Metaheuristic Algorithms with Applications

www.maa.reapress.com

Metaheur. Algor. Appl. Vol. 3, No. 2 (2025) 209–228.

Paper Type: Original Article

Metaheuristic Algorithms for Mathematical Problem Solving: A Comparative Experimental Study on Nonlinear Systems, Boundary Value Problems, and NP-Hard Combinatorial Optimization

Nadia Ghasemabadi[1,*], Zahra Joorbonyan[2]

[1] Department of Computer Engineering, Ayandegan Institute of Higher Education, Tonkabon, Iran; nadiaghasemi@aihe.ac.ir.

[2] Ayandegan Institute of Higher Education, Gilan University, International Society of Fuzzy Set Extensions and Applications (ISFSEA), Morvarid Intelligent Industrial Systems Research Group, Iran, Research Expansion Alliance (REA) Press; zahrajbn20@gmail.com.

Citation:

Received: 22 March 2025 *Revised: 13 June 2025* *Accepted: 03 July 2025*	Ghasemabadi, N., & Joorbonyan, Z. (2026). Metaheuristic algorithms for mathematical problem solving: A comparative experimental study on nonlinear systems, boundary value problems, and NP-hard combinatorial optimization. *Metaheuristic algorithms with applications*, 2(3), 209-228.

Abstract

This study presents a comprehensive experimental comparison of six metaheuristic algorithms Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Grey Wolf Optimizer (GWO), Salp Swarm Algorithm (SSA), and Sine Cosine Algorithm (SCA) applied to three fundamental classes of mathematical problems: 1) systems of nonlinear equations with dimensionality ranging from 10 to 100 variables, 2) parameter estimation in ordinary and Partial Differential Equations (PDE), and 3) NP-hard combinatorial optimization including the Traveling Salesman Problem (TSP) and graph coloring. Experiments were conducted on 23 Congress on Evolutionary Computation (CEC) 2014 benchmark functions in dimensions 10, 30, 50, and 100 under 51 independent runs with 500,000 function evaluations per run; 8 nonlinear systems drawn from the scientific and engineering literature; 5 Ordinary Differential Equation (ODE)/PDE parameter estimation problems with synthetic and semi-synthetic data; and 10 Traveling Salesman Problem Library (TSPLIB) instances alongside 5 Discrete Mathematics and Theoretical Computer Science Center (DIMACS) graph coloring instances. Rigorous statistical analysis was performed using Friedman nonparametric tests, Nemenyi post-hoc comparisons, and Wilcoxon signed-rank pairwise tests with Bonferroni correction. Results demonstrate domain-specific algorithmic superiority: DE achieves the best mean Friedman rank on continuous benchmark functions (mean rank 1.43 at D=30), GWO excels in nonlinear equation solving with a 97.8% success rate, and GA outperforms all competitors on combinatorial problems with a gap-to-optimal of 0.87% across TSP instances. No single algorithm dominates all problem domains, confirming the No Free Lunch theorem in the context of mathematical optimization. Practical guidelines for algorithm selection across mathematical problem types are provided, along with discussions of scalability, computational complexity, and directions for future hybrid and adaptive approaches.

Keywords: Metaheuristic algorithms, Mathematical optimization, Nonlinear equations, Differential equations, Combinatorial optimization, Benchmark functions, Convergence analysis, Friedman test, Swarm intelligence, Evolutionary computation.

1 | Introduction

Mathematical optimization constitutes one of the most foundational pillars of applied science and engineering. From the design of aircraft wings governed by computational fluid dynamics to the calibration

Corresponding Author: nadiaghasemi@aihe.ac.ir

https://doi.org/10.48313/maa.v2i3.47

of epidemiological models that inform public health policy, optimization problems pervade every quantitative discipline. At its core, the optimization task is to identify a point x* in a feasible region $S \subseteq \mathbb{R}n$ such that the objective function $f: S \rightarrow \mathbb{R}$ attains its minimum (or maximum). Despite this conceptual simplicity, the practical difficulty of optimization varies enormously depending on the structure of f and S, ranging from convex quadratic programs solvable in polynomial time to NP-hard combinatorial problems for which no efficient exact algorithm is known.

Classical numerical methods for mathematical optimization have a rich and distinguished history. The Newton-Raphson method [1], [2], arguably the most celebrated root-finding algorithm, offers quadratic convergence for smooth functions when initialized sufficiently close to a solution. Conjugate gradient methods, pioneered by Hestenes and Stiefel [3], extend steepest descent by exploiting conjugacy to achieve superlinear convergence on quadratic objectives. The simplex method of Dantzig [4] remains a workhorse for linear programming despite its exponential worst-case complexity. Interior-point methods, following the seminal work of Karmarkar [5], guarantee polynomial-time convergence for linear and certain nonlinear programs. For systems of nonlinear equations, Broyden's method [6] and other quasi-Newton approaches provide superlinear convergence without requiring explicit Jacobian computation.

However, these classical methods share several critical limitations that restrict their applicability to broad classes of mathematical problems: 1) gradient-based methods require differentiability of the objective function, which is violated by many real-world models involving discontinuities, non-smooth constraints, or stochastic noise, 2) Newton-type methods guarantee only local convergence, meaning that the quality of the solution depends sensitively on the initial guess a dependence that becomes increasingly problematic as dimensionality grows and the objective landscape becomes riddled with local optima, 3) classical methods for combinatorial optimization, such as branch-and-bound and cutting-plane algorithms, scale poorly with problem size; the Traveling Salesman Problem (TSP), for instance, has defied polynomial-time exact solution for over seven decades, and 4) many mathematical problems arising in scientific computing including inverse problems for Partial Differential Equations (PDE) and parameter estimation in dynamical systems exhibit ill-conditioning, non-convexity, and multimodality that render gradient-based approaches unreliable without extensive problem-specific tuning.

These limitations have motivated the development and proliferation of metaheuristic optimization algorithms over the past four decades. Metaheuristics are high-level, problem-independent algorithmic frameworks that guide subordinate heuristics toward promising regions of the search space through mechanisms inspired by natural phenomena, physical processes, or mathematical constructions. Unlike classical methods, metaheuristics impose no requirements on differentiability, convexity, or continuity of the objective function. They operate on populations of candidate solutions, enabling parallel exploration of the search space and conferring natural resilience against local-optima entrapment. The Genetic Algorithm (GA), introduced by Holland [7] and popularized by Goldberg [8], mimics Darwinian natural selection through crossover, mutation, and selection operators. Particle Swarm Optimization (PSO), proposed by Kennedy and Eberhart [9], simulates the social behavior of bird flocking and fish schooling. Differential Evolution (DE), developed by Storn and Price [10], employs differential mutation to generate trial vectors with self-adaptive step sizes. More recent algorithms include the Grey Wolf Optimizer (GWO) of Mirjalili et al. [11], which models the hierarchical hunting strategy of grey wolves; the Salp Swarm Algorithm (SSA) of Mirjalili et al. [12], inspired by the swarming behavior of salps in ocean environments; and the Sine Cosine Algorithm (SCA) of Mirjalili [13], which exploits sinusoidal functions to balance exploration and exploitation.

Despite the extensive literature on metaheuristic optimization, a notable research gap persists: few studies have systematically compared modern metaheuristic algorithms across multiple, mathematically distinct problem classes under unified experimental conditions. Most existing studies focus on a single problem type, typically continuous function optimization, using the Congress on Evolutionary Computation (CEC) benchmark suites, while neglecting the performance of these algorithms on nonlinear equation solving, differential equation parameter estimation, and combinatorial optimization. Furthermore, scalability analyses that track algorithmic performance from low-dimensional (D=10) to high-dimensional (D=100) settings remain relatively uncommon, particularly for the newer swarm-based algorithms (GWO, SSA, SCA).

The present study addresses these gaps through four principal contributions: 1) a rigorous experimental comparison of six representative metaheuristic algorithms (GA, PSO, DE, GWO, SSA, SCA) spanning evolutionary, swarm-based, and physics-inspired paradigms, 2) evaluation across three mathematically distinct problem domains continuous benchmark optimization, nonlinear system solving, ODE/PDE parameter estimation, and NP-hard combinatorial optimization, 3) a systematic scalability analysis across dimensions D = 10, 30, 50, and 100 on the CEC 2014 benchmark suite, and 4) statistical rigor ensured through Friedman nonparametric tests, Nemenyi post-hoc comparisons, and Wilcoxon signed-rank tests with Bonferroni correction. The remainder of this article is organized as follows: Section 2 reviews the relevant literature, Section 3 details the methodology, including algorithmic specifications, benchmark descriptions, and statistical testing protocols, Section 4 presents experimental results with comprehensive tables and convergence analyses, Section 5 discusses the findings and their implications, and Section 6 concludes with recommendations and future research directions.

2 | Literature Review

2.1 | Metaheuristics for Continuous Function Optimization

The systematic benchmarking of metaheuristic algorithms on continuous optimization problems has been a central activity of the computational intelligence community since the early 2000s. The IEEE CEC has played a pivotal role in this effort by publishing standardized benchmark suites at regular intervals. Liang et al. [14] introduced the CEC 2013 benchmark comprising 28 functions spanning unimodal, multimodal, and composition categories, establishing a widely adopted evaluation framework. The CEC 2014 benchmark, curated by Liang et al. [15], extended this to 30 functions with shifted and rotated variants to prevent algorithms from exploiting structural regularities. Awad et al. [16] further expanded the suite for CEC 2017 with 29 functions, including hybrid compositions of increasing complexity.

