Journal of Electronics, Computer Networking and Applied Mathematics
Vol: 05, No. 02, July-Dec 2025
ISSN: 2799-1156, DOI: 10.55529/jecnam.52.16.35

Research Paper

Benchmarking metaheuristic algorithms: a comprehensive review of test functions, real-world problems, and evaluation metrics

Saman M. Almufti[1*], Renas Rajab Asaad[2], Awaz Ahmed Shaban[3], Ridwan Boya Marqas[4]
[1*,2,3]Information Technology Department, Technical College of Informatics-Akre, Akre University for Applied Sciences, Duhok, Iraq.
[4]Department of Computer Science, College of Science, Knowledge University, Erbil.

Article Info

Article History:
Received: 11 July 2025
Revised: 20 September 2025
Accepted: 27 September 2025
Published: 13 November 2025

Keywords:
Metaheuristics
Benchmarking
Test Functions
CEC
Performance Evaluation

ABSTRACT

The proliferation of metaheuristic algorithms for solving complex optimization problems has necessitated robust and standardized benchmarking practices. A fair and comprehensive evaluation is crucial for validating an algorithm's performance, understanding its strengths and weaknesses, and guiding future algorithmic development. This paper provides a comprehensive review of the landscape of benchmarking in metaheuristic optimization. We systematically categorize and detail the primary types of benchmark problems, including the classic set of 23 mathematical functions, real-world engineering problems, specialized CEC benchmark suites, combinatorial optimization benchmarks, and multi-objective problems. For each category, we present the fundamental mathematical formulations and discuss their specific characteristics and challenges. Furthermore, we outline the standard evaluation metrics used to quantify algorithmic performance. Finally, we discuss current challenges in benchmarking, such as the no free lunch theorem, the issue of overfitting to test suites, and the need for more diverse and real-world benchmarks. This review serves as a foundational guide for researchers and practitioners in selecting appropriate benchmarks and conducting rigorous, reproducible evaluations of metaheuristic algorithms.

Corresponding Author:
Saman M. Almufti
Information Technology Department, Technical College of Informatics-Akre, Akre University for Applied Sciences, Duhok, Iraq.
Email: Saman.Almofty@gmail.com

Journal of Electronics, Computer Networking and Applied Mathematics

1. INTRODUCTION

Optimization stands as a prime process across the domains of study, engineering, and industry, focusing on the selection of the best option out of a bunch of viable alternatives [1]. In case of complicated issues where traditional gradient-based or exact methods, due to non-linearity, high dimensionality, multimodality, or non-differentiability, fail to provide a solution, metaheuristic algorithms have been recognized as powerful and fruitful solution strategies [2], [3], [4]. They are the algorithms like Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) that have shown great success guided by natural phenomena, physical laws, or even swarm intelligence [5].

On the other hand, the fast-paced growth of the new and hybrid metaheuristics has made it imperative to have thorough and uniform benchmarking as one of the main methods. It is a scientifically untenable position to claim the superiority of one algorithm over the other without a fair and comprehensive comparison against the established peers on a diverse set of problems. The objective framework that the Benchmarking provides can be used to [6], [7].

- Validate Performance: Show an algorithm's effectiveness and efficiency.
- Identify Niches: Determine the specific types of problems an algorithm works best for (e.g., exploitative vs. explorative).
- Drive Innovation: Indicate the weaknesses of the existing methods, hence leading to the development of more robust algorithms.
- Ensure Reproducibility: Make it possible to directly compare the findings of various studies.

The goal of this paper is to present a well-organized overview of the benchmark setting in material heuristics research. We classify the enormous variety of test problems into logical categories, elucidating their mathematical foundations and roles in the assessment process. We also deliberate on the criteria employed for determining success and wrap up with reflections on the current difficulties and future paths of benchmarking practices.

2. RELATED WORK

The progress of benchmarking in optimization has always been closely related to the progress of algorithms. The first evolutionary computation mainly worked with elementary test functions such as Sphere and Rosenbrock to show the basic convergence properties [8], [9]. The initial research of De Jong (1975) defined a classic set of test functions. The quality of the benchmarks kept pace with that of the algorithms, and soon the highly multimodal functions like Rastrigin and Schwefel were introduced to check the escape from local optima [10].

The foundation of the IEEE Congress on Evolutionary Computation (CEC) benchmark competitions marked an important turning point in the history of benchmarking. The first notable event was CEC 2005 [11], [12]. The organizing committee provided a selection of functions that were standardized, scalable, rotated, hybrid, and composed, thus moving from the very simple separable problems to the more realistic, difficult landscapes. This was indeed an important step in eliminating the biased reporting and enabling the direct comparisons of the algorithms. The set of 23 functions as described in the papers by Yao et al. (1999) and Suganthan et al. (2005) is still considered a fundamental benchmark for every new algorithm [13], [14].

At the same time that mathematical benchmarks were being developed, the area of optimization has always been primarily concerned with the actual application on real-world problems [15]. There have been numerous studies that compared different metaheuristics on engineering design problems such as Pressure Vessel Design and Welded Beam Design, scheduling, and logistics, which validated the practical usefulness of the algorithms. For combinatorial problems, the existence of standard instance libraries such as TSPLIB and QAPLIB has been a similar thing [16], [17].

Recent surveys conducted by Eiben and Smit (2011) and Ser et al. (2019) have highlighted one of the major trends in research the big-scale optimization which was the focus of the CEC 2013 and 2017 conferences, alongside the dynamic and uncertain environments and the computationally intensive problems. Literature has always pointed out that using one benchmark is not enough hence the need for a portfolio of various problems for fair algorithmic evaluation [18].

3. METAHEURISTIC ALGORITHMS

Metaheuristics are the top-level algorithmic frameworks which are problem-independent and allow the efficient exploring of the search space. They trade off the guarantees of finding the global optimum for the option of finding satisfactory solutions in a reasonable time for complex problems [19], [20]. According to the broad classification, they can be differentiated into:

- Trajectory-Based (Single-Solution): These algorithms are working on one candidate solution, making local moves iteratively for its improvement (e.g., Simulated Annealing, Tabu Search) [21].
- Population-Based: The population-based algorithms take a set of solutions which are maintained and improved, the collective intelligence being utilized (e.g., Genetic Algorithms, Particle Swarm Optimization, Differential Evolution) [22].
- Nature-Inspired vs. Non-Nature-Inspired: The classification based on the source of inspiration is the common taxonomy, for example, evolution (GA), swarm behavior (PSO, ACO), physical processes (SA), or human-related concepts (Teaching-Learning-Based Optimization) [23].

