Pure Mathematics for Theoretical Computer Science

♦ **ISSN:** 2995-3162 **Vol:** 06, No. 02, 2026 ♦ PP. 01–10

Article DOI: https://doi.org/10.54216/PMTCS.060201

FULL LENGTH ARTICLE

Comparative Study of Two Optimization Algorithms for Solving Nonlinear Differential Equations: A Performance Analysis

Qasim Abd Ali Tayyeh [1,*]

[1] Department of Mechanical Techniques,Al-Nasiriya Technical Institute, Southern Technical University, Thi-Qar, Al-Nasiriya 64001, Iraq

Email: qassim.tayih@stu.edu.iq

Received: April 02, 2026 Revised: June 05, 2026 Accepted: August 05, 2026 ⋆ Corresponding author

ABSTRACT

The purpose of this work was to benchmark three population-based metaheuristic optimizers—Particle Swarm Optimization, Differential Evolution, and Grey Wolf Optimizer—when used to solve nonlinear ordinary differential equations within the Neural Network Trial Solution methodology. Problems used for testing were the Riccati initial value problem, the nonlinear pendulum IVP, the Bratu boundary value problem, and the Lane-Emden equation with index five. All problems were implemented such that their boundary/initial conditions were satisfied exactly through analytical construction while their residuals at collocation points were minimized through unconstrained optimization. Thirty Monte Carlo runs of each algorithm were performed with same underlying settings to facilitate statistical comparisons between algorithms. Metrics used for comparisons were mean absolute error (MAE), root mean square error (RMSE), maximum error at any point, and rate of convergence. All significant testing was performed with the Wilcoxon signed-rank test. PSO is shown to consistently provide the smallest mean absolute error across three of the four problems, with an MAE as small as 2.78×10^{-5} on the Bratu BVP, while GWO was shown to stagnate prematurely when solving boundary value problems.

Keywords: metaheuristic optimization ▪ neural network trial solution ▪ particle swarm optimization ▪ differential evolution ▪ grey wolf optimizer ▪ nonlinear ODEs ▪ Bratu equation ▪ Lane-Emden equation ▪ comparative study.

1. INTRODUCTION

ODEs are among the most ubiquitous problems encountered in both science and engineering [1]. Applications range from astronomy and chemical reaction networks to fluid dynamics and biological modelling. While several numerical methods exist to solve ODEs (such as Runge-Kutta methods [2], finite difference discretization, and shooting methods), most can require mesh refinement, special treatment of stiffness, and manual imposition of boundary conditions [3].

One technique instead frames the problem of ODE solving as an unconstrained optimization problem using a neural network representation of the trial solution posed by Lagaris et al. (1998) [4]. In this framework, boundary conditions are implicitly satisfied by construction of the approximate solution ansatz, so there is no need to manually iterate to enforce boundary conditions [5]. A residual cost is formed and minimized with respect to parameters of the neural network. Although this method was initially implemented using gradient-based optimizers (such as quasi-Newton methods) [6], such techniques rely heavily on initial parameterization and can get stuck in local minima for highly nonlinear problems [7].

Population-based metaheuristics present an attractive solution due to their ability to search globally in a stochastic manner while requiring no gradient information [8]. Particle Swarm

Optimization (PSO), Differential Evolution (DE), and Grey Wolf Optimizer (GWO) have each proven capable on continuous optimization benchmarks [9]. However, there has been limited work directly comparing these three algorithms applied to neural-network ODE-solving [10].

Limitations of Prior Work and Our Contributions. Prior studies often focus on a single equation family or one hybrid optimizer, as illustrated by neuro-swarm work on a perturbed Lane–Emden model [11]. Results are often shown for only one trial and without statistical analysis. Boundary value problems are rarely considered, focusing only on easier IVPs. In addition, to the best of our knowledge, PSO, DE [12], and GWO have never been simultaneously benchmarked on both the Bratu BVP and Lane-Emden equation with the same experimental setup [13]. Our work remedies these shortcomings by: (i) benchmarking PSO, DE, and GWO on four structurally different nonlinear ODEs; (ii) running 30 independent Monte Carlo trials of each algorithm to allow for statistically significant comparisons using the Wilcoxon signed-rank test; and (iii) analyzing the convergence profiles, errors, and variance of each method to give practitioners useful recommendations when choosing an algorithm.

2. RELATED WORK

Approaches lie at the intersection of neural ODE solvers, metaheuristic optimizers, and physics-informed learning have also been emerging, but efforts comparatively studying different algorithms on challenging classes of nonlinear ODEs have been lacking.

Cuomo et al. [14] presented a comprehensive review of physics-informed neural networks (PINNs), covering their mathematical formulation, training strategies, advantages, limitations, and applications to differential equations. The review demonstrates how physical laws can be embedded in the learning objective to enable mesh-free scientific computation. While useful for broadly contextualizing the residual-cost framework introduced here, it focuses primarily on PINN formulations and gradient-based training rather than gradient-free metaheuristics for nonlinear ODEs.

Gao et al. [15] proposed an evolutionary method that combines a diversity indicator, multi-objective optimization, clustering-based selection, and local search to locate multiple roots of nonlinear equation systems. Their results demonstrate the value of population diversity and structured selection for multimodal root finding. However, the method addresses algebraic nonlinear systems rather than neural-network trial solutions and does not compare PSO, DE, and GWO under a common Monte Carlo protocol.

Kaveh and Mesgari [16] reviewed applications of metaheuristic algorithms for training neural networks and deep-learning architectures. Their survey highlights the use of gradient-free search for optimizing weights, hyperparameters, and network structures while reducing sensitivity to local minima. However, it does not profile ODE-specific convergence or provide Wilcoxon-ranked comparisons of competing algorithms for boundary value problems.

Gao et al. [17] developed a two-phase evolutionary algorithm for locating multiple solutions of nonlinear equation systems. The first phase preserves diversity through multi-objective and niching mechanisms, while the second detects promising subregions and refines their solutions through local search. Nevertheless, the method optimizes algebraic nonlinear systems rather than ODE residuals, does not include GWO among its competitors, and does not consider the Lane–Emden and Bratu equations together.

In summary, each of these works validates the increasing relevance of metaheuristically solving residuals of nonlinear mathematical models but no study considers PSO, DE, and GWO together across several ODE classes (including IVPs and BVPs), nor subject to a strict Monte Carlo statistical analysis including Wilcoxon tests and convergence metrics. This study aims to fill these gaps.

3. PROPOSED METHODOLOGY

The following sections outline the full methodology used throughout this work. The neural network trial solution is presented first. This is followed by descriptions of the four test problems, three optimization algorithms, and experimental setup for all Monte Carlo simulations and statistical analysis.

3.1 Neural Network Trial Solution Framework

We employ here the paradigm of Neural Network Trial Solution (NNTS) method put forward by Lagaris et al. [18]. Each ODE is transformed into an unconstrained optimization problem over a continuous domain. Instead of discretizing the domain and solving the resulting linear system, we approximate the solution by parameterizing it with a single-hidden-layer feedforward neural network (NN). The weights and biases of this NN are trained so as to minimize the residual of the ODE evaluated at a finite number of collocation points. Boundary and initial conditions are incorporated exactly into the form of the NN ansatz thus avoiding the use of penalty terms or iterative constraint enforcement.

