Sohar University Journal of Engineering and Information Technology Innovations (SUJEITI), ISSN: 3078-7513
Volume 2, Issue 2, pp 85-99, June 2024, DOI: 10.69983/SUJEITI/2247
Received: 04/05/2026 Revised: 08/06/2026 Accepted: 09/06/2026

Advances and Challenges in the Fitness Dependent Optimizer: A Systematic Survey (2019–2026)

Ardalan Awlla[1, *], Tarik Rashid[2], Ronak Abdullah[3]

[1]Information Technology Department, Technical College of Informatics, Sulaimani Polytechnic University, Sulaymaniyah, Iraq, ardalan.husin.tci@spu.edu.iq
[2]Engineering Department, Centre for Artificial Intelligence and Innovations, University of Kurdistan Hewler, Erbil, Iraq, tarik.ahmed@ukh.edu.krd
[3]Department of Mathematics, College of Science, University of Sulaimani, Sulaymaniyah, Iraq, runak.abdullah@univsul.edu.iq

* Corresponding author: Ardalan Awlla (ardalan.husin.tci@spu.edu.iq)

Abstract

Nature has always been one of the major sources of inspiration and has shown humans many ways to explore; among the best-known examples is the artificial intelligence swarm algorithm, which copies the group behavior of social animals in nature. The Fitness-Dependent Optimizer (FDO) was presented in 2019. It is a swarm intelligence algorithm that takes its inspiration from the reproductive swarming behavior of honeybees. Since its introduction, FDO has become a very popular topic for researchers, and the literature kept growing with different versions of the algorithm, hybrids, and applications specific to the domain. In accordance with the preferred reporting items for systematic reviews and meta-analyses (PRISMA) 2020 reporting standard, this paper presents a systematic review of FDO research conducted from 2019 to 2026. Initially, a total of 330 records were retrieved from six academic databases, but after applying the inclusion criteria, only 44 studies were selected and constituted the review material. Firstly, the survey discusses the biological inspiration and offers the mathematical formulation of FDO; secondly, it builds a two-axis classification of the variants and modification strategies; Thirdly, it goes through the FDO applications; Fourthly, it offers a quantitative comparison of FDO with three classical metaheuristics Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and Artificial Bee Colony (ABC) for convergence behavior, and exploration and exploitation balance; fifthly, it points out priority research gaps. The most decisive finding of this study is that the replacement of the binary weight factor by a continuous or dynamically oscillating value was, by far, the most impactful modification of FDO. Besides that, FDO's best empirical performance is in constrained engineering optimization. Among the open research challenges are scalability to high dimensions, formal convergence analysis, discrete-domain extensions, and reproducibility of benchmark evaluation.

Keywords*:* Swarm algorithm, Fitness Dependent Optimizer, Systematic review, Engineering applications, Metaheuristic algorithms.

1. Introduction

Optimization is an essential skill that is present in almost all branches of science and engineering. It facilitates the design of various solutions, such as energy-efficient power systems and the training of machine learning (ML) models. In fact, being able to effectively identify optimal or nearly optimal solutions to complex, high-dimensional, and frequently nonlinear problems is extremely important. Traditional optimization methods, such as gradient-based techniques and quadratic programming, are highly effective for smooth, differentiable, and unimodal problems. [1] But most real-world optimization problems exhibit a complex and multimodal search landscape, discrete or mixed-integer decision variables, and non-differentiable objective functions. Due to these features, classical methods become either insufficient or computationally too expensive [2]. Metaheuristic algorithms inspired by nature have become an effective tool in addressing these problems. The concept of collective behavior shown by biological organisms such as birds in flocks, ants in their colony, swarming bees, and several other forms is the foundation of these algorithms, which make use of stochastic population-based search heuristics that strike a balance between exploration and exploitation of solutions in the search space. Some well-known algorithms of this type include the GA [3], PSO [4], Ant Colony Optimization (ACO) [5], and Dragonfly Algorithm (DA) [6].

One of the recent members of this family was proposed by Abdullah and Rashid in 2019 [7]. The FDO is based on the decision-making and swarming behavior of honeybees as they search for an ideal place for their nest, comparing options against various criteria and selecting the best one by communicating with one another. Unlike other bee-based algorithms, such as ABC [8], FDO does not originate from other approaches and has a unique mechanism of fitness-dependent velocity for search agents that depends on the fitness values of both current and previous global solutions. Since then, this algorithm has been further developed and applied to solve a variety of different problems. Nevertheless, there is no comprehensive systematic literature review on FDO yet. In existing surveys, FDO is mentioned along with other methods without considering its different versions or hybrids. This paper aims to bridge this research gap.

This survey differs from the previous FDO survey by incorporating four different contributions. First, this survey focuses completely on the FDO and its related variants, unlike previous review surveys that only mentioned FDO as part of comparisons. Secondly, this survey includes studies conducted between the years 2019 and 2026, and it captures not only the early developments of FDO but also the recent advances in hybrid and multi-objective FDOs. Thirdly, we classify the studies reviewed in this survey using a methodology based on the types of studies and modifications made. Lastly, this survey makes an extensive comparative analysis between FDO and some of the most common optimization methods like PSO, GA, and ABC.

The remainder of this paper is organized as follows. First, Section 2 outlines the review methodology. Next, Section 3 provides background on the original FDO algorithm. This is followed by Section 4, which examines FDO variants and enhancements, reviews application domains, presents findings from comparative analyses, and discusses implications and limitations. Finally, Section 5 concludes the survey.

2. Methodology

The review followed the PRISMA 2020 guidelines. The stages of the systematic review included database and search strategy, study identification, screening and selection, and classification. This objective was achieved through searching the databases, screening and reviewing titles and abstracts, evaluating papers' full texts, and finally choosing the studies for qualitative analysis.

2.1. Search Strategy

The research was conducted by an extensive search in six academic databases, namely IEEE Xplore, ScienceDirect, SpringerLink, arXiv, Google Scholar, and Research Square. The main search was performed from January 2026, while a follow-up search was performed in March 2026 to include the latest publications until February 2026. The search used a Boolean search strategy customized to the syntax of each database. The main Boolean string was:
("fitness dependent optimiser" OR "fitness dependent optimizer" OR "FDO") AND ("swarm" OR "metaheuristic" OR "bee" OR "variant" OR "hybrid" OR "application" OR "engineering" OR "healthcare" OR "power system" OR "machine learning" OR "neural network")

For comprehensive searching, both forward and backward citation analysis was conducted on the original FDO article as well as on the articles chosen for inclusion in the review. Furthermore, we searched through preprint servers for newer articles that might not be indexed in mainstream databases.

2.2. Eligibility Criteria

The following inclusion and exclusion criteria were used for study selection. All selected studies should have been published from January 2019 to February 2026 and be written in the English language. Among eligible studies there were peer-reviewed journal articles, conference proceedings, and established preprints. Furthermore, eligible papers should address the topic of FDO by either introducing a new FDO variant or hybrid, or by applying FDO to some optimization problem, or FDO should be used as a primary comparative baseline in such a paper. It is necessary that the selected papers contain some empirical findings, not just theoretical considerations, like benchmarking results or practical implementation of some application based on FDO. Excluded studies were papers that briefly mentioned FDO without any experimental study, duplicates, editorials, or data.

2.3. Screening and Selection

The literature search process was carried out using Covidence software. To avoid bias, two authors independently screened the title and abstract of all the papers. Disagreements were sorted out through consultation by a third author. The full texts of those papers that passed the preliminary screening were reviewed again to confirm that they met the eligibility criteria; Figure 1 presents the final selection process in the PRISMA flow diagram.

