



OPEN ACCESS

EDITED BY
Betül Yıldız,
Bursa Uludağ Üniversitesi, Türkiye

REVIEWED BY
Zakariya Yahya Algamal,
University of Mosul, Iraq
Lucas Wekesa,
Strathmore University Institute of
Mathematical Sciences, Kenya

*CORRESPONDENCE
Aram Kamal Faraj
✉ aram.kamal@spu.edu.iq
Aso M. Aladdin
✉ aso.aladdin@chu.edu.iq

†PRESENT ADDRESS
Aso M. Aladdin
Department of Computer Engineering,
Tishk International University, Erbil, KR, Iraq

RECEIVED 03 June 2025
ACCEPTED 17 September 2025
PUBLISHED 17 October 2025

CITATION
Faraj AK, Aladdin AM and Ameen AA (2025)
EESB-FDO: enhancing the fitness-dependent
optimizer through a modified boundary
handling mechanism.
Front. Appl. Math. Stat. 11:1640044.
doi: 10.3389/fams.2025.1640044

COPYRIGHT
© 2025 Faraj, Aladdin and Ameen. This is an
open-access article distributed under the
terms of the [Creative Commons Attribution
License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or
reproduction in other forums is permitted,
provided the original author(s) and the
copyright owner(s) are credited and that the
original publication in this journal is cited, in
accordance with accepted academic practice.
No use, distribution or reproduction is
permitted which does not comply with these
terms.

EESB-FDO: enhancing the fitness-dependent optimizer through a modified boundary handling mechanism

Aram Kamal Faraj^{1,2*}, Aso M. Aladdin ^{1†} and Azad A. Ameen³

¹Department of Computer Science, College of Science, Charmo University, Sulaimani, Chamchamal, KR, Iraq, ²Department of IT, Technical College of Informatics, Sulaimani Polytechnic University, Sulaimani, KR, Iraq, ³Department of Software Engineering, College of Engineering and Computational Science, Charmo University, Chamchamal, Iraq

The fitness-dependent optimizer (FDO) has recently gained attention as an effective metaheuristic for solving different optimization problems. However, it faces limitations in exploitation and convergence speed. To overcome these challenges, this study introduces two enhanced variants: enhancing exploitation through stochastic boundary for FDO (EESB-FDO) and enhancing exploitation through boundary carving for FDO (EEBC-FDO). In addition, the ELFS strategy is proposed to constrain Levy flight steps, ensuring more stable exploration. Experimental results show that these modifications significantly improve the performance of FDO compared to the original version. To evaluate the performance of the EESB-FDO and EEBC-FDO, three primary categories of benchmark test functions were utilized: classical, CEC 2019, and CEC 2022. The assessment was further supported by the application of statistical analysis methods to ensure a comprehensive and rigorous performance evaluation. The performance of the proposed EESB-FDO and EEBC-FDO algorithms was evaluated through comparative analysis with several existing FDO modifications, as well as with other well-established metaheuristic algorithms, including the Arithmetic Optimization Algorithm (AOA), the Learner Performance-Based Behavior Algorithm (LPB), the Whale Optimization Algorithm (WOA), and the Fox-inspired Optimization Algorithm (FOX). The statistical analysis indicated that both EESB-FDO and EEBC-FDO exhibit better performance compared to the aforementioned algorithms. Furthermore, a final evaluation involved applying EESB-FDO and EEBC-FDO to four real-world optimization problems: the gear train design problem, the three-bar truss problem, the pathological igg fraction in the nervous system, and the integrated cyber-physical attack on a manufacturing system. The results demonstrate that both proposed variants significantly outperform both the FDO and the modified fitness-dependent optimizer (MFDO) in solving these complex problems.

KEYWORDS

stochastic boundary, optimization, fitness dependent optimizer, metaheuristic algorithm, Levy flight

1 Introduction

In real-world scenarios, optimization plays a vital role in solving complex problems across diverse fields such as engineering design, healthcare, transportation, energy systems, and machine learning. Many of these problems are difficult to solve using traditional mathematical approaches, particularly when the search space is large, non-linear, or contains multiple local optima.

Metaheuristic algorithms have therefore emerged as powerful tools, providing efficient and flexible methods to approximate optimal solutions in practical applications [1–4]. Over the past two decades, metaheuristic algorithms have gained prominence due to their ability to handle complex, high-dimensional problems. They generally operate by applying strategic rules to candidate solutions, starting from randomly generated points and iteratively improving them through evaluation and comparison [5]. One of the earliest milestones in this field was John Holland's genetic algorithm (GA), developed in the 1960s and later published in the 1970s and 1980s [6–8]. This was followed by simulated annealing (SA) [9], inspired by the annealing process in metallurgy, and particle swarm optimization (PSO) [5, 10], inspired by the collective behavior of birds. These pioneering algorithms demonstrated strong potential for addressing real-world optimization challenges by finding high-quality solutions, reducing computational time, and overcoming local optima.

Since then, numerous algorithms have been introduced, such as the pathfinder algorithm (PFA) [11], the mayfly algorithm (MA) [12], the bear smell search algorithm (BSSA) [13], the gradient-based optimizer (GBO) [14], the group teaching optimization algorithm (GTOA) [15], the heap-based optimizer (HBO) [16], the Henry gas solubility optimization (HGSO) [17], the marine predators algorithm (MPA) [18], the political optimizer (PO) [19], the atom search optimization (ASO) [20], the Aquila optimizer (AO) [21], the herd optimization algorithm (HOA) [22], the Ebola optimization search algorithm (EOSA) [23] and the single candidate optimizer (SCO) [24]. The large number of algorithms can be attributed to the fact that no specific algorithm can be applied to all optimization problems, as pointed out in the no free lunch (NFL) theorem [25]. This theorem has logically demonstrated that no single algorithm is universally optimal for solving all types of optimization problems. The outcomes of these algorithms show that some achieve better results than others for certain specific problems [26].

A recurring theme in the design of these algorithms is the balance between exploration and exploitation [27–29]. Achieving a balance between these concepts is critical for improving algorithmic performance, as it enhances the ability of algorithms to converge efficiently while avoiding local optima. One of the primary limitations of the fitness-dependent optimizer (FDO) algorithm, as identified in previous studies, is its underdeveloped exploration capability, which has prompted significant research efforts aimed at addressing this issue. This paper emphasizes improving exploitation while maintaining a strong focus on exploration. To confine the algorithm within a specified search space defined by lower and upper bounds after updating the bees' positions, the FDO algorithm utilizes a specialized function called the “getBoundary” function [30, 31]. This function effectively transforms outlier values of bee positions into feasible values within the defined boundaries. By introducing modifications to the equations within the “getBoundary” function, this study aims to enhance the FDO algorithm's performance, contributing to more efficient optimization outcomes.

The main contributions of this study can be summarized as follows:

1. Proposed two novel boundary handling strategies for the Fitness fitness-dependent optimizer (FDO):
 - EESB-FDO: Enhancing exploitation through stochastic repositioning, which introduces random values within bounds when scout bees exceed search space limits.
 - EEBC-FDO: Enhancing exploitation through boundary carving, which redirects bees toward feasible regions using boundary carving equations.
2. Introduced the ELFS strategy, which modifies the Levy flight mechanism to restrict step sizes within a bounded range, preventing instability due to excessive jumps.
3. Conducted extensive benchmarking using classical, CEC 2019, and CEC 2022 test functions, demonstrating significant improvements in exploitation capability and convergence behavior.
4. Applied the proposed variants to four real-world engineering problems, proving their practical effectiveness in solving complex, constrained optimization scenarios.
5. Performed comparative statistical analysis with state-of-the-art algorithms, confirming the competitiveness and reliability of EESB-FDO and EEBC-FDO under various optimization challenges.

The remaining sections of this paper are organized as follows: Section 2 presents a review of related work on the FDO. Section 3 provides a detailed description of the FDO algorithm. In Section 4, the proposed methodology is explained, followed by the introduction of the FDO modifications in Section 5. Section 6 presents the experimental results and comparisons with other algorithms using a set of benchmark test functions. Finally, Section 7 summarizes the main findings and offers directions for future research.

2 Related work

Advancing the FDO recently and effectiveness in solving real-world problems demonstrate that it is one of the most powerful metaheuristic algorithms in recent years [32–35]. Additionally, it uses random walk techniques, such as Levy flight [3], which have been employed in prior studies to improve the performance of the optimization of other algorithms. Selecting an appropriate equation or method is crucial for efficiently reaching the global optimum while avoiding local optima. Following the development of any new algorithm, the modification and enhancement of existing

TABLE 1 FDO parameter settings.

Parameters	Parameter value
Search agent	30
Weight factor	Either 0 or 1, FDO use “0” in all testing
Fitness weight	Range between (0,1)
Levy flight	Range between (1,–1)

algorithms using various methods and techniques are critical steps commonly undertaken by researchers to improve performance and adaptability. For example, the GA [36] has been extended to the

adaptive genetic algorithm (AGA) [37], and PSO [10] has evolved into the comprehensive learning PSO (CLPSO) [38]. Similarly, the ant colony optimization (ACO) [39] method was improved to the max–min ant system (MMAS) [40], while the FA [2] was adapted into the adaptive firefly algorithm (AFA) [41], and AOA [42] was refined into the enhanced AOA (EAOA) [43].

The enhancement and generation of algorithms are increasing rapidly; this rapid expansion has necessitated the need to manage and categorize them. There are various ways to classify these algorithms, and one common approach is based on their source of inspiration, which can be either natural or non-natural. Nature-inspired metaheuristic algorithms can be broadly categorized into four main classes: physics-based algorithms, evolutionary algorithms, human-based algorithms, and swarm-based algorithms [44]. The FDO is a metaheuristic algorithm that belongs to the swarm-based algorithms group, inspired by the bee swarming reproductive process. More specifically, it can be seen as an extension of PSO [31]. Various methods and techniques have been employed to enhance and improve the performance of the FDO algorithm, and these improvements must be acknowledged. Adding two parameters (alignment and cohesion) to update the position of search agents in addition to the existing pace factor [32]. Ten chaotic maps have been used for the initialization population, which is another improvement implemented in FDO [45]. After that, updating pace (velocity) by utilizing the sine-cosine scheme, modified pace-updating equations in the search phase, random weight factor, global fitness weight strategy, conversion parameter strategy, and the best solution-updating strategy [33]. Then, modifying the fitness-dependent optimizer by changing waited factor range from [0,1] to [0,0.2] is another mechanism [34].

With these developments and improvements, the FDO algorithm has been effectively applied to solve and address a variety of problems, demonstrating significant success in finding optimal solutions. For instance, the adaptive fitness-dependent optimizer has been applied to the one-dimensional bin packing problem, an NP-hard problem, producing more promising results compared to other approaches [46]. Subsequently, the automatic generation control (AGC) of a multi-source interconnected power system (IPS) was more effectively managed using FDO, reducing frequency

```

1: initialize scout bee population  $X_{t,i}$  ( $i = 1, 2, \dots, n$ )
2: while iteration (t) limit not reached
3:   for each artificial scout bee  $X_{t,i}$ 
4:     find best artificial scout bee  $X_{t,i}^*$ 
5:     generate random-walk  $r$  in  $[-1, 1]$  range
6:     if(  $X_{t,i} \text{ fitness} == 0$ ) (avoid divide by zero)
7:       fitness weight = 0
8:     else
9:       calculate fitness weight. equation (2)
10:    end if
11:    if (fitness weight = 1 or fitness weight = 0)
12:      calculate pace using equation (3)
13:    else
14:      if (random number  $\geq 0$ )
15:        calculate pace using equation (5)
16:      else
17:        calculate pace using equation (4)
18:      end if
19:    end if
20:    calculate  $X_{t+1,i}$  equation (1)
21:    if(  $X_{t+1,i} \text{ fitness} < X_{t,i} \text{ fitness}$ )
22:      move accepted and pace saved
23:    else
24:      calculate  $X_{t+1,i}$  equation (1) ...
25:      ... with previous pace
26:      if(  $X_{t+1,i} \text{ fitness} < X_{t,i} \text{ fitness}$ )
27:        move accepted and pace saved
28:      else
29:        maintain current position (don't move)
30:      end if
31:    end if
32:  end for
33: end while

```

FIGURE 1
Pseudo code of FDO described in Abdullah and Ahmed [31].

Methodology

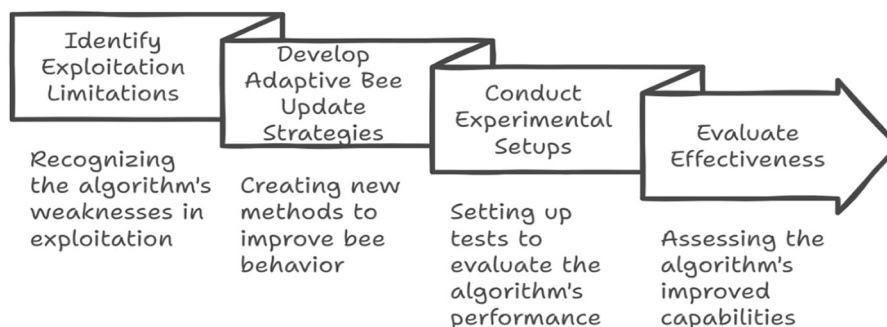


FIGURE 2
Explain methodology steps.

A	B
1: // Initialization Random value in predefined search space	1: // Initialization Random value in predefined search space
2: initialize scout bee population $X_{t,i}$ ($i = 1, 2, \dots, n$)	2: initialize scout bee population $x_{t,i}$ ($i = 1, 2, \dots, n$)
3: //Iteration Process until predefine condition	3: //Iteration Process until predefine condition
4: while iteration (t) limit not reached	4: while iteration (t) limit not reached
5: for each artificial scout bee $X_{t,i}$	5: for each artificial scout bee $X_{t,i}$
6: find best artificial scout bee $X_{t,i}^*$	6: find best artificial scout bee $X_{t,i}^*$
7: generate random-walk r in $[-1, 1]$ range	7: generate random-walk r in $[-1, 1]$ range
8: //Fitness Evaluation and Weighting	8: //Fitness Evaluation and Weighting
9: if ($X_{t,i}$ fitness == 0) (avoid divide by zero)	9: if ($X_{t,i}$ fitness == 0) (avoid divide by zero)
10: fitness weight = 0	10: fitness weight = 0
11: else	11: else
12: calculate fitness weight. equation (2)	12: calculate fitness weight. equation (2)
13: end if	13: end if
14: if (fitness weight = 1 or fitness weight = 0)	14: if (fitness weight = 1 or fitness weight = 0)
15: calculate pace using equation (3)	15: calculate pace using equation (3)
16: else	16: else
17: if (random number >= 0)	17: if (random number >= 0)
18: calculate pace using equation (5)	18: calculate pace using equation (5)
19: else	19: else
20: calculate pace using equation (4)	20: calculate pace using equation (4)
21: end if	21: end if
22: end if	22: end if
23: calculate $X_{t+1,i}$ equation (1)	23: calculate $X_{t+1,i}$ equation (1)
24: // looking for exceed value after update	24: // looking for exceed value after update
25: check boundary search space if $X_{t+1,i}$ exceed use eq (8)	25: check boundary search space if $X_{t+1,i}$ exceed use eq (9) or (10)
26: if ($X_{t+1,i}$ fitness < $X_{t,i}$ fitness)	26: if ($X_{t+1,i}$ fitness < $x_{t,i}$ fitness)
27: move accepted and pace saved	27: move accepted and pace saved
28: else	28: else
29: calculate $X_{t+1,i}$ equation (1) ...	29: calculate $X_{t+1,i}$ equation (1) ...
30: ... with previous pace	30: ... with previous pace
31: // looking for exceed value after update	31: // looking for exceed value after update
32: check boundary search space if $X_{t+1,i}$ exceed use eq (8)	32: check boundary search space if $X_{t+1,i}$ exceed use eq (9) or (10)
33: if ($X_{t+1,i}$ fitness < $X_{t,i}$ fitness)	33: if ($X_{t+1,i}$ fitness < $X_{t,i}$ fitness)
34: move accepted and pace saved	34: move accepted and pace saved
35: else	35: else
36: maintain current position (don't move)	36: maintain current position (don't move)
37: end if	37: end if
38: end if	38: end if
39: end for	39: end for
40: end while	40: end while

FIGURE 3

Pseudo code of FDO with (EESB) and (EEBC). (A) EESB-FDO. (B) EEBC-FDO.

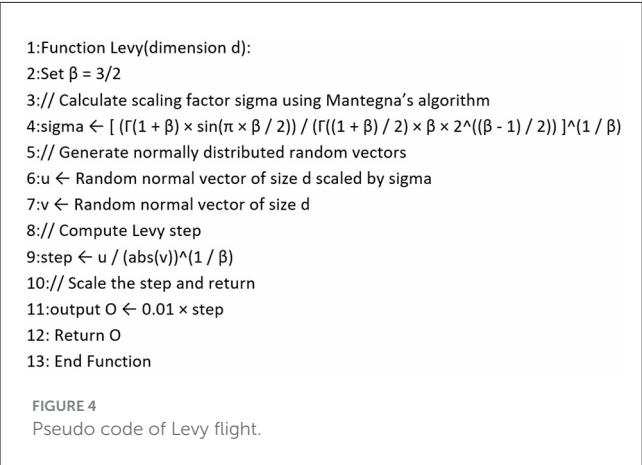
overshoot/undershoot and settling time [47]. Then Improved-FDO was applied to control the same problem and provide better results than FDO [48]. Building on the advantages of FDO in controlling Automatic Generation Control (AGC), a new controller called the “modified PID controller” is introduced, which achieves more effective control than before, again utilizing FDO [49]. The performance of this algorithm was further leveraged to enhance another area, specifically in multilayer perceptron (MLP) neural networks, resulting in an increase in the average accuracy of the model [50]. FDO has demonstrated superior efficiency in solving non-linear optimal control problems (NOCs) compared to the GA algorithm, effectively minimizing absolute error and providing better solutions [51].