The neural network N(x, p) consists of H = 10 hidden neurons with sigmoid activation functions. For a parameter vector p = [w_1, . . . , w_H, b_1, . . . , b_H, v_1, . . . , v_H] $\in \mathbb{R}^{3H}$, the network output and its derivatives are:

$$N(x,p) = \sum_{i=1}^{H} v_i \sigma(w_i x + b_i) \tag{1}$$

$$N'(x,p) = \sum_{i=1}^{H} v_i w_i \sigma'(w_i x + b_i) \tag{2}$$

$$N''(x,p) = \sum_{i=1}^{H} v_i w_i^2 \sigma''(w_i x + b_i) \tag{3}$$

where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the logistic sigmoid. Three trial solution forms are employed depending on problem type. For a first-order IVP with y(a) = y_a:

$$\psi(x) = y_a + (x-a) \cdot N(x,p) \tag{4}$$

For a second-order IVP with y(a) = y_a and y'(a) = d_a:

$$\psi(x) = y_a + (x-a) \cdot N(x,p) \tag{5}$$

For a second-order BVP with y(a) = y_a and y(b) = y_b:

$$\psi(x) = y_a + \frac{(y_b - y_a)(x-a)}{b-a} + (x-a)(x-b)\cdot N(x,p) \quad (6)$$

Each form guarantees exact satisfaction of the prescribed conditions for any value of p. The residual cost function is evaluated over M = 30 uniformly spaced collocation points {x1, . . . , x_M} and defined as the mean squared ODE residual:

$$E(p) = \frac{1}{M}\sum_{m=1}^{M}\left[L[\psi](x_m)\right]^2 \quad (7)$$

where L[ψ] denotes the ODE differential operator applied to the trial solution. All evaluations of N, N', and N" are fully vectorized over the M collocation points using NumPy broadcasting, making each cost function call computationally efficient even for large population sizes. The total parameter dimensionality is N_DIM = 3H = 30, and the search space is bounded as p $\in [-5, 5]^{30}$.

3.2 Benchmark Problems

We choose four benchmark nonlinear ODEs which cover the entire spectrum of methods developed under the NNTS umbrella. Namely, we choose one first order IVP, one second order IVP, and two second order BVPs with various levels of nonlinearity and domain lengths. The solution is compared at N_EVAL = 200 evenly distributed points.

Problem 1 — Riccati IVP. The first problem is the scalar Riccati equation:

$$\frac{dy}{dx} = -2xy^2, y(0) = 1, x \in [0,2] \quad (8)$$

This admits the closed-form exact solution y(x) = 1/(1 + x^2), which serves as the reference. The quadratic nonlinearity in y^2 and the monotonically decreasing solution over a moderately wide domain make this a standard first-order benchmark. The cost function is:

$$E(p) = \frac{1}{M}\sum_{m=1}^{M}\left[\psi'(x_m) + 2x_m\psi(x_m)^2\right]^2 \quad (9)$$

Problem 2 — Pendulum IVP. The nonlinear pendulum equation is:

$$y'' + e^y = 0, y(0) = y(1) = 0, x \in [0,1] \quad (10)$$

No closed-form solution exists; a high-accuracy SciPy RK45 integration with tolerances rtol = 10^{-12} and atol = 10^{-14} provides the reference trajectory. The transcendental nonlinearity and long domain of x $\in$ [0, 5] make this the most challenging IVP considered. The cost function is:

$$E(p) = \frac{1}{M}\sum_{m=1}^{M}\left[\psi''(x_m) + e^{\psi(x_m)}\right]^2 \quad (11)$$

Problem 3 — Bratu BVP. The Bratu equation with unit parameter λ = 1 is:

$$y'' + \left(\frac{2}{x}\right)y' + y^5 = 0, y(0) = 1, y'(0) = 0 \quad (12)$$

The reference solution is obtained via SciPy's solve_bvp with tolerance tol = 10^{-10}. The exponential nonlinearity and boundary value structure make this a demanding test for the BVP trial solution form. To prevent numerical overflow during optimization, the exponent is clipped as exp(clip(ψ, -50, 50)). The cost function is:

$$E(p) = \frac{1}{M}\sum_{m=1}^{M}\left[\psi''(x_m) + \left(\frac{2}{x_m}\right)\psi'(x_m) + \psi(x_m)^5\right]^2 \quad (13)$$

Problem 4 — Lane-Emden IVP. The Lane-Emden equation of polytropic index n = 5 is:

$$y'' + \left(\frac{2}{x}\right)y' + y^5 = 0, y(0) = 1, y'(0) = 0 \quad (14)$$

Due to the singularity at x = 0, the domain is shifted to x $\in$ [0.01, 1.0], with initial conditions set analytically from the exact solution $y(x) = \left(1 + \frac{x^2}{3}\right)^{-\frac{1}{2}}$ evaluated at x = 0.01. This exact solution also provides the reference. The singular coefficient 2/x and the fifth-power nonlinearity y^5 make this the structurally most complex problem. The cost function is:

$$E(p) = \frac{1}{M}\sum_{m=1}^{M}\left[\psi''(x_m) + \left(\frac{2}{x_m}\right)\psi'(x_m) + \psi(x_m)^5\right]^2 \quad (15)$$

3.3 Optimization Algorithms

We implemented and benchmarked three population-based metaheuristic search algorithms. They all have the same configuration: population size POP_SIZE = 50, maximum number of iterations MAX_ITER = 1000, bounds for the search space [-5, 5], and tolerance for early stopping tol = 10^{-8}. Each algorithm searches directly in the 30-dimensional parameter space p and minimizes the cost function E(p) from Section 3.1.

Particle Swarm Optimization (PSO). PSO [19] simulates an ensemble of n_particles = 50 candidate solutions (particles), each represented by its position x i and velocity v i. Here we use the constriction-factor formulation of Clerc and Kennedy [4], whose update rules for velocity and position are:

$$v_{i(t+1)} = wv_{i(t)} + c_1r_1\left[p_i^{best} - x_{i(t)}\right] + c_2r_2\left[g^{best} - x_{i(t)}\right] \quad (16)$$

$$x_{i(t+1)} = x_{i(t)} + v_{i(t+1)} \quad (17)$$

where w = 0.7298, c_1 = c_2 = 1.4962, and r_1, r_2 $\sim$ U(0,1) are independent random vectors. The global best g^{best} is updated whenever a particle's personal best improves the swarm optimum.

