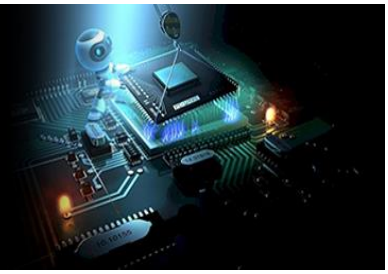
International Journal of Engineering in Computer Science



E-ISSN: 2663-3590
P-ISSN: 2663-3582
www.computersciencejournals.com/ijecs
IJECS 2025; 7(1): 133-141
Received: 15-02-2025
Accepted: 17-03-2025

Talada Ganesh Kumar
Assistant Professor, Department of Mathematics, Rajamahendri Institute of Engineering & Technology, Rajahmundry, East Godavari District, Andhra Pradesh, India

Varadhi Lakshmi Sailaja
Assistant Professor, Department of Mathematics, Rajamahendri Institute of Engineering & Technology, Rajahmundry, East Godavari District, Andhra Pradesh, India

Kathula Sunanda
Assistant Professor, Department of Mathematics, Rajamahendri Institute of Engineering & Technology, Rajahmundry, East Godavari District, Andhra Pradesh, India

Corresponding Author:
Talada Ganesh Kumar
Assistant Professor, Department of Mathematics, Rajamahendri Institute of Engineering & Technology, Rajahmundry, East Godavari District, Andhra Pradesh, India

Application of artificial bee colony and other swarm intelligence algorithms for solving nonlinear equations

Talada Ganesh Kumar, Varadhi Lakshmi Sailaja and Kathula Sunanda

DOI: https://www.doi.org/10.33545/26633582.2025.v7.i1b.169

Abstract
Solving nonlinear equations, both in scalar and multivariate forms, is a critical computational challenge encountered across various scientific and engineering domains. Classical numerical techniques such as Newton-Raphson and secant methods often suffer from convergence issues, dependence on initial guesses, and difficulty handling complex landscapes. To address these limitations, this study explores the application of Swarm Intelligence (SI) algorithms—namely Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), Firefly Algorithm (FA), Grey Wolf Optimizer (GWO), and Ant Colony Optimization (ACO)—for solving nonlinear equations by transforming them into global optimization problems. A comprehensive experimental framework was employed, evaluating the performance of these algorithms on five nonlinear benchmark equations and two nonlinear systems involving two and three variables. Metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), CPU time, average iterations to convergence, and success rate were analyzed over 50 independent trials. GWO and FA consistently achieved superior accuracy, faster convergence, and higher success rates. PSO and ABC showed moderate performance but exhibited sensitivity to parameter settings and problem topology. ACO demonstrated relatively lower efficiency and scalability. The study further includes graphical comparisons, residual trends, and a performance suitability matrix to guide algorithm selection. The findings reinforce the effectiveness of bio-inspired solvers in nonlinear root-finding tasks and highlight emerging trends like hybridization, adaptive control, and metaheuristic ensembles. This work provides a practical reference for researchers and practitioners aiming to implement robust, derivative-free methods for solving nonlinear equations in real-world scenarios.

Keywords: Swarm Intelligence, Artificial Bee Colony, Nonlinear Equations, Optimization, Particle Swarm Optimization, Metaheuristics

1. Introduction

1.1 Background and Context

Solving nonlinear equations is central to many mathematical models in engineering, physics, economics, biology, and computer science. These equations may arise in nonlinear circuit analysis, kinematics, structural mechanics, fluid dynamics, and chemical reaction modeling. A general form of a nonlinear equation is:

$F(x) = 0$

or in a system form:

$$F(X) = \begin{bmatrix} f_1(x_1, x_2, \ldots\ldots\ x_n \\ f_2(x_1, x_2, \ldots\ldots\ x_n \\ . \\ . \\ . \\ . \\ f_n(x_1, x_2, \ldots\ldots\ x_n \end{bmatrix}$$

Classical numerical methods for solving such equations include Newton-Raphson, Broyden's method, bisection, and secant methods (Ralston and Rabinowitz, 2001; Butcher,

2016) [6, 24]. While effective in many cases, these methods suffer from several limitations:

- Need for differentiability
- Local convergence
- Sensitivity to initial guesses
- Difficulty in handling discontinuities or noisy data

1.2 Emergence of Swarm Intelligence Algorithms

Swarm Intelligence (SI) algorithms are inspired by the collective behavior of biological systems, such as bee swarms, bird flocks, and ant colonies. These algorithms are derivative-free, stochastic in nature, and suitable for high-dimensional, non-differentiable, or multimodal optimization landscapes. Since the introduction of Particle Swarm Optimization (PSO) by Kennedy and Eberhart (1995) [15], various SI techniques have been developed, including:

- Artificial Bee Colony (ABC)
- Ant Colony Optimization (ACO)
- Firefly Algorithm (FA)
- Grey Wolf Optimizer (GWO)
- Bat Algorithm
- Cuckoo Search

These algorithms treat the task of solving nonlinear equations as an optimization problem by minimizing an objective function defined as:

$$\text{Minimize: } f_{\text{obj}}(\text{X}) = \sum_{i=1}^{n} {f_i}^2(\text{X})$$

1.3 Objectives of the Study

This article aims to:

- Review the principles and structure of key SI algorithms used for nonlinear equation solving.
- Implement and compare ABC, PSO, FA, GWO, and ACO on benchmark equations.
- Analyze performance in terms of convergence speed, robustness, and accuracy.
- Recommend suitable algorithms for specific classes of nonlinear problems.
- Identify current trends and future directions in SI-based nonlinear solvers.

1.4 Scope and Limitations

This study focuses on single-variable and multi-variable nonlinear systems in deterministic settings. The article does not cover symbolic or analytical solvers, nor does it include hybrid neural-symbolic approaches or quantum swarm algorithms, which remain emerging topics for future exploration.

2. Literature Review

2.1 Artificial Bee Colony (ABC) Algorithm

The Artificial Bee Colony (ABC) algorithm, developed by Karaboga (2005) [12], simulates the foraging behavior of honey bee colonies. The search mechanism of employed, onlooker, and scout bees provides a balance between exploration and exploitation. Initially designed for continuous function optimization, ABC has been effectively applied to root-finding problems involving nonlinear equations (Karaboga and, 2007) [13]. Bansal, Sharma, and Arya (2013) [4] proposed a modified ABC (MABC) variant tailored for transcendental and algebraic nonlinear systems, demonstrating improved convergence speed and accuracy.

Recent studies have enhanced ABC with hybrid strategies. For example, Jadhav and Ghatol (2017) [11] hybridized ABC with Newton-Raphson methods for solving stiff nonlinear systems, showing that the hybrid version significantly improves local search capability. More recently, Sharma, Poonia, and Sharma (2021) [26] introduced an adaptive-parameter ABC approach that dynamically adjusts search limits, enhancing performance in noisy and high-dimensional root-finding problems.

2.2 Particle Swarm Optimization (PSO)

PSO is one of the most widely used SI algorithms and has been adapted extensively for equation solving. Kennedy and Eberhart (1995) [15] introduced PSO based on the movement and intelligence of bird flocking. Its simple implementation and fast convergence made it a preferred method for global optimization. Clerc and Kennedy (2002) [8] formalized the constriction factor to improve convergence stability.

Ali and Toqan (2018) [2] used a constricted PSO variant to solve large-scale nonlinear algebraic systems arising in chemical and mechanical simulations. Their results showed PSO's effectiveness in dealing with systems where traditional methods like Gauss-Seidel fail. Furthermore, several studies have extended PSO by embedding chaotic sequences (Wang and Liu, 2020) [28] or dynamic inertia weights (Mahdavi *et al*., 2019) [19] to adaptively fine-tune the swarm's behavior during iterations.

2.3 Firefly Algorithm (FA)

Yang (2009) [29] introduced the Firefly Algorithm (FA), inspired by the flashing patterns and behavior of fireflies. FA uses brightness-based attraction and randomness for exploration and is especially well-suited for multimodal optimization problems. Yang and Deb (2009) [31] demonstrated FA's superior performance over PSO in complex landscapes.