Among the algorithms evaluated on these benchmarks, DE has consistently demonstrated strong performance. Das and Suganthan [17] provided a comprehensive survey of DE variants, documenting the algorithm's success across hundreds of benchmark experiments. Tanabe and Fukunaga [18] proposed Success-History-based Adaptive Differential Evolution (SHADE), which adapts the scaling factor F and crossover rate CR using a historical memory mechanism, achieving top rankings in the CEC 2013 competition. More recently, Mirjalili and Lewis [19] introduced the Whale Optimization Algorithm (WOA), inspired by the bubble-net feeding behavior of humpback whales, and demonstrated competitive performance on 29 benchmark functions. Mohamed et al. [20] proposed LSHADE-cnEpSin, an enhanced adaptive DE variant that won the CEC 2017 competition, further cementing DE's position as a leading continuous optimizer.

PSO has also been the subject of extensive benchmarking. Clerc and Kennedy [21] introduced the constriction coefficient approach to PSO, providing theoretical convergence guarantees under simplified assumptions. Liang et al. [22] proposed Comprehensive Learning Particle Swarm Optimization (CLPSO), which diversifies the personal best information used to update particle velocities, achieving significant improvements on multimodal functions. The GWO algorithm of Mirjalili et al. [11] has attracted considerable attention since its introduction; comparative studies by Faris et al. [23] documented GWO's competitive performance on CEC benchmarks, particularly on unimodal and simple multimodal functions, while noting limitations on highly deceptive composition functions.

2.2 | Solving Nonlinear Equations via Metaheuristics

The application of metaheuristic algorithms to the solution of systems of nonlinear equations represents a natural and increasingly productive line of research. Grosan and Abraham [24] provided one of the earliest systematic studies, demonstrating that reformulating a system of n equations in n unknowns as an unconstrained optimization problem minimizing $f(x) = \Sigma i = 1n\ [fi(x)]^2$ enables metaheuristic algorithms to locate multiple roots simultaneously, an advantage over Newton-type methods that converge to a single root per invocation. Mo et al. [25] applied a conjugate direction PSO to systems arising in chemical engineering, achieving convergence on systems with up to 20 variables where the Levenberg-Marquardt method failed due to poor initial guesses.

Pourjafari and Mojallali [26] proposed a modified GA with adaptive crossover and mutation rates for solving nonlinear systems in interval arithmetic, successfully locating all roots within specified intervals. Hirsch et al. [27] applied continuous Greedy Randomized Adaptive Search Procedure (GRASP) to nonlinear systems with up to 50 variables, demonstrating that multi-start strategies can achieve high success rates even on systems with dense Jacobian structures. More recently, Nadimi-Shahraki et al. [28] applied an improved WOA to power flow equations with 30 buses, reporting a 96.4% success rate and mean residuals below 10^{-12}.

2.3 | Ordinary Differential Equation/Partial Differential Equation Parameter Estimation

Parameter estimation in dynamical systems governed by ordinary and PDE is a classical inverse problem with applications in physics, biology, chemistry, and climate science. The problem is inherently challenging because the forward model (the numerical solution of the ODE/PDE) must be evaluated at each optimizer iteration, and the resulting objective landscape is typically non-convex and multimodal. Babaei [29] applied a hybrid GA-Nelder-Mead approach to estimate thermal diffusivity in the one-dimensional heat equation, achieving relative errors below 0.5% on synthetic data with 5% added noise. Several authors have explored Runge-Kutta-based optimization methods; notably, Ahmadianfar et al. [30] proposed the Runge-Kutta Optimizer (RUN) algorithm inspired by the slope-arithmetic structure of Runge-Kutta integration, demonstrating state-of-the-art performance on CEC 2014 benchmarks and engineering design problems.

For biological models, Mendes and Kell [31] showed that evolutionary strategies outperform gradient-based methods for estimating parameters in the Lotka-Volterra predator-prey system, particularly when data are sparse or noisy. Rodriguez-Fernandez et al. [32] compared scatter search, DE, and Levenberg-Marquardt for parameter estimation in biochemical reaction networks, concluding that DE achieved the best balance between accuracy and computational cost. In the context of epidemiological models, Godio et al. [33] applied PSO to estimate SIR model parameters for COVID-19 in Italy, obtaining excellent fits to early-phase incidence data with R^2 values exceeding 0.998.

2.4 | Combinatorial Optimization

Combinatorial optimization problems, characterized by discrete, finite (but exponentially large) search spaces, pose a qualitatively different challenge for metaheuristic algorithms. The TSP, a canonical NP-hard problem, has served as the primary benchmark since the inception of the field. Dorigo and Gambardella [34] introduced Ant Colony Optimization (ACO) and demonstrated near-optimal performance on TSP instances with up to 200 cities. Helsgaun [35] proposed the Lin-Kernighan-Helsgaun (LKH) algorithm, a sophisticated local search heuristic that has produced optimal or near-optimal solutions for instances with over 10,000 cities. For GA-based approaches to TSP, Larranaga et al. [36] provided a comprehensive survey of representation schemes and crossover operators, establishing the Ordered Crossover (OX) and Partially Matched Crossover (PMX) as standard operators for permutation-encoded problems.

Graph coloring, another NP-hard problem, has been addressed by metaheuristic algorithms with increasing success. Malaguti et al. [37] proposed a GA with greedy partition crossover for the graph coloring problem, achieving competitive results on benchmark instances from the Discrete Mathematics and Theoretical Computer Science Center (DIMACS). Lu and Hao [38] introduced a memetic algorithm combining tabu search with crossover, producing state-of-the-art results on several DIMACS instances, including the challenging flat300 and le450 families. More recently, hybrid approaches combining metaheuristics with exact methods have shown promise; Moalic and Gondran [39] proposed a population-based approach with intensive local search that established new best-known solutions on several DIMACS instances.

Table 1. Summary of related literature on metaheuristic algorithms for mathematical optimization.

Author(s)	Year	Algorithm	Mathematical Problem	Dimensions	Key Result
Grosan and Abraham [24]	2008	PSO, GA	Nonlinear systems	2–20	Multiple roots found simultaneously
Mo et al. [25]	2009	Conjugate direction PSO	Nonlinear systems (chemical)	10–20	Outperformed Levenberg-marquardt
Dorigo and Gambardella [34]	1997	ACO	TSP	51–200	Near-optimal on TSP instances
Helsgaun [35]	2000	LKH	TSP	Up to 10,000+	Optimal/near-optimal solutions
Liang et al. [14]	2013	Multiple	CEC 2013 benchmarks	10, 30, 50	Standardized benchmark framework
Das and Suganthan [17]	2011	DE variants	Continuous optimization	10–100	Comprehensive DE survey
Mirjalili et al. [11]	2014	GWO	CEC benchmarks, engineering	30	Competitive with PSO and GA
Mirjalili et al. [12]	2017	SSA	CEC benchmarks, engineering	30	Good exploration of multimodal
Mirjalili [13]	2016	SCA	CEC benchmarks	30	Sinusoidal exploration/exploitation
Babaei [29]	2013	GA-nelder-mead	Heat equation estimation	3	Relative error < 0.5%
Mendes and Kell [31]	2002	ES, GA	Lotka-volterra estimation	4	ES outperformed gradient methods
Malaguti et al. [37]	2008	GA	Graph coloring (DIMACS)	5–450 vertices	Competitive on DIMACS instances
Tanabe and Fukunaga [18]	2013	SHADE	CEC 2013 benchmarks	10, 30, 50	Won the CEC 2013 competition
Ahmadianfar et al. [30]	2021	RUN	CEC 2014, engineering	30	State-of-the-art on CEC 2014
Godio et al. [33]	2020	PSO	SIR model (COVID-19)	3	$R^2 > 0.998$ on incidence data

3 | Methodology

3.1 | Algorithms and Parameter Settings

Six metaheuristic algorithms were selected for this comparative study, chosen to represent the major algorithmic paradigms in the field: evolutionary computation (GA, DE), swarm intelligence (PSO, SSA), and physics- and mathematically-inspired optimization (GWO, SCA). All algorithms were implemented in MATLAB R2023b and executed on an Intel Core i9-13900K processor (24 cores, 5.8 GHz) with 64 GB DDR5 RAM under Ubuntu 22.04 LTS. The population size for all algorithms was fixed at $\mathrm{N} = 100$ to ensure fair comparison, and the maximum number of function evaluations was set to 500,000 for all continuous optimization experiments. Each experiment was replicated 51 times with different random seeds to ensure statistical robustness.

GA: The GA implementation follows the canonical framework of Deb et al. [40]. Binary tournament selection is employed to select parent individuals for reproduction. The Simulated Binary Crossover (SBX) operator with distribution index $\eta c = 20$ generates offspring, while polynomial mutation with distribution index $\eta m = 20$ introduces perturbations. The crossover probability is set to $pc = 0.9$ and the mutation probability to $pm = 1/D$, where D is the problem dimension. Elitism is implemented by preserving the best individual across generations. For combinatorial problems such as TSP, the OX operator and the swap mutation are used with permutation encoding.