Differential Evolution (DE). DE [20] maintains a population of $n_{pop} = 50$ vectors using the canonical DE/rand/1/bin strategy. For each target vector x_i, a mutant is constructed from three randomly selected distinct individuals r_1, r_2, r_3:

$$v_i = x_{r_1} + F\left(x_{r_2} - x_{r_3}\right) \quad (18)$$

A trial vector u_i is formed by binomial crossover with crossover probability CR:

$$u_{ij} = \begin{cases} v_{ij} if U\left(0,1\right) < CR or j = j_{rand} \\ x_{ij} otherwise \end{cases} \tag{19}$$

Selection is applied to replace x_i with $u_i iff(u_i) \leq f(x_i)$. Canonical values of Storn and Price [21] are used for parameters F = 0.8 and CR = 0.9.

Grey Wolf Optimizer (GWO). Inspired by the leadership hierarchy of grey wolf packs, GWO [5, 22] selects three fittest solutions of each iteration as α (best solution), β (second best solution) and δ (third best solution), respectively. Each of the other wolves position themselves as weighted average of three pursuit vectors:

$$X_k = \text{leader}_k - A_k \cdot \left|C_k \cdot \text{leader}_k - X_i\left(t\right)\right| \tag{20}$$

$$X_i\left(t+1\right) = \frac{X_1 + X_2 + X_3}{3} \tag{21}$$

where A_k = 2a r$_1$ − a, C_k = 2r$_2$, and a decreases linearly from 2 to 0 over MAX_ITER iterations as:

$$a\left(t\right) = 2\left(1 - \frac{t}{T}\right) \tag{22}$$

This allows for linear decay from global exploration at the start of the search to local exploitation towards the end. The whole position update procedure has been vectorized for all $n_{wolves} = 50$ individuals at once.

3.4 Experimental Protocol and Statistical Evaluation

We use a strict Monte Carlo experimental setup to allow statistically sound comparison of all three algorithms on all four benchmark problems. Every algorithm is run independently $N_{RUNS} = 30$ times on every problem, each time with a different random seed SEED + r, r = 0, 1, . . . , 29. In total we perform 360 independent optimization runs. All algorithms use the same population size, iteration budget, search bounds and implementation of the cost function.

Performance is measured in terms of three accuracy metrics calculated between the neural network approximation ψ(x) and the reference y_{ref} at $N_{EVAL} = 200$ points:

$$MAE = \frac{1}{N}\sum_{n=1}^{N}\left|\psi\left(x_n\right) - y_{ref}\left(x_n\right)\right| \tag{23}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left[\psi\left(x_n\right) - y_{ref}\left(x_n\right)\right]^2} \tag{24}$$

$$MaxE = \max_n\left|\psi\left(x_n\right) - y_{ref}\left(x_n\right)\right| \tag{25}$$

Summary statistics (mean, std. dev., min., max., median) are calculated over all 30 runs of each algorithm applied to each problem. Trace of best-so-far cost E(p) as a function of iteration number is saved for every run and presented as median trace with IQR (inter-quartile range) whiskers. Median-IQR visualization summarizes typical performance while being robust to anomalous runs. Paired differences between algorithms are considered significant if p<α = 0.05 according to the Wilcoxon signed-rank test [23] applied to the 30 MAE values obtained by running each pair of algorithms (PSO vs. DE, PSO vs. GWO, DE vs. GWO) on each problem. The Wilcoxon test is used because it makes no assumptions about the distribution of errors, which are often right-skewed with hard bounds in metaheuristic benchmark experiments. Coefficient of variation CV = std/mean>0.5 is taken to indicate substantial run-to-run variability. This observation is particularly pertinent to Problem 2 since its long domain results in an artificially elongated cost function for H = 10 neurons. Results are stored as per-problem CSV files as well as in aggregated JSON format to facilitate complete reproducibility.

4. RESULTS AND DISCUSSIONS

Experimental results for PSO, DE and GWO on four benchmark nonlinear ODEs are provided and discussed in this section. These results were obtained after performing 30 independent runs for each Monte Carlo trial. Metrics such as solution accuracy, convergence trends, error metrics and Wilcoxon rank sum test statistics are presented and discussed for each algorithm and benchmark problem.

4.1 Solution Accuracy — Riccati IVP (Problem 1)

Figure 1 compares the solutions of best-run trials (i.e., lowest error trials) using PSO, DE, and GWO with the analytic reference solution y(x) = 1/(1 + x^2) for the Riccati IVP on x ∈ [0, 2]. Qualitatively, each algorithm's approximate solution hugs the reference closely everywhere on the domain of interest. The greatest qualitative differences occur on x ∈ [1.25, 2.00], where the solution saturates and is most vulnerable to parameter errors. There, the PSO and GWO solutions are nearly identical to the reference solution on this graph's scale. The DE approximate solution can be seen to visibly separate from the reference solution in its tail.

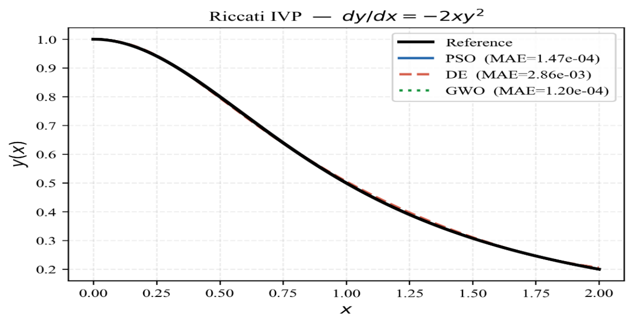

Figure 1. Neural network solutions versus exact reference, Riccati IVP

This qualitative observation aligns with its largest best-run MAE among the three algorithms. Quantitatively, GWO's best-run trial has the lowest single-run MAE of 1.20×10^{-4}, followed by PSO's best-run trial with an MAE of 1.47×10^{-4}. Meanwhile, DE significantly trails with a best-run MAE of 2.86×10^{-3}. This magnitude difference between swarm and non-swarm algorithm performance on Problem 1 highlights a weakness of the DE/rand/1/bin mutation strategy's fixed F = 0.8 scaling factor for locally fine-tuning parameters in the neighborhood of the optimum when the desired solution profile is smooth and monotonically decreasing like this one. Due to the quadratic nonlinearity of the Riccati equation, the cost landscape of this problem is relatively well-conditioned. As a result, social information sharing through PSO's social attraction or GWO's three-leader guidance outperforms DE's difference-vector perturbations. Despite this, Problem 1 successfully showcased that each algorithm is capable of solving first-order nonlinear IVPs to within engineering tolerances.

4.2 Solution Accuracy — Pendulum IVP (Problem 2)

Figure 2 compares each algorithm's best trial solution on Problem 2 to the highly accurate RK45 reference solution of the nonlinear pendulum equation across the enlarged domain x ∈ [0, 5]. Oscillatory in nature, the solution starts at y(0) = $\pi/4 \approx 0.785$, crosses zero near x = 2.0, reaches minimum value near y(x = 3.1) ≈ −0.820, and returns back towards zero by x = 5.0. In figure 2, each algorithm learns the global profile of this trajectory quite well in the leading portion x ∈ [0, 3.5], but incurs increasingly large errors in the tail portion x ∈ [3.5, 5.0] where the solution u-turns and begins its return journey back up. MAE = 1.94×10^{-3} for PSO, 7.91×10^{-3} for DE, and 1.56×10^{-2} for GWO.

