ldots \ldots . .5$ |
| $fbest_{history} \leftarrow zeros(T)$ |
| $for\ i = 1 \rightarrow [0.9T] do$ $\cdots Exploration\ phase$ |
| $for\ m = 1 \rightarrow 5\ do$ |
| $k \leftarrow random\ integer\ in\ \left[[D(8m)], \left[\frac{D}{3m}\right]\right]$ |
| for each solution $j$ in subgroup $m$ do |
| $fit \leftarrow fobj\ (x_j)$ |
| if $fit < fbestd[m] then$ |
| $fbestd[m] \leftarrow fit, sbestd[m] \leftarrow x_j$ |
| end if |
| end for |
| for each solution $j$ in subgroup $m$ do |
| $x_j \leftarrow sbestd[m]$ |
| Choose $k$ random dimensions in |
| if rand $<$ 0.9 then |
| $step \leftarrow$ Levy Flight (1.5D) |
| for each $h \in in$ do |
| $x_j$[h] $\leftarrow$ $x_j$[h]+0.01 $step$[h].$(ub[h] - lb[h]).\frac{\cos\left(i+\frac{\frac{T}{10\pi}}{T}\right)+1}{2}$ |
| Apply boundary chech |
| end for |
| else |
| for each $h \in in$ do |
| $x_j[h]$ $\leftarrow$ random element from populatin |
| $end\ for$ |
| end if |
| end for |
| if $fbestd[m] < fbest$ then |
| $fbest \leftarrow fbestd[m], sbest \leftarrow sbestd[m]$ |
| ***end if*** |
| **end for** |
| $fbest\_history[i] \leftarrow fbest$ |
| end for |
| $w_{levy} \leftarrow 0.5$ … Weight for Levy steps |
| for $i = [0.9T] + 1 \rightarrow T$ do … Exploitation phase |
| for $p = 1 \rightarrow pop$ do |
| $fit \leftarrow fobj(x_p)$ |
| if $fit < fbest$ then |
| $fbest \leftarrow fit, sbest \leftarrow x_p$ |
| ***end if*** |
| **end for** |
| for $j = 1 \rightarrow pop$ do |
| $k \leftarrow randominteger\ in$ $[2. \max\left(2\left[\frac{D}{3}\right]\right)]$ |
| $x_j \leftarrow sbest$ |
| Choose $k$ random dimensions in |
| for each $h \in$ in do |
| if rand $< w_{levy}$ then |
| $step$ $\leftarrow$Levy Flight (1.5,D) |
| $x_j$[h] $\leftarrow$ $x_j$[h]+0.01 $step$[h].$(ub[h] - lb[h]).\frac{\cos(i\pi+T)+1}{2}$ |
| $else$ |
| $x_j$[h] $\leftarrow$ $x_j$[h]+0.01 $rand.(ub[h] - lb[h]).\frac{\cos(i\pi+T)+1}{2}$ |
| **end if** |
| Apply boundary check 1 |
| **end for** |
| **end for** |
| $fbest\_history[i] \leftarrow fbest$ |

### 2.6. Code Comparison to Other Levy Flight Developed Optimization Algorithms

This paper applied the Levy Flight at the exploitation phase. The modification is as follows:

- Modify the forgetting and supplementation strategy equation using levy flight, from equation 6 of the standard DOA using equation 8 for specific variant of random walk at exploitation phase to enhance the convergence speed.
- evaluate performance against existing algorithms with parameters such as convergences speed, efficiency, and robustness.
- The Levy step size is uniquely scaled by a cosine-wave term, $\frac{\cos(.)+1}{2}$ which creates a smooth, oscillatory, and non-linear decay of step length over iterations, differing from standard static or linearly decreasing scales.
- The number of dimensions ($k$) to perturb using Levy flights is dynamically and differently calculated per subgroup during exploration, making the search effort adaptive to the subgroup's role and problem dimension.
- The use of Levy flight differs fundamentally between phases: in exploration, it's applied around subgroup-best solutions (sbestd[m]) and competes with random replacement, while in exploitation, it's applied more directly around the global best (sbest) with deterministic weighting.

#### 2.6.1. Evaluation method

The capacity of any improved or developed optimization algorithm is evaluated using benchmark functions, which are applied mathematical functions. As algorithms are created to address real-world engineering optimization issues, this is seen to be crucial. The set of ten standard test functions given in Table 3 were carefully selected to effectively measure the capacity of developed mDOA algorithms. The mDOAs performance measures of interest are listed as follows.

Execution speed: a gauge of how fast the mDOA algorithm variation can find the best convergence. The precision of the solution, which quantifies how closely each mDOA's results resemble the ideal.

The convergence rate: This is utilized to ascertain when the corresponding mDOAs find their answer.