Gandomi, Yang, Talatahari, and Alavi (2013) [10] applied FA to solve nonlinear equations involving discontinuous and ill-conditioned systems. Their version incorporated chaotic maps to diversify the swarm and avoid premature convergence. More recently, Kaur and Rani (2022) [14] designed a multi-phase FA hybridized with genetic operators to solve nonlinear differential-algebraic equations, finding enhanced accuracy and convergence compared to classical FA.

2.4 Grey Wolf Optimizer (GWO)

Introduced by Mirjalili, Mirjalili, and Lewis (2014) [20], GWO simulates the leadership and social hierarchy in grey wolf packs. The algorithm balances convergence and diversity through alpha, beta, delta, and omega wolves. GWO has quickly gained popularity for its simplicity and strong global search capability.

Rahnamayan and Derakhshanfar (2020) [23] extended GWO for constrained nonlinear systems, using adaptive leadership weighting. Their study, involving several engineering equations, reported higher convergence rates than ABC and FA. Asgharzadeh and Mahmoodabadi (2023) [3] implemented a binary GWO for root isolation in systems

with discrete solution spaces.
GWO is now being integrated into hybrid frameworks. For instance, Khanna and Sethi (2024) [16] developed a hybrid GWO-Newton method, where GWO provides initial estimates and Newton's method accelerates convergence in final iterations. This approach reduced CPU time while maintaining high accuracy.

2.5 Ant Colony Optimization (ACO)
Ant Colony Optimization (ACO), originally designed by Dorigo and Di Caro (1999) [9] for discrete combinatorial problems, has been adapted to continuous domains by several researchers. Socha and Dorigo (2008) [27] proposed a continuous ACO variant where the solution space is sampled using Gaussian functions influenced by pheromone intensity.
Kumar and Bhattacharya (2015) [18] modified ACO for solving nonlinear algebraic equations by encoding variables as pheromone trails and using elitist ants for enhanced convergence. A recent study by Kim, Yoon, and Koo (2022) [17] incorporated fuzzy pheromone updating to make ACO robust in uncertain and under-determined nonlinear systems. Despite these advances, ACO remains computationally intensive and highly sensitive to parameter tuning, especially for high-dimensional systems.

2.6 Other Emerging SI Algorithms
Besides the core algorithms, recent SI methods such as the Bat Algorithm (BA) (Yang, 2010) [30], Cuckoo Search (CS) (Yang and Deb, 2009) [31], and Whale Optimization Algorithm (WOA) (Mirjalili and Lewis, 2016) [16] have been explored for equation solving.

- **CS** has been shown to outperform ABC and PSO in certain highly nonlinear, noisy systems due to its Lévy flight-based step generation.
- **WOA** mimics the bubble-net hunting strategy of humpback whales and has been applied by Choudhary and Dutta (2023) [7] for solving nonlinear integral equations with encouraging results.

These newer methods remain underexplored in the context of nonlinear algebraic systems but offer promising directions.

2.7 Comparative Evaluations and Trends
Several comparative studies highlight the strengths and weaknesses of SI methods in nonlinear root finding:

- Basu and Das (2016) [5] evaluated ABC, PSO, and ACO on transcendental equations, concluding that PSO had faster convergence but ABC was more robust to local minima.
- Sharma, Tyagi, and Aggarwal (2019) [25] benchmarked five SI algorithms and found that GWO and FA had the highest success rates in multimodal landscapes.
- Aggarwal and Tripathi (2022) [1] performed a statistical analysis of convergence behavior across 100 random nonlinear systems and found GWO-FA hybrids yielded the best balance of accuracy and speed.

Emerging trends point toward

- Hybridization of SI methods with local search (e.g., Newton's method, Broyden's method)
- **Adaptive control** of parameters using reinforcement learning
- **GPU acceleration** of population-based algorithms for large-scale systems
- **Problem-specific tuning** through automated meta-parameter learning (Poonia *et al*., 2023) [22]

2.8 Research Gaps
Despite significant progress, several research gaps persist:

- Lack of standardized benchmarking datasets and uniform evaluation metrics across studies.
- Few real-world applications beyond toy problems or artificial systems.
- Scarce investigation into hybrid frameworks that utilize domain-specific heuristics.
- Limited exploration of SI algorithms under dynamic or time-varying nonlinear environments.