The algorithm performance is closely related to the parameters tuning and the exploration-exploitation balance that the algorithm has in its working. The benchmarks in the next section are laying down the capabilities of the algorithms for evaluation purposely to stress-test them.

4. EVALUATION FUNCTIONS AND BENCHMARK PROBLEMS

A comprehensive evaluation requires a diverse set of benchmark problems. This section categorizes the most prominent types.

4.1. Classical Benchmark Functions (F1-F23)

Table 1. Comprehensive List of Classical Benchmark Functions (F1-F23)

#	Function Name	Mathematical Equation	Search Range	Global Optimum	Characteristics
First Category Unimodal Functions (Testing Exploitation)					
F1	Sphere	$f(\mathrm{x}) = \sum_{i=1}^{n} x_i^2$	$[-100,100]^n$	0	Simple, symmetric, convex.
F2	Schwefel 2.22	$f(\mathrm{x}) = \sum_{i=1}^{n} \Vert x_i \Vert + \prod_{i=1}^{n} \Vert x_i \Vert$	$[-10,10]^n$	0	Unimodal, non-separable.
F3	Schwefel 1.2	$f(\mathrm{x}) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	$[-100,100]^n$	0	Unimodal, non-separable.
F4	Schwefel 2.21	$f(\mathrm{x}) = \max_i\{\Vert x_i \Vert, 1 \leq i \leq n\}$	$[-100,100]^n$	0	Unimodal, non-separable.
F5	Rosenbrock	$f(\mathrm{x}) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30,30]^n$	0	Non-convex, valley-shaped, hard to converge.
Second Category Multimodal Functions (Testing Exploration)					
F6	Step	$f(\mathrm{x}) = \sum_{i=1}^{n} (\lfloor x_i + 0.5 \rfloor)^2$	$[-100,100]^n$	0	Discontinuous, plate-shaped.

F7	Quartic w/ Noise	$f(\mathrm{x}) = \sum_{i=1}^{n} ix_i^4 + \mathrm{random}[0,1)$	$[-1.28,1.28]^n$	0	Unimodal with noise.
F8	Schwefel	$f(\mathrm{x}) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lVert x_i \rVert})$	$[-500,500]^n$	-418.9829n	Multimodal, deceptive, many local optima.
F9	Rastrigin	$f(\mathrm{x}) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12,5.12]^n$	0	Highly multimodal, sinusoidal, separable.
F10	Ackley	$f(\mathrm{x}) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e$	$[-32,32]^n$	0	Complex multimodal with exponential and cosine terms.
F11	Griewank	$f(\mathrm{x}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600,600]^n$	0	Multimodal, but local optima are regularly distributed.
F12	Penalized 1	$f(\mathrm{x}) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10,100,4)$ where $y_i = 1 + \frac{1}{4}(x_i + 1), u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$[-50,50]^n$	0	Multimodal with penalty terms.
F13	Penalized 2	$f(\mathrm{x}) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n-1}(x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5,100,4)$	$[-50,50]^n$	0	Multimodal with penalty terms.
Third Category Low-Dimensional Multimodal Functions (Few local optima)					
F14	Foxholes	$f(\mathrm{x}) = [\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{i}}$	$[-65.536,65.536$	~1	2D, 25 local minima.

F15	Kowalik	$f(\mathrm{x}) = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	$[-5,5]^4$	~0.0003	4D, approximation problem.
F16	Six-Hump Camel Back	$f(\mathrm{x}) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5,5]^2$	-1.0316	2D, 6 local minima.
F17	Branin	$f(\mathrm{x}) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$	$x_1 \in [-5,10], x_2 \in [0,15]$	0.398	2D, 3 global minima.
F18	Goldstein-Price	$f(\mathrm{x}) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	$[-2,2]^2$	3	2D, 4 local minima.
F19	Hartman 3	$f(\mathrm{x}) = -\sum_{i=1}^{4} c_i \exp[-\sum_{j=1}^{3} a_{ij} (- p_{ij})^2]$	$[0,1]^3$	-3.8628	3D, 4 local minima.
F20	Hartman 6	$f(\mathrm{x}) = -\sum_{i=1}^{4} c_i \exp[-\sum_{j=1}^{6} a_{ij} (- p_{ij})^2]$	$[0,1]^6$	-3.3224	6D, 4 local minima.
F21	Shekel 5	$f(\mathrm{x}) = -\sum_{i=1}^{5} [(\mathrm{x} - \mathrm{a}_i)(\mathrm{x} - \mathrm{a}_i)^T + c_i]^{-1}$	$[0,10]^4$	-10.1532	4D, 5 local minima.
F22	Shekel 7	$f(\mathrm{x}) = -\sum_{i=1}^{7} [(\mathrm{x} - \mathrm{a}_i)(\mathrm{x} - \mathrm{a}_i)^T + c_i]^{-1}$	$[0,10]^4$	-10.4028	4D, 7 local minima.
F23	Shekel 10	$f(\mathrm{x}) = -\sum_{i=1}^{10} [(\mathrm{x} - \mathrm{a}_i)(\mathrm{x} - \mathrm{a}_i)^T + c_i]^{-1}$	$[0,10]^4$	-10.5363	4D, 10 local minima.

Note: For functions F14-F23, the coefficients (a, c, p) are standard and can be found in the referenced literature [24].

4.2. Real-World Engineering Problems

Metaheuristics are often tested on practical optimization problems to validate applicability. Here is a comprehensive list of standard engineering benchmarks.

Table 2. Comprehensive List of Real-World Engineering Problems

Problem	Domain	Mathematical Formulation (Objective = Minimize Cost)	Constraints
First Category Mechanical Design			
Pressure Vessel Design (PVD)	Structural/Mechanical	$f(\mathrm{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$ $\mathrm{x} = [T_s, T_h, R, L]$	$g_1: -x_1 + 0.0193x_3 \leq 0$ $g_2: -x_2 + 0.00954x_3 \leq 0$ $g_3: -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0$
Tension/Compression Spring (TSD)	Mechanical	$f(\mathrm{x}) = (x_3 + 2)x_2x_1^2$ $\mathrm{x} = [d, D, N]$	$g_1: 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0$ $g_2: \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x} + \frac{1}{5108x_1^2} - 1 \leq 0$ $g_3: 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0$
Welded Beam Design (WBD)	Structural	$f(\mathrm{x}) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$ $\mathrm{x} = [h, l, t, b]$	Shear stress (τ), bending stress (σ), buckling load (P_c), end deflection.
Speed Reducer Design (SRD)	Mechanical	$f(\mathrm{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$	11 constraints on surface stress, bending stress, transverse deflections.
Second Category Structural Design			
Three-Bar Truss Design	Civil/Structural	$f(\mathrm{x}) = (2\sqrt{2}x_1 + x_2) \times l$ $\mathrm{x} = [A_1, A_2]$	Stress, buckling, displacement constraints.
I-Beam Design	Structural	$f(\mathrm{x}) = \frac{5000}{\frac{x_3(x_1 - 2x_4)^3}{12} + \frac{x_2x_4^3}{6} + 2x_2x_4(\frac{x_1 -}{2}}$	Cross-section area, stress constraints.
Third Category Electrical Engineering			
Gear Train Design	Mechanical	$f(\mathrm{x}) = (\frac{1}{6.931} - \frac{x_1x_2}{x_3x_4})^2$ $\mathrm{x} = [n_A, n_B, n_C, n_D]$ (integer)	Integer variables.
Brushless DC Wheel Motor	Electrical	Complex model for maximizing efficiency and minimizing mass.	Multiple electromagnetic and geometric constraints.