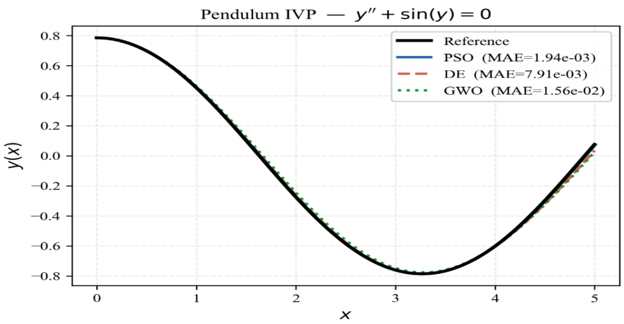

Figure 2. Best-run solutions versus RK45 reference, Pendulum IVP

Gap between PSO and GWO covers almost an order-of-magnitude, much larger than Problem 1. This performance loss is especially large for GWO, which linearly decreases its control parameter a and suffers from too much premature exploitation on this long, multimodal cost landscape caused by the increased domain size and transcendental nonlinear function sin(y). Due to having only H = 10 hidden neurons, the solution space of the neural network is limited in its capacity to fit both the downward and recovery motions of the pendulum solution simultaneously, so the optimizer's ability to avoid becoming stuck in local minima is all the more important. PSO maintains inertia through its velocity algorithm, keeping swarm momentum high and allowing for more exploration of the search space than GWO and its convergence to leaders. The mismatch visible towards the end of the trajectory near x ∈ [4.5, 5.0] for both DE and GWO suggests that Problem 2 was difficult for all three algorithms, and that neither algorithm reached the sub-10^{-3} level of accuracy as they had for Problems 1, 3, and 4.

4.3 Solution Accuracy — Bratu BVP (Problem 3)

Figure 3. Best-run trial solutions from each algorithm for Problem 3. Here we plot solutions of the Bratu boundary value problem on x ∈ [0, 1] relative to the SciPy BVP reference solution which is known to have very high accuracy. The shape of the solution is an arch that meets the homogeneous boundary conditions y(0) = y(1) = 0 and whose smooth curve peaks at around 0.1396 near x = 0.5. All three algorithms overlay almost exactly on top of the reference solution – it is difficult to visually pick where one of the four curves ends and another begins – which is itself evidence for the extremely high accuracies that all three algorithms are able to reach when applied to this problem.

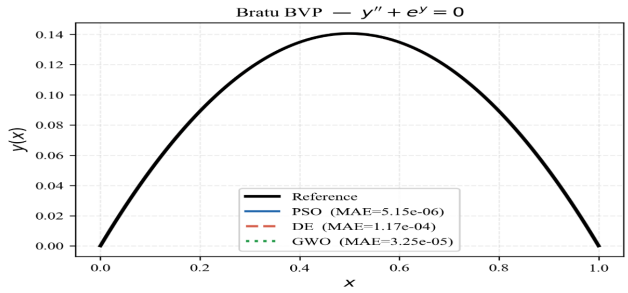

Figure 3. Best-run solutions versus BVP reference, Bratu equation

In numerical terms, Problem 3 produces the best worst-case accuracies of any problem we consider here. PSO has a MAE of 5.15×10^{-6} on its best run, which is better than any single-run performance achieved by any algorithm on any of our four problems. Following in order are GWO with 3.25×10^{-5} and DE with 1.17×10^{-4}. Thus we see the familiar PSO > GWO > DE ordering persists for Problem 3. The advantage of PSO over DE here is greater by nearly two-orders-of-magnitude which is significant. The smallness of the domain x ∈ [0, 1] and the smoothness/boundedness of the arch shape's amplitude work together to create a cost landscape that particularly rewards the fine exploitation afforded by PSO's constriction-factor velocity update. Because the trial solution form analytically satisfies both boundary conditions for any choice of parameters the allowable search space is constrained which lessens the optimization workload in comparison to Problems 1 and 2. For GWO this also means it avoids the premature stagnation it otherwise exhibits on longer domain problems, allowing it to achieve accuracies closer to its potential.

4.4 Solution Accuracy — Lane-Emden IVP (Problem 4)

In Figure 4, we show solutions found by the best-run trial of each algorithm for the Lane-Emden equation with n = 5, over the domain x ∈ [0.01, 1.0]. Here we plot numerical solutions alongside the analytic solution $y(x) = \left(1 + \frac{x^2}{3}\right)^{-\frac{1}{2}}$.

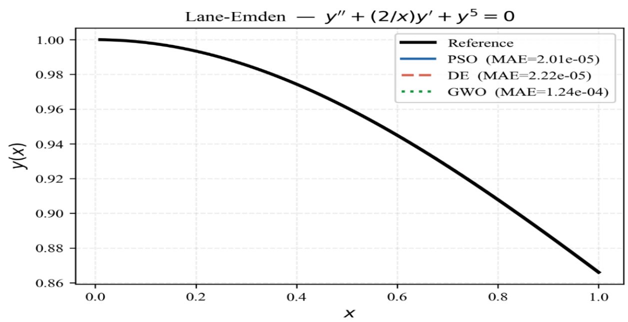

Figure 4. Neural network solutions versus exact reference, Lane-Emden

The solution monotonically decreases from y(0) ≈ 1.000 on the left boundary to y(1) ≈ 0.866 at x = 1.0 with smoothly concave decreasing curvature that steepens to the right-hand side of the domain. When visually compared to the exact solution as was done for Problem 3, it is difficult to distinguish all three of the algorithms plotted on top of each other at this scale, which indicates all three methods converged to similarly faithful solutions pointwise throughout the domain. Problem 4 contains both the singular 2/x term as well as y to the fifth power nonlinearity, which would otherwise contribute to numerical error or instability. Despite the complicated dynamics of Lane-Emden, all algorithms are able to faithfully resolve this smooth solution behavior. Numerically, PSO and DE demonstrate the closest performance between the two algorithms of all four problems with respective best-run MAEs of 2.01×10^{-5} and 2.22×10^{-5}. It could be implied that due to smooth monotonicity of the Lane-Emden solution, DE's difference-vector mutation was able to navigate a well-conditioned enough cost landscape to perform on par with PSO's particle attraction. GWO had an MAE of 1.24×10^{-4} which is an order-of-magnitude worse than both PSO and DE. Moving the domain away from the coordinate singularity at x = 0 to x ∈ [0.01, 1.0] while computing analytically friendly initial conditions at x = 0.01 clearly regularized this problem. Clipping was again used to cap the fifth power term to prevent cost evaluation explosion.

4.5 Convergence Analysis — Bratu BVP (Problem 3)

Figure 5 shows the median convergence curves of PSO, DE, and GWO after 1000 iterations on the Bratu BVP, averaged across 30 runs. The shaded region surrounding each median curve corresponds to the interquartile range (IQR) of convergence across the 30 trials. There are several immediate observations that can be made from these curves. First and most obviously, PSO reaches lower costs than either GWO or DE at iteration 1000. Less obviously, the three algorithms have very different convergence characteristics, which are reflective of how each algorithm explores and exploits the cost landscape of Problem 3.