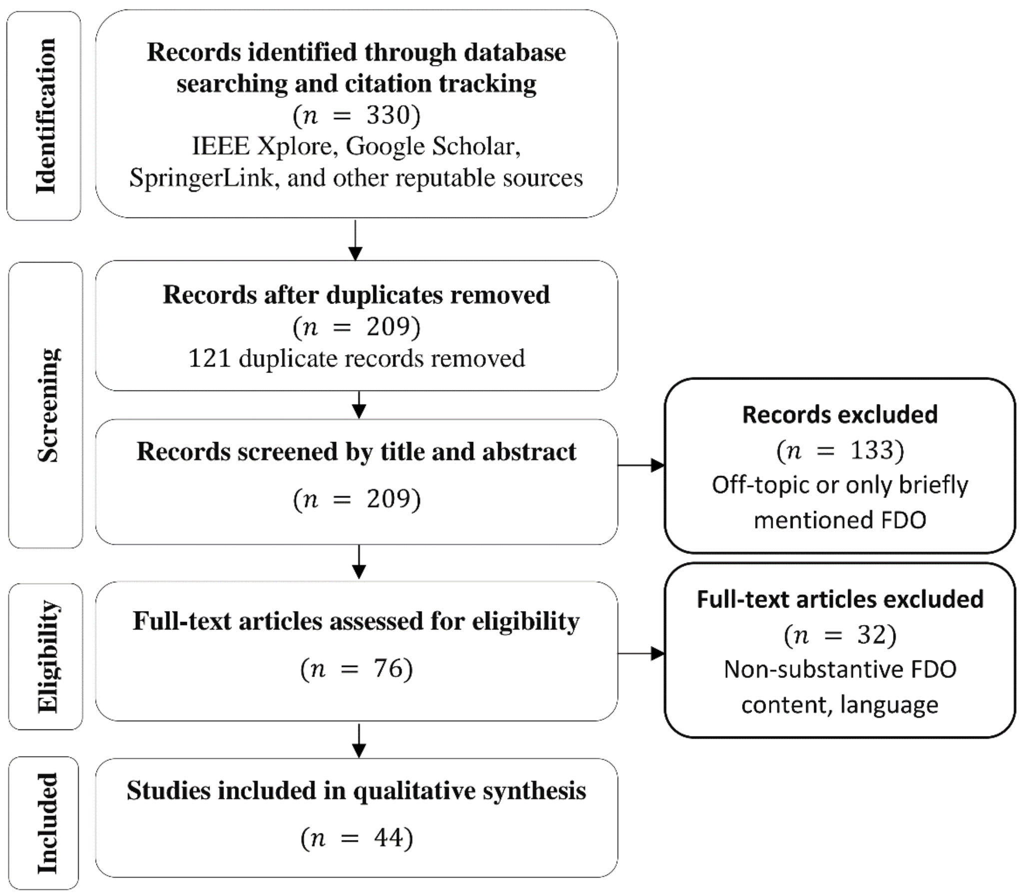

Figure 1: The systematic literature review process.

3. The Original Fitness Dependences Optimizer

The following section provides the initial definition of the algorithm along with an explanation of its theoretical background and mathematics.

3.1. Biological Inspiration

FDO is a bee-inspired algorithm based on the honeybee swarming behaviors of reproduction [7]. It refers to the queen bee leaving the hive with the help of workers and scouts to form new colonies; swarming is the way in which new colonies of bees are created inside a bee hive. When the density of the hive becomes too much and the external environment is favorable, the queen, along with the worker and scout bees, leaves the hive. At a distance of 20 to 30 meters, a temporary cluster is formed from which 20 to 50 scout bees will be sent out to locate the most suitable new nesting sites; only when such environmental and population limits are met does swarming take place. The growth of the colony is determined by how well the space is evaluated and the site is selected. FDO's main idea is built on the well-organized communication network of scout bees. Every bee evaluates prospective nesting places and communicates the direction through the waggle dance. In this case, the duration and frequency of the leg movements convey the quality of a site. When at least 20 scouts gather at one place, they continuously cross-verify distance and direction to reach an agreement. This process of accumulation is based on the evaluation of fitness; when 20 to 30 scouts verify that a site meets the defined criteria, they reach a stable decision. This fitness-dependent common decision-making process is the biological model for FDO's main mechanism.

The connection between biological organisms and FDO models is obvious and well-structured. A scout bee, for example, can be considered a search agent in the population of the algorithm; a candidate hive can be associated with a potential solution in the search space; the fitness parameters of a hive (volume, entrance size, and sunlight exposure) belong to the value of the objective function; and the decision made collectively by the scouts is equivalent to the fitness weight (fw) that controls the behavior of the algorithm's search. The comparison is shown in Table 1.

Table 1: Mapping of biological bee swarming entities to FDO algorithmic components.

Biological Entity	Algorithmic Counterpart
Scout Bee	Search Agent (artificial scout)
Hive / Nest Site	Candidate Solution
Hive Specification (size, entrance, light)	Fitness Function Value
Scout Collective Decision	Fitness Weight (fw)
Selected Hive	Optimum Solution
Bee Dance Communication	Fitness-weight guided velocity update

3.2. Mathematical Formulation

The FDO algorithm initializes a population of n artificial scout bees at random positions X_i $(x_1, x_2, x_3, \ldots, x_n)$ within the search space boundaries. At each iteration t, the position of each scout is updated by adding a computed pace (velocity) as shown in Equation 1:

$$X_{i.t+1} = X_{i.t} + pace \tag{1}$$

The $pace$ is primarily determined by the fw, which is defined in Equation 2 for minimization problems as follows:

$$fw = \left| \frac{X^*_{i.t\ fitness}}{X_{i.t\ fitness}} \right| - wf \tag{2}$$

Here, $X^*_{i.t\ fitness}$ represents the fitness of the global best solution identified up to iteration t, $X_{i.t\ fitness}$ is the fitness of the current search agent's solution, and wf is a binary weight factor (0 or 1) that controls the degree of convergence pressure. The $pace$ is then calculated based on three conditional rules outlined in Equations 3 to 5.

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

Here, r is a random number drawn from a Lévy flight distribution over $[-1, 1]$. Lévy flight is employed because it provides statistically stable movements characterized by occasional long jumps that aid global exploration while maintaining local exploitation. A key unique feature of FDO is that if a new candidate position does not yield an improved solution, the algorithm retains the previous $pace$ vector and attempts the move again in the same direction, a memory mechanism that can leverage successful search directions. Table 2 presents a summary of the algorithm parameters.

Table 2: Summarizes the notation, mathematical formulation, and parameters related to FDO.

Symbol	Meaning

n	Population size (number of scout bees).
d	Problem dimension.
t	Current iteration index.
T	Maximum number of iterations.
$X(i,t)$	Position vector of scout i at iteration t.
$X^*(t)$	Position vector of the global-best solution at iteration t.
$f(\cdot)$	Objective (fitness) function.
fw	Fitness weight.
wf	Binary weight factor $\in \{0, 1\}$.
$pace(i,t)$	Velocity vector applied to scout i at iteration t.
r	Random number $\in [-1, 1]$ drawn from a Lévy-flight distribution

3.3. Performance on Original Benchmark Tests

In the original publication, FDO was evaluated on 19 classical benchmark functions comprising unimodal (TF1–TF7), multimodal (TF8–TF13), and composite (TF14–TF19) test problems and was also tested on the IEEE CEC-C06 2019 competition benchmark suite. On classical benchmarks, FDO outperformed DA, PSO, and GA in 11 of 19 test functions, with comparative performance on the remaining cases. On the CEC-C06 suite, FDO outperformed DA, WOA, and SSA in 9 of 10 benchmark functions. Statistical validation using the Wilcoxon rank-sum test confirmed that FDO's results were statistically significant in the vast majority of comparisons. Besides, FDO was also capable of solving two real-life problems: designing an aperiodic antenna array, where it reached the optimal solution at element configuration {0.713, 1.595, 0.433, 0.130} at iteration 78; and optimizing the FM synthesizer parameters, finding almost the global optimum by iteration 64.