4.3. CEC Benchmark Suites

These are widely used in the computational intelligence community, especially for competitions. Here is a detailed breakdown of key CEC suites.

Table 3. Comprehensive CEC Benchmark Suites (2005-2022)

Suite	Year	# of Probs	Problem Types	Key Features and Example Functions
CEC 2005	2005	25	F1-F5: Unimodal F6-F12: Basic Multimodal F13-F14: Expanded Multimodal F15-F25: Hybrid Composition	Introduced standardized benchmarking. Example (F15): Hybrid Composition Function.
CEC 2011	2011	22	Real-World Problems	Parameter estimation for filters, antenna arrays, modeling, etc.
CEC 2013	2013	28	F1-F5: Unimodal F6-F20: Basic Multimodal F21-F28: Composition Functions	Focus on large-scale optimization (up to 1000D).
CEC 2014	2014	30	All Multimodal	Specifically for single objective real-parameter numerical optimization.
CEC 2017	2017	29	F1-F2: Unimodal F3-F9: Multimodal F10-F19: Hybrid F20-F29: Composition	Shifted, rotated, and hybrid functions to avoid separability. Example (F20): Shifted Rosenbrock's plus Hybrid.
CEC 2020	2020	10	All Multimodal	Focus on single objective bound constrained problems.
CEC 2021	2021	10	Single Objective Bound Constrained	New test functions with complex Pareto sets.
CEC 2022	2022	12	Single Objective Bound Constrained	Includes functions with asymmetric and complex properties.

Table 3 provides an overview of all major CEC benchmark suites used in evolutionary computation competitions from 2005 to 2022. Each suite is discussed year by year, the number of problems is indicated, and the structure is described (unimodal, multimodal, hybrid, or composition functions), the evolution was directed to a greater dimensionality and complexity was mainly depicted. Complete CEC 2017 Benchmark Functions are illustrated in Table 4. It lists the foundational mathematical functions (f_1–f_{19}) that serve as the building blocks for hybrid and composition functions in the CEC 2017 suite. Each function is briefly named and mathematically defined to show the diversity of landscapes.

Table 4. Basic Functions Used in CEC 2017

Basic Function	Function Name	Mathematical Formulation
f_1	Bent Cigar Function	$f_1(x) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$
f_2	Zakharov Function	$f_3(x) = \sum_{i=1}^{D} x_i^2 + (\sum_{i=1}^{D} 0.5x_i)^2 + (\sum_{i=1}^{D} 0.5x_i)^4$
f_3	Rosenbrock's Function	$f_4(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$
f_4	Rastrigin's Function	$f_5(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$
f_5	Expanded Scaffer's F6	$g(x,y) = 0.5 + \frac{(\sin^2(\sqrt{x^2+y^2}) - 0.5)}{(1+0.001(x^2+y^2))^2}, f_6(x) = \sum g(x_i, x_{i+1})$

f$_6$	Lunacek Bi-Rastrigin	$f_7(x) = \min(\sum(\dot{X}_i - \mu_0)^2, dD + s\sum(\dot{X}_i - \mu_1)^2) + 10(D - \sum\cos(2\pi\dot{\Sigma}_i))$
f$_7$	Non-Continuous Rastrigin	$f_8(x) = \sum(z_i^2 - 10\cos(2\pi z_i) + 10)$ with rounding operation
f$_8$	Levy Function	$f_9(x) = \sin^2(\pi w_1) + \sum_{i=1}^{D-1} (w_i - 1)^2[1 + 10\sin^2(\pi w_i + 1)] + (w_D - 1)^2[1 + \sin^2(2\pi w_D)]$
f$_9$	Modified Schwefel's	$f_{10}(x) = 418.9829 \times D - \sum g(z_i)$ with piecewise $g(z_i)$
f$_{10}$	High Conditioned Elliptic	$f_{11}(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$
f$_{11}$	Discus Function	$f_{12}(x) = 10^6 x_1^2 + \sum_{i=2}^{D} x_i^2$
f$_{12}$	Ackley's Function	$f_{13}(x) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum x_i^2}) - \exp(\frac{1}{D}\sum\cos(2\pi x_i)) + 20 + e$
f$_{13}$	Weierstrass Function	$f_{14}(x) = \sum_{i=1}^{D} (\sum_{k=0}^{20} [0.5^k\cos(2\pi \cdot 3^k(x_i + 0.5))]) - D\sum_{k=0}^{20} [0.5^k\cos(2\pi \cdot 3^k \cdot 0.5)]$
f$_{14}$	Griewank's Function	$f_{15}(x) = \sum\frac{x_i^2}{4000} - \prod\cos(\frac{x_i}{\sqrt{i}}) + 1$
f$_{15}$	Katsuura Function	$f_{16}(x) = \frac{10}{D^2}\prod_{i=1}^{D}(1 + i\sum_{j=1}^{32}\frac{\mid 2^j x_i - \mathrm{round}(2^j x_i)\mid}{2^j})^{10/D^2} - \frac{10}{D^2}$
f$_{16}$	HappyCat Function	$f_{17}(x) = \mid\sum x_i^2 - D\mid^{1/4} + (0.5\sum x_i^2 + \sum x_i)/D + 0.5$
f$_{17}$	HGBat Function	$f_{18}(x) = \mid(\sum x_i^2)^2 - (\sum x_i)^2\mid^{1/2} + (0.5\sum x_i^2 + \sum x_i)/D + 0.5$
f$_{18}$	Expanded Griewank+Rosenbrock	$f_{19}(x) = f_7(f_4(x_1, x_2)) + f_7(f_4(x_2, x_3)) + \cdots + f_7(f_4(x_D, x_1))$
f$_{19}$	Schaffer's F7 Function	$f_{20}(x) = [\frac{1}{D-1}\sum_{i=1}^{D-1}(\sqrt{s_i} \cdot (\sin(50.0s_i^{0.2}) + 1)]^2, s_i = \sqrt{x_i^2 + x_{i+1}^2}$

Table 5. Complete CEC 2017 Test Suite (Unimodal Functions)

Function	Function Name	Type	Formulation	F*	Properties
1	Shifted and Rotated Bent Cigar	Unimodal	$F_1(x) = f_1(M(x - o_1)) + 100$	100	Unimodal, Non-separable, Smooth narrow ridge
2	Shifted and Rotated Zakharov	Unimodal	$F_2(x) = f_3(M(x - o_2)) + 200$	200	Unimodal, Non-separable

Table 5 Contains unimodal functions (F1–F2) used to evaluate an algorithm's exploitation ability. Each function is shifted and rotated to remove separability, ensuring a challenging yet smooth landscape.