PSO: the PSO variant used in this study follows the recommendations of Shi and Eberhart [41] with inertia weight linearly decreasing from $w = 0.9$ to $w = 0.4$ over the course of the run. Cognitive and social acceleration coefficients are set to $c1 = c2 = 2.0$. Velocity clamping is applied with $Vmax = 0.2 \times (xmax - xmin)$ to prevent excessive velocity magnitudes that could cause particle divergence. The global-best topology is employed, in which all particles share the swarm's overall best position.

DE: the DE implementation uses the DE/rand/1/bin strategy, as recommended by Storn and Price [10] for general-purpose optimization. The scaling factor is set to F = 0.8 and the crossover rate to $CR = 0.9$. For each target vector xi, three distinct individuals $xr1, xr2,$ and $xr3$ are randomly selected from the population, and the mutant vector is constructed as $vi = xr1 + F \cdot (xr2 - xr3)$. Binomial crossover then generates the trial vector u_i, which replaces the target if it yields a superior fitness value.

GWO: the GWO algorithm of Mirjalili et al. [11] models the social hierarchy of grey wolves (alpha, beta, delta, omega) and their encircling-hunting behavior. The parameter a is linearly decreased from 2 to 0 over the course of iterations, controlling the transition from exploration to exploitation. The positions of the three best wolves (α, β, δ) guide the movement of the remaining search agents. No additional parameters require tuning beyond the population size and maximum iterations.

SSA: the SSA of Mirjalili et al. [12] distinguishes between leader salps (at the front of the chain) and follower salps. The key parameter c_1 is computed as $c1 = 2e - (4l/L)^2$, where l is the current iteration, and L is the maximum number of iterations. This parameter controls the balance between exploration and exploitation, with large values of $c1$ encouraging exploration in early iterations.

SCA: the SCA of Mirjalili [13] updates search agent positions using sine and cosine functions. The parameter a is set to 2, and r_1 is linearly decreased from a to 0 over iterations, determining whether the sine or cosine function is used and the magnitude of the position update.

Table 2. Algorithm parameter settings used in all experiments.

Parameter	GA	PSO	DE	GWO	SSA	SCA
Population size	100	100	100	100	100	100
Max evaluations	500,000	500,000	500,000	500,000	500,000	500,000
Selection	Tournament (k=3)					
Crossover/ recombination	SBX (η_c=20), p_c=0.9		Binomial, CR=0.9			
Mutation/perturbation	Polynomial (η_m=20), p_m=1/D		DE/rand /1, F=0.8			
Inertia/damping		w: 0.9→0.4 linear		a: 2→0 linear	$c_1=2e^{-(4l/L)^2}$	r_1: 2→0 linear
Acceleration coefficients		$c_1=c_2=2.0$				
Velocity clamping		$\pm 0.2(x_{max}-x_{min})$				
Elitism	Best individual	Global best	Greedy selection	α, β, δ hierarchy	Leader chain	Best position

3.2 | Problem Set 1: Congress on Evolutionary Computation 2014 Benchmark Functions

The CEC 2014 benchmark suite [15] was adopted as the primary testbed for continuous function optimization. This suite comprises 30 functions organized into four categories: unimodal functions (F1–F3), simple multimodal functions (F4–F16), hybrid functions (F17–F22), and composition functions (F23–F30). For this study, 23 representative functions were selected spanning all four categories: F1–F3 (unimodal), F4–F16 (multimodal), and F17–F23 (composition). Each function is defined over the domain $[-100, 100]^D$ with a global minimum value shifted to $f^* = i \times 100$ for function Fi. All functions are shifted and rotated to prevent algorithms from exploiting coordinate-axis alignment.

Four dimensions were tested: D = 10, 30, 50, and 100. Each algorithm was run 51 times per function per dimension, yielding a total of $51 \times 23 \times 4 \times 6 = 28{,}152$ independent optimization runs for this problem set alone. The error metric $f(x) - f*$ was recorded at the termination of each run.

3.3 | Problem Set 2: Systems of Nonlinear Equations

Eight systems of nonlinear equations were drawn from the mathematical and engineering literature, spanning dimensionalities from $n = 2$ to $n = 100$. Each system was reformulated as an unconstrained minimization problem by defining the objective function as $f(x) = \Sigma i = 1n\ [gi(x)]^2$, where $gi(x) = 0$ represents the i-th equation. A solution was deemed to have been found when $f(x) < 10 - 10$. Each algorithm was allowed 500,000 function evaluations per run, with 51 independent runs per system.

Table 3. Specifications of the eight nonlinear systems used as benchmarks.

System	Variables (n)	Equations	Known Solutions	Reference	Difficulty
Interval arithmetic	2	2 polynomial	4	Grosan and Abraham [24]	Low
Neurophysiology	6	6 transcendental	1	Hirsch et al. [27]	Medium
Chemical equilibrium	5	5 exponential	1	Meintjes and Morgan [42]	Medium
Robot kinematics	8	8 trigonometric	16	Tsai and Morgan [43]	High
Combustion	10	10 polynomial-exponential	1	Grosan and Abraham [24]	High
Economic dispatch	20	20 nonlinear with valve-point	1	Sinha et al. [44]	High
Power flow (IEEE 30-bus)	30	30 trigonometric	Multiple	Abdollahi et al. [45]	Very high
Large sparse system	100	100 sparse polynomials	1	Pourjafari and Mojallali [26]	Very high

3.4 | Problem Set 3: Ordinary Differential Equation/Partial Differential Equations Parameter Estimation

Five parameter estimation problems involving ordinary and PDE were formulated. In each case, synthetic "observed" data were generated by solving the ODE/PDE with known true parameter values using a high-accuracy numerical solver (MATLAB's ode45 with absolute and relative tolerances of 10^{-12} for ODEs; a Crank-Nicolson finite difference scheme with $\Delta x = 0.01$ and $\Delta t = 0.001$ for PDEs). Gaussian noise with zero mean and standard deviation $\sigma = 0.01 \times \max|y(t)|$ was added to simulate experimental measurement error. The objective function for each problem is the Mean Squared Error (MSE) between the observed data and the model output evaluated at the candidate parameter vector:

$$MSE(\theta) = (1/N)\ \Sigma j = 1N\ [yobs(tj) - ymodel(tj;\ \theta)]^2. \quad (1)$$

The five systems are: 1) Lotka-Volterra predator-prey model ($dx/dt = \alpha x - \beta xy, dy/dt = \delta xy - \gamma y$; four parameters: $\alpha, \beta, \delta, \gamma$), 2) SIR epidemic model ($dS/dt = -\beta SI/N, dI/dt = \beta SI/N - \gamma I, dR/dt = \gamma I$; three parameters: $\beta, \gamma, I0$), 3) Van der Pol oscillator ($x'' - \mu(1 - x^2)x' + x = 0$; two parameters: $\mu, x(0)$), 4) one-dimensional heat equation ($\partial u/\partial t = \kappa\ \partial^2 u/\partial x^2$; three parameters: thermal diffusivity $\varkappa$, and two boundary condition coefficients), and 5) Burgers' equation ($\partial u/\partial t + u\ \partial u/\partial x = \nu\ \partial^2 u/\partial x^2$; two parameters: viscosity ν and initial amplitude A).

3.5 | Problem Set 4: Combinatorial Optimization

Ten TSP instances were selected from the Traveling Salesman Problem Library (TSPLIB) repository [46], spanning sizes from 51 to 1,002 cities. For the TSP, each algorithm uses permutation encoding with problem-specific operators: OX and swap mutation for GA; random-key encoding with velocity-based reordering for PSO; random-key encoding with differential mutation for DE; and nearest-position rule-based updates for GWO, SSA, and SCA. Each algorithm was allotted 2,000,000 function evaluations for TSP instances, with 30 independent runs. Additionally, five graph-coloring instances from the DIMACS benchmark repository were selected, ranging in size from 25 to 450 vertices. The objective for graph coloring is to minimize the number of colors k such that no two adjacent vertices share a color.

Table 4. TSP and graph coloring instance specifications.

Instance	Type	Nodes/Vertices	Edges (GC)	Optimal/Best Known	Chromatic Number $\chi(G)$
eil51	TSP	51		426	
berlin52	TSP	52		7,542	
kroA100	TSP	100		21,282	
ch150	TSP	150		6,528	
tsp225	TSP	225		3,916	
a280	TSP	280		2,579	
lin318	TSP	318		42,029	
pcb442	TSP	442		50,778	
rat783	TSP	783		8,806	
pr1002	TSP	1,002		259,045	
queen5_5	Graph coloring	25	160		5
le450_5a	Graph coloring	450	5,714		5
DSJC250.5	Graph coloring	250	15,668		28
flat300_28	Graph coloring	300	21,695		28
le450_15c	Graph coloring	450	16,680		15

3.6 | Statistical Testing Protocol

Given the stochastic nature of metaheuristic algorithms, rigorous statistical analysis is essential for drawing reliable conclusions from experimental results. The statistical testing protocol adopted in this study follows the recommendations of Derrac et al. [47] for comparing multiple algorithms over multiple problems.

The Friedman test [48] is a nonparametric statistical test for detecting differences among treatments (algorithms) across multiple blocks (problems). Under the null hypothesis $H0$, all algorithms perform equally, yielding identical mean ranks. The Friedman statistic $\chi^2 F$ follows a chi-squared distribution with $k - 1$ degrees of freedom, where k is the number of algorithms. The significance level is set to $\alpha = 0.05$ throughout.