**Table 3.** Ten-dimensional standard functions [20].

| Fn | Names | Description |
|---|---|---|
| $F_1$ | Shere function [21] | $f(x)=\sum_{i=1}^{d} x_i^2$ |
| $F_2$ | Schwefel function [22] | $f(x)=\sum_{i=1}^{30}\left(x_i^2+x_{i+1}^2\right)^{0.5}\left\{\left[\sin 50\left(x_i^2+x_{i+1}^2\right)^{0.1}\right]^2\right\}$ |
| $F_3$ | Rosenbrock Function [23] | $f(x)=\sum_{i=1}^{n-1}((x_i-1)^2+100(x_{i+1}-x_i^2)^2)$ |
| $F_4$ | Rastrigin Function [24] | $f(x)=10n+\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)],$ |
| $F_5$ | Griewank Function [25] | $f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$ |
| $F_6$ | Ackley function [26] | $f(X)=-2o\exp\left[-\frac{1}{5}\sqrt{\frac{1}{n}\sum_{i=1}^{n}X_i^2}\right]-\exp\left[\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)\right]+20+e$ |
| $F_7$ | Michalewicz function [23] | $f(x)=-\sum_{i=1}^{d}\sin(x_i)\sin^{2m}\left(\frac{ix_i^2}{\pi}\right)$ |
| $F_8$ | Bohachevsky1 function [27] | $f(x_1,x_2)=x_1^2+2x_2^2-0.3\cos(3\pi x_1)-0.4\cos(4\pi x_2)+0.7$ |
| $F_9$ | Schaffer function [28] | $f(x)=\sum_{i=1}^{30}\left(x_i^2+x_{i+1}^2\right)^{0.5}\left\{\left[\sin 50\left(x_i^2+x_{i+1}^2\right)^{0.1}\right]^2\right\}$ |
| $F_{10}$ | Pathologic function [29] | $f(x)=\sum_{n=1}^{n}(0,5+\frac{\sin\left(\sqrt{100x_i^2+x_{i\neq1}^2}\right)^2-0.5)}{(1+0.001(x_i^2-2x_ix_{i\neq1}+x_{i=1}^2)^2})^2$ |

## 3. Results Analysis

### *3.1. Convergence Comparison of mDOA*

The mDOA has shown a better convergence in comparison with some of the algorithms used as shown in figures below. The mDOA shows vibrances in convergence with other algorithms used such as standard DOA, WSO, SOA, SFO, GJO and AVOA on standard functions.

For the unimodal and multimodal standard function, the standard deviation, mean, and best values attained by mDOA and other algorithms are shown in Tables 4 and 5. Furthermore, Figures 1 and 2 show how best mDOA and other algorithms perform in comparison with unimodal and multimodal standard functions.

Figure 1 shows the Sphere, Griewank, and Schwefel's 2.2 unimodal standard functions. Table 4 provides comprehensive details about their mean, standard deviation, and best values. The global minimum noted in Table 4 is quite close to the ideal value attained by mDOA. Notably, the energy function added in Equation (9) allows mDOA to perform better than the three algorithms across all standard functions of unimodal in terms of mean values and standard deviation. It's important to note, though, that mDOA shows some superior exploitation potential for the Schwefel's 2.22 function.

The multimodal standard functions depicted in Figure 2 include Rosenbrock, Rastrigin, Michalewicz, Rosenbrock, Bohachevsky1, Schaffer, and Pathologic functions. The statistical analysis regarding their standard deviation, mean, and best values can be found in Table 5.

The optimal value achieved by mDOA, as indicated in Table 5, closely approximates the global minimum highlighted, except for the Ackley, Michalewicz and Rosenbrock function. The mDOA has better exploitative ability compared to the DOA and other five algorithms in comparison. It exhibits better exploitation and exploration abilities for the Step function.