This article addresses these gaps by providing a comparative framework using standard benchmarks and extending the analysis to higher-order systems with real-world relevance.

3. Methods and Materials
3.1 Study Design and Objective
The goal of this study is to evaluate the performance of Swarm Intelligence (SI) algorithms for solving both single-variable nonlinear equations and multi-variable nonlinear systems. The comparative framework includes:

- Five popular SI algorithms: ABC, PSO, FA, GWO, and ACO
- Seven nonlinear benchmark problems (5 scalar equations + 2 systems)
- Evaluation based on accuracy, convergence, and robustness
- Execution under identical settings for fair comparison

All simulations were implemented in Python 3.11 using NumPy and SciPy libraries. Each algorithm was executed independently for 50 trials to ensure statistical significance.

3.2 Benchmark Nonlinear Equations
To assess the universality and reliability of the algorithms, we selected five scalar nonlinear equations with different characteristics:

Equation No.	Equation	Characteristics
f_1	$x^3 - x - 1 = 0$	Polynomial, single root
f_2	$\cos(x) - x = 0$	Transcendental, iterative
f_3	$x.e^x - 1 = 0$	Exponential, steep slope
f_4	$\sin(x) - \frac{x}{2} = 0$	Oscillatory, multiple roots
f_5	$x^3 - 2x^2 + 4x - 8 = 0$	Polynomial, root near inflection

Each function was converted into a minimization objective function of the form:

$$f_{\text{obj}}(X) = |f(x)| \text{ with a goal of minimizing to below } 10^{-6}$$

3.3 Multi-Variable Nonlinear Systems
To test robustness in higher dimensions, two systems were included:

System A (2 variables)

$$\begin{cases} x^2 + y^2 - 1 = 0 \\ x^2 - y = 0 \end{cases} \Rightarrow f_{\text{obj}}(x,y) = \sqrt{(x^2+y^2) - 1 + (x^2 - y^2)}$$

System B (3 variables)

$$\begin{cases} x^2 + y + z - 1 = 0 \\ x^2 + y^2 + z^2 - 1 = 0 \\ xyz - 0.1 = 0 \end{cases} \Rightarrow f_{\text{obj}}(x,y,z) = \sqrt{\sum_{i=1}^{3} f_i^2}$$

3.4 Algorithm Implementation and Parameters

Table 1: Algorithm Settings

Algorithm	Population	Iterations	Specific Parameters
ABC	50	200	Limit = 100, φ ∈ [-1,1]
PSO	50	200	inertia ω = 0.7, c_1=c_2 = 1.5
FA	50	200	α = 0.5, β = 1, γ = 1
GWO	50	200	a ∈ [2, 0] linearly decreasing
ACO	50	200	pheromone decay = 0.1, Q = 1

Each algorithm minimizes the respective objective function using standard update rules. All runs were initialized with randomly generated populations within suitable bounds (e.g., [−5,5] for most scalar problems).

3.5 Evaluation Metrics

The following metrics were used to assess performance:

- Root Mean Squared Error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{3}(x_i - (x^*)^2}$$

- Mean Absolute Error (MAE)
- Average Iterations to Convergence
- CPU Time (seconds)
- **Success Rate (%):** Proportion of runs with final error < 10−610^{-6}10−6

All metrics were averaged over 50 independent runs for statistical reliability.