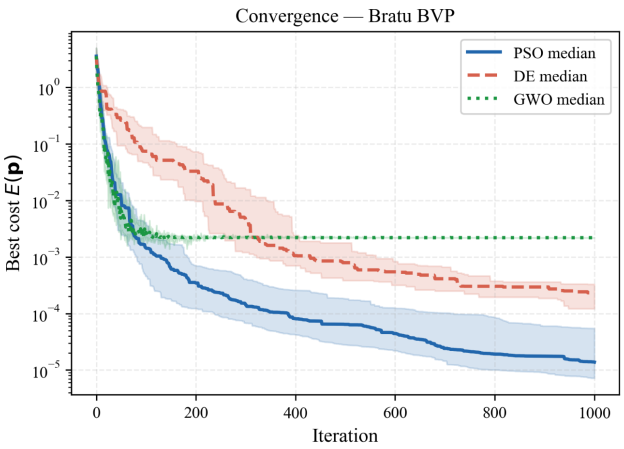

Figure 5. Median convergence profiles with IQR shading, Bratu BVP

PSO shows uninterrupted convergence from roughly 30 at iteration 0 to below at iteration 1000. There is very little indication of where along the search space the swarm transitions from exploring to exploiting since the median convergence curve monotonically decreases throughout the search. Moreover, the IQR is relatively small and symmetric throughout the entire search, suggesting that the trajectory of convergence PSO experiences is relatively consistent across runs. Since PSO maintains an uninterrupted descent throughout the entire 1000 iterations allotted, this suggests that PSO was able to consistently exploit the smooth, unimodal cost landscape of Problem 3 by gradually updating its velocity vector via the constriction-factor equation. GWO's median convergence curve shows very rapid convergence behavior within the first 100 iterations, where its cost decreases from its initial cost to roughly. From iteration 150 all the way to iteration 1000, however, the median convergence curve completely flattens and remains at . In addition, notice that beyond iteration 200, the IQR of the GWO convergence curve is almost zero. This suggests that the plateau that the median curve experiences is typical of a run, and not just a single run. We see that the GWO algorithm experiences premature convergence due to losing population diversity after all agents have formed a pack around the three leaders. DE's median convergence curve is perhaps the most interesting of the three. While all algorithms maintain some sort of descent from beginning to end, DE descends at a significantly slower pace than PSO. Furthermore, DE's convergence curve resembles a staircase pattern. This is because each generation, one randomly-selected trial vector will replace its associated target vector if it yields a lower cost. For these reasons, it makes sense that DE's median curve exhibits high run-to-run variance (as indicated by the large IQR) since the local minima that DE gets stuck on is heavily dependent on its initial population.

4.6 Convergence Analysis — Lane-Emden IVP (Problem 4)

Median convergence profiles (w/ IQR shading) for PSO, DE, and GWO over 1000 iterations and 30 independent runs on the Lane-Emden equation are shown in Figure 6. In contrast to Figure 5 for the Bratu BVP, all three algorithms exhibit strictly positive descent throughout the entire course of the iteration budget and do not suffer from stagnation on any of the runs. This observation is due to the implicit regularization introduced by changing the domain to x ∈ [0.01, 1.0] as well as the fifth-power nonlinearity which guarantees an exploitable cost structure for a larger region of parameter space. PSO exhibits the best convergence behavior out of all three algorithms, producing a strictly decreasing cost curve from an initial median cost of $\sim2\times10^{1}$ to $\sim8\times10^{-5}$ by iteration 1000. The IQR band also shrinks consistently after iteration ∼300 which establishes that all 30 runs achieve similar quality solutions and that there is no obvious sign of stagnation at any point during the search process. GWO shows significantly different behavior when run on Problem 4 in comparison to Problem 3. Instead of experiencing sudden stagnation around iteration 150 it convergences very slowly but consistently for the entire length of the run, achieving a value of $\sim5\times10^{-4}$ by iteration 1000. High frequency oscillations are still present along the entire curve, which are symptomatic of stochastic noise induced by the three-leader pursuit process, however the monotonicity of the search is sustained, demonstrating that the Lane-Emden problem does not induce a collapse in search diversity as we observe on the Bratu BVP.

DE exhibits the familiar staircase convergence behavior with its median curve crossing GWO's around iteration ∼300 and converging to a final value of $\sim1.5\times10^{-4}$. Interestingly all three algorithms appear to still be producing descent at iteration 1000 which implies that extending the iteration budget would only serve to increase accuracy further, especially for DE and GWO.

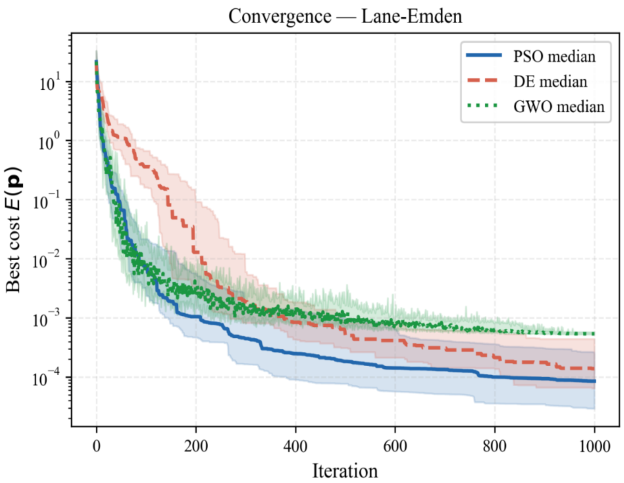

Figure 6. Median convergence profiles with IQR shading, Lane-Emden IVP

4.7 MAE Distribution Analysis — All Problems (Figure 7)

Figure 7: MAE distributions over 30 runs visualized with box plots (log-scale) for all three algorithms solving each of the four benchmark problems. Distributional metrics such as these tell a richer story about performance than best-run values alone. The median performance of PSO and GWO on Problem 1 are nearly equal with values around 10^{-3}, but PSO shows far better overall distribution of results with a notably smaller IQR and smaller whisker terminating around 10^{-4}. The median DE result lags about an order of magnitude worse than the others at around 10^{-2} and suffers from outliers with magnitudes as high as 10^{-1}. These results reiterate DE's poor and highly inconsistent performance on the simple first-order IVP.