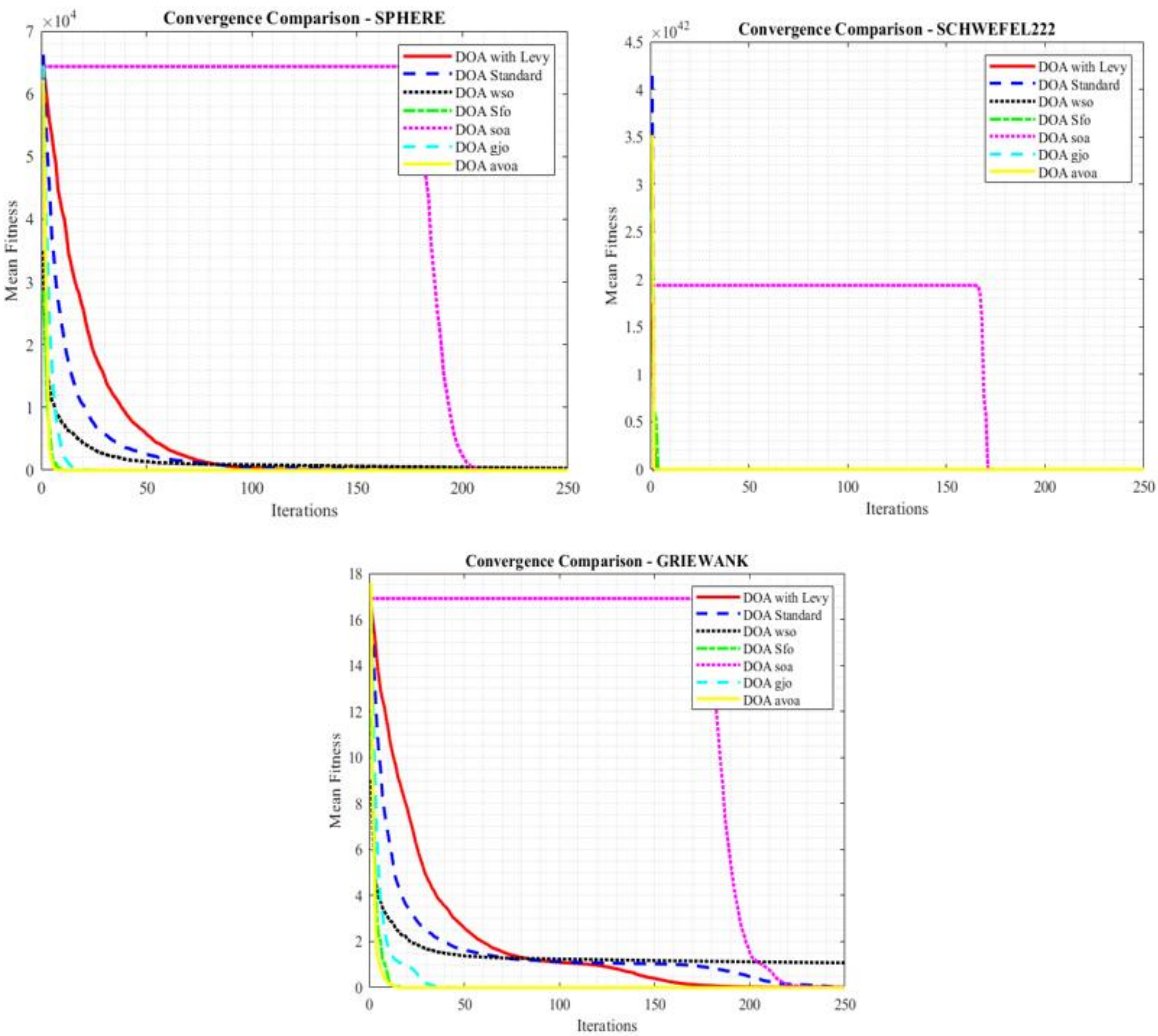


**Fig. 1**. Unimodal standard functions.

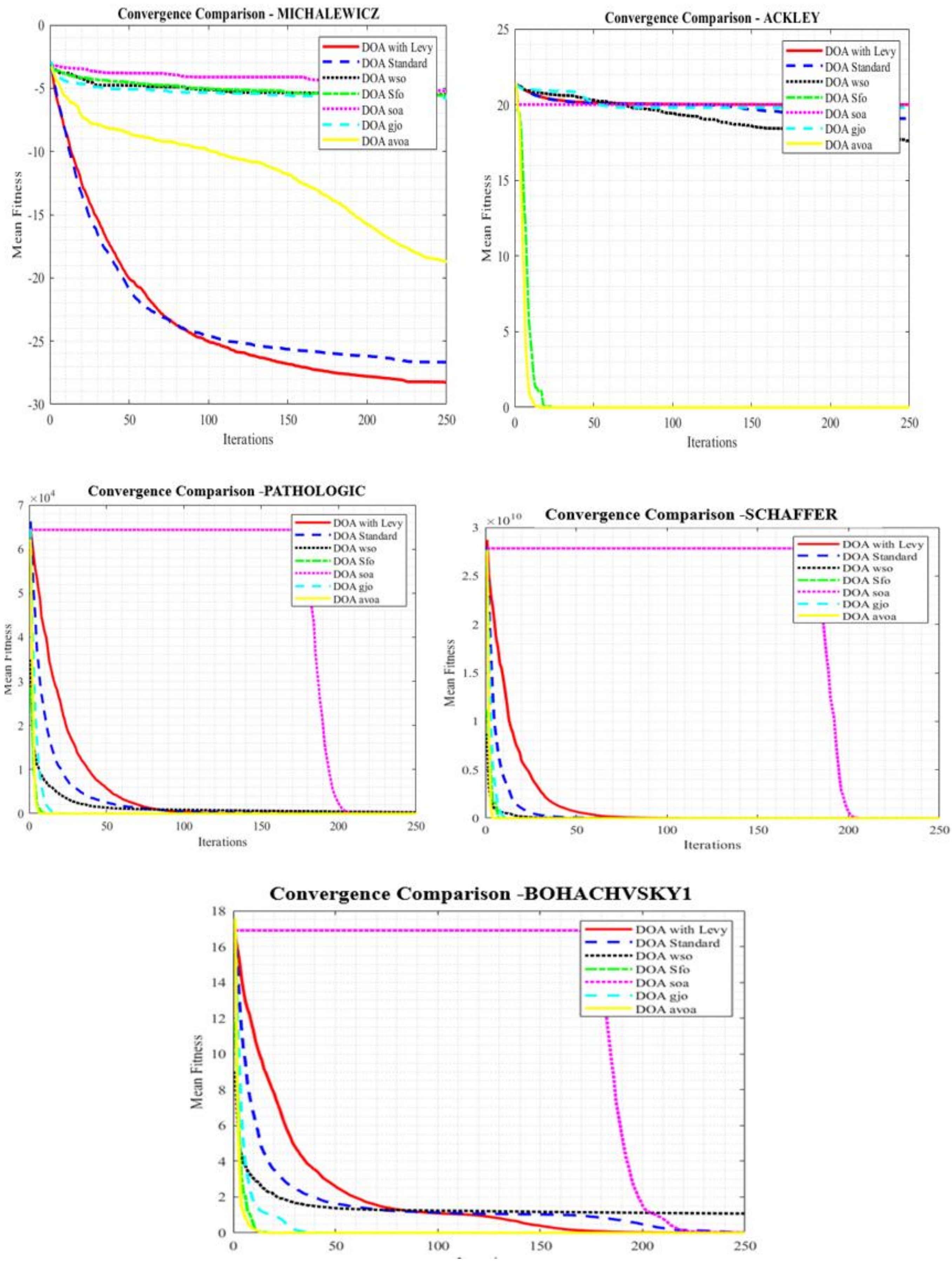


**Fig. 2**. Multimodal standard functions.

**Table 4.** Unimodal standard functions.

| Function (Fn) | | DOA-LF | DOA | WSO | SFO | SOA | GJO | AVOA | Global minimum function value |
|---|---|---|---|---|---|---|---|---|---|
| Sphere | Best | **4.0265e-03** | 1.6706e-01 | 4.0266e-03 | 1.6706e-01 | 4.0266e-03 | 1.6706e-01 | 1.6706e-01 | 0 |
| | Mean | 2.8503e-02 | 2.7535e-01 | 2.8503e-02 | 2.7535e-01 | 2.8503e-02 | 2.7535e-01 | 2.7535e-01 | |
| | Std Dev | 2.3109e-02 | 7.8468e-02 | 2.3109e-02 | 7.8468e-02 | 2.3109e-02 | 7.8468e-02 | 7.8468e-02 | |
| Schwefel 2.2 | Best | **8.8727e-02** | 1.2339e+00 | 8.8827e-02 | 1.2339e+00 | 8.8727e-02 | 1.2339e+00 | 1.2339e+00 | 0 |
| | Mean | 1.4009e+01 | 1.6050e+00 | 1.4009e+01 | 1.6050e+00 | 1.4009e+01 | 1.6050e+00 | 1.6050e+00 | |
| | Std Dev | 2.1106e+01 | 2.2127e-01 | 2.1106e+01 | 2.2127e-01 | 2.1106e+01 | 2.2127e-01 | 2.2127e-01 | |
| Griewank | Best | **4.2316e-04** | 1.4267e-02 | 4.2326e-04 | 1.4267e-02 | 4.2326e-04 | 1.4267e-02 | 1.4267e-02 | 0 |
| | Mean | 4.6401e-03 | 3.0762e-02 | 4.6401e-03 | 3.0762e-02 | 4.6401e-03 | 3.0762e-02 | 3.0762e-02 | |
| | Std Dev | 6.4389e-03 | 1.3988e-02 | 6.4389e-03 | 1.3988e-02 | 6.4389e-03 | 1.3988e-02 | 1.3988e-02 | |

**Table 5.** Multimodal standard functions.

| Function (Fn) | | DOA-LF | DOA | WSO | SFO | SOA | GJO | AVOA | Global minimum function value |
|---|---|---|---|---|---|---|---|---|---|
| Michalewicz | Best | -2.8797e+01 | **-2.7537e+01** | -2.8797e+01 | **-2.7537e+01** | -2.8797e+01 | **-2.7537e+01** | **-2.7537e+01** | −8.22015 |
| | Mean | -2.8282e+01 | -2.6672e+01 | -2.8282e+01 | -2.6672e+01 | -2.8282e+01 | -2.6672e+01 | -2.6672e+01 | |
| | Std Dev | 3.5005e-01 | 5.8910e-01 | 3.5005e-01 | 5.8910e-01 | 3.5005e-01 | 5.8910e-01 | 5.8910e-01 | |