Table 6. Complete CEC 2017 Test Suite (Simple Multimodal Functions)

Function	Function Name	Type	Formulation	F*	Properties
3	Shifted and Rotated Rosenbrock's	Multimodal	$F_3(x) = f_4(\frac{2.048(x-o_3)}{100}+1) + 300$	300	Multi-modal, Non-separable, Many local optima
4	Shifted and Rotated Rastrigin's	Multimodal	$F_4(x) = f_5(M(x-o_4)) + 400$	400	Multi-modal, Non-separable, Many local optima
5	Shifted and Rotated Expanded Scaffer's F6	Multimodal	$F_5(x) = f_6(M(x-o_5)) + 500$	500	Multi-modal, Non-separable, Asymmetrical
6	Shifted and Rotated Lunacek Bi-Rastrigin	Multimodal	$F_6(x) = f_7(\frac{M(600(x-o_6))}{100}) + 600$	600	Multi-modal, Non-separable, Asymmetrical
7	Shifted and Rotated Non-Continuous Rastrigin's	Multimodal	$F_7(x) = f_8(\frac{5.12(x-o_7)}{100}) + 700$	700	Multi-modal, Non-separable, Asymmetrical
8	Shifted and Rotated Levy Function	Multimodal	$F_8(x) = f_9(\frac{5.12(x-o_8)}{100}) + 800$	800	Multi-modal, Non-separable
9	Shifted and Rotated Schwefel's Function	Multimodal	$F_9(x) = f_{10}(\frac{1000(x-o_9)}{100}) + 900$	900	Multi-modal, Non-separable

Table 6 Summarizes functions F3–F9 that test exploration capabilities. These are non-separable, asymmetric, and possess numerous local optima, representing the core multimodal challenges.

Table 7. Complete CEC 2017 Test Suite (Hybrid Functions)

Function	Function Name	Type	N	Basic Functions (with proportions)	F*	Properties
10	Hybrid Function 1	Hybrid	3	Zakharov (20%), Rosenbrock (40%), Rastrigin (40%)	1000	Multi-modal, Non-separable subcomponents
11	Hybrid Function 2	Hybrid	3	High Cond. Elliptic (30%), Schwefel (30%), Bent Cigar (40%)	1100	Multi-modal, Non-separable subcomponents
12	Hybrid Function 3	Hybrid	3	Bent Cigar (30%), Rosenbrock (30%), Lunacek Bi-Rastrigin (40%)	1200	Multi-modal, Non-separable subcomponents
13	Hybrid Function 4	Hybrid	4	High Cond. Elliptic (20%), Ackley (20%), Schaffer F7 (20%), Rastrigin (40%)	1300	Multi-modal, Non-separable subcomponents

14	Hybrid Function 5	Hybrid	4	Bent Cigar (20%), HGBat (20%), Rastrigin (30%), Rosenbrock (30%)	1400	Multi-modal, Non-separable subcomponents
15	Hybrid Function 6	Hybrid	4	Expanded Schaffer F6 (20%), HGBat (20%), Rosenbrock (30%), Schwefel (30%)	1500	Multi-modal, Non-separable subcomponents
16	Hybrid Function 7	Hybrid	5	Katsuura (10%), Ackley (20%), Expanded Griewank+Rosenbrock (20%), Schwefel (20%), Rastrigin (30%)	1600	Multi-modal, Non-separable subcomponents
17	Hybrid Function 8	Hybrid	5	High Cond. Elliptic (20%), Ackley (20%), Rastrigin (20%), HGBat (20%), Discus (20%)	1700	Multi-modal, Non-separable subcomponents
18	Hybrid Function 9	Hybrid	5	Bent Cigar (20%), Rastrigin (20%), Expanded Griewank+Rosenbrock (20%), Weierstrass (20%), Expanded Schaffer F6 (20%)	1800	Multi-modal, Non-separable subcomponents
19	Hybrid Function 10	Hybrid	6	HappyCat (10%), Katsuura (10%), Ackley (20%), Rastrigin (20%), Schwefel (20%), Schaffer F7 (20%)	1900	Multi-modal, Non-separable subcomponents

Table 8 Complete CEC 2017 Test Suite (Composition Functions)

Function	**Function Name**	**Type**	**N**	**Basic Functions (σ, λ, bias)**	**F***	**Properties**
20	Composition Function 1	Composition	3	Rosenbrock (10,1,0), High Cond. Elliptic (20,1e-6,100), Rastrigin (30,1,200)	2000	Multi-modal, Non-separable, Asymmetrical
21	Composition Function 2	Composition	3	Rastrigin (10,1,0), Griewank (20,10,100), Schwefel (30,1,200)	2100	Multi-modal, Non-separable, Asymmetrical
22	Composition Function 3	Composition	4	Rosenbrock (10,1,0), Ackley (20,10,100), Schwefel (30,1,200), Rastrigin (40,1,300)	2200	Multi-modal, Non-separable, Asymmetrical
23	Composition Function 4	Composition	4	Ackley (10,10,0), High Cond. Elliptic (20,1e-6,100), Griewank (30,10,200), Rastrigin (40,1,300)	2300	Multi-modal, Non-separable, Asymmetrical
24	Composition Function 5	Composition	5	Rastrigin (10,10,0), HappyCat (20,1,100), Ackley (30,10,200), Discus (40,1e-6,300), Rosenbrock (50,1,400)	2400	Multi-modal, Non-separable, Asymmetrical
25	Composition Function 6	Composition	5	Expanded Schaffer F6 (10,1e-26,0), Schwefel (20,10,100), Griewank (20,1e-6,200), Rosenbrock	2500	Multi-modal, Non-separable, Asymmetrical