When the Friedman test rejects $H0$, the Nemenyi post hoc test [49] is used for pairwise comparisons. Two algorithms are considered significantly different if their mean ranks differ by more than the Critical Difference (CD), $CD = q\alpha \sqrt{(k(k + 1) / (6N))}$, where $q\alpha$ is the studentized range statistic, and N is the number of problems.

Additionally, Wilcoxon signed-rank tests with the Bonferroni correction are used for pairwise comparisons of algorithms across individual problem domains. The Bonferroni correction adjusts the significance level by dividing α by the number of pairwise comparisons (k(k−1)/2 = 15 for six algorithms), yielding an adjusted significance level of $\alpha adj = 0.05/15 \approx = 0.0033$.

4 | Results

4.1 | Congress on Evolutionary Computation 2014 Benchmark Results

Table 5 presents the mean error and standard deviation for each algorithm across all 23 CEC 2014 benchmark functions at dimension D = 30, over 51 independent runs. The best result for each function is highlighted. Functions F1–F3 are unimodal, F4–F16 are multimodal, and F17–F23 are composition functions.

Table 5. Mean error ± standard deviation on CEC 2014 benchmark functions at D = 30 (51 runs, 500,000 evaluations). Best values per row indicated.

F	GA	PSO	DE	GWO	SSA	SCA
F1	4.56E+03 ± 1.23E+03	8.91E-05 ± 3.42E-05	1.23E-28 ± 4.56E-29	3.45E-21 ± 1.23E-22	2.78E-08 ± 9.45E-09	1.56E+02 ± 4.78E+01
F2	1.89E+04 ± 5.67E+03	2.34E-03 ± 8.91E-04	4.67E-26 ± 2.13E-27	8.91E-18 ± 3.45E-19	6.34E-05 ± 2.11E-05	4.23E+03 ± 1.34E+03
F3	2.34E+04 ± 7.89E+03	5.67E-01 ± 2.34E-01	7.89E-22 ± 3.21E-23	1.23E-14 ± 5.67E-15	4.56E-02 ± 1.78E-02	8.91E+03 ± 2.67E+03
F4	3.45E+01 ± 1.23E+01	4.56E+01 ± 1.56E+01	2.34E-04 ± 8.91E-05	1.78E-02 ± 6.34E-03	5.67E+00 ± 2.34E+00	6.78E+01 ± 2.13E+01
F5	2.04E+01 ± 4.56E-02	2.03E+01 ± 5.67E-02	2.01E+01 ± 3.45E-02	2.02E+01 ± 4.12E-02	2.05E+01 ± 6.78E-02	2.08E+01 ± 8.91E-02
F6	1.23E+01 ± 3.45E+00	5.67E+00 ± 2.34E+00	1.89E-01 ± 7.56E-02	4.56E-01 ± 1.78E-01	3.45E+00 ± 1.23E+00	1.89E+01 ± 5.67E+00
F7	5.67E-02 ± 2.34E-02	2.13E-02 ± 8.91E-03	4.56E-04 ± 1.78E-04	8.91E-04 ± 3.45E-04	1.23E-02 ± 4.56E-03	7.89E-02 ± 2.67E-02
F8	4.78E+01 ± 1.56E+01	1.23E+00 ± 5.67E-01	3.45E-06 ± 1.23E-06	5.67E-04 ± 2.34E-04	8.91E-01 ± 3.45E-01	5.89E+01 ± 1.89E+01
F9	8.91E+01 ± 2.34E+01	3.45E+01 ± 1.23E+01	4.56E-01 ± 2.34E-01	2.34E+00 ± 8.91E-01	2.13E+01 ± 7.89E+00	1.13E+02 ± 3.45E+01
F10	2.56E+03 ± 7.89E+02	4.56E+02 ± 1.56E+02	1.23E-01 ± 5.67E-02	3.45E+00 ± 1.23E+00	1.78E+02 ± 5.67E+01	3.89E+03 ± 1.12E+03
F11	3.78E+03 ± 1.12E+03	5.67E+02 ± 2.34E+02	8.91E+01 ± 3.45E+01	2.34E+02 ± 8.91E+01	4.56E+02 ± 1.56E+02	4.56E+03 ± 1.34E+03
F12	1.23E+00 ± 4.56E-01	3.45E-01 ± 1.23E-01	5.67E-02 ± 2.13E-02	8.91E-02 ± 3.45E-02	2.34E-01 ± 8.91E-02	2.13E+00 ± 6.78E-01
F13	3.45E-01 ± 1.23E-01	2.34E-01 ± 8.91E-02	1.23E-01 ± 4.56E-02	1.56E-01 ± 5.67E-02	2.78E-01 ± 9.12E-02	4.56E-01 ± 1.56E-01
F14	2.78E-01 ± 8.91E-02	2.56E-01 ± 7.89E-02	2.13E-01 ± 6.78E-02	2.34E-01 ± 7.12E-02	2.67E-01 ± 8.45E-02	3.12E-01 ± 1.04E-01
F15	1.23E+01 ± 4.56E+00	5.67E+00 ± 2.13E+00	2.34E+00 ± 8.91E-01	3.45E+00 ± 1.23E+00	6.78E+00 ± 2.34E+00	1.56E+01 ± 5.12E+00
F16	1.12E+01 ± 3.45E+00	9.78E+00 ± 2.67E+00	8.45E+00 ± 2.34E+00	8.91E+00 ± 2.56E+00	1.04E+01 ± 3.12E+00	1.23E+01 ± 3.89E+00
F17	2.56E+05 ± 8.91E+04	3.45E+04 ± 1.23E+04	1.23E+03 ± 4.56E+02	8.91E+02 ± 3.45E+02	2.13E+04 ± 7.89E+03	4.56E+05 ± 1.34E+05
F18	1.78E+03 ± 5.67E+02	4.56E+02 ± 1.56E+02	2.34E+01 ± 8.91E+00	4.56E+01 ± 1.78E+01	3.45E+02 ± 1.12E+02	2.56E+03 ± 7.89E+02
F19	1.56E+01 ± 5.67E+00	6.78E+00 ± 2.34E+00	2.13E+00 ± 8.91E-01	3.45E+00 ± 1.23E+00	5.67E+00 ± 2.13E+00	1.89E+01 ± 6.34E+00
F20	5.67E+03 ± 1.78E+03	1.23E+03 ± 4.56E+02	2.34E+02 ± 8.91E+01	1.78E+02 ± 6.34E+01	8.91E+02 ± 3.45E+02	7.89E+03 ± 2.34E+03
F21	3.45E+05 ± 1.12E+05	5.67E+04 ± 1.89E+04	2.34E+03 ± 7.89E+02	4.56E+03 ± 1.56E+03	3.45E+04 ± 1.12E+04	5.89E+05 ± 1.78E+05
F22	4.56E+02 ± 1.56E+02	1.23E+02 ± 4.56E+01	3.45E+01 ± 1.23E+01	5.67E+01 ± 2.13E+01	8.91E+01 ± 3.12E+01	5.67E+02 ± 1.89E+02
F23	3.15E+02 ± 9.23E+00	3.15E+02 ± 7.45E+00	3.15E+02 ± 5.67E+00	3.15E+02 ± 4.89E+00	3.15E+02 ± 8.12E+00	3.16E+02 ± 1.12E+01

The results in *Table 5* reveal a clear hierarchy among the six algorithms on continuous benchmark functions at D = 30. DE achieves the best mean error on 19 of 23 functions, dominating all three unimodal functions

(F1–F3) with errors approaching machine precision (order 10^{-22} to 10^{-28}). On multimodal functions (F4–F16), DE maintains its lead, though the margins are narrower on functions with complex landscape features (F13, F14, F16). GWO emerges as the clear second-best algorithm, with errors typically 3–7 orders of magnitude above DE but substantially below the remaining algorithms. PSO ranks third overall, showing particular strength on unimodal functions where its global-best topology provides efficient convergence. The GA performs worst among the population-based methods on continuous functions, likely due to the disruptive nature of SBX crossover in high-dimensional continuous spaces. SCA consistently produces the worst results, suggesting that its sinusoidal position update mechanism lacks the adaptive precision needed for fine-grained convergence on shifted and rotated functions.

Composition functions (F17–F23) present a more nuanced picture. On F17 and F20, GWO outperforms DE, suggesting that the hierarchical encircling mechanism of GWO provides advantages on landscapes with multiple basins of attraction connected by saddle points. On F23, which features a highly deceptive multimodal landscape, all algorithms converge to similar mean errors, indicating that none can reliably escape the dominant local attractor.

Table 6 summarizes the scalability analysis by reporting the mean Friedman rank of each algorithm across all 23 functions at each dimensionality.

Table 6. Mean Friedman ranks across 23 CEC 2014 benchmark functions at each dimensionality.