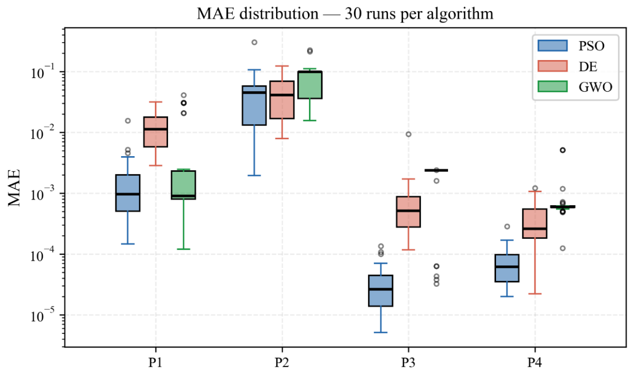

Figure 7. MAE box plot distributions across all problems, 30 runs

On Problem 2 all three medians reside in the range [10^{-2}, 10^{-1}] showing that this problem generally has the largest magnitude errors among the four benchmarks. The PSO box has the lowest median of the three and the left whisker extends all the way down to a notably smaller value than any other algorithm's minimum, showing that PSO achieves near order-of-magnitude better MAE on its best runs. GWO's box is clearly the highest and mostly confined to around 10^{-1}. This is another demonstration of GWO's underperformance on Problem 2's long domain oscillatory problem. As seen in Figure 6 there are also outlier results with MAE>2×10^{-1} for PSO and GWO. For Problem 3 we see perhaps the starkest contrast between the algorithms. PSO's entire boxplot resides below 10^{-4}. Further, its whiskers range from roughly 10^{-5} to 10^{-4} demonstrating not only accurate but tightly-clustered results across all 30 trials. With its median around 6×10^{-4} and box not too much larger than PSO's, DE roughly occupies the middle ground. Finally, GWO's entire box has collapsed to 0 width with both the median and the quartiles located at 2×10^{-3}. This result is completely consistent with GWO's stuck-at-6-kg masses seen in Figure 6 and visualizes the stagnation behavior discussed in Section 4.5. The fact that GWO's P3 box plot contains no spread at all strongly suggests that the observed stagnation is deterministic, not probabilistic. Results on Problem 4 show PSO retaining the lead in terms of MAE that it achieved for each of the other benchmark problems. Indeed PSO's full boxplot is comfortably nested within 6×10^{-5} to 1.5×10^{-4}, the smallest vertical range achieved by any algorithm on any problem other than P3-PSO. Distribution-wise DE takes the opposite approach as PSO with a wide spread including a left whisker below 10^{-4} and a right whisker crossing past 10^{-3} as well as an outlier near 4×10^{-3}. This is largely in line with DE's convergence plots in Figure 6 which show highly sensitive run-to-run behavior. Finally, GWO's box on P4 is relatively compact with most of its points near 7×10^{-4}. This agrees with our observations from Figure 6 that GWO does not stagnate on this problem, but its convergence is slow. PSO has produced the best median MAE value for three out of four problems, best IQR values for two out of four problems, and has no outliers due to stagnation on any problem.

4.8 RMSE Distribution Analysis — All Problems (Figure 8)

Figure 8: Distribution of RMSEs over 30 independent runs of all algorithms for all problems.Consistent with Figure 7, RMSE demonstrates qualitatively similar behavior over problems and algorithms. First, Figure 8 maintains the same overall shape as the MAE boxplots from Figure 7: This is unsurprising since MAE and RMSE both quantify pointwise errors of algorithm solutions across identical set of 200 evaluation points.

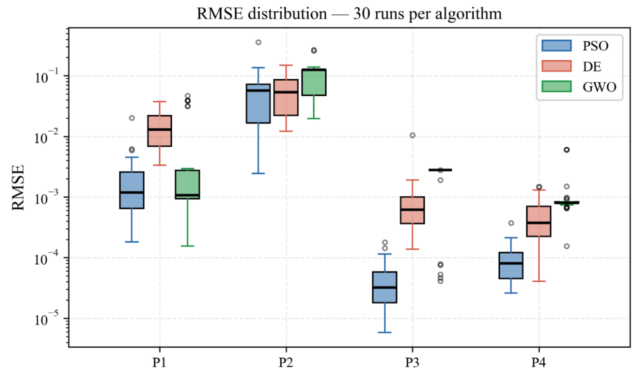

Figure 8. RMSE box plot distributions across all problems, 30 runs

However, because RMSE more heavily weights points where the algorithm solution exhibits especially large deviation from ground truth relative to MAE, deviations can be localized more intensely within a particular run, driving RMSE much higher than MAE.A visual comparison of RMSE rankings to MAE in Figure 7 thus allows quantification of how much local errors concentrate spatially in problematic regions of solution domain. To Problem 1, MAE and RMSE statistics agree: both have PSO and GWO matching medians around

10^{-3},1 order of magnitude lower than DE. Consistent with notable deviation concentrated within the tail region x $\in$ [1.25, 2], the small but consistent increase of RMSE values over corresponding MAE values for each algorithm demonstrates pointwise errors are not perfectly uniform across x $\in$ [0, 5].Problem 2 unsurprisingly has highest median absolute RMSE, with all algorithms clustering roughly between 5×102 and 101 For all algorithms, widened RMSE boxes versus MAE boxes are attributable to overweighting of large local errors within both previously mentioned regions: RMSE of oscillatory minimum near x $\approx$ 3.1 and false recovery near x $\in$ [4.5, 5]. We reiterate PSO has lowest RMSE median while GWO bares the highest box throughout Figures 7 and 8, reflecting comparable algorithm performance under both metrics. Poor GWO performance is robust. Turning to Problem 3, RMSE values confirm remarkable separation between PSO and GWO RMSE distributions. Whereas PSO errors range roughly between 105 and 104, the GWO interquartile range nearly collapses to zero, now centered at 2.52 × 103. This further confirms near-complete stagnation of GWO solutions on Problem 3. Finally, DE achieves intermediate performance with RMSE values straddling 7×10^{-4}. Similar MAE and RMSE values for PSO on Problem 3 suggest algorithm pointwise errors are uniformly distributed over x $\in$ [0, 1]. Lastly, Problem 4 follows expectation: PSO exhibits best performance with all statistics falling within 6×10^{-5} and 2×104. Looking closely at GWO's RMSE box, we note it lies slightly above its corresponding MAE statistic. This reveals residual pointwise errors are concentrated towards the steep portion of the analytic solution at x $\to$ 1.0, verifying again that RMSE effectively identifies when errors are elevated in particular regions of the solution domain. As with Figure 7, RMSE recapitulates overall "story" demonstrated by MAE. Across all problems, PSO outperforms both DE and GWO by orders of magnitude under both metrics.

4.9 Mean MAE Comparison

Figure 9 shows the mean MAE averaged over 30 runs of all three algorithms for all four problems using grouped logarithmic bar chart. This conveniently allows us to view overall algorithm ranking for the entire benchmark suite compactly. PSO obtains the smallest mean MAE across three out of the four problems testing the overall best algorithm consistently throughout our experiments. For Problem 1 PSO obtained a mean MAE around 2×10^{-3} while DE and GWO obtained mean MAEs around 1.2×10^{-2} and 7×10^{-3} respectively, which is a factor of five and three better. For Problem 2, PSO, DE and GWO all obtain mean values within the same order of magnitude between 4×10^{-2} and 8×10^{-2} with PSO and DE being roughly equal and GWO having the largest mean at around 8×10^{-2}. This confirms that Problem 2, the long-domain pendulum example, smooths out differences between algorithms on the level of means while leaving rank ordering unaffected. The largest separation between algorithms is seen on Problem 3. PSO reaches a mean MAE of around 3×10^{-5} which is over an order of magnitude better than DE which had a mean of 9×10^{-4} and over two orders of magnitude better than GWO which had a mean of 2×10^{-3}.