4. Systematic Analysis and Taxonomy of the FDO

This section reviews the studies on FDO and its variants. It is organized into six parts: publication trends, how the FDO variants are classified (the main modification methods and the areas where they are applied), how they are tested and evaluated, how they compare in performance, weaknesses and solutions, and the research gaps found.

4.1. Publication Distribution Over Time

A total of 44 studies published between 2020 and 2026 were analyzed, showing clear growth in research interest in FDO-based optimization. Figure 2 shows how the papers are spread over time, and Table 3 gives a year-by-year breakdown. The first studies appeared in 2020 (five papers), mainly applying FDO to automatic generation control (AGC) in power systems, improving its basic mechanics, and extending it to reactive power and discrete problems. Activity rose sharply in 2021 with 11 studies, the largest single-year jump in the dataset. This was driven by the growing use of FDO as a comparison benchmark and by new FDO-inspired methods such as the Ant Nesting Algorithm (ANA) and the SC-FDO variant. Publications peaked in 2023 with 13 studies covering the widest range of applications. The slight drop in 2024–2026 may suggest the field is maturing, with researchers refining existing methods rather than producing new work at the earlier pace.

Table 3: Shows how review articles were published over time, from 2020 to 2026.

Year	No. of Papers	Primary Application Domains
2020	5	Power systems, Reactive optimization, Discrete optimization
2021	11	Control, ML, Power, Engineering, Benchmark
2022	9	Power systems, Engineering, ML, IoT
2023	13	Broad cross-domain (widest coverage)
2024	3	Power distribution, Sensor networks
2025	2	Forex forecasting, General optimization
2026	1	Fog computing scheduling

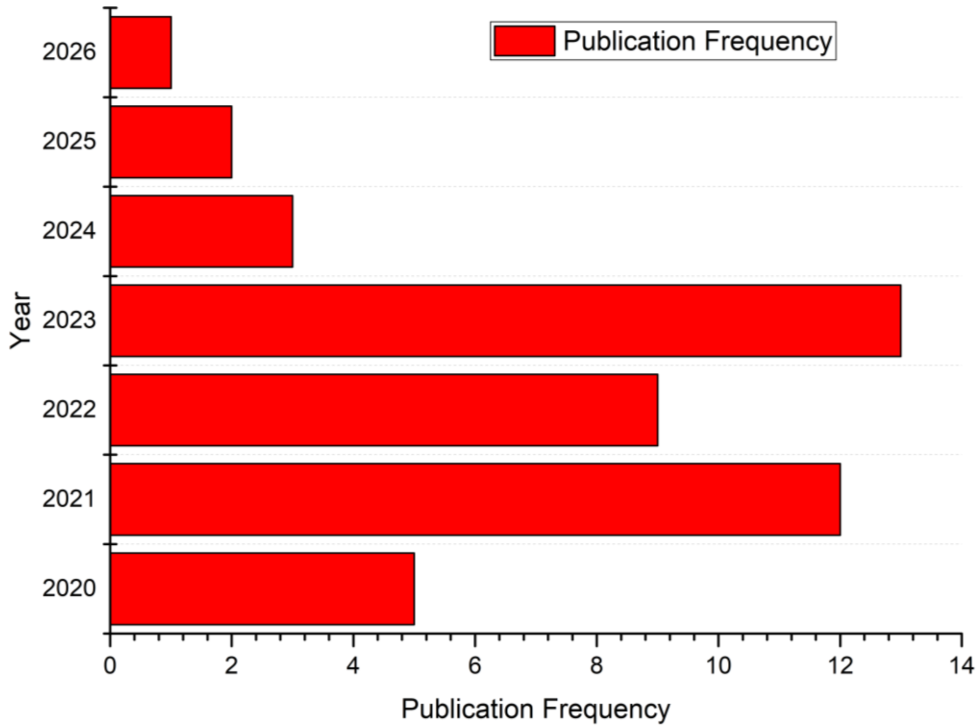

Figure 2: The number of related paper FDOs from 2020 to 2026.

4.2. Classification of FDO Related Studies

Based on the connections between the FDO algorithm, the types of structures involved, and the contributions made by the research, the 44 selected papers were divided into seven different categories. Table 4 clearly highlights the specific areas related to the FDO algorithm in these categories.

Table 4: Classification overview of FDO studies by research type.

Category	Description	Reference	Count	% of Total
FDO Core Variants & Improvements	Papers that directly modify or improve the FDO algorithm structure, operators, or parameters.	[9], [10], [11], [12], [13], [14], [15], [16]	8	18.2%
FDO + Power Systems & Control	FDO (or its variants) is applied to power system control, AGC, LFC, and DG placement.	[17], [18], [19], [20], [21]	5	11.4%
FDO + Machine Learning hybrid	FDO is used as a training/optimization algorithm for neural networks and deep learning models.	[22], [23], [24], [25], [26]	5	11.4%
FDO in Specialized Applications	FDO applied to specific engineering, combinatorial, IoT, and scientific domains.	[27], [28], [29], [30], [31], [32], [33]	7	15.9%
Multi-objective FDO	Extension of FDO to multi-objective optimization with Pareto-based mechanisms.	[34]	1	2.3%

Novel Metaheuristics (FDO as baseline)	New metaheuristic algorithms were proposed independently; FDO was used only as a comparison algorithm.	[35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49]	15	34%
Survey / Statistical / Other	Large-scale algorithm surveys, statistical evaluation frameworks, and standalone deep learning work.	[50], [51], [52]	3	6.8%
Total			44	100%

According to Table 4, FDO has proven to be an integral part of contemporary scientific research, serving as the primary algorithm for performance comparisons in approximately 34% of the studies examined. This popularity indicates significant scientific progress, allowing FDO to be classified as the universally acknowledged benchmark for metaheuristic algorithms. Additionally, nearly 18% of the studies focus on enhancing the mechanics of the algorithm, particularly concerning convergence speed and the quality of the solutions produced. Together, these areas of research constitute more than 50% of the total number of papers reviewed, reflecting a strong scientific interest in both the validation and improvement of the framework utilized. Beyond these foundational enhancements, FDO continues to demonstrate its growing potential as a versatile and efficient algorithm. Since 2021, the integration of FDO with machine and deep learning techniques has emerged as a compelling research topic, appearing in 11% of the papers. Furthermore, FDO has established itself as a valuable tool for addressing real-world challenges, especially in the fields of power systems and engineering tasks. Its applications also encompass optimization problems, resource allocation, and even the training of neural networks. We will discuss the specifics of these advancements in detail in the following subsections.