| Rosenbrock | Best | 2.7636e+02 | **1.5119e+02** | 2.7636e+02 | **1.5119e+02** | 2.7636e+02 | 1.5119e+02 | **1.5119e+02** | 0 |
| | Mean | 2.8630e+04 | 3.9598e+02 | 2.8630e+04 | 3.9598e+02 | 2.8630e+04 | 3.9598e+02 | 3.9598e+02 | |
| | Std Dev | 8.5402e+04 | 2.2181e+02 | 8.5402e+04 | 2.2181e+02 | 8.5402e+04 | 2.2181e+02 | 2.2181e+02 | |
| Rastrigin | Best | **2.3966e+01** | 2.7677e+01 | 2.3966e+01 | 2.7677e+01 | 2.3966e+01 | 2.7677e+01 | 2.7677e+01 | 0 |
| | Mean | 3.7232e+01 | 4.2162e+01 | 3.7232e+01 | 4.2162e+01 | 3.7232e+01 | 4.2162e+01 | 4.2162e+01 | |
| | Std Dev | 8.6681e+00 | 8.1255e+00 | 8.6681e+00 | 8.1255e+00 | 8.6681e+00 | 8.1255e+00 | 8.1255e+00 | |
| Ackley | Best | 1.9998e+01 | **1.6679e+00** | 1.9998e+01 | **1.6679e+00** | 1.9998e+01 | 1.6679e+00 | 1.6679e+00 | 0 |
| | Mean | 2.0000e+01 | 1.9077e+01 | 2.0000e+01 | 1.9077e+01 | 2.0000e+01 | 1.9077e+01 | 1.9077e+01 | |
| | Std Dev | 8.2108e-04 | 4.0976e+00 | 8.2108e-04 | 4.0976e+00 | 8.2108e-04 | 4.0976e+00 | 4.0976e+00 | |
| Bohachevsky1 | Best | **4.2316e-04** | 1.4268e-02 | 4.2326e-04 | 1.4277e-02 | 4.2326e-04 | 1.4277e-02 | 1.4287e-02 | 0 |
| | Mean | 4.6401e-03 | 3.0762e-02 | 4.6411e-03 | 3.0762e-02 | 4.6401e-03 | 3.0762e-02 | 3.0762e-02 | |
| | Std Dev | 6.4389e-03 | 1.3988e-02 | 6.4389e-03 | 1.3988e-02 | 6.4389e-03 | 1.3988e-02 | 1.3988e-02 | |
| Schaffer | Best | **1.4636e+02** | 1.6119e+02 | 2.7636e+02 | 1.6119e+02 | 2.7636e+02 | 1.6119e+02 | 1.6119e+02 | |
| | Mean | 2.8630e+04 | 3.9598e+02 | 2.8630e+04 | 3.9598e+02 | 2.8630e+04 | 3.9598e+02 | 3.9598e+02 | |
| | Std Dev | 8.5402e+04 | 2.2181e+02 | 8.5402e+04 | 2.2181e+02 | 8.5402e+04 | 2.2181e+02 | 2.2181e+02 | |
| Pathologic | Best | **4.0065e-03** | 1.6706e-01 | 4.0266e-03 | 1.6206e-01 | 4.0266e-03 | 1.6206e-01 | 1.6206e-01 | 0 |
| | Mean | 2.8503e-02 | 2.7535e-01 | 2.8503e-02 | 2.7535e-01 | 2.8503e-02 | 2.7535e-01 | 2.7535e-01 | |
| | Std Dev | 2.3109e-02 | 7.8468e-02 | 2.3109e-02 | 7.8468e-02 | 2.3109e-02 | 7.8468e-02 | 7.8468e-02 | |