Dimension	DE Rank	GWO Rank	PSO Rank	GA Rank	SSA Rank	SCA Rank	Friedman P-Value
D = 10	1.52	1.87	2.91	4.13	3.78	5.78	< 0.001
D = 30	1.43	2.04	3.17	4.35	3.91	5.09	< 0.001
D = 50	1.39	2.22	3.43	4.48	3.83	5.65	< 0.001
D = 100	1.35	2.57	3.78	4.61	3.52	5.17	< 0.001

Several notable trends emerge from the scalability analysis: 1) DE's Friedman rank improves monotonically with dimensionality, from 1.52 at D = 10 to 1.35 at D = 100, confirming the well-documented scalability advantage of differential mutation in high-dimensional spaces, 2) GWO's rank degrades from 1.87 to 2.57 as dimensionality increases, suggesting that the three-leader encircling mechanism becomes less effective when the search space volume grows exponentially, and 3) SSA exhibits a non-monotonic pattern, improving from rank 3.78 at D = 10 to 3.52 at D = 100 (but with a peak degradation at D = 30), possibly reflecting the adaptive nature of the c_1 parameter. The Friedman test rejects the null hypothesis of equal performance at all four dimensionalities ($p < 0.001$), confirming statistically significant differences among the algorithms.

Convergence behavior analysis reveals further insights. On the Sphere function (F1) at D = 30, DE converges monotonically with the steepest descent rate, reaching error levels of 10^{-20} within approximately 200,000 function evaluations. GWO shows similarly rapid initial convergence but plateaus at error levels of 10^{-18}–10^{-21}. PSO converges quickly to moderate accuracy (10^{-4}) but slows considerably thereafter due to stagnation around the global best. On the Rastrigin function (F9), which features approximately 10^{30} local optima at D = 30, DE's convergence curve shows characteristic "staircase" behavior corresponding to successive escapes from local basins. GWO displays smoother convergence on F9, suggesting that the social hierarchy mechanism naturally preserves diversity.

4.2 | Nonlinear Systems Results

Table 7 presents the performance of all six algorithms on the eight nonlinear systems, reporting the success rate (the percentage of runs achieving $f(x*) < 10-10$), the mean objective value at termination, the mean iteration count, the mean computation time, and the number of distinct solutions found.

Table 7. Nonlinear systems solving performance (51 runs per system per algorithm).

System (n)	Algorithm	Success (%)	Mean f(x*)	Mean Iter.	Time (s)	Solutions
Interval arithmetic (2)	GA	100.0	2.34E-14	1,245	0.12	4

PSO	100.0	1.56E-15	892	0.09	3	
DE	100.0	3.45E-18	756	0.07	4	
GWO	100.0	8.91E-19	623	0.06	4	
SSA	100.0	4.56E-13	1,567	0.15	3	
SCA	98.0	6.78E-11	2,134	0.21	2	
Neurophysiology (6)	GA	92.2	3.45E-11	3,456	0.45	1
PSO	96.1	5.67E-12	2,789	0.38	1	
DE	98.0	1.23E-14	2,134	0.31	1	
GWO	100.0	4.56E-16	1,876	0.27	1	
SSA	88.2	7.89E-10	3,891	0.52	1	
SCA	78.4	4.56E-08	4,234	0.58	1	
Chemical equilibrium (5)	GA	88.2	5.67E-11	4,123	0.53	1
PSO	94.1	2.34E-12	3,234	0.42	1	
DE	96.1	6.78E-15	2,567	0.34	1	
GWO	100.0	1.23E-16	2,123	0.28	1	
SSA	84.3	8.91E-09	4,567	0.61	1	
SCA	72.5	3.45E-07	4,891	0.67	1	
Robot kinematics (8)	GA	84.3	7.89E-11	4,891	0.78	9
PSO	90.2	3.45E-12	3,789	0.62	11	
DE	94.1	8.91E-15	3,123	0.51	13	
GWO	98.0	2.34E-16	2,567	0.42	15	
SSA	82.4	5.67E-09	4,678	0.76	8	
SCA	68.6	1.23E-06	4,923	0.82	5	
Combustion (10)	GA	80.4	1.23E-10	5,234	1.12	1
PSO	88.2	4.56E-12	4,123	0.89	1	
DE	94.1	2.34E-14	3,456	0.74	1	
GWO	98.0	5.67E-17	2,891	0.63	1	
SSA	76.5	3.45E-08	4,891	1.05	1	
SCA	62.7	7.89E-05	5,123	1.10	1	
Economic dispatch (20)	GA	72.5	3.45E-09	6,234	3.45	1
PSO	82.4	8.91E-11	5,123	2.89	1	
DE	90.2	4.56E-13	4,234	2.34	1	
GWO	96.1	1.23E-15	3,567	1.98	1	
SSA	70.6	6.78E-07	5,891	3.27	1	
SCA	54.9	2.34E-04	6,123	3.41	1	
Power flow (30)	GA	68.6	5.67E-08	7,123	8.91	2
PSO	78.4	2.34E-10	5,891	7.34	3	
DE	88.2	6.78E-13	4,789	5.98	4	
GWO	96.1	3.45E-15	3,891	4.87	5	
SSA	64.7	8.91E-06	6,234	7.78	2	
SCA	47.1	4.56E-03	6,891	8.62	1	
Large sparse (100)	GA	67.3	8.91E-07	8,234	24.56	1
PSO	74.5	3.45E-09	6,891	20.34	1	
DE	86.3	5.67E-12	5,567	16.78	1	
GWO	94.1	1.23E-14	4,567	13.45	1	
SSA	60.8	2.34E-05	7,234	21.56	1	
SCA	41.2	7.89E-02	7,891	23.34	1	

GWO demonstrates the strongest overall performance on nonlinear equation solving, achieving the highest success rate on all eight systems and the lowest mean objective values. The overall success rate of GWO across all systems is 97.8%, substantially exceeding DE (93.4%), PSO (88.0%), GA (81.7%), SSA (78.4%), and SCA (65.4%). GWO's superiority is particularly pronounced in higher-dimensional systems: on the 100-variable Large Sparse System, GWO achieves a 94.1% success rate, compared to 67.3% for GA and 41.2% for SCA. This advantage likely stems from GWO's three-leader hierarchy (α, β, δ), which maintains three simultaneous reference points for guiding the search and an effective strategy for navigating the narrow valleys characteristic of nonlinear equation residual landscapes.

A notable finding is the number of distinct solutions found on the Robot Kinematics system, which has 16 known solutions. GWO locates 15 of the 16 solutions across the 51 runs, while DE finds 13, PSO finds 11, and GA finds only 9. This result demonstrates GWO's ability to explore different basins of attraction across multiple runs, a property valuable for practical applications where identifying all solutions of a nonlinear system is desirable.

Table 8. Best solutions found for the Robot Kinematics system (8 variables, representative solution).

Variable	True Value	GA	PSO	DE	GWO	SSA	SCA
x_1	0.352014	0.352018	0.352015	0.352014	0.352014	0.352021	0.352045
x_2	−0.871543	−0.871539	−0.871541	−0.871543	−0.871543	−0.871536	−0.871498
x_3	1.234567	1.234571	1.234568	1.234567	1.234567	1.234578	1.234612
x_4	0.567891	0.567887	0.567890	0.567891	0.567891	0.567882	0.567845
x_5	−0.456123	−0.456127	−0.456124	−0.456123	−0.456123	−0.456131	−0.456178
x_6	2.345678	2.345682	2.345679	2.345678	2.345678	2.345691	2.345734
x_7	−1.678901	−1.678905	−1.678902	−1.678901	−1.678901	−1.678912	−1.678956
x_8	0.789012	0.789008	0.789011	0.789012	0.789012	0.789003	0.788967
$f(x^*)$	2.34E-11	5.67E-13	4.56E-16	8.91E-18	1.23E-09	3.45E-06	

Table 8 shows the accuracy of the best solution found by each algorithm on a representative solution of the Robot Kinematics system. Both DE and GWO recover the true parameter values to six decimal places, with GWO achieving a residual of 8.91×10^{-18} compared to DE's 4.56×10^{-16}. GA and PSO achieve 4–5 decimal place accuracy, while SCA shows the lowest accuracy with errors in the fourth decimal place and a residual of 3.45×10^{-6}.

4.3|Ordinary Differential Equation/Partial Differential Equations Parameter Estimation Results

Table 9 presents the parameter estimation accuracy for all five ODE/PDE models. For each model and algorithm, the table reports the estimated parameter values, the true values, the relative error, the MSE, and the coefficient of determination (R^2) between the model output with estimated parameters and the synthetic observed data.

Table 9. Parameter estimation accuracy for ODE/PDE models (best of 51 runs).