4.2.1. FDO Core Variants and Improvements

An analysis of 44 papers has revealed eight recurring modification strategies that show the adaptation and improvement of FDO. These strategies are important not only for finding the design choices of FDO variants but also for leading to the development of new algorithms. The fw parameter is the most significant and commonly modified, as it directly regulates the exploration-exploitation balance. In the original FDO, fw is fixed to either 0 or 1. Setting fw to 0 results in stable search at the cost of convergence speed, while setting fw to 1 speeds up convergence but significantly hampers exploration. IFDO [13] has dealt with the problem by randomly varying fw within the range of $[0, 1]$. MFDO [15] has further developed it by restricting the range to $[0, 0.2]$ and introducing a sinc cardinal function for smoother transitions and more precise solution refinement. M-IFDO [10] replaced the heavy alignment and cohesion terms with the Lambda parameter, which is simpler and has less overhead while still preserving quality. This series of changes indicates that fw modification is the most frequent change, and it has occurred in at least six distinct ways. As another main approach, CFDO [12] incorporates chaos theory into the FDO's position update method, primarily by increasing the diversity of the population during initialization and throughout the search. By carrying out a detailed test of ten chaotic maps used both for random variable generation and population seeding, it was discovered that the Singer map was the only one able to perform well on all CEC2019 benchmark functions. Therefore, the overall performance of the functions varied at different chaos locations, indicating that chaos could be used at one point to execute a specific function and at other points to execute different functions. Eight strategies for modification emerged across the 44 studies. Table 5 illustrates how each of the modifications addresses a failure mode of the original FDO.

Table 5: Core modification strategies applied to the original FDO.

Modification Strategy	Main Idea	Example Variants	Main Improvement
Fitness-dependent weight factor	Uses a continuous random weight factor instead of binary values.	IFDO, M-IFDO, SC-FDO, MFDO, CFDO, I-FDO	Provides smoother balance between exploration and exploitation and improves search performance.
Sine–cosine update mechanism	Updates movement using sine and cosine mathematical functions.	SC-FDO, MFDO, Sobol-FDO-SC	Significantly reduces runtime and improves convergence speed.

Chaotic-map initialization	Uses chaotic maps to initialize the population.	CFDO, MCYSGA, ZGBBO	Increases population diversity and avoids local optima.
Alignment and cohesion factors	Adds swarm-based social interaction similar to PSO.	IFDO, M-IFDO	Enhances cooperation among search agents and improves exploration.
Opposition-based learning (OBL)	Generates opposite candidate solutions during search.	SHO-OBL	Improves convergence quality and solution accuracy.
Sobol sequence initialization	Uses Sobol quasi-random sequences for population generation.	AFDO, EFDO	Produces better initial solutions and reduces optimization loss.
Lambda (λ) parameter control	Introduces a continuous control parameter in the pace equation.	MFDO, M-IFDO	Reduces execution time and improves optimization efficiency.
Discrete solution encoding	Represents solutions using integer-based encoding for discrete problems.	AFDO (bin packing)	Improves performance in combinatorial and discrete optimization tasks.

4.2.2. Application Domain Analysis

In this study, the reviewed studies are classified based on their main real-world application areas; overall, FDO and its variant types have been exploited to a significant extent in different application areas. Figure 3 presents the number of papers related to each area; first, engineering design and control is the largest domain among those listed in the figure, with a total of 11 papers. FDO-family algorithms in this area are being used to solve engineering optimization problems or as a tool for benchmarking other algorithms. These problems encompass structural design tasks, for example, welded beam, pressure vessel, and spring design; antenna array design; nonlinear optimal control issues; task assignment; and bin packing [10, 12, 13, 14, 15, 16, 27, 29, 32, 35, 37]. The Power Systems and Energy domain, with its 8 papers, is the second largest area in terms of practical applications, to which several problems such as load frequency control, automatic generation control, economic load dispatch, maximum power point tracking, reactive power optimization, and distributed generation placement are associated, as can be seen through publications [11, 17, 18, 19, 20, 21, 23, 45]. In the machine learning and prediction domain, five papers are using FDO approaches to handle problems such as fault detection, student performance prediction, gear diagnosis, COVID-related classification, and Forex forecasting, including papers [22, 23, 24, 25, 26]. Five papers make up the IoT, Networking, and Computing section with a focus on wireless sensor networks, underground network optimization, and fog computing scheduling as demonstrated by papers [28, 30, 33, 41, 44]. Likewise, the Healthcare and Medical domain with 2 papers deals with COVID-19 classification, smart healthcare systems, and pathological detection, which can be found in papers [25] and [28]. The other specialized applications section, consisting of 3 papers, picks up on very distinctive applications such as weather data imputation, neural style transfer, and formal verification, as shown in the papers [9, 31, 51].

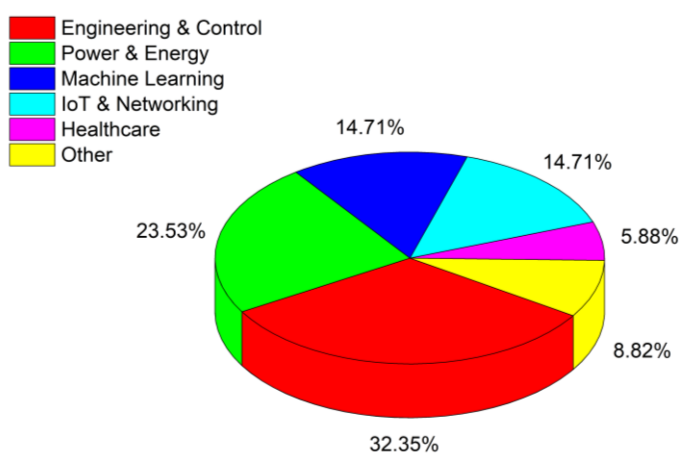

Figure 3: Distribution of FDO applications by sector.

4.3. Benchmarking and Evaluation Methods

The reviewed papers employ a wide range of evaluation approaches, drawing on established benchmark functions, statistical validation techniques, and performance metrics to gauge how well each algorithm performs, as shown in Figure 4. Benchmark function suites are the principal tool for testing algorithm effectiveness. The CEC-2019 suite is the most widely adopted, appearing in 23 of the reviewed papers, followed by classical benchmark functions, which served as the basis for comparison in 18. Among the more specialized options, the 100-Digit Challenge was featured in 6 papers and the CEC-2017 suite in 2, while a single paper drew on the CEC-2014 and CEC-2005 suites. These benchmarks present various mathematical problems that let researchers assess how well an algorithm balances exploration and exploitation. To confirm their findings statistically, the studies apply several formal tests. The Wilcoxon rank-sum test is the preferred choice, used in 8 papers, with the Friedman test selected in 3 and ANOVA in 2. Tests of this kind are valuable for determining whether the performance gains reported for a newly proposed algorithm are genuine rather than the product of random variation.

The metrics used to report success are closely tied to each study's application domain. Research in power systems generally relies on error-based measures such as ITSE, ISE, ITAE, and IAE, alongside transient-response indicators like overshoot and settling time. Machine learning studies assess performance through metrics such as accuracy and sensitivity. Work on engineering design problems centers on physical and economic outcomes, including cost, weight, stress, and constraint violations. For general benchmark optimization, the standard reporting metrics are the mean function value, standard deviation, best and worst results, and total computational runtime. Taken together, these benchmarks, statistical tests, and metrics form a comprehensive framework for evaluating new advances in the field.