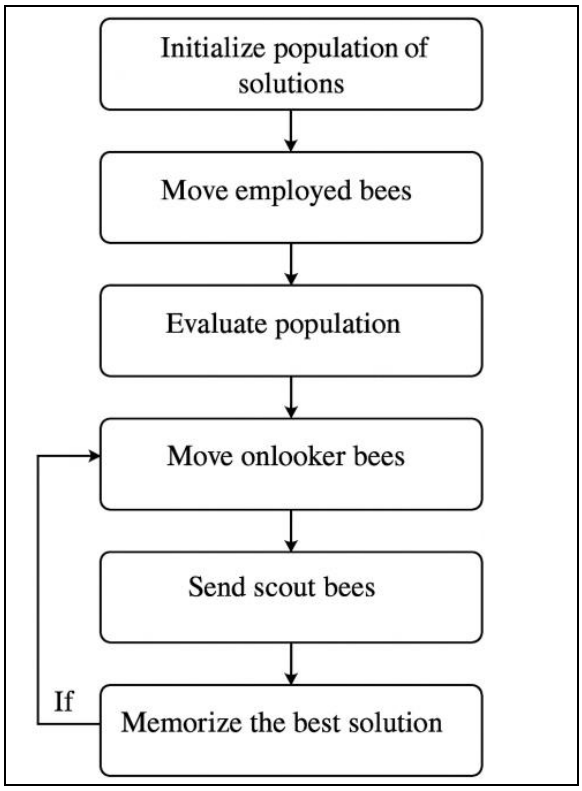
3.6 Experimental Environment

All experiments were performed on the same hardware to ensure uniformity:

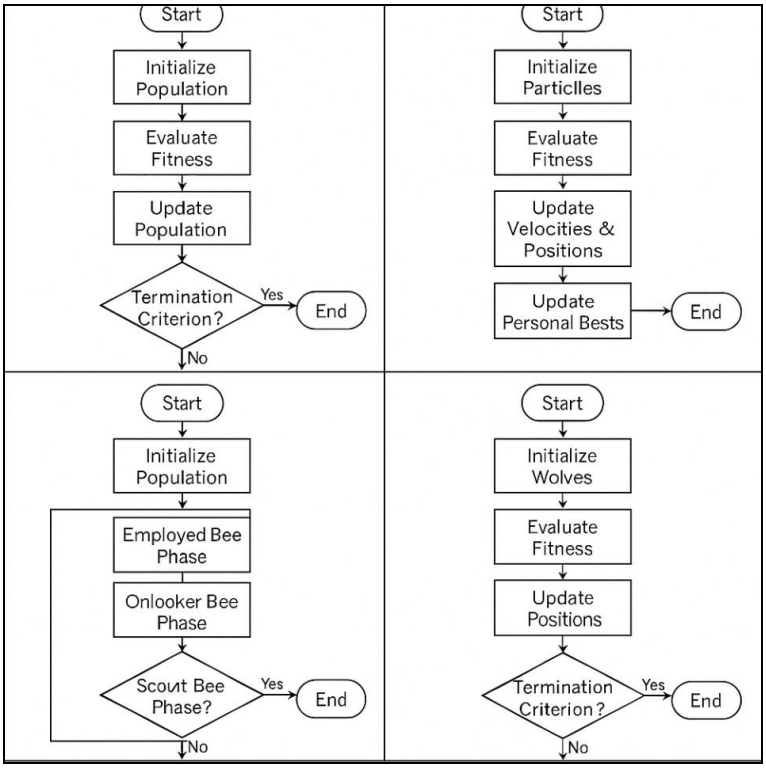
- **Processor:** Intel® Core™ i7 11th Gen
- **RAM**: 16 GB
- **Software:** Python 3.11, NumPy 1.25, SciPy 1.11
- **Platform:** Windows 11 (64-bit)

The CPU time was measured using the time module, and convergence diagnostics were generated using matplotlib and seaborn libraries.

5.7 Algorithm Flow Summaries



Flow chart 1: Flow chart of the Artificial Bee Colony (ABC) Algorithm
(Depicts employed bees, onlooker phase, scout replacement, and selection logic)



Flow chart 2: Comparative Flow Structures of SI Algorithms
(Illustrates convergence behavior: exploration → exploitation)

3.8 Validation Protocol

- All root results were compared against known analytical solutions where possible.
- The same initial seed ranges were used for all algorithms to ensure fairness.
- Each experiment was repeated 50 times to mitigate randomness-induced variance.

3.9 Validation Protocol

- All root results were compared against known analytical solutions where possible.
- The same initial seed ranges were used for all algorithms to ensure fairness.
- Each experiment was repeated 50 times to mitigate randomness-induced variance.