A new technique called generalized regression neural network combined with fitness-dependent optimization (GRNNFDO) has been used to control the behavior of the thermoelectric generator

(TEG) system, and this technique produces more power compared to other algorithms [52]. The economic load dispatch (ELD) problem is another area where the improved FDO has been used to effectively reduce emission allocation and fuel costs [53]. Another effective method for classifying COVID-19 cases as positive or negative, where FDO combined with neural networks was tested on three different datasets, enhanced accuracy rates and minimized error rates [54]. A hybrid approach combining genetic algorithm and FDO (GA-MFDO) has been used to solve the NP-hard problem of workflow scheduling, achieving a significant p -value compared to other methods [55]. Similarly, the edge server allocation problem, an NP-hard challenge, was effectively addressed using the effective fitness-dependent optimizer (EFDO). When tested on Shanghai Telecom’s dataset, which includes factors such as access delay, energy consumption, and workload balance, EFDO outperformed competing algorithms [56]. Finally, FDO integrated

with neural networks has also been employed for detecting steel plate faults, achieving higher classification accuracy than other methods [57].

Levy flight is a type of random walk where step lengths follow a Levy distribution, characterized by its heavy-tailed probability distribution. It is one of the most effective approaches for generating a diverse and well-distributed random initialization set, thereby enhancing the exploration capabilities of the algorithm when metaheuristic algorithms begin with a randomly initialized set of values, structured according to the algorithm’s design and procedural steps [2]. Levy flight is a fundamental component of the cuckoo search (CS) algorithm, which mimics the foraging behavior of certain animals and insects [58]. In this work, Levy flight is utilized to simulate insect movement, significantly boosting the global search capability of the flower pollination algorithm (FPA) [59]. For algorithms like the dragonfly algorithm (DA), Levy flight guides stochastic behavior, allowing for better traversal of the search space [60]. Additionally, the combination of differential evolution (DE) and Levy flight (LF) techniques has been shown to improve the performance of the harmony search algorithm (HSA) [61]. The slap swarm algorithm (SSA) also benefits from Levy flight mutation, which enhances its randomness and exploration capabilities during the search process [62]. Similarly, Levy flight-based mutation has been introduced to improve the exploration ability and overall efficiency of the slime mould algorithm (SMA) [63]. Furthermore, the use of Levy flight in the bat algorithm (BA) has been shown to improve its ability to escape local optima and achieve faster convergence [64]. Similarly, the incorporation of Levy flight in the gray wolf optimizer (GWO) enhances its global search ability, making it more effective for solving complex optimization problems [65].



3 FDO algorithm

The FDO is inspired by the swarming behavior of bees during colony reproduction. When bees seek to establish a new hive, they follow a systematic set of rules to identify a suitable location. These decision-making processes within the bee community can be categorized into two fundamental components: searching for an appropriate site and moving toward it. Further details regarding FDO are provided through its parameter settings, as summarized in Table 1. These parameters significantly influence the algorithm’s performance, and deviations beyond their predefined values may adversely affect the quality of the results.

In optimization algorithms, the search process is typically conducted within a predefined search space to identify optimal solutions. To ensure that the search remains within these boundaries, some algorithms implement a boundary handling mechanism. This mechanism involves iterating through potential solutions and adjusting any values that exceed the predefined limits by reassigning them to a random boundary value. This section provides a detailed exploration of these three key aspects: searching, movement, and boundary handling. Showed Pseudo code of FDO in Figure 1.

3.1 Bee searching

Scout bees initiate the search for an optimal hive by exploring various potential locations. This process represents a fundamental stage in the algorithm, which begins with the random generation of the scout bee population within a predefined search space, denoted as X_i (where $i = 1, 2, 3, \dots, n$). Each scout bee’s position corresponds to a newly discovered solution within the search space. To identify a more suitable hive, scout bees perform a stochastic search for alternative locations. If a newly identified position offers a more favorable solution, the previous locations are disregarded. Conversely, if the new position does not yield an improvement, the scout bees continue moving in their prior direction while disregarding the less optimal locations.

3.2 Bee movement

In the natural world, scout bees search for a hive randomly to identify a more suitable location for reproduction. Similarly, artificial scout bees adopt this principle by moving and updating their current position to a potentially improved one through the addition of a pace value. If the newly discovered solution proves to be superior to the previous one, the scouts proceed in the

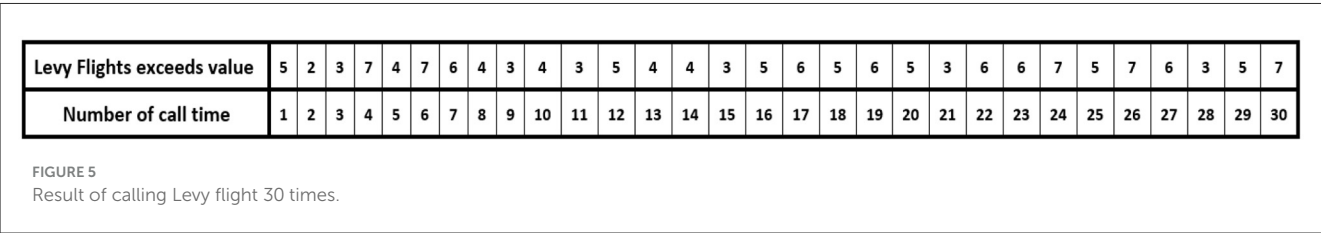


TABLE 2 Unimodal benchmark functions [80].

Function	Range	fmin
$TF1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]$	0
$TF2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_{ij} $	$[-10, 10]$	0
$TF3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	$[-100, 100]$	0
$TF4(x) = \max\{ x , 1 \leq i \leq n\}$	$[-100, 100]$	0
$TF5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_{i-1})^2 \right]$	$[-30, 30]$	0
$TF6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	$[-100, 100]$	0
$TF7(x) = \sum_{i=1}^n ix_i^2 + \text{random}[0, 1]$	$[-1.28, 1.28]$	0

TABLE 3 Multi-modal benchmark functions [80].

Function	Range	Shift position	fmin
$TF8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	$[-500, 500]$	$[-300, \dots -300]$	-418.9829
$TF9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	$[-2, -2, \dots -2]$	0
$TF10(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]$		0
$TF11(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]$	$[-400, \dots -400]$	0
$TF12(x) = \frac{\pi}{n} \left\{ 10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 \left[1 + 10 \sin^2(\pi y_{i+1}) \right] + (y_{n-1})^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4} \cdot u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-50, 50]$	$[-30, 30, \dots 30]$	0
$TF13(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 \left[1 + \sin^2(3\pi x_i + 1) \right] + (x_n - 1)^2 \left[1 + \sin^2(2\pi x_n) \right] \right\} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	$[-50, 50]$	$[-100, \dots -100]$	0

TABLE 4 Fixed-dimension multimodal benchmark functions [80].

Function	Dim	Range	fmin
$TF14(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{25} (x_i - a_{ij})^6} \right)^{\frac{1}{25}}$	2	$[-65, 65]$	1
$TF15(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	$[-5, 5]$	0.00030
$TF16(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]$	-1.0316
$TF17(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	$[-5, 5]$	0.398
$TF18(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]$	3
$TF19(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2\right)$	3	$[1, 3]$	-3.86
$TF20(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2\right)$	6	$[0, 1]$	-3.32
$TF21(x) = -\sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.1532
$TF22(x) = -\sum_{i=1}^7 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.4028
$TF23(x) = -\sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]$	-10.5363

current direction; otherwise, they continue following their prior trajectory. The movement of artificial scout bees is determined using Equation 1.

$$X_{i,t+1} = X_{i,t} + \text{Pace} \quad (1)$$

where i represents the current search agent, t represents the current iteration, x represents an artificial scout bee (search agent),

and pace is the movement rate and direction of the artificial scout bee. The fitness weight (fw) as expressed in Equation 2 is used to calculate and manage the pace when the direction of pace is completely dependent on a random mechanism.

$$fw = \left| \frac{X_{i,t}^* \text{fitness}}{X_{i,t} \text{fitness}} \right| - wf \quad (2)$$

TABLE 5 Results of comparing EESB-FDO with FDO and other chosen algorithms using classical benchmark functions.

TF	Metrics	EESB-FDO	FDO	LPB	WOA	GSA	GWO
F1	Mean	1.53E-22	1.74E-18	6.27E-02	9.53E-76	1.83E-02	1.34E-27
	SD	7.14E-22	9.39E-18	5.06E-02	5.11E-75	9.86E-02	1.71E-27
F2	Mean	2.68E+00	2.34E+00	3.58E-02	1.02E-52	5.97E-02	9.16E-17
	SD	7.49E-01	6.92E-01	1.44E-02	5.15E-52	1.40E-01	6.35E-17
F3	Mean	2.23E-08	7.27E-08	2.06E+02	1.38E+02	9.95E+02	1.25E-05
	SD	4.13E-08	2.61E-07	1.43E+02	2.92E+02	4.74E+02	2.83E-05
F4	Mean	6.43E-08	6.90E-08	7.17E-01	4.05E+00	6.96E+00	1.01E-06
	SD	1.96E-07	5.88E-07	2.37E-01	6.31E+00	2.12E+00	1.09E-06
F5	Mean	1.46E+01	2.62E+01	5.43E+01	6.95E+00	9.31E+01	2.70E+01
	SD	2.96E+01	3.88E+01	3.78E+01	6.51E-01	1.22E+02	8.67E-01
F6	Mean	9.49E-19	1.30E-17	4.83E-02	9.19E-03	8.10E+00	7.64E-01
	SD	5.19E-18	7.13E-17	4.44E-02	4.13E-02	1.17E+01	3.50E-01
F7	Mean	6.29E-01	6.33E-01	1.57E-02	2.69E-03	8.01E-02	1.86E-03
	SD	3.12E-01	3.36E-01	9.18E-03	3.24E-03	4.31E-02	9.14E-04
F8	Mean	−4.23E+03	−8.30E+03	−3.71E+03	−3.17E+03	−2.55E+03	−5.80E+03
	SD	3.96E+03	1.14E+04	2.43E+02	1.28E+03	3.79E+02	9.72E+02
F9	Mean	7.34E+00	7.05E+00	3.51E-02	2.69E+00	2.80E+01	2.32E+00
	SD	4.00E+00	3.11E+00	2.73E-02	8.84E+00	8.81E+00	3.70E+00
F10	Mean	7.07E-15	6.95E-15	1.19E-01	9.35E-15	1.29E-08	1.05E-13
	SD	1.21E-15	2.61E-15	6.49E-02	1.92E-14	3.83E-09	1.82E-14
F11	Mean	1.31E-01	1.56E-01	1.96E-01	6.21E-02	2.80E+01	2.84E- 03
	SD	9.74E-02	9.93E-02	1.75E-01	1.05E-01	6.13E+00	7.18E- 03
F12	Mean	8.66E+00	8.02E+00	3.01E-03	4.64E-03	1.62E+00	4.44E-02
	SD	4.29E+00	5.60E+00	6.13E-03	7.75E-03	1.01E+00	2.29E-02
F13	Mean	3.37E+00	3.70E+00	7.09E-03	1.27E-01	8.03E+00	6.46E-01
	SD	5.45E+00	4.10E+00	6.12E-03	1.82E-01	4.63E+00	2.01E-01
F14	Mean	2.52E+00	2.72E+00	9.98E-01	4.75E+00	5.09E+00	4.65E+00
	SD	1.32E+00	1.46E+00	4.44E-16	4.07E+00	4.24E+00	4.37E+00
F15	Mean	8.24E-04	1.35E-03	8.63E-03	1.04E-03	4.49E-03	3.80E-03
	SD	5.86E-04	2.06E-03	9.32E-03	1.63E-03	2.24E-03	7.41E-03
F16	Mean	−1.03E+00	−1.03E+00	−1.03E+00	−9.97E-01	−1.03E+00	−1.03E+00
	SD	6.66E-16	6.66E-16	6.66E-16	1.85E-01	6.66E-16	6.66E-16
F17	Mean	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	SD	1.67E-16	1.67E-16	3.16E-06	1.91E-05	1.67E-16	2.49E-06
F18	Mean	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	SD	0.00E+00	0.00E+00	1.23E-04	5.12E-05	0.00E+00	4.00E-05
F19	Mean	−3.86E+00	−3.86E+00	−3.20E+00	−3.85E+00	−3.86E+00	−3.86E+00
	SD	3.11E-15	3.11E-15	1.33E-15	1.89E-02	3.11E-15	2.32E-03
F20	Mean	−3.25E+00	−2.86E+00	−3.25E+00	−3.22E+00	−3.32E+00	−3.27E+00
	SD	8.41E-02	1.14E+00	5.89E-02	8.60E-02	1.78E-15	6.69E-02
F21	Mean	−5.05E+00	−5.05E+00	−5.39E+00	−8.94E+00	−5.94E+00	−8.55E+00
	SD	5.33E-04	3.61E-04	3.25E+00	2.23E+00	3.69E+00	2.48E+00
F22	Mean	−5.24E+00	−4.70E+00	−5.56E+00	−8.86E+00	−1.04E+01	−1.02E+01
	SD	3.09E+00	3.38E+00	3.23E+00	2.78E+00	0.00E+00	9.54E-01

(Continued)

TABLE 5 (Continued)

TF	Metrics	EESB-FDO	FDO	LPB	WOA	GSA	GWO
F23	Mean	−4.66E+00	−4.32E+00	−6.56E+00	−6.75E+00	−9.93E+00	−1.02E+01
	SD	3.30E+00	2.91E+00	3.76E+00	4.60E+00	2.49E+00	1.35E+00

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

where $X_{i,t}^{* fitness}$ denotes the best global solution's fitness function value that has been revealed so far; $X_{i,t}^{* fitness}$ denotes the current solution's value of the fitness function; and wf denotes a weight factor, randomly set between 0 and 1, which is used for controlling the fw .

Later, the algorithm considers some settings for fw , for instance, if $fw = 1$ or 0, and $X_{i,t}^{* fitness} = 0$, the algorithm sets the pace randomly according to Equation 3. On the other hand, if $fw > 0$ and $fw < 1$, then the algorithm generates a random number in the $(-1, 1)$ range to make the scout search in every direction; when $r < 0$, pace is calculated according to Equation 4, and when $r \geq 0$, pace is calculated according to Equation 5.

$$\begin{cases} fw = 1 \text{ or } fw = 0 \text{ or } X_{i,t}^{* fitness} = 0, \text{ pace} = X_{i,t}^{*} r & (3) \\ fw > 0 \text{ and } fw < 1 \begin{cases} r < 0, \text{ pace} = (X_{i,t} - X_{i,t}^{*}, t) fw - 1 & (4) \\ r \geq 0, \text{ pace} = (X_{i,t} - X_{i,t}^{*}, t)^{*} fw & (5) \end{cases} \end{cases}$$

where r denotes a random number in the range of $[-1, 1]$, $X_{i,t}$ denotes the current solution, and $X_{i,t}^{*}$ denotes the global best solution achieved thus far. Among various applications for random numbers, the FDO selects Levy flight because it considers further stable movement via its fair distribution curve [10].

3.3 Bee boundary

In optimization algorithms, the search space refers to the set of all possible solutions within which the algorithm seeks to identify the optimal solution. The algorithm explores this space to determine the most suitable outcome. Each algorithm operates within a defined search space, constrained by specific boundaries and limitations to ensure both efficiency and feasibility. The performance of an algorithm is significantly influenced by the structure of the search space, and maintaining the algorithm's operations within the prescribed boundaries is crucial for its success. For instance, the FDO utilizes mechanisms, such as Equations 6 and 7 to adjust values that exceed the defined range, ensuring that all solutions remain within the limited search space. This approach enhances the algorithm's ability to converge toward optimal solutions effectively.

$$X_{i,t+1} = \text{upper bound} * \text{Levy} \quad (6)$$

$$X_{i,t+1} = \text{lower bound} * \text{Levy} \quad (7)$$

where $X_{i,t+1}$ represent the new bee value after the update in each iteration, the upper bound represents the upper boundary value of search space, the lower bound represent the lower boundary value

of search space, and Levy shows Levy flight to generate a random value between $(-1, 1)$.

4 Methodology

This section of the research outlines the study aimed at enhancing the FDO algorithm. It encompasses addressing exploitation limitations, improving FDO through adaptive bee update strategies, experimental setup, performance evaluation, and effectiveness assessment. The primary objective of this study is to improve the algorithm's exploitation capability, which has been identified as one of its main weaknesses, illustrated in Figure 2.