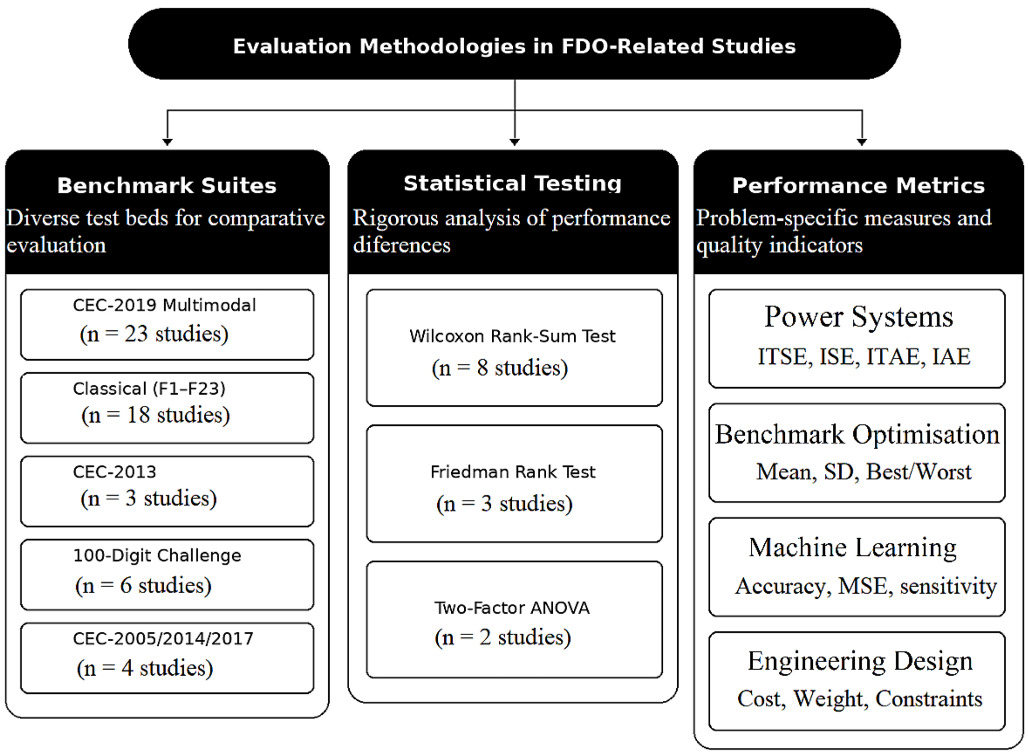

Figure 4: Overview of evaluation methodologies in FDO research.

4.4. Comparison of FDO with Other Well-Known Algorithms

In this section the study compares the FDO with the PSO, GA, and ABC. The original FDO only includes comparisons with the PSO and GA and does not address the ABC, even though both the FDO and the ABC are inspired by bee behavior. We close that gap here using independently reported ABC results. However, we will evaluate the FDO against the ABC based on algorithms that were benchmarked using two representative classical functions from the F1–F19 suite [53]. Specifically, this section explains how the four algorithms differ, how fast they converge, and which kinds of problems each one suits best.

Essentially, the four algorithms work the same way: they maintain a set of candidate solutions and enhance them gradually over iterations. What distinguishes them is how each candidate is changed. For example, FDO directs each search agent (a scout bee), to the best solution found so far; the length of each step automatically decreases as the agent gets closer to the best solution, so providing large steps early on for broad searching and small ones later for fine-tuning. It only has $pace$ parameter to adjust. PSO changes the position of each particle using a combination of its own personal best position and the global best position of the entire swarm, together with the momentum from its previous move; this momentum allows it to overshoot and move back, which helps it to escape local optima but results in some loss of precision, and it takes three parameters to be set. GA does not move solutions at all. Rather, in each generation, GA keeps the top solutions, combines pairs of them to generate new solutions through crossover, and randomly alters a few through mutation, so progressively producing a superior population. ABC uses three types of bees: worker bees explore around existing solutions, onlooker bees focus their attention on the most promising solutions, and scout bees give up any solution that has not been improved for a certain number of times and reinitialize it at a random position, which, in effect, is a built-in mechanism to escape from dead ends. Table 6 presents the result of classical benchmarking (F1-F19), which can be divided into two standard benchmarks. Sphere benchmarks is a simple and straightforward single-peak benchmark that measures tuning ability. However, the Rastrigin benchmarks are a complex and difficult multiple-peaks benchmark that measures the escaping ability.

Table 6: Comparative performance of the four algorithms (lower values indicate better performance).

Algorithm	How it searches	Sphere (easy problem)	Rastrigin (hard problem)	Best suited for
FDO	Steps toward the best; step shrinks near the target.	7.47×10^{-21}	14.57	Smooth, single-peak problems.
PSO	Blends personal best + swarm best, plus momentum.	4.20×10^{-18}	10.45	Problems where momentum helps cross ridges.
GA	Breeds new solutions by mixing and mutating.	7.49×10^{2}	25.52	Discrete / combinatorial problems.
ABC	Local search; abandons and restarts stuck solutions.	2.66×10^{-15}	3.98×10^{-1}	Many-peaked, tricky problems.

The numbers reveal a clear pattern. On the easy Sphere problem, FDO reaches the most accurate result, roughly a thousand times better than PSO and a million times better than ABC, while GA fails badly because mixing two similar solutions produces little improvement. But on the hard Rastrigin problem, the order flips: ABC wins by a wide margin because its scout bees keep escaping local traps, while FDO, which has no equivalent restart mechanism, finishes well behind. This reversal is a textbook example of the "No Free Lunch" principle: no single algorithm is best at everything. Figure 5 reveals the reason for these differences by tracking the movement of one agent from each algorithm toward the target across iterations.

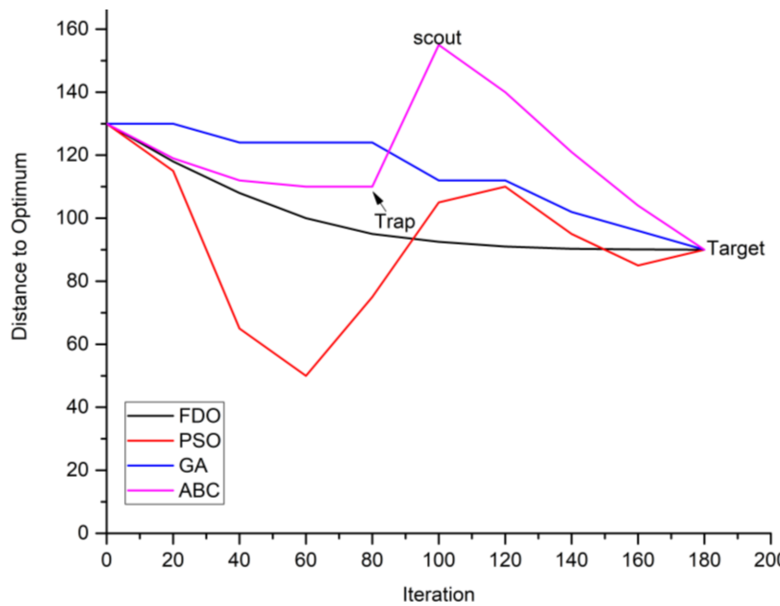

Figure 5: Convergence and search behaviors of optimization algorithms.

As shown in Figure 5, every algorithm follows a unique approach to obtain the optimal solution (target). FDO uses adaptive steps where the steps become smaller closer to the target, thus ensuring precise convergence when dealing with smooth problems. However, while the use of momentum by PSO allows it to navigate ridges, this technique can also lead to overshooting. On the other hand, GA relies on population diversity, making it better suited for discrete rather than smooth optimization problems. Finally, ABC's scout-driven reset mechanism makes it the most robust for escaping local optima in rugged landscapes. In general, each algorithm finds a balance between exploration and exploitation in a different way; FDO accomplishes this task using a single self-adjusting weight, while PSO uses personal and social attractors, GA employs genetic recombination, and ABC utilizes its abandonment rule.