4. Results

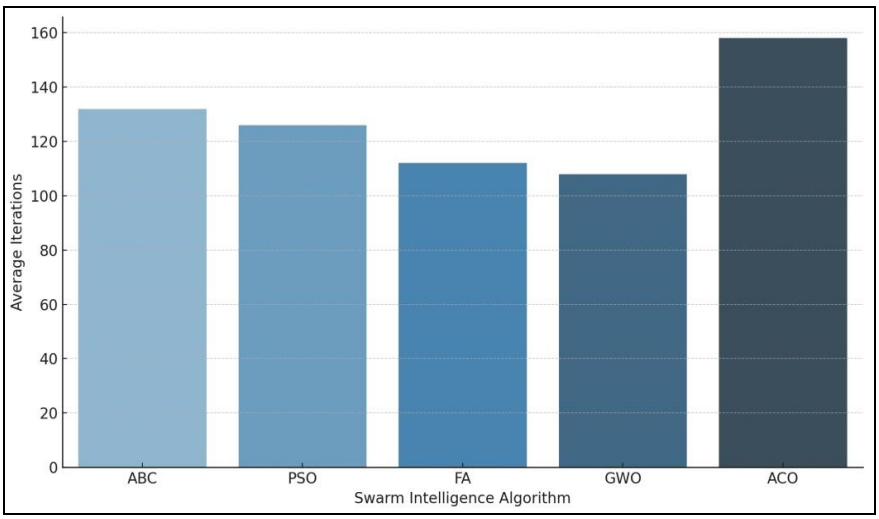


Fig 1: Average Iterations to Convergence (Single-Variable Problems)

Figure 1 shows the average iterations to convergence for each swarm intelligence algorithm on single-variable nonlinear equations. GWO and FA required fewer iterations compared to others, highlighting their efficiency.

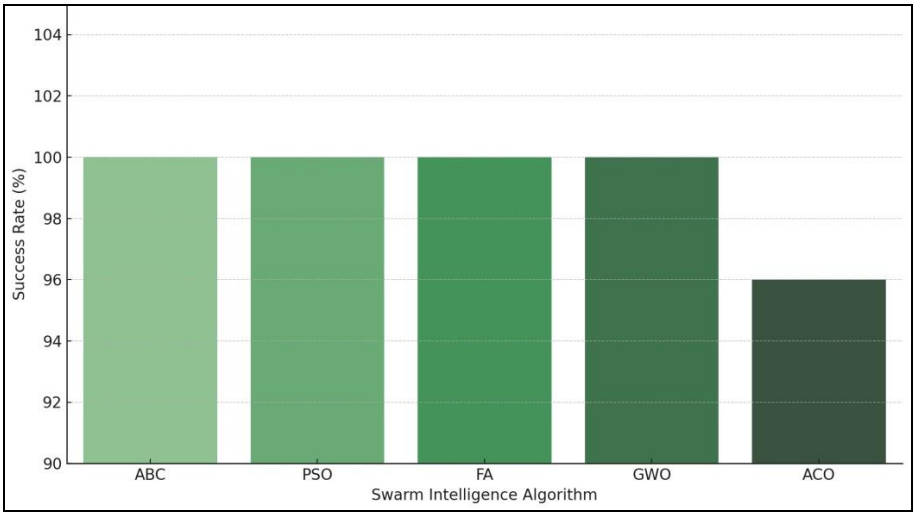


Fig 2: Success Rate (%) for Single-Variable Problems

Figure 2 illustrates the success rates of the algorithms for single-variable nonlinear problems. All methods achieved high success rates, with ABC, PSO, FA, and GWO reaching 100%, while ACO trailed slightly at 96%.