				(30,10,300), Rastrigin (40,5e-4,400)		
26	Composition Function 7	Composition	6	HGBat (10,10,0), Rastrigin (20,10,100), Schwefel (30,2.5,200), Bent Cigar (40,1e-26,300), High Cond. Elliptic (50,1e-6,400), Expanded Schaffer F6 (60,5e-4,500)	2600	Multi-modal, Non-separable, Asymmetrical
27	Composition Function 8	Composition	6	Ackley (10,10,0), Griewank (20,10,100), Discus (30,1e-6,200), Rosenbrock (40,1,300), HappyCat (50,1,400), Expanded Schaffer F6 (60,5e-4,500)	2700	Multi-modal, Non-separable, Asymmetrical
28	Composition Function 9	Composition	3	Hybrid 6 (10,1,0), Hybrid 7 (30,1,100), Hybrid 8 (50,1,200)	2800	Multi-modal, Non-separable, Asymmetrical
29	Composition Function 10	Composition	3	Hybrid 5 (10,1,0), Hybrid 8 (30,1,100), Hybrid 9 (50,1,200)	2900	Multi-modal, Non-separable, Asymmetrical

Table 7 describes functions F10–F19, where multiple basic functions are combined in varying proportions to simulate heterogeneous search spaces. Each hybrid function evaluates how well algorithms adapt to problems containing subcomponents with differing characteristics.

Table 9. Complete CEC 2017 Test Suite (Key Experimental Settings)

Parameter	Value
Dimensions	D = 10, 30, 50, 100
Search Range	[-100, 100]^D
MaxFES	10,000 × D
Runs per Problem	51
Initialization	Uniform random in search range
Termination	MaxFES reached or error < 10^{-8}
Shift Data	Each function has unique shift vector o in [-80,80]^D
Rotation	Each function has unique rotation matrix M

Table 8 details functions F20–F29 constructed by blending several hybrid or basic functions with scaling, bias, and weight factors. These are the most complex benchmarks, designed to test robustness, adaptability, and convergence accuracy in highly irregular, non-separable landscapes.

4.4. Combinatorial Optimization Benchmarks

Used for discrete or combinatorial metaheuristics. Below is a list of the most common simple combinatorial problems. Table 9 indicates the main parameters that were common for CEC 2017, such as dimension (10-100), search ranges, maximum evaluations, and the random initialization process. Reproducibility is guaranteed and algorithm comparison is done fairly through these settings.

Table 10. Comprehensive Combinatorial Optimization Benchmarks

Problem	Description	Mathematical Formulation	Standard Instances
Routing Problems			

Travelling Salesman (TSP)	Find shortest tour visiting each city once.	$\min \sum_{i=1}^{n} d_{\pi(i),\pi(i+1)}$ (with $\pi(n+1)=\pi(1)$)	TSPLIB (e.g., berlin52, eil101, att532)
Vehicle Routing (VRP)	Optimize routes for a fleet of vehicles.	$\min \sum_{k=1}^{K} \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} x_{ijk}$ s.t. capacity and tour constraints.	VRPWEB, Solomon instances
Scheduling Problems			
Flow Shop (FSP)	Schedule jobs on machines in same order.	$\min C_{max}$ (makespan)	Taillard benchmarks, OR-Library
Job Shop (JSP)	Schedule jobs on machines with different orders.	$\min C_{max}$	FT06, FT10, LA01-LA40
Assignment Problems			
Quadratic Assignment (QAP)	Assign facilities to locations.	$\min \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{p(i)p(j)}$	QAPLIB (e.g., nug12, nug30, tai64c)
Generalized Assignment (GAP)	Assign tasks to agents with capacity.	$\min \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} x_{ij}$ s.t. $\sum_{j=1}^{m} x_{ij} = 1$ and capacity constraints.	OR-Library
Packing/Covering			
Knapsack (0/1, Multi)	Select items with max value within capacity.	$\max \sum_{i=1}^{n} v_i x_i$ s.t. $\sum_{i=1}^{n} w_i x_i \le W, x_i \in \{0,1\}$	OR-Library, SAC-94 suite
Bin Packing	Pack items into minimum number of bins.	$\min \sum_{j=1}^{n} y_j$ s.t. $\sum_{i=1}^{n} w_i x_{ij} \le C y_j$	OR-Library, BPPLIB

4.5. Multi-Objective Benchmarks

For algorithms that are able to optimize several objectives opposing one another, here is a comprehensive list of the standard multi-objective benchmarks.

ZDT Suite

The ZDT (Zitzler-Deb-Thiele) test suite, which consist of five standard bi-objective benchmarking problems ZDT1-ZDT6, is the first in the list. Each function is described in terms of its mathematical formulation, Pareto front shape (convex, concave, or disconnected), and main purpose testing convergence, diversity preservation, and the ability to maintain multiple subpopulations in multimodal and discontinuous environments. See Table 10.

Table 11. ZDT Suite Functions Primarily For 2-Objective Problems. The Number of Decision Variables Is N.

Fn	Obj	Search Space	Mathematical Formulation	Pareto Front Shape	Key Feature & Difficulty
ZDT1	2	$[0,1]^n$	$f_1(x) = x_1$	Convex	Simple, convex front. Tests convergence.

			$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$ $f_2(x) = g(x)(1 - \sqrt{\frac{f_1(x)}{g(x)}})$		
ZDT2	2	$[0,1]^n$	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$ $f_2(x) = g(x)(1 - (\frac{f_1(x)}{g(x)})^2)$	Non-convex (Concave)	Non-convex front. Tests algorithm's ability to handle concave geometries.
ZDT3	2	$[0,1]^n$	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$ $f_2(x)$ $= g(x)[1 - \sqrt{\frac{f_1(x)}{g(x)}}$ $- \frac{f_1(x)}{g(x)}\sin(10\pi f_1(x))]$	Disconnected	Convex, disconnected segments. Tests ability to find and maintain multiple sub-populations.
ZDT4	2	$x_1 \in [0,1]$ $x_{2..n}$ $\in [-5,5]$	$f_1(x) = x_1$ $g(x)$ $= 1 + 10(n-1)$ $+ \sum_{i=2}^{n} [x_i^2 - 10\cos(4\pi x_i)]$ $f_2(x) = g(x)(1 - \sqrt{\frac{f_1(x)}{g(x)}})$	Convex	Multimodal. Many local Pareto fronts. Tests convergence and escape from local optima.
ZDT6	2	$[0,1]^n$	$f_1(x)$ $= 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $g(x) = 1 + 9[\frac{\sum_{i=2}^{n} x_i}{n-1}]^{0.25}$ $f_2(x) = g(x)(1 - (\frac{f_1(x)}{g(x)})^2)$	Non-convex	Non-uniformly dense solutions (biased). Tests algorithms' ability to maintain diversity.