All algorithms are trying to achieve a balance between searching the global search space (exploration) and focusing only on the best solution found so far (exploitation); each one has a different way of doing it: FDO through one self-adjusting step size, PSO through tunable pull toward personal and group bests, GA through mixing and mutation, and ABC through an explicit abandon-and-restart rule. FDO's main strengths are its simplicity, since it has only one parameter to tune, and its automatic, highly accurate convergence on smooth continuous problems, where it matches or beats PSO, GA, and ABC. Its main weakness is the lack of a built-in escape mechanism, which makes it less effective on extremely rugged, many-peaked landscapes where ABC's scout resets or PSO's momentum pays off. The best choice therefore depends on the problem: FDO for smooth continuous problems, PSO when momentum helps, GA for discrete problems, and ABC for highly multi-peaked problems.

4.5. Weaknesses and Solutions in FDO and Related Algorithms

A systematic review of the literature shows that FDO and its variants have several common weaknesses, and researchers have proposed different methods to address them. Over 60% of the studies attribute the main difficulty, premature convergence, as the cause. Various methods have been implemented to solve it, for example, Sobol sequence initialization [11], chaotic maps [12, 43], opposition-based learning in SHO-OBL [41], and lifespan control mechanisms in IGWO [45]. However, there is no single methodology that outperforms others in all scenarios. Thus, the "No Free Lunch" theorem states that all methods can be used successfully, depending on the problem's nature and complexity. The second major problem is the imbalance between exploration and exploitation, particularly in the original FDO that employs a simple binary weight factor. Solutions that have resulted in an improved state of affairs over the years revolve around the transition from a binary to a continuous and then to an adaptive weight factor. The sine-cosine method implemented in SC-FDO represents one effective way of resolving this problem of the trade-off and, at the same time, reducing computational complexity [10]. The third concern is the high computational expense, which has been lowered in a few versions [40, 29] by making certain components and parameters simpler, which in turn increases efficiency, particularly in the context of adaptive weight factors that enhance performance in control systems. Similarly, the limitation of a fixed or binary weight factor (0, 1) has been addressed by introducing random or adaptive strategies, which provide greater flexibility and responsiveness to varying conditions in control systems [13, 14].

There are also some papers showing parameter sensitivity, especially in control systems. Analysis of sensitivity plus selection of objective functions are two wide ways to manage this sort of problem. An issue that is hardly the focus of attention is that FDO is not inherently appropriate for discrete optimization problems. Only one variant has been developed [32] to address this issue by changing the algorithm to work with integer-based solutions, specifically by incorporating discrete optimization problem constraint handling techniques. This remains an open research area despite the importance of discrete problems in fields like scheduling and logistics. In general, these findings show that FDO research is gradually improving, with well-developed solutions for common problems. However, discrete optimization remains the most important area for future work, particularly in developing more efficient algorithms and addressing complex real-world applications that remain underexplored, such as those in supply chain management and resource allocation [54].

4.6. Research Gaps and Future Direction

After reviewing the 44 studies on FDO-based optimization, it is clear that the FDO suffers from a number of major research gaps that currently define the primary directions for future research in FDO-based optimization. The most significant problem is the lack of examination of high-dimensional problems. Although few papers deal with the dimension of 100 to 300, the majority of reviewed studies restrict themselves to 10 or 30 variables. Therefore, FDO's efficiency for large-scale problems, for example, those with more than 100 variables that are very common in modern engineering and data science, is still an open question; hence, future research may be directed towards increasing the scalability of solutions through features such as dimensionality reduction or cooperative strategies.

Additionally, there is a significant gap in the lack of proof for theoretical convergence; for instance, unlike other algorithms such as PSO, FDO still lacks a formal mathematical proof of convergence [55]. Resolving this problem will strengthen the algorithm's theoretical basis and improve parameter settings. Moreover, most papers just consider the case of a static optimization problem, where the objective function remains the same throughout. However, real-world problems often differ from these assumptions, such as the need for constant algorithm adaptation or reflection [56], which can lead to dynamic changes in the target function based on varying conditions and requirements. Therefore, this direction for FDO is unexplored, and practical implementations on real hardware are very limited as well.

One more serious drawback is that FDO barely ascends to a reasonable level in various discrete optimization problems where a limited set of options serves as a source of choices for the solutions; for example, scheduling, task allocation, or bin assignment problems require integer or categorical decisions, but to date, only one study [32] has modified FDO to these problems. Hence, discrete optimization becomes a crucial field for future research. Furthermore, there is a lack of thorough use of ablation studies, which are indispensable to identify the true impact of different FDO's components, especially the fitness-dependent weight factor. Eventually, these shortcomings present an unambiguous outline for future research, such as enhancing scalability, conducting theoretical studies, moving FDO into discrete and dynamic problems, and normalizing evaluation procedures.

5. Conclusion

In this review study, we conducted a systematic review of the FDO algorithm; in the first step, we gathered 330 studies on the FDO algorithm from various academic databases such as IEEE, Springer, etc., from 2019 to 2026. However, after revision and filtration based on questions such as whether this study is about an FDO variant or an FDO application, 44 papers remain to complete the systematic review process. The objective of our review is to investigate the development, applications, strengths, and weaknesses of the FDO metaheuristic algorithm. We found that researchers have extensively studied the well-developed FDO algorithm. First of all, the FDO algorithm has improved enormously over time. For example, the original version based on honeybee behavior provided a simple and efficient fitness-based updating mechanism; despite this strength, some limitations of the early versions, such as the tendency toward premature convergence and the binary weight factor, led researchers to develop improved versions. These include adaptive strategies, chaotic methods, opposition-based learning, and sine–cosine techniques. These developments showcase the flexibility of FDO and its potential for improvement without altering its core concept. In terms of applications, FDO has been successfully used in many fields, including engineering design, power systems, machine learning, IoT, and healthcare. Examples include load frequency control, antenna design, disease classification, and task scheduling; it is also increasingly used as a comparison benchmark in many studies, which shows its growing importance and acceptance in the research community. Despite these advances, several important challenges remain. Most studies focus on small-scale problems, and FDO has not been properly tested for high-dimensional cases. In addition, there is no formal mathematical proof of convergence for FDO. Most applications also assume that the environment is static, but in the real world, problems are often dynamic. This can lead to less-than-ideal solutions when using FDO in these situations, particularly because the algorithm may not adapt effectively to changing conditions or account for the variability in dynamic environments. Furthermore, FDO is rarely applied to discrete problems, such as scheduling or routing, where solutions must be chosen from fixed options. In conclusion, FDO has developed from a new bio-inspired idea into a strong and widely used optimization algorithm. However, future research should focus on improving its theoretical foundation, scalability, and ability to handle discrete and dynamic problems. This survey provides a clear overview and direction for future work in this area.

Author contribution: All authors have contributed, read, and agreed to the published version of the manuscript results.

Conflict of interest: The authors declare no conflict of interest.