4.1 Addressing exploitation limitations

Optimization problems are prevalent across various specialized domains, where the primary objective is to identify the optimal solution among numerous possible alternatives. In the field of metaheuristic algorithms, enhancing both exploration and exploitation is essential for improving performance. Exploitation focuses on refining solutions within promising regions. The FDO has been observed to exhibit weaknesses in exploitation; theoretically, exploitation can be improved by concentrating the search around high-quality solutions. In the context of swarm-based algorithms, this involves reinforcing local search behavior while maintaining feasibility. The proposed modifications aim to increase solution density within promising regions, thereby enhancing the probability of convergence toward global optima. According to convergence theory in metaheuristics, increasing selective pressure around fitter solutions helps refine accuracy in late-stage optimization, limiting the algorithm's ability to converge efficiently toward optimal solutions. Therefore, this research aims to address and enhance the exploitation capability of the FDO algorithm to improve its overall effectiveness in solving complex optimization problems.

4.2 Improving FDO through adaptive bee update strategies

As discussed in the preceding sections, various methods and solutions have been employed to address this issue, contributing to a relative improvement in the exploitation capability of the FDO. This enhancement of the algorithm adds new progression to previous improvements by introducing some novel equations to update bees: that exceed the boundary of the search space. In this research, two new modifications are proposed for updating

TABLE 6 Results of comparing EESB-FDO with modifications Of FDO using classical benchmark functions.

TF	Metrics	EESB-FDO	M_IFDO	IFDO	SC-FDO	CFDO
F1	Mean	1.53E-22	3.44E-24	5.38E-24	0.00E+00	6.44E-51
	SD	7.14E-22	1.12E-23	2.74E-23	0.00E+00	3.46E-50
F2	Mean	5.95E-02	5.17E-01	7.34E-01	0.00E+00	4.02E-34
	SD	1.22E-01	2.71E-01	1.62E+00	0.00E+00	2.02E-33
F3	Mean	2.23E-08	1.06E-13	2.88E-07	0.00E+00	1.81E-06
	SD	4.20E-08	3.93E-13	6.90E-07	0.00E+00	5.60E-06
F4	Mean	6.43E-08	5.00E-05	3.60E-04	0.00E+00	7.95E-06
	SD	1.96E-07	4.27E-03	8.11E-04	0.00E+00	4.27E-05
F5	Mean	1.46E+01	3.10E+01	1.84E+02	2.69E-01	1.68E+01
	SD	2.96E+01	4.00E+01	3.45E+02	2.69E-01	3.14E+01
F6	Mean	9.49E-19	4.15E+06	4.22E+06	2.02E-02	2.01E-01
	SD	5.19E-18	1.10E+03	8.15E-09	5.53E+02	5.02E-01
F7	Mean	6.29E-01	7.20E-01	5.68E-01	4.49E-02	5.75E-01
	SD	3.12E-01	3.17E-01	3.14E-01	5.53E+02	3.54E-01
F8	Mean	−4.23E+03	−3.00E+06	−2.92E+06	−1.02E+04	−1.04E+04
	SD	3.96E+03	1.48E+05	2.34E+05	2.90E+04	1.29E+04
F9	Mean	7.34E+00	8.98E+00	1.35E+01	0.00E+00	5.64E-01
	SD	4.00E+00	9.85E+00	6.66E+00	0.00E+00	3.04E+00
F10	Mean	7.07E-15	3.89E-15	5.18E-15	3.26E-15	1.48E-15
	SD	1.21E-15	5.88E-16	1.67E-15	3.66E-15	1.32E-15
F11	Mean	1.31E-01	7.35E-02	5.26E-01	0.00E+00	9.26E-02
	SD	9.74E-02	3.91E-02	8.90E-02	0.00E+00	1.25E-01
F12	Mean	8.66E+00	1.75E+01	1.81E+01	4.54E-02	1.19E+00
	SD	4.29E+00	1.86E+01	2.57E+01	5.61E-02	7.97E-01
F13	Mean	3.37E+00	4.18E+09	4.10E+09	1.73E-01	6.79E-01
	SD	5.45E+00	3.13E+07	1.50E-05	1.90E-02	2.87E-01
F14	Mean	2.52E+00	8.50E-07	2.68E-07	6.36E+01	5.50E+00
	SD	1.32E+00	3.45E-05	4.68E-07	9.01E+01	3.92E+00
F15	Mean	8.24E-04	2.00E-19	4.03E-16	2.28E+02	3.10E-03
	SD	5.86E-04	6.73E-04	9.25E-16	2.37E+02	1.12E-02
F16	Mean	−1.03E+00	1.94E-16	9.14E-16	3.55E+02	−1.03E+00
	SD	6.66E-16	3.34E-02	3.61E-16	3.61E+02	2.80E-03
F17	Mean	3.98E-01	2.20E+01	2.38E+01	5.17E+02	−1.03E+00
	SD	1.67E-16	3.21E-01	1.24E-01	5.25E+02	3.01E+01
F18	Mean	3.00E+00	2.23E+ 02	2.24E+02	1.56E+02	1.93E+01
	SD	0.00E+00	1.34E-02	2.68E-05	1.68E+02	3.51E-03
F19	Mean	−3.86E+00	3.15E+01	3.15E+01	7.09E+02	−2.64E+00
	SD	3.11E-15	7.90E-02	1.32E-03	7.31E+02	5.17E-01

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

these exceeding bees, the first one, EESB-FDO, is designed to update bees, and the second, EEBC-FDO, also aims to update bees, both contributing to improving the exploitation capability of the FDO algorithm. From a theoretical standpoint, boundary handling mechanisms play a critical role in shaping the topology of the solution space. The proposed modifications replace rigid boundary

TABLE 7 Results of comparing EESB-FDO with FDO for (20 and 30) dimensions.

TF	Metrics	20-Dimensional		30-Dimensional	
		EESB-FDO	FDO	EESB-FDO	FDO
F1	Mean	5.51E-21	1.22E-19	7.13E-05	8.70E-05
	SD	1.5E-20	4.16E-19	1.08E-04	1.43E-04
F2	Mean	5.24E+00	5.71E+00	1.45E+01	1.65E+01
	SD	2.91E+00	3.00E+00	3.43E+00	3.49E+00
F3	Mean	7.40E-01	1.01E+00	4.40E+02	4.30E+02
	SD	4.75E-01	1.02E+00	2.93E+02	2.54E+02
F4	Mean	1.86E+00	1.45E+00	1.40E+01	1.47E+01
	SD	1.08E+00	7.90E-01	3.84E+00	3.08E+00
F5	Mean	3.43E+01	3.66E+01	7.33E+01	7.94E+01
	SD	2.89E+01	2.94E+01	4.66E+01	6.51E+01
F6	Mean	1.71E-20	2.01E-20	6.31E-04	7.95E-04
	SD	5.56E-20	8.10E-20	5.44E-04	6.60E-04
F7	Mean	6.07E-01	7.68E-01	8.97E-01	8.47E-01
	SD	2.92E-01	2.82E-01	3.32E-01	2.49E-01
F8	Mean	−5.19E+03	−8.87E+03	−8.20E+03	−9.50E+03
	SD	4.46E+03	2.66E+04	9.83E+03	1.11E+04
F9	Mean	2.15E+01	2.27E+01	3.45E+01	3.70E+01
	SD	1.14E+01	8.05E+00	1.16E+01	1.18E+01
F10	Mean	1.78E+00	1.84E+00	2.18E+00	1.98E+00
	SD	9.21E-01	1.21E+00	1.18E+00	9.69E-01
F11	Mean	2.20E-02	3.65E-02	1.76E-02	1.53E-02
	SD	2.02E-02	2.80E-02	1.74E-02	1.73E-02
F12	Mean	6.55E+00	5.12E+00	3.13E+00	4.79E+00
	SD	4.60E+00	3.87E+00	2.79E+00	2.81E+00
F13	Mean	1.89E+00	2.53E+00	1.61E+00	3.98E+00
	SD	4.61E+00	5.68E+00	4.94E+00	6.29E+00

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

clipping with stochastic repositioning or controlled correction strategies. These approaches improve the algorithm’s capacity to recover from constraint violations while maintaining stable and coherent search dynamics. Such strategies are consistent with the principles of constrained optimization and contribute to preserving the continuity and efficiency of the search trajectory.

4.3 Experimental setup and performance evaluation

The algorithm was implemented in the MATLAB 2022 environment on a laptop equipped with a 10th-generation Intel Core i5 processor and 8 GB of RAM. After incorporating modifications, the algorithm was executed more than 30 times, each for 500 iterations. The performance was evaluated using a

set of benchmark functions and statistical methods. The results demonstrate the improved efficiency of the algorithm with the proposed updates compared to its previous version.

4.4 Effectiveness assessment

The testing stage is crucial for evaluating the effectiveness and capability of the algorithm. This process consists of several key steps, including benchmark test functions, comparative analysis, statistical test methods, and real-world problem testing. Benchmark testing functions, such as classical, CEC2019, and CEC2022, are used to assess the performance and capability of the new improvements. Comparative analysis measures the enhanced algorithm’s results against the original version and other existing algorithms. Additionally, statistical methods, including the Wilcoxon rank-sum test, standard deviation, and mean,

TABLE 8 Results of comparing EEBC-FDO with FDO and other chosen algorithms Using classical benchmark functions.

TF	Metrics	EEBC-FDO	FDO	LPB	WOA	GSA	GWO
F1	Mean	2.78E-21	1.74E-18	6.27E-02	9.53E-76	1.83E-02	1.34E-27
	SD	1.50E-20	9.39E-18	5.06E-02	5.11E-75	9.86E-02	1.71E-27
F2	Mean	1.87E-03	2.34E+00	3.58E-02	1.02E-52	5.97E-02	9.16E-17
	SD	5.98E-03	6.92E-01	1.44E-02	5.15E-52	1.40E-01	6.35E-17
F3	Mean	5.67E-07	7.27E-08	2.06E+02	1.38E+02	9.95E+02	1.25E-05
	SD	2.83E-06	2.61E-07	1.43E+02	2.92E+02	4.74E+02	2.83E-05
F4	Mean	3.12E-08	5.09E-08	7.17E-01	4.05E+00	6.96E+00	1.01E-06
	SD	4.04E-08	5.88E-08	2.37E-01	6.31E+00	2.12E+00	1.09E-06
F5	Mean	1.96E+01	2.62E+01	5.43E+01	6.95E+01	9.31E+01	2.70E+01
	SD	3.15E+01	3.88E+01	3.78E+01	6.51E+ 01	1.22E+02	8.67E-01
F6	Mean	2.61E-20	1.30E-17	4.83E-02	9.19E-03	8.10E+00	7.64E-01
	SD	1.43E-19	7.13E-17	4.44E-02	4.13E-02	1.17E+01	3.50E-01
F7	Mean	5.53E-01	6.33E-01	1.57E-02	2.69E-03	8.01E-02	1.86E-03
	SD	3.35E-01	3.36E-01	9.18E-03	3.24E-03	4.31E-02	9.14E-04
F8	Mean	−4.20E+03	−8.30E+03	−3.71E+03	−3.17E+03	−2.55E+03	−5.80E+03
	SD	3.86E+03	1.14E+04	2.43E+02	1.28E+03	3.79E+02	9.72E+02
F9	Mean	7.50E+00	7.05E+00	3.51E-02	2.69E+00	2.80E+01	2.32E+00
	SD	3.77E+00	3.11E+00	2.73E-02	8.84E+00	8.81E+00	3.70E+00
F10	Mean	6.73E-15	6.95E-15	1.19E-01	9.35E-15	1.29E-08	1.05E-13
	SD	3.03E-15	2.61E-15	6.49E-02	1.92E-14	3.83E-09	1.82E-14
F11	Mean	1.92E-01	1.56E-01	1.96E-01	6.21E-02	2.80E+01	2.84E- 03
	SD	1.42E-01	9.93E-02	1.75E-01	1.05E-01	6.13E+00	7.18E- 03
F12	Mean	7.72E+00	8.02E+00	3.01E-03	4.64E-03	1.62E+00	4.44E-02
	SD	5.53E+00	5.60E+00	6.13E-03	7.75E-03	1.01E+00	2.29E-02
F13	Mean	5.14E+00	3.70E+00	7.09E-03	1.27E-01	8.03E+00	6.46E-01
	SD	4.95E+00	4.10E+00	6.12E-03	1.82E-01	4.63E+00	2.01E-01
F14	Mean	2.47E+00	2.72E+00	9.98E-01	4.75E+00	5.09E+00	4.65E+00
	SD	2.90E+00	1.46E+00	4.44E-16	4.07E+00	4.24E+00	4.37E+00
F15	Mean	1.72E-03	1.35E-03	8.63E-03	1.04E-03	4.49E-03	3.80E-03
	SD	3.91E-03	2.06E-03	9.32E-03	1.63E-03	2.24E-03	7.41E-03
F16	Mean	−1.03E+00	−1.03E+00	−1.03E+00	−9.97E-01	−1.03E+00	−1.03E+00
	SD	6.78E-16	6.66E-16	6.66E-16	1.85E-01	6.66E-16	6.66E-16
F17	Mean	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	SD	1.69E-16	1.67E-16	3.16E-06	1.91E-05	1.67E-16	2.49E-06
F18	Mean	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	SD	0.00E+00	0.00E+00	1.23E-04	5.12E-05	0.00E+00	4.00E-05
F19	Mean	−3.86E+00	−3.86E+00	−3.20E+00	−3.85E+00	−3.86E+00	−3.86E+00
	SD	3.16E-15	3.11E-15	1.33E-15	1.89E-02	3.11E-15	2.32E-03
F20	Mean	−2.89E+00	−2.86E+00	−3.25E+00	−3.22E+00	−3.32E+00	−3.27E+00
	SD	1.16E+00	1.14E+00	5.89E-02	8.60E-02	1.78E-15	6.69E-02
F21	Mean	−5.92E+00	−5.05E+00	−5.39E+00	−8.94E+00	−5.94E+00	−8.55E+00
	SD	2.50E+00	3.61E-04	3.25E+00	2.23E+00	3.69E+00	2.48E+00
F22	Mean	−5.39E+00	−4.70E+00	−5.56E+00	−8.86E+00	−1.04E+01	−1.02E+01
	SD	2.73E+00	3.38E+00	3.23E+00	2.78E+00	0.00E+00	9.54E-01

(Continued)

TABLE 8 (Continued)

TF	Metrics	EEBC-FDO	FDO	LPB	WOA	GSA	GWO
F23	Mean	−5.69E+00	−4.32E+00	−6.56E+00	−6.75E+00	−9.93E+00	−1.02E+01
	SD	3.12E+00	2.91E+00	3.76E+00	4.60E+00	2.49E+00	1.35E+00

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

are used to verify the significance, performance, consistency, and reliability of the algorithm. Real-world testing is conducted by applying the algorithm to practical problems. The testing phase ultimately confirms that the modifications have led to meaningful enhancements.

5 FDO modifications

In this section, this study focuses on three modifications: EESB-FDO, EEBC-FDO, and ELFS, which are proposed as extensions of the FDO. These modifications and FDO are substantive to parameter setting and change results by changing one of them [19]. The fundamental concept of FDO is inspired by the reproductive process and collective decision-making behavior exhibited by bees. Specifically, it mimics the way bees search for multiple available hives and collaboratively select the most optimal ones.

5.1 Enhancing FDO performance through scout bee boundary repositioning

The artificial scout bee operates on the same principle, searching for the optimal solution among multiple available options within the defined boundary search space. Maintaining the search within a restricted area is a crucial factor influencing the algorithm's performance and effectiveness. To enforce this condition, the FDO employs Equations 6 and 7. When the position of an artificial scout bee exceeds the predefined boundaries, the FDO updates its value according to these equations, ensuring that the search remains within the permitted limits. In this enhancement, two new modifications are proposed to update scout bees that exceed the upper or lower boundary, ensuring they remain within the predefined search space as described below.

5.1.1 Enhancing exploitation through stochastic boundary for FDO

The first modification employs Equation 8 to regulate values that surpass either the upper or lower boundary, meaning using just one Equation 8 instead of Equations 6, 7. By using this equation, EESB-FDO updates the bee value that exceeds the boundaries (upper and lower); the proposed approach utilizes a randomized value within a predefined fixed range. This involves generating random values within the defined search space limits, thereby introducing greater diversity and adaptability into the search process, as shown in Figure 3A. This approach leverages uniform random sampling within the valid bounds, which ensures all feasible regions are equally likely to be explored

upon violation. This strategy preserves population diversity and prevents premature convergence caused by agents clustering near hard boundary limits. It also avoids stagnation by introducing controlled randomness, a key requirement for ergodicity in metaheuristic convergence.

$$\text{update } X_{i,t+1} = \text{Random}(\text{boundary}) * \text{Levy} (1) \quad (8)$$

Here, $\text{update } X_{i,t+1}$ represents the new bee value after update in each iteration, random (boundary) mean generate any random value in the predefined search space between the upper boundary and lower boundary, Levy flight to generate a random value between $(-1,1)$.