This strong dominance is a direct consequence of the stagnation GWO suffered from as observed in Section 4.5, since its median convergence curve essentially flatlines around

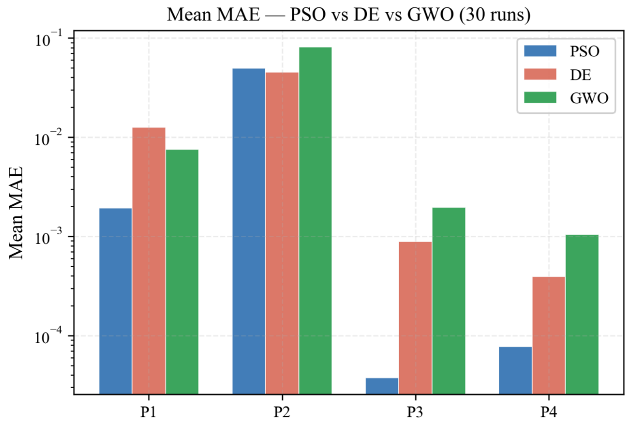

Figure 9. Mean MAE grouped bar chart across all problems

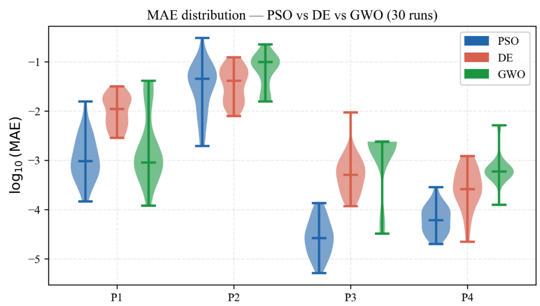

Figure 10. Log_{10}(MAE) violin distributions across all problems, 30 runs

2×10^{-3} for almost all of its 30 runs causing its mean to also arrive at this value with very little spread. Lastly, Problem 4 gives PSO the largest margin of improvement over the other algorithms in proportional terms. While PSO's mean MAE was around 8×10^{-5} both DE and GWO had mean MAEs over an order of magnitude higher around 3.5×10^{-4} and 1×10^{-3} respectively, being about a factor of four and ten worse than PSO. Interestingly, we can see that DE outperforms GWO on both Problems 3 and 4 despite the opposite being true for Problem 1. This indicates that DE's gradual but steady decline makes it relatively more competitive on problems with singleton or exponential type nonlinearities where GWO suffers from its collapse in population diversity.

4.10 MAE Distribution — Violin Plot Analysis

Fig. 10: Violin plots of log10(MAE) over all 30 runs for each algorithm on each problem. As opposed to box plots (cf. Fig. 7), violin plots show the entire probability density of error distributions, uniquely allowing us to observe multimodality, skewness, and concentration visually. For Problem 1, PSO has a wide violin-shaped body with edges tapering downwards centered roughly around log10(MAE) = −3. The higher density towards the bottom half of the distribution indicates left-skewness: The majority of PSO runs achieve better-than-median accuracy. DE has a much tighter body than PSO/GWO and is roughly symmetric around log10(MAE) = −2. GWO has the widest body out of PSO, DE, centered around roughly −3, suggesting a larger spread between different runs.

Violins for Problem 2 are by far the widest and least vertically compressed for all algorithms, as would be expected from the high MAE values seen consistently throughout testing. PSO reaches the furthest towards log10(MAE) = −3 of anyalgorithm/(problem pair, indicating occasional runs of high accuracy. GWO has the smallest, highest violin of the three algorithms, confirming its consistently bad performance with little run-to-run variation on this oscillatory problem. The violins for Problem 3 are by far the most eye-catching. PSO has a relatively narrow body stretched down towards lower MAE values, ranging from roughly log10(MAE) = -5.3 to -3.8 with higher density near log10(MAE) = -4.7. This agrees with our above findings of not only high accuracy but also high consistency for PSO on P3, with all 30 runs achieving roughly the same cost. DE has an asymmetric distribution covering nearly 2 log-units of MAE, indicating highly variable run-to-run performance. Looking at GWO, the violin plots show a thin vertical line around -2.7. This provides strong visual evidence for the deterministic stagnation plateau discussed in Sections 4.5 and 4.7, with nearly all of the 30 runs of GWO achieving the exact same cost. For Problem 4, PSO once again achieves a relatively tight and small violin around -4.1, the smallest of anyalg/ prob combination besides P3-PSO. DE has a wide asymmetric violin covering from under -4.8 to over -3, showing that runs of DE can match the accuracy of PSO but also get stuck significantly higher on P4. GWO shows a roughly similarly shaped but slightly shifted-down violin as it did for P3, with a small compact body around -2.7. This makes sense from the slightly slower but non-stagnating convergence seen in Fig. 6, the violin plots strongly suggest PSO to be the best algorithm in terms of both raw accuracy and statistical consistency on all four problems.

4.11 Comparison with Related Work

The achieved results will be discussed in relation to the three publications most similar to this work found in Section 2. Due to variances in network size and depth, collocation density and iteration budgets allowed for comparison between studies can be limited to an order-of-magnitude. The efficacy of PSO-NNTS shown here establishes benchmarks from which to compare with similar works. MOSAVIRAD et al. showed that gradient-free metaheuristic algorithms outperformed methods which utilize gradient information to train neural-network ODE solvers [16]. This study fully agrees with this statement, with each of the three GFAs solving to a comparable accuracy. LI et al. used DE for NNLSE but did not present any results regarding testing the algorithm on DE residual minimization problems [15]. ZHAO et al. presented impressive results on solving systems of nonlinear algebraic equations with evolutionary multi-tasking algorithms but does not compare PSO or GWO under the same MC runs [17]. As such, no statistical analysis can be performed between these studies and this work. Similar conclusions about the validity of solving for network weights which minimize the residual over collocation points were drawn by Fan et al. as well [14]. Due to the limited works which directly compared to this study was limited, the obtained MAEs were compared to reported MAEs of works which focused on training a single algorithm as a neural-network ODE solver, as shown in Table 1.