DTLZ Suite

The DTLZ (Deb–Thiele–Laumanns–Zitzler) suite, designed for scalable multi-objective optimization with two or more objectives. The paper describes the mathematical structure, Pareto front geometry, and main difficulties of functions such as DTLZ1 (linear, multimodal), DTLZ2 (concave, continuous), and DTLZ7 (disconnected). These functions are a prerequisite for the assessment of both scalability and algorithm performance in the high-dimensional objective spaces. DTLZ suite functions are summarized in Table 11.

Table 12. DTLZ Suite Functions

Fn	Objectives (M)	Search Space	Mathematical Formulation (Word Equation Format)	Pareto Front Shape	Key Features & Difficulties
DTLZ1	2 – Many	$[0, 1]^n$	$f_1(x)$ = ½ (1 + $g(x_m)$) × ∏_{i=1}^{M−1} x_i $f_j(x)$ = ½ (1 + $g(x_m)$) × (∏_{i=1}^{M−j} x_i) × (1 − x_{M−j+1}) $f_m(x)$ = ½ (1 + $g(x_m)$) × (1 − x_1) $g(x_m)$ = 100 [$\lvert x_m \rvert$ +	Linear (hyperplane Σ f_m = 0.5)	Multimodal with linear correlation among objectives. Challenging due to many local

			$\Sigma\_\{x_i \in x_m\}$ $((x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)))$]		optima and slow convergence.
DTLZ2	2 – Many	$[0, 1]^n$	$f_1(x) = (1 + g(x_m)) \times \prod\_\{i=1\}\^\{M-1\} \cos(x_i^\alpha \pi / 2)$ $f_j(x) = (1 + g(x_m)) \times (\prod\_\{i=1\}\^\{M-_j\} \cos(x_i^\alpha \pi / 2)) \times \sin(x\_\{M-_j+1\}^\alpha \pi / 2)$ $f_m(x) = (1 + g(x_m)) \times \sin(x_1^\alpha \pi / 2)$ $g(x_m) = \Sigma\_\{x_i \in x_m\} (x_i - 0.5)^2$, typically $\alpha = 1$	Concave (on a unit hypersphere)	Smooth, scalable benchmark for evaluating convergence and diversity in multi-objective algorithms.
DTLZ7	2 – Many	$[0, 1]^n$	$f_i(x) = x_i$, $i = 1 \ldots M-1$ $f_m(x) = (1 + g(x_m)) \times h(f_1, f_2, \ldots, f\_\{M-1\}, g)$ $g(x_m) = 1 + (9 / \lvert x_m \rvert) \times \Sigma\_\{x_i \in x_m\} x_i$ $h = M - \Sigma\_\{i=1\}\^\{M-1\} [(f_i / (1 + g)) \times (1 + \sin(3\pi f_i))]$	Disconnected	Produces 2^(M−1) disconnected Pareto-optimal regions. Tests algorithm robustness in maintaining diversity and handling discontinuities.

Scalable to M objectives. The letter k represents the number of parameters related to the position (normally k = n - M + 1). The notation x_M symbolizes the last k variables.

WFG Suite

The WFG (Walking Fish Group) benchmark suite, consisting of nine complex, parameterized functions (WFG1–WFG9). Each is characterized by specific transformations such as bias, non-separability, and deception. The Pareto front shapes range from convex to concave and mixed forms. WFG suite is used to measure an algorithm's weight, its flexibility and dealing ability with human-like problems in multi-objective optimization. WFG suite functions summarized in
Table *12*.

Table 13. WFG Suite Functions

Function	**Obj. (M)**	**Search Space**	**Mathematical Formulation (Key Characteristics)**	**Pareto Front Shape**	**Key Feature & Difficulty**
WFG1	2-Many	$[0, 2i]^n$	Shape: Convex & Mixed Transformations: Polynomial bias, flat regions, and parameter dependence.	Mixed Convex/ Concave	Biased & multimodal. Difficult due to flat regions and complex parameter interactions.
WFG2	2-Many	$[0, 2i]^n$	Shape: Convex Transformations: Non-separable reduction (sub-problems require grouping variables).	Convex, Disconnected	Non-separable & disconnected. Tests convergence on a convex front with disconnected parts.
WFG3	2-Many	$[0, 2i]^n$	Shape: Linear Transformations: Non-	Linear, Degenerate	Degenerate front (true Pareto set has

			separable reduction and a linear degenerate front.		a lower dimension than the objective space).
WFG4	2-Many	$[0, 2i]^n$	Shape: Concave Transformations: Multi-modal (many local optima) via sin functions.	Concave	Multi-modal. Concave front with many local optima, testing convergence.
WFG5	2-Many	$[0, 2i]^n$	Shape: Concave Transformations: Deceptive minima.	Concave	Deceptive. Hard to find the true global Pareto front.
WFG6	2-Many	$[0, 2i]^n$	Shape: Concave Transformations: Non-separable reduction without using sin functions.	Concave	Non-separable. Tests algorithms on a concave front with variable dependencies.
WFG7	2-Many	$[0, 2i]^n$	Shape: Concave Transformations: Parameter dependence (shift) causing bias.	Concave	Biased. Creates a non-uniform distribution of solutions.
WFG8	2-Many	$[0, 2i]^n$	Shape: Concave Transformations: Parameter dependence (shift) and non-separability.	Concave	Biased & non-separable. A more difficult version of WFG7.
WFG9	2-Many	$[0, 2i]^n$	Shape: Concave Transformations: Combines bias (shift), non-separability, and multi-modality.	Concave	Composite difficulties. The most complex WFG problem, combining multiple challenges.

Very adaptive and customizable. Stands for the number of position parameters (which should be a multiple of M - 1) and l is the number of distance parameters. n = k + l.

A comparative summary of the three main multi-objective benchmark families ZDT, DTLZ, and WFG is provided in Table 13. The discussion addresses their scalability, complexity, and main areas of research. The ZDT suite handles easy two-objective problems, DTLZ is for large many-objective cases, and WFG is for algorithm testing under various conditions such as composite, biased, and deceptive.

Table 14. Summary and Use Cases

Function Family	Primary Use	Scalability	Key Strengths	Typical Use Cases
ZDT	2-Objective Testing	Fixed (2 only)	Simple, intuitive, fast to compute.	Basic algorithm validation, testing convergence and diversity mechanisms.
DTLZ	Many-Objective Testing	Highly Scalable	Systematic design for M objectives; diverse front shapes (linear, concave, disconnected).	Testing scalability to many objectives, performance on specific geometries (e.g., disconnected, linear).