5.1.2 Enhancing exploitation through boundary carving for FDO

The second modification utilizes Equation 9 to adjust values that exceed the upper boundary and Equation 10 to regulate values that fall below the lower boundary. By formulating two equations for updating scout bees, notable progress has been achieved in enhancing the algorithm's performance. These equations contribute to more effective handling of boundary violations and improve the overall exploitation capability of the algorithm, as explained in Figure 3B.

$$\text{update } X_{i,t+1} = (X_{i,t+1} - \text{upper}_{\text{bound}}) * \text{Levy} (1) \quad (9)$$

$$\text{update } X_{i,t+1} = (X_{i,t+1} - \text{lower}_{\text{bound}}) * \text{Levy} (1) \quad (10)$$

where $\text{update } X_{i,t+1}$ represents the new bee value after the update in each iteration, $X_{i,t+1}$ represents the old bee value that exceed boundaries, the upper bound represents the upper boundary value of the search space, the lower bound represents the lower boundary value of the search space, and Levy shows Levy flight to generate random values between $(-1,1)$.

5.2 Eliminating levy flight shortcomings

Levy flight is a type of random walk where step lengths follow a Levy distribution, known for its heavy-tailed probability distribution, as shown in the pseudo code in Figure 4. This property makes Levy flight highly effective for exploring large and unknown search spaces [2].

Levy flight represents a type of random walk and plays a crucial role in metaheuristic algorithms by generating random values within the range of $(-1,1)$. However, these values

TABLE 9 Comparing EEBC-FDO with modifications Of FDO using classical benchmark functions.

TF	Metrics	EEBC-FDO	M_IFDO	IFDO	SC-FDO	CFDO
F1	Mean	2.78E-21	3.44E-24	5.38E-24	0.00E+00	6.44E-51
	SD	1.50E-20	1.12E-23	2.74E-23	0.00E+00	3.46E-50
F2	Mean	1.87E-03	5.17E-01	7.34E-01	0.00E+00	4.02E+34
	SD	5.98E-03	2.71E-01	1.62E+00	0.00E+00	2.02E-33
F3	Mean	5.67E-07	1.06E-13	2.88E-07	0.00E+00	1.81E-06
	SD	2.83E-06	3.93E-13	6.90E-07	0.00E+00	5.60E-06
F4	Mean	3.12E-08	5.00E-05	3.60E-04	0.00E+00	7.95E-06
	SD	4.04E-08	4.27E-03	8.11E-04	0.00E+00	4.27E-05
F5	Mean	1.96E+01	3.10E+01	1.84E+02	2.69E-01	1.68E+01
	SD	3.15E+01	4.00E+01	3.45E+02	2.69E-01	3.14E+01
F6	Mean	2.61E-20	4.15E+06	4.22E+06	2.02E-02	2.01E-01
	SD	1.43E-19	1.10E+03	8.15E-09	5.53E+02	5.02E-01
F7	Mean	5.53E-01	7.20E-01	5.68E-01	4.49E-02	5.75E-01
	SD	3.35E-01	3.17E-01	3.14E-01	5.53E+02	3.54E-01
F8	Mean	-4.20E+03	-3.00E+06	-2.92E+06	-1.02E+04	-1.04E+04
	SD	3.86E+03	1.48E+05	2.34E+05	2.90E+04	1.29E+04
F9	Mean	7.50E+00	8.98E+00	1.35E+01	0.00E+00	5.64E-01
	SD	3.77E+00	9.85E+00	6.66E+00	0.00E+00	3.04E+00
F10	Mean	6.73E-15	3.89E-15	5.18E-15	3.26E-15	1.48E-15
	SD	3.03E-15	5.88E-16	1.67E-15	3.66E-15	1.32E-15
F11	Mean	1.92E-01	7.35E-02	5.26E-01	0.00E+00	9.26E-02
	SD	1.42E-01	3.91E-02	8.90E-02	0.00E+00	1.25E-01
F12	Mean	7.72E+00	1.75E+01	1.81E+01	4.54E-02	1.19E+00
	SD	5.53E+00	1.86E+01	2.57E+01	5.61E-02	7.97E-01
F13	Mean	5.14E+00	4.18E+09	4.10E+09	1.73E-01	6.79E-01
	SD	4.95E+00	3.13E+07	1.50E-05	1.90E-02	2.87E-01
F14	Mean	2.47E+00	8.50E-07	2.68E-07	6.36E+01	5.50E+00
	SD	2.90E+00	3.45E-05	4.68E-07	9.01E+01	3.92E+00
F15	Mean	1.72E-03	2.00E-19	4.03E-16	2.28E+02	3.10E-03
	SD	3.91E-03	6.73E-04	9.25E-16	2.37E+02	1.12E-02
F16	Mean	-1.03E+00	1.94E-16	9.14E-16	3.55E+02	-1.03E+00
	SD	6.78E-16	3.34E-02	3.61E-16	3.61E+02	2.80E-03
F17	Mean	3.98E-01	2.20E+01	2.38E+01	5.17E+02	-1.03E+00
	SD	1.69E-16	3.21E-01	1.24E-01	5.25E+02	3.01E+01
F18	Mean	3.00E+00	2.23E+ 02	2.24E+02	1.56E+02	1.93E+01
	SD	0.00E+00	1.34E-02	2.68E-05	1.68E+02	3.51E-03
F19	Mean	-3.86E+00	3.15E+01	3.15E+01	7.09E+02	-2.64E+00
	SD	3.16E-15	7.90E-02	1.32E-03	7.31E+02	5.17E-01

should not exceed the defined limits. To assess the behavior of Levy flight, it was executed 30 times, each with 10,000 iterations. The results, as presented in [Figure 5](#), indicate that in each execution, some randomly generated values exceeded the specified range of $(-1,1)$. To address this issue, [Equation 11](#)

was incorporated into the Levy flight mechanism. This equation ensures that any generated value exceeding the upper limit of 1 is adjusted to 1, while any value falling below the lower limit of -1 is set to -1 . In [Figure 4](#), explaining Levy flight before enhancement, it was shown that O was represented

TABLE 10 Results of comparing EEBC-FDO with FDO for 20- and 30-dimensional algorithms using classical benchmark functions.

TF	Metrics	20-Dimensional		30-Dimensional	
		EEBC-FDO	FDO	EEBC-FDO	FDO
F1	Mean	3.33E-21	1.22E-19	8.71E-05	8.70E-05
	SD	1.15E-20	4.16E-19	1.05E-04	1.43E-04
F2	Mean	5.08E+00	5.71E+00	1.41E+01	1.65E+01
	SD	3.25E+00	3.00E+00	5.06E+00	3.49E+00
F3	Mean	9.65E-01	1.01E+00	4.94E+02	4.30E+02
	SD	1.15E+00	1.02E+00	3.37E+02	2.54E+02
F4	Mean	9.92E-01	1.45E+00	1.28E+01	1.47E+01
	SD	5.19E-01	7.90E-01	4.04E+00	3.08E+00
F5	Mean	3.32E+01	3.66E+01	7.29E+01	7.94E+01
	SD	3.83E+01	2.94E+01	6.16E+01	6.51E+01
F6	Mean	1.60E-20	2.01E-20	6.30E-04	7.95E-04
	SD	2.55E-20	8.10E-20	5.81E-04	6.60E-04
F7	Mean	7.39E-01	7.68E-01	7.75E-01	8.47E-01
	SD	2.87E-01	2.82E-01	3.12E-01	2.49E-01
F8	Mean	−1.37E+07	−8.87E+03	−1.98E+09	−9.50E+03
	SD	6.19E+07	2.66E+04	5.17E+09	1.11E+04
F9	Mean	2.25E+01	2.27E+01	3.56E+01	3.70E+01
	SD	9.44E+00	8.05E+00	1.42E+01	1.18E+01
F10	Mean	1.55E+00	1.84E+00	1.92E+00	1.98E+00
	SD	1.26E+00	1.21E+00	1.21E+00	9.69E-01
F11	Mean	2.25E-02	3.65E-02	2.09E-02	1.53E-02
	SD	2.30E-02	2.80E-02	2.05E-02	1.73E-02
F12	Mean	4.13E+00	5.12E+00	4.88E+00	4.79E+00
	SD	5.09E+00	3.87E+00	3.23E+00	2.81E+00
F13	Mean	2.04E+00	2.53E+00	1.53E+00	3.98E+00
	SD	6.86E+00	5.68E+00	5.14E+00	6.29E+00

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

TABLE 11 CEC-C06 2019 benchmark functions [80].

No.	Functions	Dim	Range	fmin
CEC01	Storn's Chebyshev polynomial fitting problem	9	[−8,192, 8,192]	1
CEC02	Inverse Hilbert matrix problem	16	[−16,384, 16,384]	1
CEC03	Lennard–Jones minimum energy cluster	18	[−4, 4]	1
CEC04	Rastrigin's function	10	[−100, 100]	1
CEC05	Griewangk's function	10	[−100, 100]	1
CEC06	Weierstrass function	10	[−100, 100]	1
CEC07	Modified Schwefel's function	10	[−100,100]	1
CEC08	expanded Schaffer's F6 function	10	[−100,100]	1
CEC09	Happy cat function	10	[−100, 100]	1
CEC10	Ackley function	10	[−100, 100]	1

TABLE 12 Results of comparing EESB-FDO with FDO and other chosen algorithms Using CEC 2019 benchmark functions.

CEC	Metrics	EESB-FDO	FDO	AOA	LPB	WOA	FOX
CEC01	Mean	1.91E+09	1.74E+09	7.67E+09	7.03E+10	3.81E+10	1.83E+04
	SD	2.49E+09	1.77E+09	2.68E+10	7.40E+10	5.07E+10	2.25E+04
CEC02	Mean	1.73E+01	1.73E+01	1.93E+01	3.76E+01	1.74E+01	1.83E+01
	SD	1.08E-14	1.08E-14	3.76E-01	2.00E+01	9.95E-03	3.95E-04
CEC03	Mean	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.37E+01
	SD	5.42E-15	5.42E-15	9.08E-04	5.33E-15	5.33E-15	7.11E-15
CEC04	Mean	3.73E+01	4.21E+01	1.09E+04	7.14E+01	3.62E+02	1.70E+03
	SD	1.26E+01	2.15E+01	3.35E+03	3.00E+01	1.28E+02	9.84E+02
CEC05	Mean	1.17E+00	1.20E+00	3.84E+00	1.27E+00	1.78E+00	6.48E+00
	SD	1.07E-01	1.35E-01	9.36E-01	1.29E-01	3.64E-01	1.32E+00
CEC06	Mean	1.08E+00	1.09E+01	9.02E+00	6.34E+00	9.43E+00	4.45E+00
	SD	8.50E-01	9.21E-01	1.22E+00	7.50E-01	1.20E+00	1.04E+00
CEC07	Mean	3.30E+02	3.47E+02	2.52E+02	2.68E+02	6.15E+02	3.86E+02
	SD	1.93E+02	2.12E+02	1.63E+02	1.66E+02	2.62E+02	2.48E+02
CEC08	Mean	5.40E+00	5.23E+00	5.57E+00	5.57E+00	5.91E+00	5.63E+00
	SD	6.67E-01	6.41E-01	5.31E-01	5.60E-01	6.13E-01	5.88E-01
CEC09	Mean	2.58E+00	2.62E+00	9.36E+02	3.09E+00	4.82E+00	3.77E+00
	SD	1.65E-01	2.98E-01	4.66E+02	2.97E-01	9.07E-01	3.55E-01
CEC10	Mean	1.93E+01	2.00E+01	2.01E+01	2.01E+01	2.01E+01	2.10E+01
	SD	3.59E+00	1.21E-02	6.09E-02	2.79E-02	1.10E+00	5.06E-03

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

by $(0.01 \times \text{step})$ and then changed to Equation 11 after the update.

$$O = \max(\min(0.01 * \text{step}, 1), -1) \quad (11)$$

where O represents the output of the Levy flight function, Equation 11 ensures that the generated values remain within the defined range of $(-1, 1)$ without exceeding these limits.

6 Result and discussion

The key point of algorithm evaluation is the comparison with other established optimization methods. To ensure a comprehensive assessment, various evaluation metrics are employed, including benchmark functions, statistical methods, and real-world problem scenarios. In this study, the proposed EESB-FDO and EEBC-FDO were compared with seven metaheuristic algorithms: FDO [31], arithmetic optimization algorithm (AOA) [42], gravitational search algorithm (GSA) [66], GWO [65], learner performance-based behavior algorithm (LPB) [67], whale optimization algorithm (WOA) [68], and fox-inspired optimization algorithm (FOX) [69] using classical benchmark functions, CEC-2019, and CEC-2022. The performance evaluation is conducted based on statistical measures such as mean, standard deviation, and the Wilcoxon rank-sum test to

ensure the performance and capability. To ensure fair and unbiased comparisons, all algorithms were evaluated under identical experimental conditions. Each algorithm was executed independently for 30 runs to account for the stochastic nature of metaheuristic methods. In every run, a population of 30 search agents was employed, and the optimization process was conducted for a maximum of 500 iterations. This consistent configuration across all algorithms ensures that observed performance differences are attributable to algorithmic effectiveness rather than variations in experimental settings. Following the execution, the average and standard deviation were calculated, and the results are presented in this section.

6.1 Classical benchmark test functions

The performance of EESB-FDO and EEBC-FDO was evaluated using three categories of test function groups, as outlined in Zervoudakis and Tsafarakis [12]. These test function groups possess distinct characteristics that assess different aspects of the algorithms' performance. Unimodal test functions (f1–f7) are employed to evaluate the algorithms' exploitation capability and convergence efficiency, as they contain only a single global minimum (or maximum) without any local optima, as shown in Table 2. In contrast, multimodal test functions (f8–f13) are utilized to assess the algorithms' exploration ability, ensuring they can escape local optima and effectively search for the global solution

TABLE 13 Results of comparing EEBC-FDO with FDO and other chosen algorithms using CEC 2019 benchmark functions.

CEC	Metrics	EEBC-FDO	FDO	AOA	LPB	WOA	FOX
CEC01	Mean	1.80E+09	1.74E+09	7.67E+09	7.03E+10	3.81E+10	1.83E+04
	SD	2.21E+09	1.77E+09	2.68E+10	7.40E+10	5.07E+10	2.25E+04
CEC02	Mean	1.73E+01	1.73E+01	1.93E+01	3.76E+01	1.74E+01	1.83E+01
	SD	1.08E-14	1.08E-14	3.76E-01	2.00E+01	9.95E-03	3.95E-04
CEC03	Mean	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.27E+01	1.37E+01
	SD	5.42E-15	5.42E-15	9.08E-04	5.33E-15	5.33E-15	7.11E-15
CEC04	Mean	4.51E+01	4.21E+01	1.09E+04	7.14E+01	3.62E+02	1.70E+03
	SD	2.76E+01	2.15E+01	3.35E+03	3.00E+01	1.28E+02	9.84E+02
CEC05	Mean	1.21E+00	1.20E+00	3.84E+00	1.27E+00	1.78E+00	6.48E+00
	SD	1.26E-01	1.35E-01	9.36E-01	1.29E-01	3.64E-01	1.32E+00
CEC06	Mean	1.10E+00	1.09E+01	9.02E+00	6.34E+00	9.43E+00	4.45E+00
	SD	1.10E+00	9.21E-01	1.22E+00	7.50E-01	1.20E+00	1.04E+00
CEC07	Mean	3.84E+02	3.47E+02	2.52E+02	2.68E+02	6.15E+02	3.86E+02
	SD	1.80E+02	2.12E+02	1.63E+02	1.66E+02	2.62E+02	2.48E+02
CEC08	Mean	5.14E+00	5.23E+00	5.57E+00	5.57E+00	5.91E+00	5.63E+00
	SD	6.20E-01	6.41E-01	5.31E-01	5.60E-01	6.13E-01	5.88E-01
CEC09	Mean	2.58E+00	2.62E+00	9.36E+02	3.09E+00	4.82E+00	3.77E+00
	SD	1.30E-01	2.98E-01	4.66E+02	2.97E-01	9.07E-01	3.55E-01
CEC10	Mean	2.00E+01	2.00E+01	2.01E+01	2.01E+01	2.01E+01	2.10E+01
	SD	3.15E-02	1.21E-02	6.09E-02	2.79E-02	1.10E+00	5.06E-03

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

TABLE 14 CEC-C06 2022 benchmark functions [87].

Type	No.	Functions	Dim	Range	<i>fmin</i>
Unimodal function	1	Shifted and full rotated Zakharov function	10	[−100,100]	300
Basic functions	2	Shifted and full rotated Rosenbrock's function	10	[−100,100]	400
	3	Shifted and full rotated expanded Schaffer's F6 function	10	[−100,100]	600
	4	Shifted and full rotated non-continuous Rastrigin's function	10	[−100,100]	800
	5	Shifted and full rotated Levy function	10	[−100,100]	900
Hybrid functions	6	Hybrid function 1 (<i>N</i> = 3)	10	[−100,100]	1,800
	7	Hybrid function 2 (<i>N</i> = 6)	10	[−100,100]	2,000
	8	Hybrid function 3 (<i>N</i> = 5)	10	[−100,100]	2,200
Composition functions	9	Composition function 1 (<i>N</i> = 5)	10	[−100,100]	2,300
	10	Composition function 2 (<i>N</i> = 4)	10	[−100,100]	2,400
	11	Composition function 3 (<i>N</i> = 5)	10	[−100,100]	2,600
	12	Composition function 4 (<i>N</i> = 6)	10	[−100,100]	2,700

in complex landscapes with multiple optimal points, as presented in Table 3. Additionally, composite benchmark functions (f14–f23) are represent advanced test functions designed to simulate real-world optimization challenges, providing a more rigorous evaluation of the algorithms' robustness and adaptability in highly complex and irregular search spaces, as presented in Table 4.