### *3.2. Statistical Value Analysis on The Stability of mDOA*

The mDOA statistical value on common benchmark functions is shown in this subsection. For the unimodal and multimodal benchmark function, the P values attained by mDOA, DOA, WSO, SFO, SOA, GJO and AVOA are shown in Tables 6 and 7, along with comparisons with mDOA and other algorithms. Furthermore, Figures 3 and 4 show how mDOA stability performs best in comparison with ten classes of the benchmark functions. Figure 3 above for Sphere, Schwefel's 2.2, and Griewank unimodal standard show that the box plot of the DOA-Levy method has almost negligible spread and clusters tightly around very low Final Fitness levels, indicating far more stability than other algorithm approaches.

DOA-Standard, SOA, AVOA on the other hand, shows less stability because of its significant variability and considerable variation in Final Fitness among runs. While Table 6 provides comprehensive details about their P value. The P value noted in Table 6 is quite close to the ideal value attained by mDOA which shows that it is more stable compared to other algorithms with only exception in Greiwalk benchmark function. It's important to note that mDOA shows some superior exploitation potential for the Sphere and Schwefel's2.22 function.

The multimodal Standard functions distribution performance box stability depicted in Figure 4 include Rosenbrock, Rastrigin, Michalewicz, Bohachevsky1, Schaffer, Rosenbrock and Pathologic functions. The statistical analysis regarding their P values can be found in Table 7. The extremely small range of Final Fitness values around a mean close to 20 indicates that the DOA-Levy approach exhibits more stability. In contrast, WSO, SOA, GJO and AVOA exhibits far less stability across runs, with a much wider range in Final Fitness, from close to zero to roughly 20. Meanwhile, mDOA performed more stable in the whole run with the exception in two benchmark functions (Rosenbrock and Rastrign) which account for 71.4% performance output. A statistically significant difference in stability between the two approaches is confirmed by the modest p-value as highlighted. except for the Rastrigin and Rosenbrock function. The mDOA has better exploitative stability compared to standard DOA, WSO, SFO, SOA, GJO and AVOA. Also has better exploitation and exploration stability for the Step function.