References

[1] Sadrehaghighi, I. (2022). Gradient (Derivative) Based Shape Optimization with Case Studies. CFD Open Series, 2(55). https://doi.org/10.1016/J.ASOC.2017.09.030

[2] Cuesta Ramirez, J., Le Riche, R., Roustant, O., & others. (2022). A comparison of mixed-variables Bayesian optimization approaches. Advances in Modeling and Simulation in Engineering Sciences, 9(6). https://doi.org/10.1186/s40323-022-00218-8

[3] Bettinger, P. (2025). Genetic algorithm search. In Forest harvest scheduling (pp. 217–235). Springer. https://doi.org/10.1007/978-3-031-89432-9_11

[4] Nayak, J., Swapnarekha, H., Naik, B., & others. (2023). 25 years of particle swarm optimization: Flourishing voyage of two decades. Archives of Computational Methods in Engineering, 30, 1663–1725. https://doi.org/10.1007/s11831-022-09849-x

[5] Awadallah, M. A., Makhadmeh, S. N., Al-Betar, M. A., & others. (2025). Multi-objective ant colony optimization: Review. Archives of Computational Methods in Engineering, 32, 995–1037. https://doi.org/10.1007/s11831-024-10178-4

[6] Tomar, V., Bansal, M., & Singh, P. (2023). Metaheuristic Algorithms for Optimization: A Brief Review. Engineering Proceedings, 59(1), 238. https://doi.org/10.3390/engproc2023059238

[7] Abdullah, J. M., & Ahmed, T. (2019). Fitness dependent optimizer: Inspired by the bee swarming reproductive process. IEEE Access, 7, 43473–43486. https://doi.org/10.1109/ACCESS.2019.2907012

[8] Kaya, E., Gorkemli, B., Akay, B., & Karaboga, D. (2022). A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems. Engineering Applications of Artificial Intelligence, 115, 105311. https://doi.org/10.1016/j.engappai.2022.105311

[9] Chiu, P. C., Selamat, A., Krejcar, O., & Kuok, K. K. (2021). Hybrid sine cosine and fitness dependent optimizer for global optimization. IEEE Access, 9, 128601–128622. https://doi.org/10.1109/ACCESS.2021.3111033

[10] Hamarashid, H. K., Hassan, B. A., & Rashid, T. A. (2024). Modified-improved fitness dependent optimizer for complex and engineering problems. Knowledge-Based Systems, 300, 112098. https://doi.org/10.1016/j.knosys.2024.112098

[11] Tahir, B. H., Rashid, T. A., Rauf, H. T., Bacanin, N., Chhabra, A., Vimal, S., & Yaseen, Z. M. (2022). Improved fitness-dependent optimizer for solving economic load dispatch problem. Computational Intelligence and Neuroscience, 2022, 7055910. https://doi.org/10.1155/2022/7055910

[12] Mohammed, H. M., & Rashid, T. A. (2021). Chaotic fitness-dependent optimizer for planning and engineering design. Soft Computing, 25, 14281–14295. https://doi.org/10.1007/s00500-021-06135-z

[13] Muhammed, D. A., Saeed, S. A. M., & Rashid, T. A. (2020). Improved fitness-dependent optimizer algorithm. IEEE Access, 8, 19074–19088. https://doi.org/10.1109/ACCESS.2020.2968064

[14] Faraj, A. K., Aladdin, A. M., & Ameen, A. A. (2025). EESB-FDO: Enhancing the fitness-dependent optimizer through a modified boundary handling mechanism. Frontiers in Applied Mathematics and Statistics, 11, 1640044. https://doi.org/10.3389/fams.2025.1640044

[15] Salih, J. F., Mohammed, H. M., & Abdul, Z. K. (2022). Modified fitness dependent optimizer for solving numerical optimization functions. IEEE Access, 10, 83916–83930. https://doi.org/10.1109/ACCESS.2022.3197290

[16] Hamarashid, H. K., Hassan, B. A., & Rashid, T. A. (2024). Modified-improved fitness dependent optimizer for complex and engineering problems. Knowledge-Based Systems, 300, 112098. https://doi.org/10.1016/j.knosys.2024.112098

[17] Daraz, A., Malik, S. A., Azar, A. T., Aslam, S., Alkhalifah, T., & Alturise, F. (2022). Optimized fractional order integral-tilt derivative controller for frequency regulation of interconnected diverse renewable energy resources. IEEE Access, 10, 43514–43527. https://doi.org/10.1109/ACCESS.2022.3167811

[18] Daraz, A., Malik, S. A., Mokhlis, H., Haq, I. U., Laghari, G. F., & Mansor, N. N. (2020). Fitness dependent optimizer-based automatic generation control of multi-source interconnected power system with non-linearities. IEEE Access, 8, 100989–101003. https://doi.org/10.1109/ACCESS.2020.2998127

[19] Daraz, A., Malik, S. A., Mokhlis, H., Haq, I. U., Zafar, F., & Mansor, N. N. (2020). Improved fitness-dependent optimizer based FOI-PD controller for automatic generation control of multi-source interconnected power system in deregulated environment. IEEE Access, 8, 197757–197775. https://doi.org/10.1109/ACCESS.2020.3033983

[20] Riaz, M. U., Malik, S. A., Daraz, A., Alrajhi, H., Alahmadi, A. N. M., & Afzal, A. R. (2024). Advanced Energy Management in a Sustainable Integrated Hybrid Power Network Using a Computational Intelligence Control Strategy. Energies, 17(20), 5040. https://doi.org/10.3390/en17205040

[21] Daraz, A., Malik, S. A., Waseem, A., Azar, A. T., Haq, I. U., Ullah, Z., & Aslam, S. (2021). Automatic Generation Control of Multi-Source Interconnected Power System Using FOI-TD Controller. Energies, 14(18), 5867. https://doi.org/10.3390/en14185867

[22] Farahmand-Tabar, S., & Rashid, T. A. (2024). Steel plate fault detection using the fitness-dependent optimizer and neural networks. In A. J. Kulkarni & A. H. Gandomi (Eds.), Handbook of formal optimization. Springer. https://doi.org/10.1007/978-981-97-3820-5_41

[23] Mirza, A. F., Haider, S. K., Ahmed, A., Rehman, A. U., Shafiq, M., Bajaj, M., Zawbaa, H. M., Szczepankowski, P., & Kamel, S. (2022). Generalized regression neural network and fitness dependent optimization: Application to energy harvesting of centralized TEG systems. Energy Reports, 8, 6332–6346. https://doi.org/10.1016/j.egyr.2022.05.003

[24] Abbas, D., Rashid, T. A., Abdalla, K., Bacanin, N., & Alsadoon, A. (2021). Using fitness dependent optimizer for training multi-layer perceptron. Journal of Internet Technology, 22(7), 1575–1585. https://jit.ndhu.edu.tw/article/view/2628/2648

[25] Abdulkhaleq, M. T., Rashid, T. A., Hassan, B. A., Alsadoon, A., Bacanin, N., Chhabra, A., & Vimal, S. (2023). Fitness dependent optimizer with neural networks for COVID-19 patients. Computer Methods and Programs in Biomedicine Update, 3, 100090. https://doi.org/10.1016/j.cmpbup.2022.100090

[26] Dash, S., Sahu, P. K., & Mishra, D. (2025). Enhancing Forex market trend analysis with bidirectional long short-term memory and transformer attention network. Journal of the Korean Statistical Society, 54, 825–857. https://doi.org/10.1007/s42952-025-00322-6

[27] Laghari, G. F., Malik, S. A., Daraz, A., Ullah, A., Alkhalifah, T., & Aslam, S. (2022). A numerical approach for solving nonlinear optimal control problems using the hybrid scheme of fitness dependent optimizer and Bernstein polynomials. IEEE Access, 10, 50298–50313. https://doi.org/10.1109/ACCESS.2022.3173285

[28] Aladdin, A. M., Abdullah, J. M., Salih, K. O. M., Rashid, T. A., Sagban, R., Alsadoon, A., Bacanin, N., Chhabra, A., Vimal, S., & Banerjee, I. (2022). Fitness dependent optimizer for IoT healthcare using adapted parameters: A case study implementation. arXiv. https://arxiv.org/abs/2207.04846

[29] Laghari, G. F., Malik, S. A., Khan, I. A., Daraz, A., AlQahtani, S. A., & Ullah, H. (2023). Fitness dependent optimizer based computational technique for solving optimal control problems of nonlinear dynamical systems. IEEE Access, 11, 38485–38501. https://doi.org/10.1109/ACCESS.2023.3267434

[30] Nawaz, M.Z., Hasan, O., Nawaz, M.S., Fournier-Viger, P., & Sun, M. (2020). Proof searching in HOL4 with genetic algorithm. Proceedings of the 35th Annual ACM Symposium on Applied Computing.