All algorithms presented in Tables 5–10 were evaluated over 30 independent runs, each consisting of 500 iterations. The optimization process was initiated with 30 randomly generated search agents in 10, 20, and 30-dimensional search spaces for the first two test function groups (unimodal and multimodal functions). For the composite test function group, the

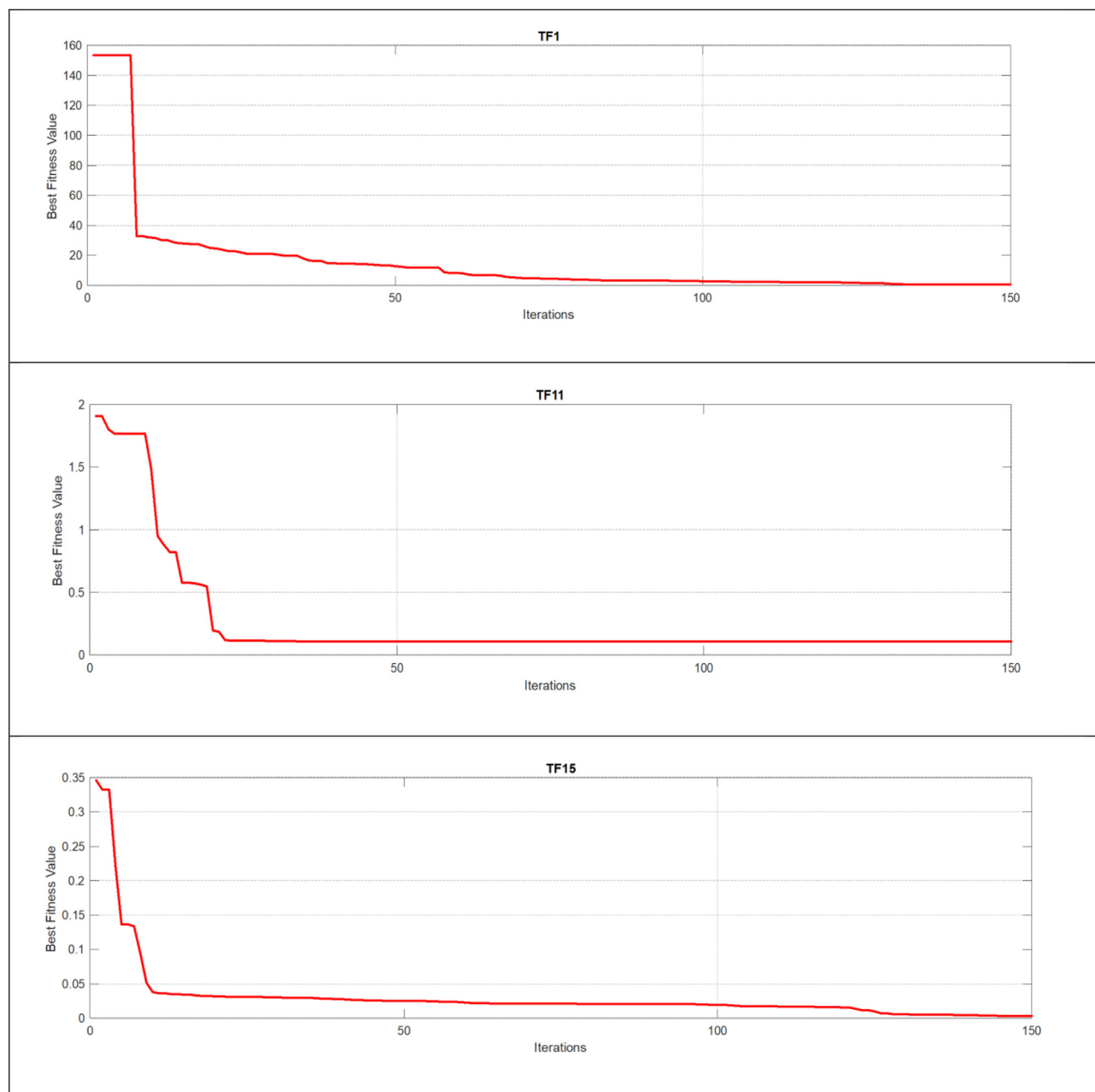


FIGURE 6
Using unimodal, multimodal, and composite test functions for the FDO algorithm convergence curve.

dimensionality was set according to the structural requirements of each function. The performance of the algorithms was assessed by computing statistical measures, including the mean and standard deviation, to ensure a comprehensive evaluation of their optimization capabilities.

6.1.1 Classical benchmark test for enhancing exploitation through stochastic boundary for FDO

The results of EESB-FDO, FDO, LPB, WOA, GSA, and GWO, as shown in Table 5 for test functions TF1 to TF7, indicate that EESB-FDO generally outperforms FDO, with

the exception of TF2. Specifically, EESB-FDO demonstrates superior performance over FDO, LPB, and GSA for TF1. Additionally, for TF3, TF4, and TF6, EESB-FDO outperforms all other algorithms. However, in TF2 and TF5, EESB-FDO shows comparatively poor performance, with other algorithms outperforming it. These results, by enhancing F1 to F7 static values, mean minimizing values and prove that the new modification in EESB-FDO significantly improves the algorithm's exploitation capability.

On the other hand, in TF8, TF11, TF14, and TF15, the results indicate that EESB-FDO outperformed FDO. However, when compared to other algorithms, the performance of these

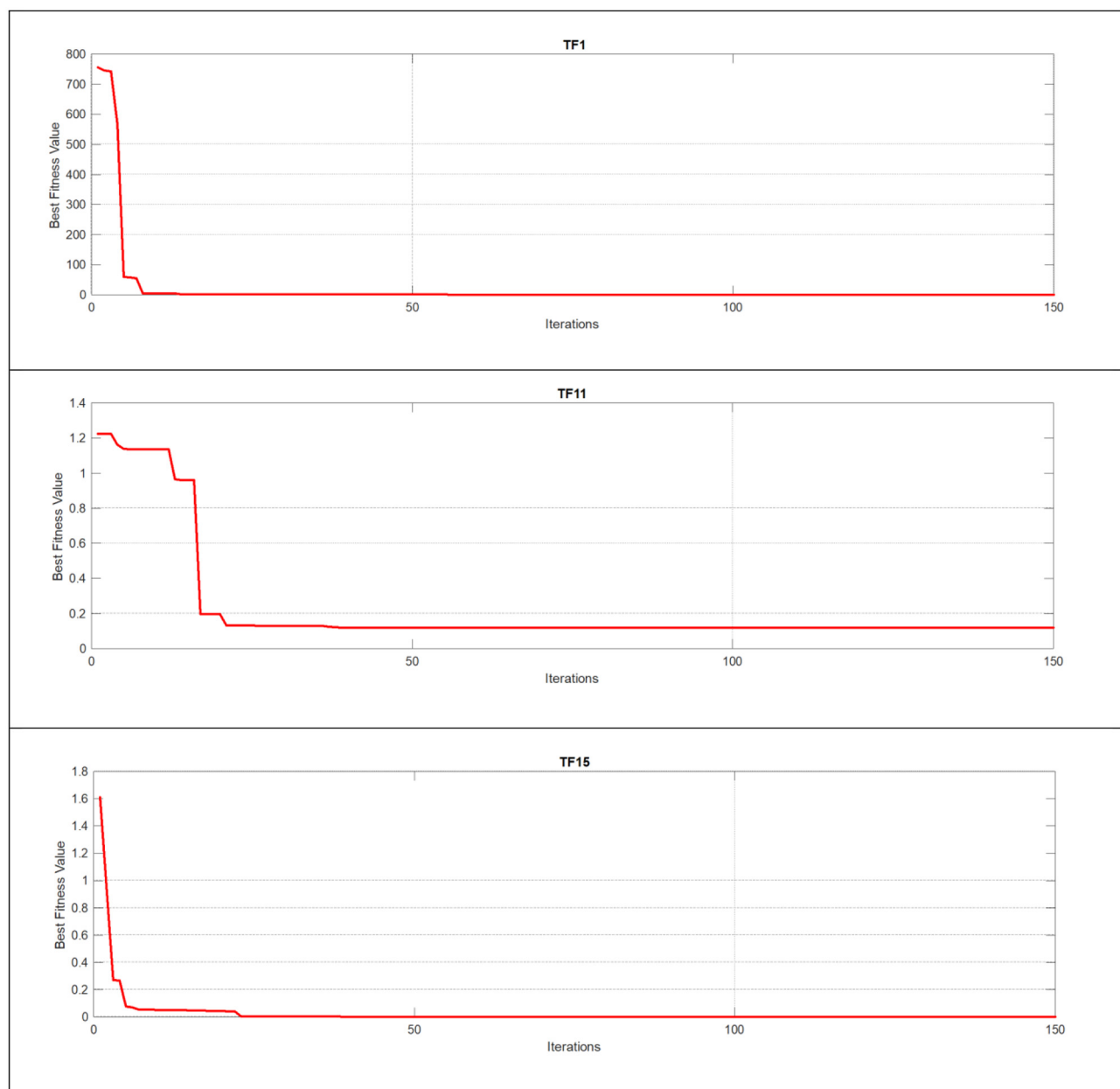


FIGURE 7
Using unimodal, multimodal, and composite test functions for the EESB-FDO algorithm convergence curve.

alternatives was slightly superior. Furthermore, in TF16, TF17, TF18, and TF19, both EESB-FDO and FDO achieved the optimal value and demonstrated comparable performance. In general, their results were more favorable than those of other algorithms.

In addition to the previous comparisons, Table 6 presents the results of EESB-FDO in comparison with M-IFDO, IFDO, SC-FDO, and CFDO. The findings indicate that EESB-FDO performed better than M-IFDO and IFDO in TF2 and TF4. In TF5, it outperformed all algorithms except SC-FDO, while in TF6, it demonstrated superior performance compared to the others. However, for TF1, TF3, and TF7, its results were

the weakest among the algorithms. Furthermore, regarding multimodal functions, EESB-FDO achieved better performance than M-IFDO and IFDO in TF9, TF12, and TF13. However, in TF8, TF11, TF14, and TF15, its results fluctuated, showing slight weaknesses. Additionally, from TF16 to TF19, EESB-FDO successfully attained the optimal values, demonstrating its effectiveness in these test cases.

The final evaluation of EESB-FDO, in comparison with FDO, was conducted using both 20- and 30-dimensional settings across unimodal and multimodal benchmark test functions, as presented in Table 7. In the 20-dimensional case, EESB-FDO outperformed FDO on several functions, specifically F1, F2, F3, F5, F6, F10, and

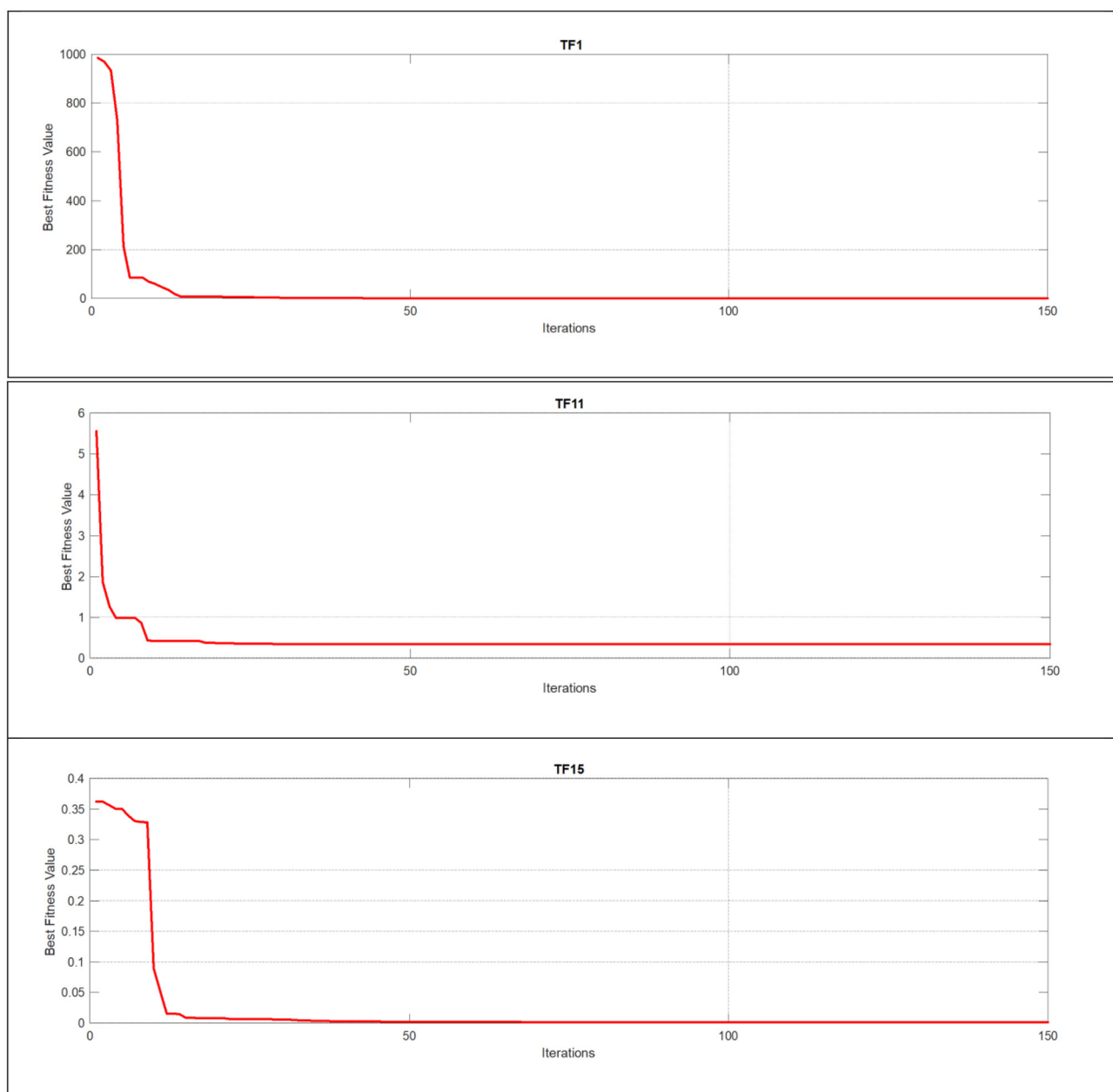


FIGURE 8
Using unimodal, multimodal, and composite test functions for the EEBC-FDO algorithm convergence curve.

F11, indicating a clear advantage in terms of both convergence speed and exploitation capability within this dimensional context. In the 30-dimensional case, EESB-FDO demonstrated superior performance compared to FDO on several benchmark functions, namely F1, F2, F5, F6, F9, F12, and F13. These results highlight the algorithm's enhanced convergence speed and exploitation capability, underscoring its effectiveness in handling complex optimization tasks within this dimensional setting. The results clearly indicate that several benchmark functions showed significant improvement, approaching near-optimal solutions, following dimension modification and the discovery of new optimal points.

6.1.2 Classical benchmark test for enhancing exploitation through boundary carving for FDO

For the second modification, EEBC-FDO, the results presented in Table 8 clearly demonstrate its superiority over the FDO in TF1 to TF7, with the exception of TF3, where it outperformed other algorithms but performed worse than FDO. These outcomes indicate an improvement in the algorithm's exploration capability. Additionally, in TF1 and TF2, EEBC-FDO performed better than LPB and GSA but was outperformed by WOA and GWO. In TF4, TF5, and TF6, EEBC-FDO outperformed all other algorithms, demonstrating its competitive advantage. However, from TF9 to TF15, its performance showed slight weaknesses, except for

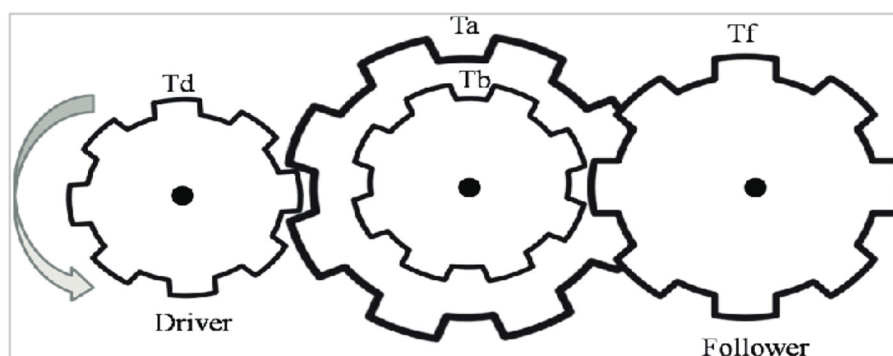


FIGURE 9
The gear train design problem.