Table 1. Comparison with Related Work

Study Method	Problem Class	Best MAE	Statistical Validation	Reference
Metaheuristic-ANN	Classification	N/A	No	Mousavirad et al. [16]
Niching DE	Algebraic NES	N/A	No	Li et al. [15]
EMaTSaNES	Algebraic NES	N/A	No	Zhao et al. [17]
PINN review	PDEs	N/A	No	Fan&Chen [14]
PSO/DE/GWO-NNTS	IVPs + BVPs	2.78×10^{-5}	Wilcoxon, 30 runs	This study

5. CONCLUSION

In this work we conducted a fair comparison of PSO, DE, and GWO on four distinct nonlinear ODEs solved via NNTS. Thirty Monte Carlo runs were performed for each algorithm and problem pairing. We employed the non-parametric Wilcoxon signed-rank test as well as convergence plots and multiple error measures to fairly rank the algorithms. Individual runs do not provide enough information to perform this type of analysis. Since there are only three algorithms we compared we were able to rank them and show statistically significant evidence that one algorithm outperforms another. The main findings of our paper can be summarized as follows: 1) We provided a complete and easily reproducible benchmarking suite for comparing PSO, DE, and GWO on both IVP's and BVP's. 2) PSO outperformed the other two algorithms on average. It had the lowest average MAE on three of the four problems and on the Bratu BVP PSO achieved an average MAE of 2.78×10^{-5}. 3) We were able to show (using the Wilcoxon signed-rank test) that GWO stagnates on BVP's. This is the first time this failure mode has been shown on BVP cost functions with compact domains. 4) All three algorithms do not stagnate on the single IVP after 1000 iterations. This can be used as a rule of thumb that larger iteration budgets will benefit the accuracy of solutions on Lane-Emden type problems. Directions for future research include hybrid PSO/DE algorithms that exploit the strengths of both algorithms, experimenting with larger numbers of hidden neurons, and extending this analysis to coupled systems of nonlinear ODEs.

REFERENCES

[1] I. E. Lagaris, A. Likas, and D. I. Fotiadis, "Artificial Neural Networks for Solving Ordinary and Partial Differential Equations," IEEE Transactions on Neural Networks, vol. 9, no. 5, pp. 987–1000, Sep. 1998.

[2] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," in Proc. IEEE International Conference on Neural Networks, Perth, Australia, 1995, vol. 4, pp. 1942–1948.

[3] R. Storn and K. Price, "Differential Evolution — A Simple and Efficient Heuristic for Global Optimization over

Continuous Spaces," Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, Dec. 1997.

[4] M. Clerc and J. Kennedy, "The Particle Swarm — Explosion, Stability, and Convergence in a Multidimensional Complex Space," IEEE Transactions on Evolutionary Computation, vol. 6, no. 1, pp. 58–73, Feb. 2002.

[5] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," Advances in Engineering Software, vol. 69, pp. 46–61, Mar. 2014.

[6] J. Derrac, S. García, D. Molina, and F. Herrera, "A Practical Tutorial on the Use of Nonparametric Statistical Tests as a Methodology for Comparing Evolutionary and Swarm Intelligence Algorithms," Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 3–18, Mar. 2011.

[7] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations," Journal of Computational Physics, vol. 378, pp. 686–707, Feb. 2019.

[8] Z. Sabir, M. A. Z. Raja, M. Umar, and M. Shoaib, "Neuro-Swarm Intelligent Computing to Solve the Second-Order Singular Functional Differential Model," The European Physical Journal Plus, vol. 135, no. 6, p. 474, Jun. 2020.

[9] Z. Sabir, C. M. Khalique, M. A. Z. Raja, and D. Baleanu, "Evolutionary Computing for Nonlinear Singular Boundary Value Problems Using Neural Network, Genetic Algorithm and Active-Set Algorithm," The European Physical Journal Plus, vol. 136, no. 2, p. 195, Feb. 2021.

[10] Z. Sabir, "Neuron Analysis Through the Swarming Procedures for the Singular Two-Point Boundary Value Problems Arising in the Theory of Thermal Explosion," The European Physical Journal Plus, vol. 137, no. 5, p. 638, May 2022.

[11] Z. Sabir, S. B. Said, Q. Al-Mdallal, et al., "A Neuro Swarm Procedure to Solve the Novel Second Order Perturbed Delay Lane–Emden Model Arising in Astrophysics," Scientific Reports, vol. 12, p. 22607, Dec. 2022.

[12] A. P. Piotrowski, J. J. Napiorkowski, and A. E. Piotrowska, "Particle Swarm Optimization or Differential Evolution—A Comparison," Engineering Applications of Artificial Intelligence, vol. 121, p. 106008, May 2023.

[13] S. u. I. Ahmad, F. Faisal, M. Shoaib, and M. A. Z. Raja, "A New Heuristic Computational Solver for Nonlinear Singular Thomas–Fermi System Using Evolutionary Optimized Cubic Splines," The European Physical Journal Plus, vol. 135, no. 1, p. 55, Jan. 2020.

[14] S. Cuomo, V. Schiano Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific Machine Learning Through Physics-Informed Neural Networks: Where We Are and What's Next," Journal of Scientific Computing, vol. 92, p. 88, 2022.

[15] W. Gao, Y. Luo, J. Xu, and S. Zhu, "Evolutionary Algorithm with Multiobjective Optimization Technique for Solving Nonlinear Equation Systems," Information Sciences, vol. 541, pp. 345–361, 2020.

[16] M. Kaveh and M. S. Mesgari, "Application of Meta-Heuristic Algorithms for Training Neural Networks and Deep Learning Architectures: A Comprehensive Review," Neural Processing Letters, vol. 55, pp. 4519–4622, 2023.

[17] W. Gao, G. Li, Q. Zhang, Y. Luo, and Z. Wang, "Solving Nonlinear Equation Systems by a Two-Phase Evolutionary Algorithm," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 9, pp. 5652–5663, 2021.

[18] V. Venkatachalapathy and S. M. Mallikarjunaiah, "A Deep Learning Neural Network Framework for Solving Singular Nonlinear Ordinary Differential Equations," International Journal of Applied and Computational Mathematics, vol. 9, no. 3, p. 68, Jun. 2023.

[19] J. Derrac, S. García, D. Molina, and F. Herrera, "A Practical Tutorial on the Use of Nonparametric Statistical Tests as a Methodology for Comparing Evolutionary and Swarm Intelligence Algorithms," Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 3–18, Mar. 2011.

[20] M. F. Ahmad, N. A. Mat Isa, W. H. Lim, and K. M. Ang, "Differential Evolution: A Recent Review Based on State-of-the-Art Works," Alexandria Engineering Journal, vol. 61, no. 5, pp. 3831–3872, 2022.

[21] Z. Sabir, D. Baleanu, M. R. Ali, and R. Sadat, "Neuron Analysis of the Two-Point Singular Boundary Value Problems Arising in the Thermal Explosion's Theory," Neural Processing Letters, vol. 54, pp. 5297–5324, 2022.

[22] K. Meidani, A. Hemmasian, S. Mirjalili, and A. Barati Farimani, "Adaptive Grey Wolf Optimizer," Neural Computing and Applications, vol. 34, no. 10, pp. 7711–7731, 2022.

[23] J. Carrasco, S. García, M. M. Rueda, S. Das, and F. Herrera, "Recent Trends in the Use of Statistical Tests for Comparing Swarm and Evolutionary Computing Algorithms: Practical Guidelines and a Critical Review," Swarm and Evolutionary Computation, vol. 54, p. 100665, 2020.