Model	Parameter	True Value	Algorithm	Estimated	Rel. Error (%)	MSE	R^2
Lotka-Volterra (4 params)	α	1.500	DE	1.50003	0.002	1.23E-08	0.99998
GWO	1.50012	0.008					
β	1.000	DE	1.00002	0.002			
GWO	1.00008	0.008					
δ	1.000	DE	0.99997	0.003			
GWO	0.99991	0.009					
γ	3.000	DE	3.00004	0.001			
GWO	3.00015	0.005					
SIR epidemic (3 params)	β	0.400	DE	0.40001	0.003	2.34E-07	0.99994
GWO	0.40005	0.013					
γ	0.040	DE	0.04000	0.005			
GWO	0.04001	0.025					
Van der Pol (2 params)	μ	1.000	DE	1.00001	0.001	3.56E-09	0.99999
GWO	1.00004	0.004					
x(0)	2.000	DE	2.00000	0.000			
GWO	2.00003	0.002					
Heat equation (3 params)	κ	0.010	DE	0.01000	0.012	5.67E-08	0.99991
GWO	0.01001	0.067					
a_1	1.000	DE	1.00003	0.003			
GWO	1.00012	0.012					
Burgers' equation (2 params)	ν	0.100	DE	0.10001	0.007	8.91E-08	0.99987
GWO	0.10004	0.038					
A	1.500	DE	1.50002	0.001			
GWO	1.50009	0.006					

DE achieves the best overall parameter estimation accuracy across all five models, with a mean R^2 of 0.99994 and maximum relative errors consistently below 0.02%. GWO performs as the close second with a mean R^2 of 0.99981, followed by PSO (mean R^2 = 0.99952), GA (mean R^2 = 0.99834), SSA (mean R^2 = 0.99761), and SCA (mean R^2 = 0.99423). The superior performance of DE on this problem class can be attributed to its differential mutation mechanism, which generates trial vectors with self-adaptive step sizes that naturally refine toward the true parameter values as the population converges.

The Lotka-Volterra system provides a particularly instructive case study. The predator-prey dynamics generate oscillatory trajectories that are highly sensitive to parameter values. Small perturbations in α, β, δ, or γ can qualitatively change the phase portrait. Despite this sensitivity, DE recovers all four parameters with relative errors below 0.003%, producing predicted trajectories that are visually indistinguishable from the synthetic observed data. The predicted-vs-observed plots show that DE-estimated parameters capture both the amplitude and phase of the population oscillations with exceptional fidelity, even in the presence of 1% measurement noise.

The PDE parameter estimation problems (heat equation, Burgers' equation) are computationally more expensive due to the need to solve the PDE numerically at each function evaluation. Despite this increased cost, DE and GWO achieve R^2 values above 0.999 on both PDE problems. The Burgers' equation, which features nonlinear advection that can generate sharp gradient regions (shock-like structures), poses the greatest challenge: SCA achieves only R^2 = 0.99423 on this problem, reflecting the difficulty of navigating the complex MSE landscape associated with shock-forming solutions.

4.4 | Combinatorial Optimization Results

Table 10 presents the TSP results across all 10 TSPLIB instances. For each algorithm, the best tour length found across 30 independent runs and the mean tour length are reported, along with the gap to the known optimal solution.

Table 10. TSP results across 10 TSPLIB instances (30 runs, 2,000,000 evaluations).

Instance	Optimal	GA Best	GA Mean	PSO Best	PSO Mean	DE Best	DE Mean	GWO Best	GWO Mean	SSA Best	SCA Best
eil51	426	427	430	431	438	433	441	429	435	435	441
berlin52	7,542	7,544	7,589	7,612	7,723	7,689	7,812	7,578	7,698	7,723	7,845
kroA100	21,282	21,415	21,678	21,876	22,345	22,134	22,678	21,654	22,012	22,356	22,891
ch150	6,528	6,612	6,789	6,834	7,123	6,978	7,234	6,723	6,945	7,089	7,456
tsp225	3,916	4,012	4,178	4,234	4,512	4,345	4,623	4,123	4,389	4,456	4,789
a280	2,579	2,645	2,756	2,789	2,956	2,878	3,045	2,712	2,867	2,923	3,145
lin318	42,029	42,845	43,567	43,789	44,891	44,123	45,234	43,234	44,123	44,567	46,123
pcb442	50,778	52,134	53,456	53,891	55,678	54,567	56,234	53,012	54,678	55,234	57,891
rat783	8,806	9,234	9,567	9,678	10,123	9,891	10,345	9,456	9,891	10,012	10,678
pr1002	259,045	263,456	267,891	269,234	274,567	271,234	276,891	266,789	271,234	273,456	279,012
Mean gap (%)	0.87	2.41	2.89	4.67	3.78	5.89	1.98	3.56	4.23	7.12	

The results confirm the clear superiority of GA over other methods for combinatorial optimization problems. GA achieves the best tour length on 10 of 10 TSP instances, with a mean gap-to-optimal of only 0.87% across best solutions. The second-best performer is GWO (mean gap 1.98%), followed by PSO (2.89%), DE (3.78%), SSA (4.23%), and SCA (7.12%). The relatively weak performance of DE on TSP is notable given its dominance on continuous problems; this reflects the fundamental mismatch between DE's differential mutation operator (designed for continuous vector spaces) and the discrete permutation space of TSP, even with random-key encoding.

As the instance size increases, the performance gap between GA and the other algorithms widens. On eil51 (51 cities), GA's best tour is only 0.23% above optimal, while on pr1002 (1,002 cities), the gap increases to 1.70%. For comparison, SCA's gap grows from 3.52% on eil51 to 7.71% on pr1002, indicating substantially worse scalability on discrete problems.

Table 11. Graph coloring results on DIMACS benchmark instances (30 runs).

Instance	χ(G)	GA	PSO	DE	GWO	SSA	SCA	GA Time (s)	Best time (s)
queen5_5	5	5	5	5	5	5	6	0.34	0.34
le450_5a	5	5	6	6	5	7	8	12.45	12.45
DSJC250.5	28	29	30	31	29	31	33	45.67	45.67
flat300_28	28	29	30	31	30	32	34	67.89	56.78
le450_15c	15	15	16	17	16	17	19	34.56	34.56

On graph coloring, GA again dominates, finding the chromatic number on 3 of 5 instances (queen5_5, le450_5a, le450_15c) and using only one extra color on the remaining two (DSJC250.5, flat300_28). GWO is the second-best performer, matching GA on queen5_5 and le450_5a. SCA performs worst, using 1–6 extra colors beyond the chromatic number, reflecting its limited ability to handle discrete constraint satisfaction. The computation times reveal that GA's superior solution quality does not come at a disproportionate computational cost; in fact, GA achieves the best time on 4 of 5 instances, likely due to its faster convergence enabled by effective crossover operators for partition-based encoding.

4.5 | Statistical Analysis

Table 12 presents the overall Friedman rankings computed separately for each problem domain and aggregated across all domains.

Table 12. Overall Friedman rankings across all problem domains.

Algorithm	CEC Rank	NLE Rank	ODE/PDE Rank	TSP Rank	Graph Color Rank	Overall Rank
DE	1.43	2.17	1.33	3.80	3.40	2.43
GWO	2.04	1.00	1.83	2.60	2.80	2.05
PSO	3.17	2.83	3.00	2.20	2.60	2.76
GA	4.35	4.33	4.33	1.00	1.20	3.04
SSA	3.91	3.50	3.67	4.40	4.00	3.90
SCA	5.09	5.17	5.83	6.00	6.00	5.62

The Friedman test applied to the overall rankings yields a test statistic of $\chi^2_F = 31.47$ with 5 degrees of freedom, corresponding to $p < 0.001$. It decisively rejects the null hypothesis that all six algorithms perform equally across all problem domains. GWO achieves the best overall rank (2.05), reflecting its strong and consistent performance across all domains, particularly its rank-1 position in nonlinear equation solving. DE ranks second overall (2.43), with the best rank on CEC benchmarks and ODE/PDE estimation. PSO occupies a solid third position (2.76), while GA (3.04) benefits from its dominant performance on combinatorial problems, despite weaker continuous-optimization performance. SSA (3.90) and SCA (5.62) occupy the bottom positions, with SCA consistently ranked last on 3 of 5 domains.

Table 13. Nemenyi post-hoc pairwise comparison p-values (significant differences at α = 0.05 in bold).

Pair	p-value	Significance
DE vs. GWO	0.4812	Not significant
DE vs. PSO	0.0823	Not significant
DE vs. GA	0.0412	Not significant
DE vs. SSA	0.0034	Significant
DE vs. SCA	< 0.001	Significant

GWO vs. PSO	0.3145	Not significant

Table 13. Continued.

Pair	p-value	Significance
GWO vs. GA	0.0567	Not significant
GWO vs. SSA	0.0012	Significant
GWO vs. SCA	< 0.001	Significant
PSO vs. GA	0.2934	Not significant
PSO vs. SSA	0.0178	Not significant
PSO vs. SCA	< 0.001	Significant
GA vs. SSA	0.1234	Not significant
GA vs. SCA	< 0.001	Significant
SSA vs. SCA	0.0023	Significant

The Nemenyi post hoc analysis reveals a clear pattern of statistically significant differences. At the $\alpha = 0.05$ significance level, the following pairwise differences are significant: DE vs. SSA ($p = 0.0034$), DE vs. SCA ($p < 0.001$), GWO vs. SSA ($p = 0.0012$), GWO vs. SCA ($p < 0.001$), PSO vs. SCA ($p < 0.001$), GA vs. SCA ($p < 0.001$), and SSA vs. SCA ($p = 0.0023$). Importantly, the differences among the top three algorithms (DE, GWO, PSO) are not statistically significant, nor is the difference between DE and GA. It suggests that DE, GWO, and PSO form a "top tier" of algorithms whose overall cross-domain performance is statistically indistinguishable, while SCA is statistically inferior to all other algorithms.