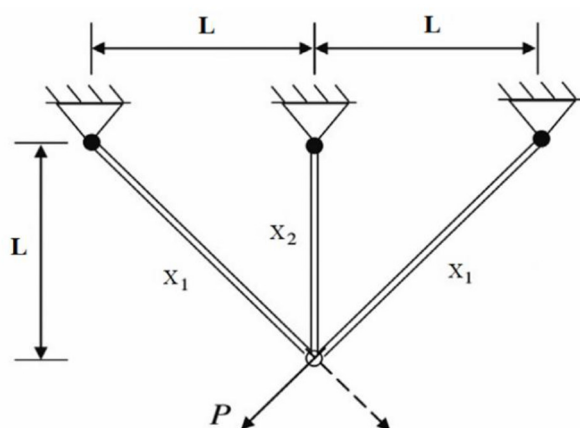


FIGURE 10
The three-bar truss problem.

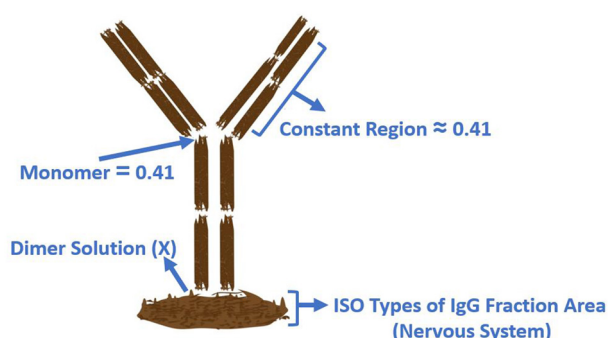


FIGURE 11
Pathological IgG fraction design problem.

TF10, where it performed favorably compared to other algorithms. Nonetheless, in TF16 to TF19, both FDO and EEBC-FDO achieved equal performance, obtaining the optimal value. On the other hand, Table 9 presents a comparison of EEBC-FDO with previous

modifications of FDO, including M-IFDO, IFDO, SC-FDO, and CFDO. The results indicate that the performance of EEBC-FDO is closely aligned with that of EESB-FDO when compared to these modifications.

The use of higher-dimensional evaluations is a critical aspect of validating the effectiveness of EEBC-FDO. To this end, both 20- and 30-dimensional benchmark functions were employed to compare the performance of EEBC-FDO against FDO. In the 20-dimensional case, as shown in Table 10, EEBC-FDO outperformed FDO on functions F1, F4, F6, and F11, while also demonstrating superior average performance on functions F2, F3, F5, F7, F9, F10, and F13. These results confirm the robustness and improved optimization capabilities of EEBC-FDO in more complex, high-dimensional problem spaces. In the second case, using the 30-dimensional benchmark functions, EEBC-FDO exhibited some limitations. While it outperformed FDO on only three functions—F5, F6, and F13—it still achieved better average performance on functions F4, F7, F9, and F10. These results suggest that although EEBC-FDO maintains competitive performance in certain high-dimensional scenarios, its effectiveness may diminish as problem dimensionality increases further.

6.2 CEC2019 benchmark test functions

Testing EESB-FDO and EEBC-FDO on a single benchmark test function is not a sufficient measure to ensure their effectiveness. Therefore, to thoroughly evaluate their performance and capability, the algorithm was tested on the CEC2019 benchmark functions presented in Table 11, which are commonly used for assessing optimization algorithms. This benchmark set includes ten multimodal test functions, which serve as enhanced evaluation functions for optimization purposes [70].

The results of the proposed EESB-FDO, presented in Table 12, demonstrate its success in outperforming other algorithms, achieving the highest ranking in CEC02, CEC04, CEC05, CEC06, and CEC09. Additionally, in CEC03, CEC08, and CEC10, EESB-FDO obtained the best performance based on the average calculation. Furthermore, the LPB algorithm secured second place and showed competitive performance, particularly in CEC03 and

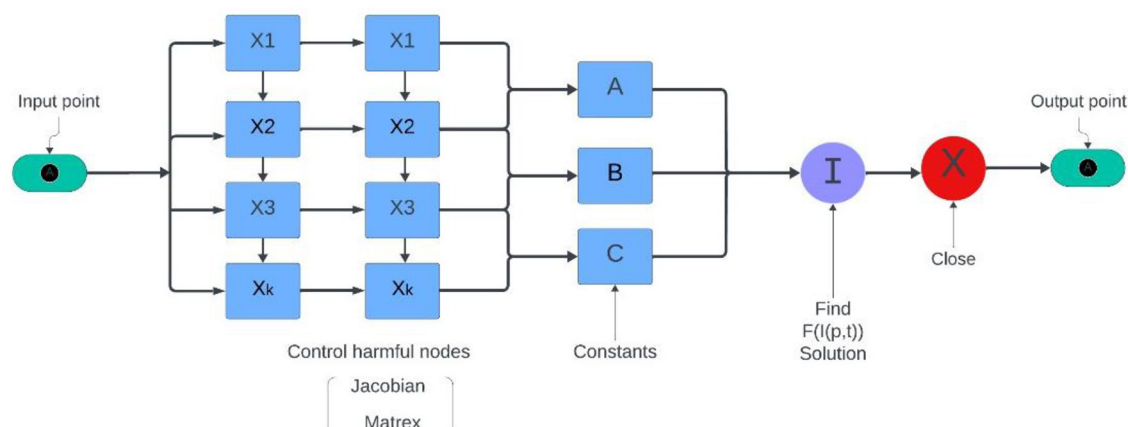


FIGURE 12
Design problem for a Cyber-physical attack mitigation system.

CEC07. Meanwhile, WOA and FOX exhibited similar performance, ranking third overall. Finally, AOA demonstrated the weakest performance among the compared algorithms. On the other hand, Table 13 presents the results of EEBC-FDO, clearly indicating that FDO and EEBC-FDO did not achieve superior performance in the competition, as their results were nearly identical. However, overall, the outcomes demonstrated better performance compared to the other algorithms discussed in this section.

6.3 CEC2022 benchmark test functions

To further evaluate the feasibility of the proposed algorithm, the CEC2022 benchmark test functions were utilized, as presented in Table 14. This benchmark set comprises 12 functions, labeled CEC01 to CEC12, which are organized into four groups: unimodal, multimodal, hybrid, and composite functions. The search space for these benchmark problems is defined within specific ranges of decision variables, constrained within the interval $[-100, 100]$ for all test functions. Therefore, $d = 10$ and 20 dimensions were used to assess the proposed modification [71].

The results of EESB-FDO, FDO, AOA, LPB, WOA, and FOX were evaluated by $d = 10$ dimension and are presented in Table 15. The outcomes of the proposed EESB-FDO modification indicate that it achieved the highest rank, outperforming other algorithms in CEC01, CEC02, CEC07, CEC09, and CEC12. Additionally, it obtained the best average performance in CEC05, CEC08, and CEC10. Meanwhile, the LPB algorithm secured second place, demonstrating superior performance in CEC03, CEC04, and CEC06. The FDO ranked third, while the remaining algorithms exhibited the lowest performance. To further evaluate the proposed modifications, Table 16 presents comparative results for both 10- and 20-dimensional problem instances. The outcomes demonstrate that EESB-FDO continues to perform successfully in the 20-dimensional setting, particularly on benchmark functions CEC01, CEC02, CEC05, CEC07, CEC08, CEC11, and CEC12, thereby confirming the robustness and scalability of the algorithm across higher-dimensional search spaces.

On the other hand, as illustrated in Table 17, the results indicate that the EEBC-FDO algorithm outperforms the other algorithms on the CEC03, CEC05, and CEC09 benchmark functions. Conversely, the remaining functions show superior performance by the other algorithms compared to EEBC-FDO. Subsequently, EEBC-FDO was evaluated using 20-dimensional benchmark functions, with the results presented in Table 18. The outcomes indicate that EEBC-FDO achieved superior performance, outperforming all competing algorithms on CEC01, CEC02, CEC06, CEC07, CEC08, and CEC11, thereby demonstrating its effectiveness and competitiveness in high-dimensional optimization tasks.

6.4 Wilcoxon rank-sum test

One of the primary statistical methods used for hypothesis testing in this study is the Wilcoxon rank-sum test, which evaluates whether the performance differences between two algorithms are statistically significant. This non-parametric test is well-suited for optimization problems, as it does not assume a normal distribution of results and instead relies on rank comparisons. The test produces a p -value (probability value) that quantifies the strength of evidence against the null hypothesis. A smaller p -value (typically $p < 0.05$) indicates strong evidence to reject the null hypothesis, suggesting that one algorithm significantly outperforms the other, whereas larger p -values imply that the performance difference is not statistically significant [72–74].

Table 19 presents the Wilcoxon test results comparing FDO, EESB-FDO, and EEBC-FDO against MFDO on the CEC2022 benchmark suite. The findings show that, for many functions (e.g., CEC01, CEC03, CEC04, CEC05), the extremely small p -values confirm that the proposed variants achieve statistically significant improvements. In contrast, higher p -values in functions such as CEC02 and CEC09 indicate no significant difference, suggesting that MFDO retains competitiveness in these landscapes, where exploration dominates over exploitation. Notably, EESB-FDO and EEBC-FDO reduce the number of

TABLE 15 Results of comparing EESB-FDO with FDO and Other chosen algorithms using CEC 2022 benchmark functions.

CEC	Metrics	EESB-FDO	FDO	AOA	LPB	WOA	FOX
CEC01	Mean	3.52E+03	3.93E+03	1.45E+04	8.77E+03	2.75E+04	3.99E+03
	SD	3.31E+03	3.39E+03	5.77E+03	5.59E+03	1.30E+04	3.56E+03
CEC02	Mean	4.07E+02	4.09E+02	1.33E+03	4.21E+02	4.57E+02	4.43E+02
	SD	1.77E+01	2.13E+01	6.85E+02	3.05E+01	1.04E+02	3.79E+01
CEC03	Mean	6.13E+02	6.14E+02	6.42E+02	6.00E+02	6.23E+02	6.55E+02
	SD	7.37E+00	8.94E+00	9.24E+00	2.25E-01	1.07E+02	9.58E+00
CEC04	Mean	8.25E+02	8.26E+02	8.37E+02	8.22E+02	8.44E+02	8.36E+02
	SD	7.73E+00	7.99E+00	7.36E+00	7.35E+00	1.70E+01	1.16E+01
CEC05	Mean	9.05E+02	9.57E+02	1.39E+03	1.14E+03	1.60E+03	1.48E+03
	SD	1.72E+02	1.06E+02	1.66E+02	2.63E+02	5.00E+02	5.49E+01
CEC06	Mean	4.41E+03	5.08E+03	1.57E+07	3.91E+03	6.84E+03	4.36E+03
	SD	3.16E+03	2.75E+03	4.11E+07	1.96E+03	4.91E+03	2.12E+03
CEC07	Mean	2.04E+03	2.04E+03	2.11E+03	2.04E+03	2.08E+03	2.15E+03
	SD	2.06E+01	2.32E+01	4.05E+01	4.60E+01	3.52E+01	8.26E+01
CEC08	Mean	2.23E+03	2.23E+03	2.28E+03	2.25E+03	2.23E+03	2.40E+03
	SD	2.26E+01	3.15E+01	8.27E+01	5.03E+01	5.94E+00	1.51E+02
CEC09	Mean	2.53E+03	2.53E+03	2.76E+03	2.56E+03	2.60E+03	2.63E+03
	SD	1.04E+01	2.68E+01	6.70E+01	3.43E+01	5.91E+01	4.91E+01
CEC10	Mean	2.55E+03	2.57E+03	2.78E+03	2.57E+03	2.71E+03	3.10E+03
	SD	6.35E+01	6.20E+01	2.82E+02	6.37E+01	3.73E+02	6.87E+02
CEC11	Mean	2.71E+03	2.68E+03	3.49E+03	2.80E+03	2.73E+03	3.10E+03
	SD	1.26E+02	1.30E+02	3.66E+02	1.44E+02	3.90E+02	6.87E+02
CEC12	Mean	2.88E+03	2.88E+03	3.03E+03	2.88E+03	2.90E+03	3.02E+03
	SD	1.06E+01	1.65E+01	8.58E+01	1.61E+01	4.39E+01	1.15E+02

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

non-significant cases compared to the original FDO, confirming the effectiveness of the proposed modifications in enhancing exploitation and convergence speed. These variations highlight that algorithm performance is problem-dependent: functions requiring stronger exploitation benefit from EESB-FDO and EEBC-FDO, while functions emphasizing broad exploration (e.g., CEC02, CEC09) show less pronounced differences. Overall, the Wilcoxon rank-sum test not only validates the robustness of the proposed variants but also provides insights into the problem characteristics where they achieve the most significant advantages.

6.5 Quantitative measurement metrics

Quantitative analysis is a crucial metric that provides detailed insights and in-depth observations of new algorithms and their modifications. Figures 6–8 illustrate this analysis, where the convergence of iterations represents the measurement of an agent’s global best performance. Among the unimodal test functions (TF1–TF7), TF1 was selected as

the representative function. For multimodal test functions (TF8–TF13), TF11 was chosen, while for composite test functions (TF14–TF19), TF15 was nominated. To further investigate performance, a two-dimensional search space was explored over 150 iterations using 10 search agents. As the number of iterations increased, X_i^* (the global best agent) demonstrated greater precision. Additionally, when the scout bee emphasized exploitation and local search, the proposed modifications achieved superior results within fewer iterations, thereby demonstrating the algorithm’s adaptability and effectiveness.

6.6 Real-world applications of EESB-FDO and EEBC-FDO

6.6.1 Gear train design problem

The primary objective of the gear train design (GTD) problem is to minimize the cost associated with the transmission ratio of the gear train configuration illustrated in Figure 9 [75, 76]. This

TABLE 16 Results of comparing EESB-FDO with FDO for 10 and 20 dimensions using CEC 2022 benchmark functions.

CEC	Metrics	10-Dimensional		20-Dimensional	
		EESB-FDO	FDO	EESB-FDO	FDO
CEC01	Mean	3.52E+03	3.93E+03	4.88E+04	5.11E+04
	SD	3.31E+03	3.39E+03	1.55E+04	1.57E+04
CEC02	Mean	4.07E+02	4.09E+02	4.63E+02	4.67E+02
	SD	1.77E+01	2.13E+01	2.12E+01	2.33E+01
CEC03	Mean	6.13E+02	6.14E+02	6.47E+02	6.40E+02
	SD	7.37E+00	8.94E+00	8.71E+00	9.67E+00
CEC04	Mean	8.25E+02	8.26E+02	8.70E+02	8.85E+02
	SD	7.73E+00	7.99E+00	1.28E+01	2.20E+01
CEC05	Mean	9.05E+02	9.57E+02	2.16E+03	2.29E+03
	SD	1.72E+02	1.06E+02	6.04E+02	6.20E+02
CEC06	Mean	4.41E+03	5.08E+03	6.35E+03	5.45E+03
	SD	3.16E+03	2.75E+03	5.88E+03	4.62E+03
CEC07	Mean	2.04E+03	2.04E+03	2.14E+03	2.20E+03
	SD	2.06E+01	2.32E+01	6.33E+01	1.10E+02
CEC08	Mean	2.23E+03	2.23E+03	2.42E+03	2.48E+03
	SD	2.26E+01	3.15E+01	3.24E+02	4.01E+02
CEC09	Mean	2.53E+03	2.53E+03	2.49E+03	2.42E+03
	SD	1.04E+01	2.68E+01	6.20E+00	3.24E+02
CEC10	Mean	2.55E+03	2.57E+03	3.57E+03	3.37E+03
	SD	6.35E+01	6.20E+01	6.96E+02	9.67E+02
CEC11	Mean	2.71E+03	2.68E+03	2.94E+03	2.97E+03
	SD	1.26E+02	1.30E+02	4.78E+01	7.40E+02
CEC12	Mean	2.88E+03	2.88E+03	2.99E+03	3.09E+03
	SD	1.06E+01	1.65E+01	5.57E+02	1.09E+02

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

optimization problem involves four discrete decision variables, T_a, T_b, T_d , and T_f , each corresponding to the number of teeth on a specific gear within the train. The problem is formulated mathematically as follows:

Suppose : $\bar{x} = (x_1, x_2, x_3, x_4) = T_a, T_b, T_d, T_f$ (12)

Minimize : $f(\bar{x}) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4} \right)^2$ (13)

Subject to : $\begin{cases} g1 - 4(\bar{x}) = 12 - x_i \leq 0 \\ g5 - 8(\bar{x}) = (60 - x) \leq 0 \\ x_1, x_2, x_3, x_4 \in [12, 60] \end{cases}$ (14)

(15)

6.6.2 Three-bar truss problem

As illustrated in Figure 10, the three-bar truss structure is considered for structural optimization. The objective is to minimize the total volume of the truss while ensuring that the design

adheres to specified stress constraints [77, 78]. The corresponding mathematical formulation is provided below.

Minimize : $f(X) = \left(2\sqrt{2x_1} + x_2 \right)^* L$ (16)

subject to : $\begin{cases} g1(x) = \frac{\sqrt{2x_1} + x_2}{\sqrt{2x_1^2 + 2x_1 x_2}} P - \sigma \leq 0 \\ g2(x) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1 x_2}} P - \sigma \leq 0 \end{cases}$ (17)

(18)

$\begin{cases} g3(x) = \frac{1}{\sqrt{2x_2 + x_1}} P - \sigma \leq 0 \end{cases}$ (19)

where $0 \leq x_1, x_2 \leq 1, l = 100\text{ cm}, P = 2 \frac{KN}{cm^2}$, and $\sigma = 2 \frac{KN}{cm^2}$.