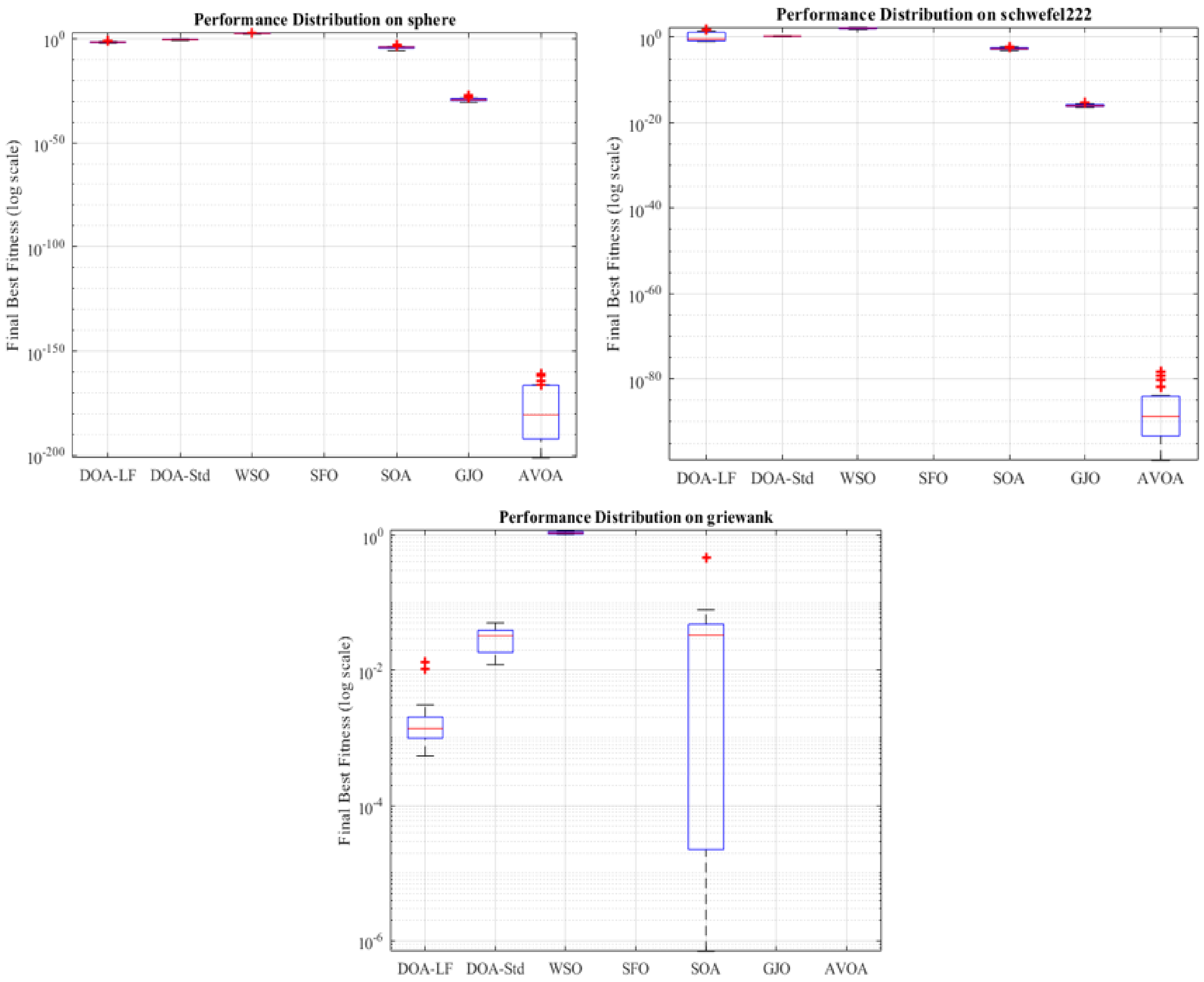


**Fig. 3**. Statistical graph of unimodal standard functions.

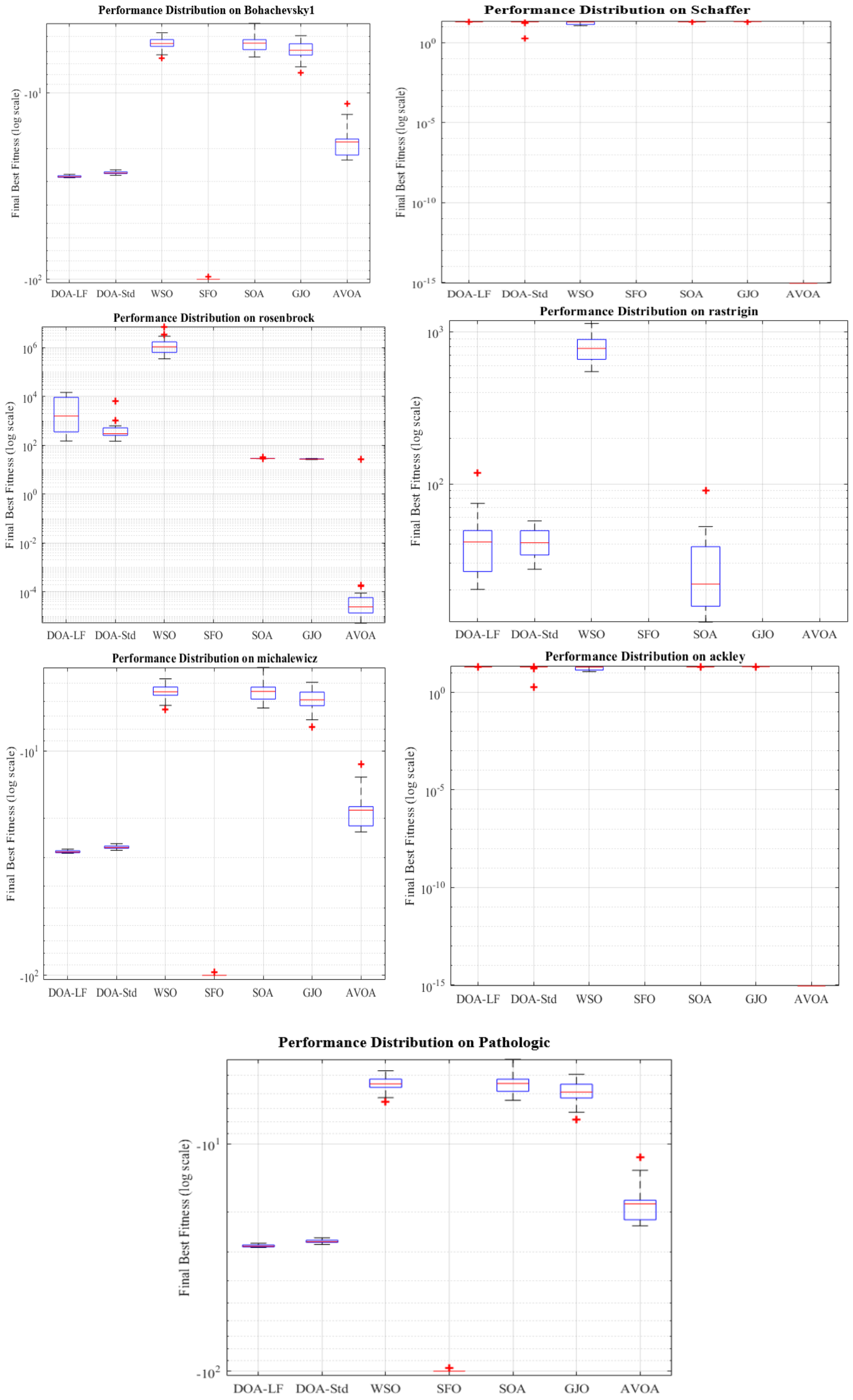


**Fig. 4**. Statistical graph of multimodal standard functions.

**Table 6.** Unimodal statistical P-Value

| **Function (Fn)** | **p-value(h=1)- wilcoxon** |
|---|---|
| Sphere | 6.7956e-08 |
| Schwefel2.2 | 6.7856e-08 |
| Griewank | 6.3803e-08 |

**Table 7.** Multimodal statistical P-value.

| **Function (Fn)** | **p-value(h=1)-wilcoxon** |
|---|---|
| Ackley | 1.7294e-07 |
| Michalewicz | 1.12668e-08 |
| Rosenbrock | 8.0065e-09 |
| Rastrigin | 8.0065e-09 |
| Bohachevsky1 | 1.22668e-08 |
| Schaffer | 6.4803e-08 |
| Pathologic | 6.2956e-08 |

### *3.3. Computational Complexity*

The computational complexity of proposed mDOA algorithm is:

$$k \leftarrow randominteger\ in\ \left[2.\max\left(2\left[\frac{D}{3}\right]\right)\right] \tag{10}$$

Whereas, the space complexity of all algorithms represents the maximum quantity of allocated space at any given moment, which is evaluated during the initialization phase. In this study, the space complexity for all methodologies is examined as $i = [0.9T] + 1 \rightarrow T$.

However, the mean execution duration of the proposed mDOA algorithm in comparison to other algorithms is delineated in Table 8. It is evident that mDOA exhibits a reduced temporal requirement relative to alternative methodologies, measured in seconds. Consequently, one may deduce that the computational efficiency of the proposed enhanced algorithm significantly surpasses that of its competing counterparts.

**Table 8.** Average running time of inproved mDOA and competitive approaches.

| **Algorithms** | **Average time (in seconds)** |
|---|---|
| Modified Dream Optimization Algorithm (mDOA) | 1.5245 |