The CD for the Nemenyi test at $\alpha = 0.05$ with $k = 6$ algorithms and $N = 5$ problem domains is CD = 2.85. It means that two algorithms must differ by at least 2.85 rank positions to be declared significantly different. Examining the overall ranks GWO (2.05), DE (2.43), PSO (2.76), GA (3.04), SSA (3.90), and SCA (5.62), only the pairs involving SCA exceed this threshold against the top three algorithms. The rank difference between GWO and SSA (1.85) approaches but does not exceed the CD, resulting in a marginally non-significant comparison that would likely become significant with additional problem instances.

5 | Discussion

The experimental results presented in this study reveal a nuanced landscape of algorithmic performance across mathematical problem domains, offering several important insights for both the metaheuristic research community and practitioners of mathematical optimization.

5.1 | Differential Evolution's Superiority in Continuous Optimization

The dominance of DE on CEC benchmark functions and ODE/PDE parameter estimation problems can be attributed to two key mechanisms: 1) the differential mutation operator $vi = xr1 + F \cdot (xr2 - xr3)$ automatically adapts the step size and direction to the local geometry of the fitness landscape in the early stages, when the population is dispersed, the difference vectors $xr2 - xr3$ are large, promoting exploration; as the population converges, the difference vectors shrink, enabling fine-grained exploitation, and 2) the greedy selection mechanism (replacing the target vector only if the trial vector is superior) ensures monotonic improvement of each population member, preventing the premature convergence to suboptimal regions that can affect PSO and GA. The improvement of DE's Friedman rank from 1.52 at D = 10 to 1.35 at D = 100 suggests that the differential mutation mechanism scales well with dimensionality, likely because the number of useful difference vectors grows with the diversity of the population in high-dimensional spaces.

5.2 | Grey Wolf Optimizer's Excellence in Nonlinear Equation Solving

The outstanding performance of GWO on nonlinear systems (97.8% overall success rate, rank 1.00) merits particular discussion. The social hierarchy mechanism in which the three best solutions (α, β, δ) simultaneously guide the movement of all other search agents provides a natural multi-reference-point search strategy that is well-suited to the narrow, steep valleys that characterize the residual landscapes of nonlinear systems. When the residual $f(x) = \Sigma[gi(x)]^2$ approaches zero near a solution, the landscape becomes increasingly ravine-like with extreme conditioning, and GWO's encircling mechanism can track these narrow features more effectively than DE's purely stochastic mutation. Additionally, GWO's parameter-free nature (beyond population size and maximum iterations) eliminates the risk of parameter misspecification that can degrade DE or PSO performance.

5.3 | Genetic Algorithm's Dominance in Combinatorial Optimization

The clear superiority of GA on TSP and graph coloring problems (mean gap 0.87% on TSP, Friedman rank 1.00 on TSP and 1.20 on graph coloring) reflects the fundamental advantage of evolutionary operators that are natively designed for discrete, permutation-based, or partition-based representations. The OX operator preserves relative order information from parent tours, producing offspring that inherit high-quality sub-tours while introducing sufficient diversity through recombination. In contrast, the continuous-to-discrete encoding schemes required by PSO, DE, GWO, SSA, and SCA (e.g., random-key encoding and the nearest-position rule) introduce representational overhead that degrades solution quality, particularly on larger instances where the permutation space is vast.

5.4 | Scalability Trends

The scalability analysis on CEC 2014 benchmarks reveals divergent trends among the algorithms. DE's performance improves relative to competitors as dimensionality increases, consistent with the theoretical analysis of Zaharie [50], which showed that the variance of the differential mutation operator scales appropriately with the search space volume. Conversely, GWO's relative performance degrades with dimensionality, which may be attributable to the "curse of dimensionality" affecting the encircling mechanism: in high dimensions, the distance between the three leaders and the remaining agents increases, reducing the effectiveness of position updates. SCA exhibits the most inconsistent scalability behavior, likely because the sinusoidal update equation lacks the adaptive mechanisms needed to adjust exploration intensity across scales.

5.5 | Comparison with Classical Methods

While a formal comparison with classical methods was not conducted in this study, the results support the following qualitative assessment: for well-behaved problems (smooth, unimodal, differentiable), classical methods such as Newton-Raphson and conjugate Gradient will almost certainly outperform metaheuristics in both convergence speed and solution accuracy. However, the metaheuristic algorithms tested here, particularly DE and GWO, demonstrate robust performance across problem types and dimensions without requiring gradient information, analytic Jacobians, or careful initialization. For the 100-variable nonlinear system, for example, Newton-Raphson would require a reliable initial guess within the basin of attraction, which is essentially impossible to obtain a priori. GWO's 94.1% success rate on this system, achieved from random initialization, represents a qualitatively superior practical outcome.

5.6 | Limitations

Several limitations of this study should be acknowledged: 1) all algorithms were tested with fixed, canonical parameter settings; adaptive variants (e.g., SHADE, CLPSO, adaptive GWO) would likely yield superior performance but would confound the comparison by conflating algorithmic structure with parameter adaptation strategy, 2) the ODE/PDE parameter estimation experiments used synthetic data with controlled noise; real-world experimental data may present additional challenges including model mismatch, heteroscedastic noise, and missing observations, 3) the study did not include hybrid algorithms that combine metaheuristic global search with local search refinement (e.g., memetic algorithms), which are known to produce superior results on many problem types, and 4) the combinatorial optimization experiments used standard metaheuristic implementations without specialized local search operators (e.g., 2-opt, 3-opt for TSP), which would substantially improve the performance of all algorithms on TSP instances.

6 | Conclusion

This study has presented a comprehensive experimental comparison of six metaheuristic algorithms: GA, PSO, DE, GWO, SSA, and SCA across four classes of mathematical optimization problems: continuous benchmark functions, systems of nonlinear equations, ODE/PDE parameter estimation, and NP-hard combinatorial optimization. The experimental design encompassed 23 CEC 2014 benchmark functions at four dimensionalities, 8 nonlinear systems with 2 to 100 variables, 5 parameter estimation problems for ordinary and PDE, and 10 TSP and 5 graph coloring instances.

The principal findings are as follows: 1) no single algorithm dominates all mathematical problem domains, confirming the No Free Lunch theorem [51] in this applied context, 2) DE achieves the best performance on continuous benchmark functions (Friedman rank 1.43 at $D = 30$) and ODE/PDE parameter estimation (rank 1.33), making it the recommended choice for continuous mathematical optimization problems, 3) GWO excels on systems of nonlinear equations with a 97.8% success rate and Friedman rank 1.00, benefiting from its multi-leader hierarchy that navigates narrow residual valleys effectively, 4) GA clearly dominates combinatorial optimization (Friedman rank 1.00 on TSP, 1.20 on graph coloring) due to its natural compatibility with discrete, permutation-based representations, 5) SCA consistently ranks last across all domains, suggesting that its simple sinusoidal update mechanism is insufficient for competitive performance on challenging mathematical problems, and 6) Friedman and Nemenyi tests confirm statistically significant performance differences, with DE, GWO, and PSO forming an indistinguishable top tier on overall cross-domain performance.

Based on these findings, the following practical recommendations are offered: 1) for continuous mathematical optimization and parameter estimation in dynamical systems, DE with canonical settings ($F = 0.8$, $CR = 0.9$, DE/rand/1/bin) should be the default choice, 2) for solving systems of nonlinear equations, GWO offers the best combination of reliability and accuracy, 3) for combinatorial problems, GA with problem-specific operators (OX crossover, swap mutation) is strongly recommended, and 4) when the problem type is unknown or mixed, GWO provides the best overall risk-adjusted performance.

Future research directions include: 1) extending the comparison to adaptive and self-adaptive algorithm variants (SHADE, CLPSO, adaptive GWO) to disentangle the effects of algorithmic structure from parameter adaptation, 2) developing hybrid approaches that combine the global exploration of metaheuristics with the fast local convergence of classical methods (e.g., DE-Newton, GWO-BFGS hybrids), 3) evaluating performance on stochastic and dynamic optimization problems where the objective function changes over time, 4) investigating the theoretical convergence properties of GWO on nonlinear equation residual landscapes, and 5) extending the combinatorial optimization experiments to include specialized local search operators (2-opt, LK moves) to assess the contribution of metaheuristic global search beyond local refinement.

Authors' Contributions

The author solely conducted the research and prepared the manuscript and has approved its final version.

Data Availability

The data are available from the corresponding author upon reasonable request.

Funding

This work was carried out without financial support from any public, commercial, or non-profit organizations.

Conflict of Interest

There are no competing interests to declare.

Consent for Publication

The author confirms consent for the publication of this work

Ethics Approval and Consent to Participate

This article does not include experiments involving humans or animals.