6.6.3 The pathological IgG fraction in the nervous system

This real-world problem, described in Ismaeel Ghareb et al. [79], aims to determine the optimal solution for pathological conditions in humans or animals. It is known as the dimmer

TABLE 17 Results of comparing EEBC-FDO with FDO and Other chosen algorithms using CEC 2022 benchmark function.

CEC	Metrics	EEBC-FDO	FDO	AOA	LPB	WOA	FOX
CEC01	Mean	3.84E+03	3.93E+03	1.45E+04	8.77E+03	2.75E+04	3.99E+03
	SD	4.17E+03	3.39E+03	5.77E+03	5.59E+03	1.30E+04	3.56E+03
CEC02	Mean	4.16E+02	4.09E+02	1.33E+03	4.21E+02	4.57E+02	4.43E+02
	SD	2.79E+01	2.13E+01	6.85E+02	3.05E+01	1.04E+02	3.79E+01
CEC03	Mean	6.12E+02	6.14E+02	6.42E+02	6.00E+02	6.23E+02	6.55E+02
	SD	7.94E+00	8.94E+00	9.24E+00	2.25E-01	1.07E+02	9.58E+00
CEC04	Mean	8.28E+02	8.26E+02	8.37E+02	8.22E+02	8.44E+02	8.36E+02
	SD	1.03E+01	7.99E+00	7.36E+00	7.35E+00	1.70E+01	1.16E+01
CEC05	Mean	9.47E+02	9.57E+02	1.39E+03	1.14E+03	1.60E+03	1.48E+03
	SD	5.13E+01	1.06E+02	1.66E+02	2.63E+02	5.00E+02	5.49E+01
CEC06	Mean	5.56E+03	5.08E+03	1.57E+07	3.91E+03	6.84E+03	4.36E+03
	SD	3.96E+03	2.75E+03	4.11E+07	1.96E+03	4.91E+03	2.12E+03
CEC07	Mean	2.04E+03	2.04E+03	2.11E+03	2.04E+03	2.08E+03	2.15E+03
	SD	2.68E+01	2.32E+01	4.05E+01	4.60E+01	3.52E+01	8.26E+01
CEC08	Mean	2.24E+03	2.23E+03	2.28E+03	2.25E+03	2.23E+03	2.40E+03
	SD	5.17E+01	3.15E+01	8.27E+01	5.03E+01	5.94E+00	1.51E+02
CEC09	Mean	2.53E+03	2.53E+03	2.76E+03	2.56E+03	2.60E+03	2.63E+03
	SD	9.25E-13	1.04E+01	6.70E+01	3.43E+01	5.91E+01	4.91E+01
CEC10	Mean	2.57E+03	2.55E+03	2.78E+03	2.57E+03	2.71E+03	3.10E+03
	SD	6.15E+01	6.20E+01	2.82E+02	6.37E+01	3.73E+02	6.87E+02
CEC11	Mean	2.70E+03	2.68E+03	3.49E+03	2.80E+03	2.73E+03	3.10E+03
	SD	1.23E+02	1.30E+02	3.66E+02	1.44E+02	3.90E+02	6.87E+02
CEC12	Mean	2.88E+03	2.88E+03	3.03E+03	2.88E+03	2.90E+03	3.02E+03
	SD	2.72E+01	1.65E+01	8.58E+01	1.61E+01	4.39E+01	1.15E+02

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

solution and is used to identify the IgG fraction in the nervous system. The mathematical formulation is presented below.

$$\begin{aligned}
 \text{IgGp} &= X_i \text{ So, IgG (IgGp)} = Y(X_i) \text{ then:} \\
 Y(X_i) &= \sum_{i=1}^n (0.41 + 0.0014 X_i) \quad (20)
 \end{aligned}$$

where X_i represents the input albumin quotient values, and $Y(X_i)$ denotes the corresponding optimized IgG values. These equations, supported by Figure 11, demonstrate a real-world application of statistical modeling and optimization in medical diagnostics, offering an improved framework for evaluating immune dysfunctions in the nervous system.

6.6.4 Integrated cyber-physical attack for manufacturing system

This real-world application, supported by Figure 12, is designed to identify potentially harmful nodes within network systems, with a particular focus on software and hardware engineering design

phases. The primary objective is to optimize the detection and regulation of these nodes within a normalized range of [0, 1], as their presence can introduce critical vulnerabilities, including system bugs and risk-prone components [80].

The fitness function is defined as

$$F((I(p, t)) = \sum X^3 + A \sum X^2 + B \sum X + C \quad (21)$$

where X represents the node state values, and the coefficients A , B , and C are functions of system parameters, calculated as

$$\begin{aligned}
 A &= 0.0283 \left(1 + \frac{1}{d} - k_2 \right) \\
 B &= \frac{0.0283 - 1.0283k_2}{d} \\
 C &= \frac{0.0013 k_1 - 0.0283k_2}{d}
 \end{aligned}$$

in these equations, d denotes the total number of nodes (ranging from 15 to 36), while $k_1 \in [0, 1]$ and $k_2 \in [0.1, 0.5]$ are system-specific constants reflecting infection rate and risk thresholds. The

TABLE 18 Results of comparing EEBC-FDO with FDO for (10 and 20) dimensions using CEC 2022 benchmark function.

CEC	Metrics	10 Dim		20 Dim	
		EEBC-FDO	FDO	EEBC-FDO	FDO
CEC01	Mean	3.84E+03	3.93E+03	4.94E+04	5.11E+04
	SD	4.17E+03	3.39E+03	1.56E+04	1.57E+04
CEC02	Mean	4.16E+02	4.09E+02	4.64E+02	4.67E+02
	SD	2.79E+01	2.13E+01	1.72E+01	2.33E+01
CEC03	Mean	6.12E+02	6.14E+02	6.40E+02	6.40E+02
	SD	7.94E+00	8.94E+00	1.13E+01	9.67E+00
CEC04	Mean	8.28E+02	8.26E+02	8.86E+02	8.85E+02
	SD	1.03E+01	7.99E+00	1.98E+01	2.20E+01
CEC05	Mean	9.47E+02	9.57E+02	2.51E+03	2.29E+03
	SD	5.13E+01	1.06E+02	7.02E+02	6.20E+02
CEC06	Mean	5.56E+03	5.08E+03	4.57E+03	5.45E+03
	SD	3.96E+03	2.75E+03	3.65E+03	4.62E+03
CEC07	Mean	2.04E+03	2.04E+03	2.17E+03	2.20E+03
	SD	2.68E+01	2.32E+01	7.81E+01	1.10E+02
CEC08	Mean	2.24E+03	2.23E+03	2.39E+03	2.48E+03
	SD	5.17E+01	3.15E+01	1.36E+02	4.01E+02
CEC09	Mean	2.53E+03	2.53E+03	2.48E+03	2.42E+03
	SD	9.25E-13	1.04E+01	2.81E+00	3.24E+02
CEC10	Mean	2.57E+03	2.55E+03	3.46E+03	3.37E+03
	SD	6.15E+01	6.20E+01	9.32E+02	9.67E+02
CEC11	Mean	2.70E+03	2.68E+03	2.92E+03	2.97E+03
	SD	1.23E+02	1.30E+02	4.05E+01	7.40E+02
CEC12	Mean	2.88E+03	2.88E+03	3.16E+03	3.09E+03
	SD	2.72E+01	1.65E+01	1.93E+02	1.09E+02

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

aim of this fitness function is to identify configurations in which $F(I(p, t)) \rightarrow 0$, indicating an optimal system state with minimal harmful node influence.

6.6.5 Real-world application comparison and discussion

The final step in evaluating any newly proposed algorithm or enhancement involves its application to real-world problems. To assess performance, standard evaluation metrics such as the mean, standard deviation, and *p*-value are employed for comparative analysis. In this section, four real-world applications were utilized to fulfill this objective. The corresponding results are presented in Tables 20–22.

The results of EESB-FDO and EEBC-FDO, presented in Tables 20, 21, demonstrate the performance of these two modified versions in comparison with the FDO and MFDO algorithms. In three real-world applications, both EESB-FDO and EEBC-FDO consistently achieved the best rankings, with lower mean and standard deviation values, indicating improved stability and

reliability, outperforming the other algorithms. However, one of the modified versions exhibited a slight weakness in a specific case. On the other hand, the *p*-value, which quantifies the strength of evidence against the null hypothesis, indicates that a smaller *p*-value, typically $p < 0.05$, indicates strong evidence to reject the null hypothesis, suggesting that the observed results are unlikely to have occurred by chance. Table 22 shows the results of the comparison between FDO and EESB-FDO, which has just one case that fails to reject the null hypothesis for the gear train design, and FDO with EEBC-FDO, which shows the same result but with a different real-world problem’s integrated cyber-physical attack for manufacturing system.

7 Conclusion

This study proposed two enhanced variants of the fitness-dependent optimizer (FDO) designed to address its limitations in exploitation and convergence speed. The first variant, EESB-FDO, introduces a stochastic boundary repositioning mechanism that

TABLE 19 WILCOXON rank-sum test for CEC2022 benchmark functions.

CEC	FDO_vs_MFDO	EESB-FDO_vs_MFDO	EEBC-FDO_vs_MFDO
CEC01	0.000002	0.000002	0.000002
CEC02	0.942604	0.734325	0.110878
CEC03	0.000002	0.000002	0.000002
CEC04	0.000011	0.000006	0.000003
CEC05	0.000003	0.000037	0.000002
CEC06	0.000002	0.000031	0.000004
CEC07	0.000063	0.000028	0.000022
CEC08	0.000012	0.000031	0.000002
CEC09	0.626221	0.194059	0.367185
CEC10	0.298944	0.028486	0.068714
CEC11	0.662004	0.007816	0.007734
CEC12	0.177907	0.040702	0.393334

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

TABLE 20 The results of comparison FDO with EESB-FDO and MFDO.

Real-world problems	Metrics	FDO	EESB-FDO	MFDO
Pathological IgG fraction in the nervous system	Mean	4.91E+00	4.90E+00	4.91E+00
	SD	1.98E-02	1.50E-02	1.47E-02
Integrated cyber-physical attack for manufacturing system	Mean	3.46E-04	−3.72E+05	4.52E-04
	SD	1.44E-04	1.91E+06	1.86E-04
Three-bar truss	Mean	1.62E+02	6.03E+01	−2.39E+02
	SD	1.97E+02	1.48E+02	9.23E+02
Gear train design	Mean	1.08E-07	4.31E-08	−2.88E+04
	SD	3.02E-07	2.32E-07	3.27E+04

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

reassigns scout bees that exceed the search space to random positions within the valid range. The second variant, EEBC-FDO, employs a boundary carving strategy that redirects out-of-bound solutions toward feasible regions using customized correction equations. In addition, a third enhancement, the ELFS strategy, constrains the Levy flight step range to $(-1, 1)$, ensuring more stable and bounded exploration behavior. To validate the effectiveness of these modifications, extensive experiments were conducted using classical benchmark functions and CEC 2019 and CEC 2022 test suites. The results demonstrated that both EESB-FDO and EEBC-FDO significantly improve exploitation capability and convergence performance compared to the FDO and other state-of-the-art algorithms.

In the future, various enhancements and extensions of the current limited work can be developed. An important aspect to improve is the application of multi-objective vector optimization, utilizing functions specifically related to multi-objective optimization techniques [81]. A major limitation of this study lies in the inability to test certain benchmark functions

in higher dimensions, which limits the generalizability of the findings. Future work could explore dimensionality reduction techniques or alternative benchmark suites that support high-dimensional testing. While classical benchmark functions can typically be tested across various dimensions, including 50, many CEC2019 functions are restricted and do not support high-dimensional testing. This constraint limits the generalizability and robustness analysis of the proposed algorithm. However, to partially address this issue, the CEC2022 benchmark set, which allows testing with 20 dimensions, can be used as an alternative for future evaluations. Despite these limitations, the enhancements should be applied to classification tasks and parameter tuning to improve the new model. This would also provide an opportunity to test the model on datasets related to education, healthcare, and other fields [79, 82]. Furthermore, the modification of this algorithm has not yet been validated on classification tasks involving real-world healthcare datasets, such as those related to sleep disorders or chronic disease diagnosis [83, 84]. As a result, its adaptability and robustness in managing

TABLE 21 The results of comparison FDO with EEBC-FDO and MFDO.

Real-world problems	Metrics	FDO	EEBC-FDO	MFDO
Pathological IgG fraction in the nervous system	Mean	4.91E+00	4.79E+00	4.91E+00
	SD	1.98E-02	1.93E-01	1.47E-02
Integrated cyber-physical attack for manufacturing system	Mean	3.46E-04	3.30E-04	4.52E-04
	SD	1.44E-04	2.96E-04	1.86E-04
Three-bar truss	Mean	1.62E+02	1.61E+02	−2.39E+02
	SD	1.97E+02	1.15E+02	9.23E+02
Gear train design	Mean	1.08E-07	4.14E-11	−2.88E+04
	SD	3.02E-07	6.57E-11	3.27E+04

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

TABLE 22 WILCOXON rank-sum test for four real-world problems.

Real-world problem	FDO_vs_ EESB-FDO	FDO_vs_ EEBC-FDO
Pathological IgG fraction in the nervous system	0.013194	0.000174
Integrated cyber-physical attack for manufacturing system	0.000002	0.149443
Three-bar truss	0.000979	0.005386
Gear train design	0.205888	0.049836

The bold values in each row indicate the best performance among all compared algorithms for that test function (TF). Specifically, the lowest mean value represents the best solution quality*, while for the standard deviation (SD), the lowest value reflects the highest stability/consistency of results. *The lowest mean value represents the best solution quality.

noisy, imbalanced, and high-dimensional medical data remain uncertain due to the absence of dedicated benchmarking in medical classification scenarios. Although improvements have been demonstrated across several real-world applications, further development is needed by verifying dataset authenticity and incorporating security-focused algorithms to better guide the exploitation and exploration strategies in searching for new solutions [85, 86].

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

AF: Formal analysis, Funding acquisition, Methodology, Writing – original draft, Writing – review & editing, Conceptualization. AMA: Conceptualization, Methodology, Writing – review & editing, Resources. AAA: Conceptualization, Data curation, Writing – review & editing, Project administration.

Funding

The author(s) declare that no financial support was received for the research and/or publication of this article.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declare that no Gen AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher’s note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