WFG	Complex Challenge Testing	Highly Scalable	Highly configurable; can introduce real-world-like challenges (bias, non-separability, deception).	Stress-testing algorithms on complex, composite problems that mimic real-world difficulties.

4.6. Evaluation Metrics

Quantifying Algorithmic Performance Requires Multiple Metrics.

- Solution Quality: Best, worst, mean, and median fitness value over multiple runs.
- Convergence Speed: Function Evaluation (NFE) or iterations count to reach a target solution quality.
- Robustness: The quality standard deviation of the final solution over a number of runs.
- Statistical Significance: Non-parametric tests (like Wilcoxon signed-rank test) are used to establish that the performance differences are not due to chance.

Multi-Objective Metrics:

- Hypervolume (HV): Area of the objective space that is controlled by the Pareto front.
- Inverted Generational Distance (IGD): Mean length from the real Pareto front to the discovered solutions.
- Spread (Δ): Indicates the degree of dispersion reached among the acquired solutions.

5. DISCUSSION

The very large variety of benchmarks laid out here emphasizes their indispensable part in the progress of metaheuristics research. However, some significant problems and considerations are still present:

The No Free Lunch (NFL) Theorem: This theorem states that no method can be the best in all problem contexts. The performance across different benchmark categories (unimodal vs. multimodal, continuous vs. combinatorial, single vs. multi-) multi-objective is one area where this is clearly seen; a particular algorithm may outperform in one case but it may also be the worst in the next. Thus, benchmarking should be regarded as a way of the algorithm domain of competence through which the declaring of a universal winner is avoided.

- Overfitting and Benchmark-Specific Tuning: One major concern in this area is that in the process of tuning an algorithm's parameters for a particular benchmark suite, e.g., CEC 2017 functions, one may end up with an impressive performance on that suite alone and non-generalization in other problems including real-world applications. It is imperative for the researchers to carry out cautious tuning of parameters and evaluating performance on an independent test set or through an entirely different benchmark category.
- The Gap between Synthetic and Real-World Problems: The CEC functions and other mathematical benchmarks do a great job of exposing the specific algorithmic traits; however, they often do it at the cost of not being able to depict the scenario in which many real-world simulations are "black-box", expensive, noisy, and with multiple constraints. There is a need for the community to present more and more application-driven benchmarks that portray these features.
- Performance Metrics and Reproducibility: A full assessment should include not only the mean best fitness but also other factors. These factors include statistical significance tests, convergence curves, and robustness measures, to name a few. Additionally, employing standard benchmark suites and code open-sourcing are crucial to deliver reproducibility and fair comparison.
- The Proliferation of Novel Algorithms: Nature-inspired metaheuristics that are new to the field keep showing up, sometimes with strong assertions. The skilful and rigorous benchmarking against the

established comprehensive suites described in this paper is the most effective means of preventing the publication of inferior and/or duplicate methods.

- Benchmarking Complexity and Resources: A full-scale benchmark that covers all these categories of algorithms and problems will require a vast amount of computational resources and a lot of time. Researchers will need to choose wisely which sets of benchmarks will be the most appropriate ones to validate their claims regarding the algorithm's performance.

6. CONCLUSION

This paper has provided a structured and comprehensive review of the complete benchmark ecosystem used in metaheuristic optimization. We have categorized and detailed the mathematical foundations of the classic 23 test functions, numerous real-world engineering problems, the evolving CEC competition suites, fundamental combinatorial benchmarks, and essential multi-objective problems. This complete classification provides researchers with a thorough and unambiguous road map to the choice of proper benchmarks for a strict algorithmic evaluation.

The dialogue underlined that good benchmarking is not simply about getting the best numbers from a single suite but rather a comprehensive performance evaluation of the entire problem portfolio, with great attention to statistical significance, real-world applicability, and the risk of overfitting. As the discipline changes, the following areas should be the focus of future benchmarking:

A. Closing the gap between artificial and real-world problems,
B. Setting up standards for dynamic, large-scale, and uncertain optimization,
C. Establishing benchmarks in the new areas such as multi-task optimization and transfer optimization,
D. Supporting open-source frameworks that assure reproducibility and equitable comparison.

The application of these strict and wide-ranging benchmarking methods will help the metaheuristics community continue to be a breeding ground for true innovation and to provide solid optimization solutions to the intricate problems of science and technology.

Acknowledgments

The authors have no specific acknowledgments to make for this research.

Author Contributions Statement

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Saman M. Almufti	✓	✓	✓	✓	✓	✓		✓	✓	✓		✓	✓	
Renas Rajab Asaad		✓		✓			✓			✓				
Awaz Ahmed Shaban		✓		✓	✓			✓			✓			✓
Ridwan Boya Marqas			✓		✓	✓			✓				✓	

C : **Co**nceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

Funding Information

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of Interest Statement

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Informed Consent

All participants were informed about the purpose of the study, and their voluntary consent was obtained prior to data collection.

Ethical Approval

The study was conducted in compliance with the ethical principles outlined in the Declaration of Helsinki and approved by the relevant institutional authorities.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] A. Rodan, A.-K. Al-Tamimi, L. Al-Alnemer, and S. Mirjalili, 'Stellar oscillation optimizer: a nature-inspired metaheuristic optimization algorithm', Cluster Comput, vol. 28, no. 6, 2025. doi.org/10.1007/s10586-024-04976-5

[2] S. Mohapatra and P. Mohapatra, 'An Improved Golden Jackal Optimization Algorithm Using Opposition-Based Learning for Global Optimization and Engineering Problems', International Journal of Computational Intelligence Systems, vol. 16, no. 1, 2023. doi.org/10.1007/s44196-023-00320-8

[3] S. M. Almufti, Metaheuristics Algorithms: Overview, Applications, and Modifications. 2025. doi.org/10.70593/978-93-7185-454-2

[4] A. A. Shaban, S. M. Almufti, and R. R. Asaad, 'Metaheuristic Algorithms for Engineering and CombinatorialOptimization: A Comparative Study Across Problems Categories and Benchmarks', International Journal of Scientific World, vol. 11, no. 2, pp. 38-49, 2025. doi.org/10.14419/0hndc578

[5] S. M. Almufti, U-Turning Ant Colony Algorithm powered by Great Deluge Algorithm for the solution of TSP Problem. 2015.