| White Shark Optimization (WSO) | 1.8000 |
| Seagull Optimization Algorithm (SOA) | 1.7636 |
| Sunflower Optimization (SFO) | 1.6246 |
| Golden Jackal Optimization (GJO) | 2.7891 |
| African Vultures Optimization Algorithm (AVOA) | 2.7333 |

## 4. Conclusion

Using levy flight, this research introduces an enhance version of the Dream Optimisation Algorithm.The levy flight inherent in the position update phase was used to develop modified DOA algorithm called the mDOA algorithm. ten standard optimization functions were used to assess the improved mDOA's performance. Multimodal and unimodal standard optimization functions (Ackley, Griewangk, Michalwicz, Rastrigin Rosenbrock, Shwefel, Bohachevsky1 Schaffer, Pathologic and Sphere) comprise these functions.

According to the simulation results, mDOA outperformed and attaining the optimal global solution in 7 out of 10 cases, constituting 70.0% of the benchmark functions. it demonstrated 85% enhancement in its convergence towards global optima. Except for the Ackley and Rosenbrock function, where other algorithms have outperformed mDOA, this provides a greater capacity to escape local minima than the normal DOA, WSO. AVOA, GJO, SOA and SFO. Overall performance on the benchmark test suites the mDOA, showing better convergence, stability, significance, and reliability, over most of the algorithms compired with in this reserch.

The Modified Dream Optimization Algorithm (mDOA) presents a promising new metaheuristic that effectively balances exploration and exploitation through its novel memory and forgetting-inspired mechanisms, outperforming state-of-the-art algorithms on standard benchmarks. Its successful application to complex engineering problems, including photovoltaic cell parameter optimization, underscores its practical versatility and robustness.