- Sörensen K, Glover F. *METAHEURISTICS 1.1 Definition*. Available online at: <http://metaheuristics.eu> (Accessed March 4, 2025).
- Yang X-S. *Nature-Inspired Optimization Algorithms*. London: Elsevier (2014). doi: 10.1016/B978-0-12-416743-8.00010-5
- Fister I, Yang X-S, Fister I, Brest J, Fister D. *A Brief Review of Nature-Inspired Algorithms for Optimization* (2013). Available online at: <http://arxiv.org/abs/1307.4186> (Accessed April 8, 2025).
- Talbi E. *Metaheuristics: From Design to Implementation*. Hoboken, NJ: Wiley (2009). doi: 10.1002/9780470496916
- Mustafa AI, Aladdin AM, Mohammed-Taha SR, Hasan DO, Rashid TA. *Is PSO the Ultimate Algorithm or Just Hype?* Piscataway, NJ: TechRxiv (2025). doi: 10.36227/techrxiv.173609965.50383071v1
- Aladdin AM, Rashid TA. A new Lagrangian problem crossover—a systematic review and meta-analysis of crossover standards. *Systems*. (2023) 11:144. doi: 10.3390/systems11030144
- Deb K. An introduction to genetic algorithms. *Sadhana*. (1999) 24:293–315. doi: 10.1007/BF02823145
- Hassanat A, Almohammadi K, Alkafaween E, Abunawas E, Hammouri A, Prasath VBS. choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. *Information*. (2019) 10:390. doi: 10.3390/info10120390
- Dokeroglu T, Sevinc E, Kucukylmaz T, Cosar A. A survey on new generation metaheuristic algorithms. *Comput Ind Eng*. (2019) 137:106040. doi: 10.1016/j.cie.2019.106040
- Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN'95 - International Conference on Neural Networks*. IEEE, p. 1942–8. doi: 10.1109/ICNN.1995.488968
- Yapici H, Cetinkaya N. A new meta-heuristic optimizer: pathfinder algorithm. *Appl Soft Comput J*. (2019) 78:545–68. doi: 10.1016/j.asoc.2019.03.012
- Zervoudakis K, Tsafarakis S. A mayfly optimization algorithm. *Comput Ind Eng*. (2020) 145:106559. doi: 10.1016/j.cie.2020.106559
- Ghasemi-Marzbali A. A novel nature-inspired meta-heuristic algorithm for optimization: bear smell search algorithm. *Soft comput*. (2020) 24:13003–35. doi: 10.1007/s00500-020-04721-1
- Ahmadianfar I, Bozorg-Haddad O, Chu X. Gradient-based optimizer: a new metaheuristic optimization algorithm. *Inf Sci (N Y)*. (2020) 540:131–59. doi: 10.1016/j.ins.2020.06.037
- Zhang Y, Jin Z. Group teaching optimization algorithm: a novel metaheuristic method for solving global optimization problems. *Expert Syst Appl*. (2020) 148:113246. doi: 10.1016/j.eswa.2020.113246
- Askari Q, Saeed M, Younas I. Heap-based optimizer inspired by corporate rank hierarchy for global optimization. *Expert Syst Appl*. (2020) 161:113702. doi: 10.1016/j.eswa.2020.113702
- Hashim FA, Houssein EH, Mabrouk MS, Al-Atabany W, Mirjalili S. Henry gas solubility optimization: a novel physics-based algorithm. *Fut Gen Comput Syst*. (2019) 101:646–67. doi: 10.1016/j.future.2019.07.015
- Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH. *Marine Predators Algorithm: A Nature-inspired Metaheuristic*. Available online at: <https://linkinghub.elsevier.com/retrieve/pii/S0957417420302025>
- Askari Q, Younas I, Saeed M. Political optimizer: a novel socio-inspired meta-heuristic for global optimization. *Knowl Based Syst*. (2020) 195:105709. doi: 10.1016/j.knsys.2020.105709
- Zhao W, Wang L, Zhang Z. Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowl Based Syst*. (2019) 163:283–304. doi: 10.1016/j.knsys.2018.08.030
- Abualigah L, Younsri D, Abd Elaziz M, Ewees AA, Al-qaness MAA, Gandomi AH. *Aquila Optimizer: a novel meta-heuristic optimization algorithm*. *Comput Ind Eng*. (2021) 157:107250. doi: 10.1016/j.cie.2021.107250
- MiarNaeimi F, Azizyan G, Rashki M. Horse herd optimization algorithm: a nature-inspired algorithm for high-dimensional optimization problems. *Knowl Based Syst*. (2021) 213:106711. doi: 10.1016/j.knsys.2020.106711
- Oyelade ON, Ezugwu AES, Mohamed TIA, Abualigah L. Ebola optimization search algorithm: a new nature-inspired metaheuristic optimization algorithm. *IEEE Access*. (2022) 10:16150–77. doi: 10.1109/ACCESS.2022.3147821
- Shami TM, Grace D, Burr A, Mitchell PD. Single candidate optimizer: a novel optimization algorithm. *Evol Intell*. (2024) 17:863–87. doi: 10.1007/s12065-022-00762-7
- Wolpert DH, Macready WG. No free lunch theorems for optimization. *IEEE Trans Evol Comput*. (1997) 1:67–82. doi: 10.1109/4235.585893
- Arora S, Singh S. A conceptual comparison of firefly algorithm, bat algorithm and cuckoo search in 2013. In: *International Conference on Control, Computing, Communication and Materials (ICCCCM)*. Allahabad: IEEE (2013). p. 1–4. doi: 10.1109/ICCCCM.2013.6648902
- Cohen JD, McClure SM, Yu AJ. Should I stay or should I go? How the human brain manages the trade-off between exploitation and exploration. *Philos Trans R Soc B: Biol Sci*. (2007) 2007:933–42. doi: 10.1098/rstb.2007.2098
- Mehlhorn K, Newell BR, Todd PM, Lee MD, Morgan K, Braithwaite VA, et al. Unpacking the exploration-exploitation tradeoff: a synthesis of human and animal literatures. *Decision*. (2015) 2:191–215. doi: 10.1037/dec0000033
- March JG. Exploration and exploitation in organizational learning. *Organiz Sci*. (1991) 2:71–87. doi: 10.1287/orsc.2.1.71
- Aladdin M, Abdullah JM, Salih KOM, Rashid TA, Sagban R, Alsaddan A, et al. Fitness dependent optimizer for IoT healthcare using adapted parameters: a case study implementation. In: *Practical Artificial Intelligence for Internet of Medical Things*. Boca Raton, FL: CRC Press (2023). 45–61. doi: 10.1201/9781003315476-3
- Abdullah JM, Ahmed T. Fitness dependent optimizer: inspired by the bee swarming reproductive process. *IEEE Access*. (2019) 7:43473–86. doi: 10.1109/ACCESS.2019.2907012
- Muhammed DA, Saeed SAM, Rashid TA. Improved fitness-dependent optimizer algorithm. *IEEE Access*. (2020) 8:19074–88. doi: 10.1109/ACCESS.2020.2968064
- Chiu PC, Selamat A, Krejcar O, Kuok KK. Hybrid sine cosine and fitness dependent optimizer for global optimization. *IEEE Access*. (2021) 9:128601–22. doi: 10.1109/ACCESS.2021.3111033
- Salih JE, Mohammed HM, Abdul ZK. Modified fitness dependent optimizer for solving numerical optimization functions. *IEEE Access*. (2022) 10:83916–30. doi: 10.1109/ACCESS.2022.3197290
- Abdulla HS, Ameen AA, Saeed SI, Mohammed IA, Rashid TA. MRSO: balancing exploration and exploitation through modified rat swarm optimization for global optimization. *Algorithms*. (2024) 17:423. doi: 10.3390/a17090423
- Deb K. *An Introduction to Genetic Algorithms*. Heidelberg: Springer-Verlag.
- Srinivas M, Patnaik LM. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Trans Syst Man Cybern*. (1994) 24:656–67. doi: 10.1109/21.286385
- Liang JJ, Qin AK, Suganthan PN, Baskar S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Trans Evol Comput*. (2006) 10:281–95. doi: 10.1109/TEVC.2005.857610
- Blum C. Ant colony optimization: introduction and recent trends. *Phys Life Rev*. (2005) 2:353–73. doi: 10.1016/j.plev.2005.10.001
- Stützle T, Hoos HH. Ant system. *Fut Gen Comput Syst*. (2000) 16:889–914. doi: 10.1016/S0167-739X(00)00043-1
- Cheung NJ, Ding X-M, Shen H-B. Adaptive firefly algorithm: parameter analysis and its application. *PLoS ONE*. (2014) 9:e112634. doi: 10.1371/journal.pone.0112634
- Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH. The arithmetic optimization algorithm. *Comput Methods Appl Mech Eng*. (2021) 376:113609. doi: 10.1016/j.cma.2020.113609
- Zhang J, Zhang G, Huang Y, Kong M. A novel enhanced arithmetic optimization algorithm for global optimization. *IEEE Access*. (2022) 10:75040–62. doi: 10.1109/ACCESS.2022.3190481
- Bhattacharyya T, Chatterjee B, Singh PK, Yoon JH, Geem ZW, Sarkar R. Mayfly in harmony: a new hybrid meta-heuristic feature selection algorithm. *IEEE Access*. (2020) 8:195929–45. doi: 10.1109/ACCESS.2020.3031718
- Mohammed HM, Rashid TA. Chaotic fitness-dependent optimizer for planning and engineering design. *Soft Comput*. (2021) 25:14281–95. doi: 10.1007/s00500-021-06135-z
- Abdul-Minaam DS, Al-Mutairi MES W, Awad MA, El-Ashmawi WH. An adaptive fitness-dependent optimizer for the one-dimensional bin packing problem. *IEEE Access*. (2020) 8:97959–74. doi: 10.1109/ACCESS.2020.2985752
- Daraz A, Malik SA, Mokhlis H, Haq IU, Laghari GF, Mansor NN. Fitness dependent optimizer-based automatic generation control of multi-source interconnected power system with non-linearities. *IEEE Access*. (2020) 8:100989–1003. doi: 10.1109/ACCESS.2020.2998127
- Daraz A, Malik SA, Mokhlis H, Haq IU, Zafar F, Mansor NN. Improved-fitness dependent optimizer based FOI-PD controller for automatic generation control of multi-source interconnected power system in deregulated environment. *IEEE Access*. (2020) 8:197757–75. doi: 10.1109/ACCESS.2020.3033983
- Daraz A, Abdullah Malik S, Haq IU, Khan KB, Laghari GF, Zafar F. Modified PID controller for automatic generation control of multi-source interconnected power system using fitness dependent optimizer algorithm. *PLoS ONE*. (2020) 15:e0242428. doi: 10.1371/journal.pone.0242428

50. Abbas DK, Rashid TA, Abdalla KH, Bacanin N, Alsadoon A. Using fitness dependent optimizer for training multi-layer perceptron. *J Internet Technol.* (2021) 22:1575–85. doi: 10.53106/160792642021122207011
51. Laghari GF, Malik SA, Daraz A, Ullah A, Alkhalifah T, Aslam S. A numerical approach for solving nonlinear optimal control problems using the hybrid scheme of fitness dependent optimizer and bernstein polynomials. *IEEE Access.* (2022) 10:50298–313. doi: 10.1109/ACCESS.2022.3173285
52. Mirza F, Haider SK, Ahmed A, Rehman AU, Shafiq M, Bajaj M, et al. Generalized regression neural network and fitness dependent optimization: application to energy harvesting of centralized TEG systems. *Energy Rep.* (2022) 8:6332–46. doi: 10.1016/j.egyrs.2022.05.003
53. Tahir H, Rashid TA, Rauf HT, Bacanin N, Chhabra M, Vimal S, et al. *Improved Fitness-Dependent Optimizer for Solving Economic Load Dispatch Problem.* London: Hindawi Limited (2022). doi: 10.1155/2022/7055910
54. Abdulkhaleq MT, Rashid TA, Hassan BA, Alsadoon A, Bacanin N, Chhabra A, et al. Fitness dependent optimizer with neural networks for COVID-19 patients. *Comput Methods Programs Biomed.* (2023). 3:100090. doi: 10.1016/j.cmpbup.2022.100090
55. Rathi S, Nagpal R, Srivastava G, Mehrotra D. A multi-objective fitness dependent optimizer for workflow scheduling. *Appl Soft Comput.* (2024) 152:111247. doi: 10.1016/j.asoc.2024.111247
56. El-Ashmawi WH, Slowik A, Ali AF. An effective fitness dependent optimizer algorithm for edge server allocation in mobile computing. *Soft Comput.* (2024) 28:6855–77. doi: 10.1007/s00500-023-09582-y
57. Farahmand-Tabar S, Rashid TA. Steel plate fault detection using the fitness dependent optimizer and neural networks. *arXiv [Preprint].* arXiv:2405.00006. (2024). doi: 10.48550/arXiv.2405.00006
58. Roy S, Sinha Chaudhuri S. Cuckoo search algorithm using lévy flight: a review. *Int J Modern Educ Comput Sci.* (2013) 5:10–5. doi: 10.5815/ijmecs.2013.12.02
59. Yang X-S. Flower pollination algorithm for global optimization. In: Durand-Lose J, Jonoska N, editors. *Unconventional Computation and Natural Computation.* Vol. 7445. London: Springer (2012). 240–9. doi: 10.1007/978-3-642-32894-7_27
60. Mirjalili S. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput Appl.* (2016) 27:1053–73. doi: 10.1007/s00521-015-1920-1
61. Qin F, Zain AM, Zhou K-Q, Yusup NB, Prasetya DD, Abdul R. Hybrid harmony search algorithm integrating differential evolution and Lévy flight for engineering optimization. *IEEE Access.* (2025) 13:13534–72. doi: 10.1109/ACCESS.2025.3529714
62. Mirjalili S, Gandomi AH, Mirjalili SZ, Saremi S, Faris H, Mirjalili SM. Salp Swarm algorithm: a bio-inspired optimizer for engineering design problems. *Adv Eng Softw.* (2017) 114:163–91. doi: 10.1016/j.advengsoft.2017.07.002
63. Li S, Chen H, Wang M, Heidari AA, Mirjalili S. Slime mould algorithm: a new method for stochastic optimization. *Fut Gen Comput Syst.* (2020) 111:300–23. doi: 10.1016/j.future.2020.03.055
64. Xie J, Zhou Y, Chen H. A Novel Bat Algorithm based on differential operator and Lévy flights trajectory. *Comput Intell Neurosci.* (2013) 2013:1–13. doi: 10.1155/2013/453812
65. Mirjalili S, Mirjalili SM, Lewis A. Grey Wolf optimizer. *Adv Eng Softw.* (2014) 69:46–61. doi: 10.1016/j.advengsoft.2013.12.007
66. Rashedi E, Nezamabadi-pour H, Saryazdi S. GSA: a gravitational search algorithm. *Inf Sci (N Y).* (2009) 179:2232–48. doi: 10.1016/j.ins.2009.03.004
67. Rahman CM, Rashid TA. A new evolutionary algorithm: learner performance based behavior algorithm. *Egypt Inform J.* (2021) 22:213–23. doi: 10.1016/j.eij.2020.08.003
68. Mirjalili S, Lewis A. The Whale optimization algorithm. *Adv Eng Softw.* (2016) 95:51–67. doi: 10.1016/j.advengsoft.2016.01.008
69. Mohammed HM, Rashid TA. *Applied Intelligence.* London: Springer (2022). doi: 10.21203/rs.3.rs-1939478/v1
70. Price KV, Awad NH, Ali MZ, Suganthan PN. *Problem Definitions and Evaluation Criteria for the 100-Digit Challenge Special Session and Competition on Single Objective Numerical Optimization.* Nanyang Technol. Univ. Singapore, Singapore, Tech. Rep. (2018).
71. Ahrari A, Elsayed SM, Sarker R, Essam D, Coello CAC. *Problem Definitions and Evaluation Criteria for the CEC 2022 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization.* Canberra, ACT: University of New South Wales (2022).
72. de Barros RSM, Hidalgo JIG, de Cabral DRL. Wilcoxon rank sum test drift detector. *Neurocomputing.* (2018) 275:1954–63. doi: 10.1016/j.neucom.2017.10.051
73. Thiese MS, Ronna B, Ott U. P value interpretations and considerations. *J Thorac Dis.* (2016) 8:E928–31. doi: 10.21037/jtd.2016.08.16
74. Amin AAH, Aladdin AM, Hasan DO, Mohammed-Taha SR, Rashid TA. Enhancing algorithm selection through comprehensive performance evaluation: statistical analysis of stochastic algorithms. *Computation.* (2023) 11:231. doi: 10.3390/computation11110231
75. Bidar M, Mouhoub M, Sadaoui S, Kanan HR. A novel nature-inspired technique based on mushroom reproduction for constraint solving and optimization. *Int J Comput Intell Appl.* (2020) 19:108. doi: 10.1142/S1469026820500108
76. EKER E. Performance evaluation of capuchin search algorithm through non-linear problems, and optimization of gear train design problem. *Eur J Technic.* (2023) 13:142–9. doi: 10.36222/ejt.1391524
77. Ray T, Saini P. Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Eng Optimiz.* (2001) 33:735–748. doi: 10.1080/03052150108940941
78. Fauzi H, Batool U. A Three-bar Truss Design using single-solution simulated Kalman filter optimizer. *Mekatronika.* (2019) 1:98–102. doi: 10.15282/meatronika.v1i2.4991
79. Ismael Ghareb M, Ali Ahmed Z, Abdullah Ameen A. planning strategy and the use of information technology in higher education: a review study in Kurdistan region government. *Int J Inform Visualiz.* (2019) 3:283–7. doi: 10.30630/joiv.3.3.263
80. Aladdin AM, Rashid TA. Leo: Lagrange Elementary Optimization. *Neural Comput Appl.* (2025). 37:14365–97. doi: 10.1007/s00521-025-11225-2
81. Rashid TA, Ahmed AM, Hassan BA, Yaseen ZM, Mirjalili S, Bacanin N, et al. *Multi-Objective Optimization Techniques.* CRC Press (2025).
82. Hasan DO, Aladdin AM, Hama Amin AA, Rashid TA, Ali YH, Al-Bahri M, et al. Perspectives on the impact of e-learning pre- and post-COVID-19 pandemic—the case of the Kurdistan Region of Iraq. *Sustainability.* (2023) 15:4400. doi: 10.3390/su15054400
83. Mohammed-Taha SR, Aladdin AM, Mustafa AI, Hasan DO, Muhammed RK, Rashid TA. Analyzing the impact of the pandemic on insomnia prediction using machine learning classifiers across demographic groups. *Netw Model Anal Health Inform Bioinform.* (2025) 14:51. doi: 10.1007/s13721-025-00552-y
84. Hasan DO, Aladdin AM. Sleep-related consequences of the COVID-19 pandemic: a survey study on insomnia and sleep apnea among affected individuals. *Insights Public Health J.* (2025) 5:12972. doi: 10.20884/1.iphj.2024.5.2.12972
85. Abdulla HS, Aladdin AM. Enhancing design and authentication performance model: a multilevel secure database management system. *Fut Internet.* (2025) 17:74. doi: 10.3390/fi17020074
86. Muhammed RK, et al. Comparative analysis of AES, Blowfish, Twofish, Salsa20, and ChaCha20 for image encryption. *Kurdistan J Appl Res.* (2024) 9:52–65. doi: 10.24017/science.2024.1.5
87. Hasan DO, Mohammed HM, Abdul ZK. Griffon vultures optimization algorithm for solving optimization problems. *Expert Syst Appl.* (2025) 276:127206. doi: 10.1016/j.eswa.2025.127206

Appendix

This appendix provides a detailed of the benchmark test functions employed to assess the performance of the proposed

algorithms. The evaluation includes a comprehensive set of functions categorized into three main groups: classical benchmark functions (unimodal, multimodal, and composite types), as well as the CEC 2019 and CEC 2022 benchmark suites.