[6] T. Hamadneh, 'Social Rumor Propagation Optimization (SRPO): A Novel Metaheuristic Approach Inspired by Human Behaviour in Social Networks', International Journal of Intelligent Engineering and Systems, vol. 18, no. 5, pp. 585-597, 2025. doi.org/10.22266/ijies2025.0630.41

[7] A. Tang, H. Zhou, T. Han, and L. Xie, "A modified manta ray foraging optimization for global optimization problems," IEEE Access, vol. 9, pp. 128702-128721, 2021 doi.org/10.1109/ACCESS.2021.3113323

[8] N. H. Awad, M. Z. Ali, P. N. Suganthan, J. J. Liang, and B. Y. Qu, Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. 2016.

[9] S. Liu, Q. Lin, K. C. Tan, and Q. Li, Benchmark Problems for CEC2021 Competition on Evolutionary Transfer Multiobjectve Optimization. 2021.

[10] A. Zraiqat, 'Driver and Navigator Algorithm: A Novel Parameter-Free Human-inspired Metaheuristic for Efficient Global Optimization', International Journal of Intelligent Engineering and Systems, vol. 18, no. 9, pp. 555-569, 2025. doi.org/10.22266/ijies2025.1031.36

[11] V. Plevris and G. Solorzano, 'A Collection of 30 Multidimensional Functions for Global Optimization Benchmarking', Data (Basel), vol. 7, no. 4, 2022. doi.org/10.3390/data7040046

[12] K. Hussain, M. N. Salleh, S. Cheng, and R. Naseem, 'Common Benchmark Functions for Metaheuristic Evaluation: A Review', JOIV : International Journal on Informatics Visualization, vol. 1, no. 4-2, 2017. doi.org/10.30630/joiv.1.4-2.65

[13] S. M. Almufti, 'Fusion of Water Evaporation Optimization and Great Deluge: A Dynamic Approach for Benchmark Function Solving', Fusion: Practice and Applications, vol. 13, pp. 19-36, 2023. doi.org/10.54216/FPA.130102
[14] J. Kudela and R. Matousek, 'New Benchmark Functions for Single-Objective Optimization Based on a Zigzag Pattern', IEEE Access, vol. 10, pp. 8262-8278, 2022. doi.org/10.1109/ACCESS.2022.3144067
[15] S. M. Almufti, R. B. Marqas, P. S. Othman, and A. B. Sallow, 'Single-based and population-based metaheuristics for solving np-hard problems', Iraqi Journal of Science, vol. 62, no. 5, pp. 1710-1720, 2021. doi.org/10.24996/10.24996/ijs.2021.62.5.34
[16] S. Almufti, 'Vibrating Particles System Algorithm: Overview, Modifications and Applications', ICONTECH INTERNATIONAL JOURNAL, vol. 6, no. 3, pp. 1-11, 2022. doi.org/10.46291/ICONTECHvol6iss3pp1-11
[17] S. M. Almufti, 'Exploring the Impact of Big Bang-Big Crunch Algorithm Parameters on Welded Beam Design Problem Resolution', Academic Journal of Nawroz University, vol. 12, no. 4, pp. 1-16, 2023. doi.org/10.25007/ajnu.v12n4a1903
[18] O. K. Erol, O. Hasançebi, and S. Kazemzadeh Azad, "Evaluating efficiency of big-bang big-crunch algorithm in benchmark engineering optimization problems," INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING, vol. 3, pp. 495–505, 2011.
[19] S. M. Almufti, 'Historical survey on metaheuristics algorithms', Int. J. Sci. World, vol. 7, no. 1, p. 1, Nov. 2019. doi.org/10.14419/ijsw.v7i1.29497
[20] S. M. Almufti and A. A. Shaban, 'Advanced Metaheuristic Algorithms for Structural Design Optimization', FMDB Transactions on Sustainable Intelligent Networks, vol. 2, no. 1, pp. 33-48, 2025. doi.org/10.69888/FTSIN.2025.000368
[21] C. Zhong, G. Li, Z. Meng, H. Li, A. R. Yildiz, and S. Mirjalili, 'Starfish optimization algorithm (SFOA): a bio-inspired metaheuristic algorithm for global optimization compared with 100 optimizers', Neural Comput Appl, vol. 37, no. 5, pp. 3641-3683, 2025. doi.org/10.1007/s00521-024-10694-1
[22] L. Xie, T. Han, H. Zhou, Z. R. Zhang, B. Han, and A. Tang, "Tuna Swarm Optimization: A Novel Swarm-Based Metaheuristic Algorithm for Global Optimization," 2021, Hindawi Limited. doi.org/10.1155/2021/9210050
[23] S. M. Almufti, R. Boya, and V. A. Marqas, 'Taxonomy of bio-inspired optimization algorithms', Journal of Advanced Computer Science & Technology, vol. 8, no. 2, 2019. doi.org/10.14419/jacst.v8i2.29402
[24] S. Mohapatra, D. Kaliyaperumal, and F. S. Gharehchopogh, "A revamped black winged kite algorithm with advanced strategies for engineering optimization," Sci Rep, vol. 15, no. 1, Dec. 2025. doi.org/10.1038/s41598-025-93370-1

How to Cite: Saman M. Almufti, Renas Rajab Asaad, Awaz Ahmed Shaban, Ridwan Boya Marqas (2025). Benchmarking metaheuristic algorithms: a comprehensive review of test functions, real-world problems, and evaluation metrics. Journal of Electronics, Computer Networking and Applied Mathematics, 5(2), 16–35. https://doi.org/10.55529/jecnam.52.16.35

BIOGRAPHIES OF AUTHORS

Saman M. Almufti, as a researcher at both the Technical College of Informatics–Akre, Akre University for Applied Sciences, Duhok, Iraq, and the Department of Computer Science, College of Science, Knowledge University, Erbil. His research focuses on metaheuristic algorithms, swarm intelligence, and applied machine learning. He has published numerous papers in high-impact international journals. Email: Saman.Almofty@gmail.com

	**Renas Rajab Asaad**, is a researcher at the Technical College of Informatics–Akre, Akre University for Applied Sciences, and the Department of Computer Science, College of Science, Knowledge University, Erbil. His primary research areas include optimization, computer vision, and intelligent computing systems.
	**Awaz Ahmed Shaban**, is a Researcher at the Information Technology Department, Technical College of Informatics–Akre, Akre University for Applied Sciences, Duhok, Iraq. Her research interests include artificial intelligence, optimization algorithms, and data-driven decision systems. She has contributed to several studies in computational intelligence and applied informatics. Email: Awaz.Amedy1@gmail.com
	**Ridwan Boya Marqas,** is a researcher at the Department of Computer Science, College of Science, Knowledge University, and Erbil, Iraq. His research interests include computer networks, cybersecurity, and intelligent information systems. He has contributed to several academic and applied research projects focusing on enhancing network security and developing intelligent computing solutions.