Future research could explore hybridizing mDOA with other optimization techniques to further enhance convergence speed and solution quality, as well as extending its framework to multi-objective and dynamic optimization problems. Additionally, investigating adaptive mechanisms for memory and forgetting parameters may improve mDOA's flexibility across diverse problem domains.

## Author Contributions

K.L. and D.D. carried out the full research work, including problem formulation, system modelling, simulation, data analysis, and manuscript preparation. A.S., K.O., and A.E. provided academic supervision, technical guidance, and critical review of the work. U.Y. and J.E. contributed through discussions, assistance with simulations, and review of the results. All authors reviewed and approved the final version of the manuscript.

## Acknowledgements

Not applicable.

## Conflict of Interest

The authors declare no conflict of interest.

## References

[1] X.-S. Yang, "Nature-inspired metaheuristic algorithms: Success and new challenges," 2012, arXiv:1211.6658.

[2] S. Yılmaz and E. U. Küçüksille, "A new modification approach on bat algorithm for solving optimization problems," *App Soft Comp*, vol. 28, pp. 259–275, 2015.

[3] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimization," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010.

[4] M. Shehab, A. T. Khader, and M. A. Al-Betar, "A survey on applications and variants of the cuckoo search algorithm," *App Soft Comp*, 2017.

[5] P. Civicioglu and E. Besdok, "A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," *Artif. Intell. Rev.*, pp. 1–32, 2013.

[6] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proc. World Congress on Nature & Biologically Inspired Computing*, 2009.

[7] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. IEEE Int. Symp. Micro Machine and Human Science*, 1995.

[8] V. K. Pathak and A. K. Singh, "Optimization of morphological process parameters in contactless laser scanning system using modified particle swarm algorithm," *Measurement*, 2017.

[9] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Proc. International Symposium on Stochastic Algorithms*, 2019.

[10] S.-C. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization," *Pacific Rim Conf. Artif. Intell.*, 2016.

[11] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization*, pp. 65–74, Springer, 2010.

[12] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA: MIT Press, 2012.

[13] Y. Lang and Y. Gao, "Dream Optimization Algorithm (DOA): A novel metaheuristic optimization algorithm inspired by human dreams and its applications to real-world engineering problems," *Comput. Methods Appl. Mech. Eng.*, vol. 436, 2025.

[14] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: An overview," *Soft Computing*, vol. 22, pp. 387–408, 2018.

[15] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.

[16] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: Sparrow search algorithm," *Syst. Sci. Control Eng.*, vol. 8, no. 1, pp. 22–34, 2020.

[17] P. Duankhan, K. Sunat, S. Chiewchanwattana, and P. Nasa-ngium, "The differentiated creative search (DCS): Leveraging differentiated knowledge-acquisition and creative realism to address complex optimization problems," *Expert Syst. Appl.*, 2024, Art. no. 123734.

[18] Z. Guan, C. Ren, J. Niu, P. Wang, and Y. Shang, "Great Wall Construction Algorithm: A novel meta-heuristic algorithm for engineering problems," *Expert Syst. Appl.*, vol. 233, p. 120905, 2023.

[19] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, 2016.

[20] A. T. Salawudeen, M. B. Mu'azu, A. Yusuf, and A. E. Adedokun, "A novel smell agent optimization (SAO): An extensive CEC study and engineering application," *Knowledge-Based Systems*, vol. 232, p. 107486, 2021.

[21] Y. Wu, X. Z. Gao, and K. Zenger, "Knowledge-based artificial fish swarm algorithm," *18th IFAC World Congress*, Milano, Italy, 2011.

[22] K. Tang, X. Yao, P. N. Suganthan, C. MacNish, Y.-P. Chen, C.-M. Chen, and Z. Yang, "Benchmark functions for the CEC 2008 special session and competition on large-scale global optimization," Nature Inspired Computation and Applications Laboratory, China, 2007.

[23] J. Momin and X.-S. Yang, "A literature survey of benchmark functions for global optimization problems," *Int. J. Math. Model. Numer. Optim.*, vol. 4, no. 2, pp. 150–194, 2013.

[24] Y. Wang, J. Ma, and Y. Wang, "Application of ant colony algorithm in path planning of the data center room robot," in *AIP Conf. Proc.*, vol. 1839, no. 1, p. 020193, 2017.

[25] N. Hansen, "Compilation of results on the 2005 CEC benchmark function set," 2006.

[26] A. T. Salawudeen, "Development of an improved cultural artificial fish swarm algorithm with crossover," M.S. thesis, Ahmadu Bello Univ., Zaria, Nigeria, 2015.

[27] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization*, Studies in Computational Intelligence, vol. 284, Springer, 2010.

[28] X. Li, K. Tang, M. N. Omidvar, Z. Yang, and K. Qin, "Benchmark functions for the CEC 2013 special session and competition on large-scale global optimization," Technical Report, Evolutionary Computation and Machine Learning Group, RMIT University, Melbourne, Australia, 2013.

[29] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "A novel population initialization method for accelerating evolutionary algorithms," *Comput. Math. Appl.*, vol. 53, no. 10, pp. 1605–1614, 2007.