[31] Abdul, Z. K. (2012). Gear Fault Detection Based on Time and Non-Time Series Feature Representation Using Machine Learning (Doctoral dissertation, Polytechnic University).

[32] Abdul-Minaam, D. S., Al-Mutairi, W. M. E. S., Awad, M. A., & El-Ashmawi, W. H. (2020). An adaptive fitness-dependent optimizer for the one-dimensional bin packing problem. IEEE Access, 8, 97959–97974. https://doi.org/10.1109/ACCESS.2020.2985752

[33] Dehshid, A., Khorsand, R., & Mohebbi, K. (2026). A volunteer-supported fog computing environment for DVFS based workflow scheduling. Future Generation Computer Systems, 178, 108301. https://doi.org/10.1016/j.future.2025.108301

[34] Abdullah, J. M., Rashid, T. A., Maaroof, B. B., et al. (2023). Multi-objective fitness-dependent optimizer algorithm. Neural Computing and Applications, 35, 11969–11987. https://doi.org/10.1007/s00521-023-08332-3

[35] Hama Rashid, D. N., Rashid, T. A., & Mirjalili, S. (2021). ANA: Ant Nesting Algorithm for Optimizing Real-World Problems. Mathematics, 9(23), 3111. https://doi.org/10.3390/math9233111

[36] Villuendas-Rey, Y., Velázquez-Rodríguez, J. L., Alanis-Tamez, M. D., Moreno-Ibarra, M.-A., & Yáñez-Márquez, C. (2021). Mexican Axolotl Optimization: A Novel Bioinspired Heuristic. Mathematics, 9(7), 781. https://doi.org/10.3390/math9070781

[37] Hamad, R. K., & Rashid, T. A. (2024). GOOSE algorithm: A powerful optimization tool for real-world engineering challenges and beyond. Evolving Systems, 15, 1249–1274. https://doi.org/10.1007/s12530-023-09553-6

[38] Aladdin, A. M., & Rashid, T. A. (2025). LEO: Lagrange elementary optimization. Neural Computing and Applications, 37, 14365–14397. https://doi.org/10.1007/s00521-025-11225-2

[39] Ahmed, M., Sulaiman, M. H., & Mohamad, A. J. (2023). Improved barnacle mating optimizer-based least square support vector machine to predict COVID-19 confirmed cases with total vaccination. Cybernetics and Information Technologies, 23(1), 125–140. https://doi.org/10.2478/cait-2023-0007

[40] Wu, D., Li, T., & Wan, Q. (2021). A hybrid deep kernel incremental extreme learning machine based on improved coyote and beetle swarm optimization methods. Complex & Intelligent Systems, 7, 3015–3032. https://doi.org/10.1007/s40747-021-00486-8

[41] Houssein, E. H., Saad, M. R., Çelik, E., et al. (2024). An enhanced sea-horse optimizer for solving global problems and cluster head selection in wireless sensor networks. Cluster Computing, 27, 7775–7802. https://doi.org/10.1007/s10586-024-04368-9

[42] Zhang, L., Ren, Z., Liu, T., & Tang, J. (2022). Improved artificial bee colony algorithm based on Harris hawks optimization. Journal of Internet Technology, 23(2), 379–389. https://jit.ndhu.edu.tw/article/view/2675

[43] Kashyap, D., Singh, B., & Kaur, M. (2021). Chaotic approach for improving global optimization in yellow saddle goatfish. Engineering Reports, 3(9), e12381. https://doi.org/10.1002/eng2.12381

[44] Singh, P., Singh, R. P., Singh, Y., Shafi, J., & Ijaz, M. F. (2021). An Enhanced Naked Mole Rat Algorithm for Optimal Cross-Layer Solution for Wireless Underground Sensor Networks. Mathematics, 9(22), 2942. https://doi.org/10.3390/math9222942

[45] Singh, P., & Mittal, N. (2021). Optimized localization of sensor nodes in 3D WSNs using modified learning enthusiasm-based teaching learning based optimization algorithm. IET Communications, 15(9), 1223–1239. https://doi.org/10.1049/cmu2.12155

[46] Chaudhary, K. C. (2023). A modified version of the ABC algorithm and evaluation of its performance. Heliyon, 9(5), e16086. https://doi.org/10.1016/j.heliyon.2023.e16086

[47] de Freitas, J. G., & Yamanaka, K. (2021). An accelerated and robust algorithm for ant colony optimization in continuous functions. Journal of the Brazilian Computer Society, 27, 16. https://doi.org/10.1186/s13173-021-00116-8

[48] Tang, R., Yang, J., Fong, S., Wong, R., Vasilakos, A. V., & Chen, Y. (2021). Dynamic group optimization algorithm with a mean–variance search framework. Expert Systems with Applications, 183, 115434. https://doi.org/10.1016/j.eswa.2021.115434

[49] Zhang, Z., Gao, Y., Li, J., et al. (2021). An improved biogeography-based optimization with hybrid migration and feedback differential evolution and its performance analysis. Research Square. https://doi.org/10.21203/rs.3.rs-1064223/v1

[50] Amin, A. A. H., Aladdin, A. M., Hasan, D. O., Mohammed-Taha, S. R., & Rashid, T. A. (2023). Enhancing Algorithm Selection through Comprehensive Performance Evaluation: Statistical Analysis of Stochastic Algorithms. Computation, 11(11), 231. https://doi.org/10.3390/computation11110231

[51] Sayed, H.I. (2022). Designing New Tools for Curing Cardiovascular Diseases to Avoid Angioplasty, Coronary Bypass and Open Heart Surgery: Simulation and Modeling. In: Gupta, G., Wang, L., Yadav, A., Rana, P., Wang, Z. (eds) Proceedings of Academia-Industry Consortium for Data Science. Advances in Intelligent Systems and Computing, vol 1411. Springer, Singapore. https://doi.org/10.1007/978-981-16-6887-6_15

[52] Si-Ma, S., Liu, H.-M., Zhan, H.-X., et al. (2023). Swarm intelligence algorithms evaluation. Research Square. https://doi.org/10.21203/rs.3.rs-3240432/v1

[53] Chaudhary, K. C. (2023). A modified version of the ABC algorithm and evaluation of its performance. Heliyon, 9(5), e16086. https://doi.org/10.1016/j.heliyon.2023.e16086

[54] Loucks, D. P. (2022). Discrete optimization modeling. In Public systems modeling (International Series in Operations Research & Management Science, Vol. 318). Springer. https://doi.org/10.1007/978-3-030-93986-1_7

[55] Wang, X., Chen, H., Heidari, A. A., Zhang, X., Xu, J., Xu, Y., & Huang, H. (2020). Multi-population following behavior-driven fruit fly optimization: A Markov chain convergence proof and comprehensive analysis. Knowledge-Based Systems, 210, 106437. https://doi.org/10.1016/j.knosys.2020.106437

[56] Awlla, A. H., Rashid, T. A., & Abdullah, R. M. (2026). A dynamic-weight multi-objective sloth-inspired clustering algorithm for wireless sensor networks. International Journal of Communication Systems, 39(4), e70395. https://doi.org/10.1002/dac.70395