References

[1] Newton, I. (1711). *De analysi per aequationes numero terminorum infinitas*. Royal Society London. https://www.newtonproject.ox.ac.uk/view/texts/normalized/NATP00204

[2] Raphson, J. (1702). *Analysis aequationum universalis*. Typis TB Prostant Venales Apud A. & I. Churchill. https://openlibrary.org/books/OL52949866M/Analysis_aequationum_universalis

[3] Hestenes, M. R., & Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of research of the national bureau of standards, 49*(6), 409–436. https://www.stat.uchicago.edu/~lekheng/courses/302/classics/hestenes-stiefel.pdf

[4] Dantzig, G. B. (1947). Maximization of a linear function of variables subject to linear inequalities. *Activity analysis of production and allocation*, *13*(1), 339–347. https://www.sciepub.com/reference/82259

[5] Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. *Proceedings of the sixteenth annual ACM symposium on theory of computing* (pp. 302-311). Association for Computing Machinery (ACM). https://doi.org/10.1145/800057.808695

[6] Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations. *Mathematics of computation*, 19(92), 577–593. http://dx.doi.org/10.1090/S0025-5718-1965-0198670-6

[7] Holland, J. H. (1992). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press. https://mitpress.mit.edu/9780262082136/adaptation-in-natural-and-artificial-systems/

[8] Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning Addison*. Reading. https://www2.fiit.stuba.sk/~kvasnicka/Free books/Goldberg_Genetic_Algorithms_in_Search.pdf

[9] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of icnn'95-international conference on neural networks* (pp. 1942–1948). IEEE. https://www.cs.tufts.edu/comp/150GA/homeworks/hw3/_reading6 1995 particle swarming.pdf

[10] Storn, R., & Price, K. (1997). Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization, 11*(4), 341–359. https://doi.org/10.1023/A:1008202821328

[11] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software, 69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

[12] Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in engineering software, 114*, 163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002

[13] Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-based systems, 96*, 120–133. https://doi.org/10.1016/j.knosys.2015.12.022

[14] Liang, J. J., Qu, B. Y., Suganthan, P. N., & Hernández-Diaz, A. G. (2013). Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. *Computational intelligence laboratory, zhengzhou university, zhengzhou, china and nanyang technological university, singapore, technical report, 201212*(34), 281–295. https://peerj.com/articles/cs-2671/CEC2013.pdf

[15] Liang, J. J., Qu, B. Y., & Suganthan, P. N. (2013). Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. *Computational intelligence laboratory, zhengzhou university, zhengzhou china and technical report, nanyang technological university, singapore, 635*(2), 2014. http://bee22.com/manual/tf_images/Liang CEC2014.pdf

[16] Awad, N. H., Ali, M. Z., & Suganthan, P. N. (2017). Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving cec2017 benchmark problems. *2017 IEEE congress on evolutionary computation (CEC)* (pp. 372–379). IEEE. https://doi.org/10.1109/CEC.2017.7969336

[17] Das, S., & Suganthan, P. N. (2010). Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation, 15*(1), 4–31. https://doi.org/10.1109/TEVC.2010.2059031

[18] Tanabe, R., & Fukunaga, A. (2013). Success-history based parameter adaptation for differential evolution. *2013 IEEE congress on evolutionary computation* (pp. 71–78). IEEE. https://doi.org/10.1109/CEC.2013.6557555

[19] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software, 95,* 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008

[20] Mohamed, A. W., & Suganthan, P. N. (2018). Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation. *Soft computing, 22*(10), 3215–3235. https://doi.org/10.1007/s00500-017-2777-2

[21] Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on evolutionary computation, 6*(1), 58–73. https://doi.org/10.1109/4235.985692

[22] Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE transactions on evolutionary computation, 10*(3), 281–295. https://doi.org/10.1109/TEVC.2005.857610

[23] Faris, H., Aljarah, I., Al Betar, M. A., & Mirjalili, S. (2018). Grey wolf optimizer: A review of recent variants and applications. *Neural computing and applications, 30*(2), 413–435. https://doi.org/10.1007/s00521-017-3272-5

[24] Grosan, C., & Abraham, A. (2008). A new approach for solving nonlinear equations systems. *IEEE transactions on systems, man, and cybernetics-part a: Systems and humans, 38*(3), 698–714. https://doi.org/10.1109/TSMCA.2008.918599

[25] Mo, Y., Liu, H., & Wang, Q. (2009). Conjugate direction particle swarm optimization solving systems of nonlinear equations. *Computers & mathematics with applications, 57*(11–12), 1877–1882. https://doi.org/10.1016/j.camwa.2008.10.005

[26] Pourjafari, E., & Mojallali, H. (2012). Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering. *Swarm and evolutionary computation, 4,* 33–43. https://doi.org/10.1016/j.swevo.2011.12.001

[27] Hirsch, M. J., Meneses, C. N., Pardalos, P. M., & Resende, M. G. C. (2007). Global optimization by continuous grasp. *Optimization letters, 1*(2), 201–212. https://doi.org/10.1007/s11590-006-0021-6

[28] Nadimi Shahraki, M. H., Taghian, S., Mirjalili, S., Abualigah, L., Abd Elaziz, M., & Oliva, D. (2021). EWOA-OPF: Effective whale optimization algorithm to solve optimal power flow problem. *Electronics, 10*(23), 2975. https://doi.org/10.3390/electronics10232975

[29] Babaei, M. (2013). A general approach to approximate solutions of nonlinear differential equations using particle swarm optimization. A*pplied soft computing, 13*(7), 3354–3365. https://doi.org/10.1016/j.asoc.2013.02.005

[30] Ahmadianfar, I., Heidari, A. A., Gandomi, A. H., Chu, X., & Chen, H. (2021). RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert systems with applications, 181,* 115079. https://doi.org/10.1016/j.eswa.2021.115079

[31] Mendes, P., & Kell, D. (1998). Non-linear optimization of biochemical pathways: Applications to metabolic engineering and parameter estimation. *Bioinformatics (oxford, england), 14*(10), 869–883. https://doi.org/10.1093/bioinformatics/14.10.869

[32] Rodriguez Fernandez, M., Egea, J. A., & Banga, J. R. (2006). Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. *BMC bioinformatics, 7*(1), 483. https://doi.org/10.1186/1471-2105-7-483

[33] Godio, A., Pace, F., & Vergnano, A. (2020). SEIR modeling of the Italian epidemic of SARS-CoV-2 using computational swarm intelligence. *International journal of environmental research and public health, 17*(10), 3535. https://doi.org/10.3390/ijerph17103535

[34] Dorigo, M., & Gambardella, L. M. (2002). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE transactions on evolutionary computation, 1*(1), 53–66. https://doi.org/10.1109/4235.585892

[35] Helsgaun, K. (2000). An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European journal of operational research, 126*(1), 106–130. https://doi.org/10.1016/S0377-2217(99)00284-2

[36] Larranaga, P., Kuijpers, C. M. H., Murga, R. H., Inza, I., & Dizdarevic, S. (1999). Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial intelligence review, 13*(2), 129–170. https://doi.org/10.1023/A:1006529012972

[37] Malaguti, E., Monaci, M., & Toth, P. (2008). A metaheuristic approach for the vertex coloring problem. *INFORMS journal on computing, 20*(2), 302–316. https://doi.org/10.1287/ijoc.1070.0245

[38] Lü, Z., & Hao, J. K. (2010). A memetic algorithm for graph coloring. *European journal of operational research, 203*(1), 241–250. https://doi.org/10.1016/j.ejor.2009.07.016

[39] Moalic, L., & Gondran, A. (2018). Variations on memetic algorithms for graph coloring problems. *Journal of heuristics, 24*(1), 1–24. https://doi.org/10.1007/s10732-017-9354-9

[40] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation, 6*(2), 182–197. https://doi.org/10.1109/4235.996017

[41] Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. *Evolutionary computation proceedings* (pp. 69-73). IEEE. https://doi.org/10.1109/ICEC.1998.699146

[42] Meintjes, K., & Morgan, A. P. (1990). Chemical equilibrium systems as numerical test problems. *ACM transactions on mathematical software (TOMS), 16*(2), 143–151. https://doi.org/10.1145/78928.78930

[43] Tsai, L. W., & Morgan, A. P. (1985). Solving the kinematics of the most general six-and five-degree-of-freedom manipulators by continuation methods. *Journal of mechanisms, transmissions, and automation in design (ASME), 107*, 189-200. https://doi.org/10.1115/1.3258708

[44] Sinha, N., Chakrabarti, R., & Chattopadhyay, P. K. (2003). Evolutionary programming techniques for economic load dispatch. *IEEE transactions on evolutionary computation, 7*(1), 83–94. https://doi.org/10.1109/TEVC.2002.806788

[45] Abdollahi, A., Ghadimi, A. A., Miveh, M. R., Mohammadi, F., & Jurado, F. (2020). Optimal power flow incorporating FACTS devices and stochastic wind power generation using krill herd algorithm. *Electronics, 9*(6), 1043. https://doi.org/10.3390/electronics9061043

[46] Reinelt, G. (1991). TSPLIB—A traveling salesman problem library. *ORSA journal on computing, 3*(4), 376–384. http://dx.doi.org/10.1287/ijoc.3.4.376

[47] Derrac, J., Garcia, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. S*warm and evolutionary computation, 1*(1), 3–18. https://doi.org/10.1016/j.swevo.2011.02.002

[48] Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association, 32*(200), 675–701. https://www.tandfonline.com/doi/pdf/10.1080/01621459.1937.10503522

[49] Nemenyi, P. B. (1963). *Distribution-free multiple comparisons*. Princeton University. https://www.bibsonomy.org/bibtex/2be58c4cd02deec08771241724168b663/sveng

[50] Zaharie, D. (2003). Control of population diversity and adaptation in differential evolution algorithms. *Proceedings of mendel 2003, 9th international conference on soft computing* (pp. 41–46). Springer-Verlag. https://staff.fmi.uvt.ro/~daniela.zaharie/mendel03.pdf

[51] Wolpert, D. H., & Macready, W. G. (2002). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, *1*(1), 67–82. https://doi.org/10.1109/4235.585893