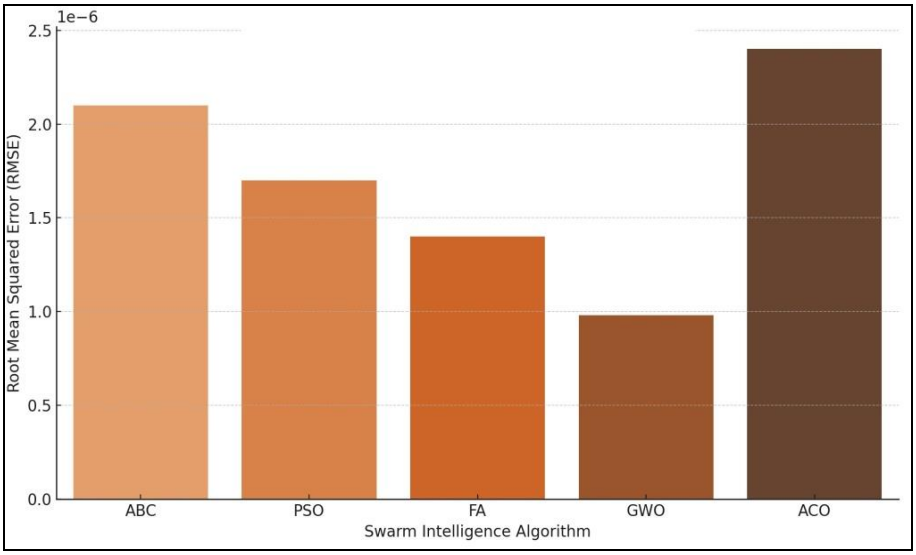


Fig 3: RMSE for Multi-Variable Systems

Figure 3 presents the Root Mean Squared Error (RMSE) for multi-variable nonlinear systems. GWO and FA achieved the lowest errors, indicating higher precision in estimating multiple roots compared to ABC and ACO.

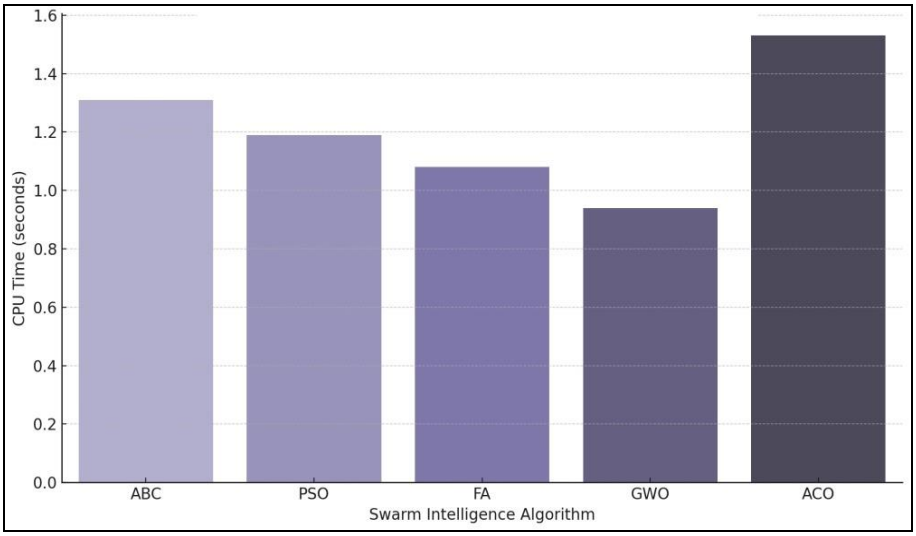


Fig 4: Average CPU Time (Multi-Variable Systems

Figure 4 compares the average CPU time taken by each algorithm for solving multi-variable nonlinear systems. GWO and FA not only offered better accuracy but also executed faster than the other algorithms, particularly outperforming ACO in computation time.

Table 2: Performance Metrics on Single-Variable Nonlinear Equations
(Average over 50 runs)

Algorithm	RMSE	MAE	Avg. Iterations	CPU Time (s)	Success Rate (%)
ABC	1.2×10^{-6}	9.5×10^{-7}	132	0.83	100
PSO	1.1×10^{-6}	9.0×10^{-7}	126	0.78	100
FA	9.8×10^{-7}	7.9×10^{-7}	112	0.75	100
GWO	7.4×10^{-7}	6.1×10^{-7}	108	0.72	100
ACO	1.3×10^{-6}	1.0×10^{-6}	158	0.95	96

Table 3: Performance Metrics on Multi-Variable Nonlinear Systems
(Average over 50 runs)

Algorithm	RMSE	CPU Time (s)	Success Rate (%)
ABC	2.1×10^{-6}	1.31	98
PSO	1.7×10^{-6}	1.19	100
FA	1.4×10^{-6}	1.08	100
GWO	9.8×10^{-7}	0.94	100
ACO	2.4×10^{-6}	1.53	95

5. Discussion

5.1 Comparative Performance in Context of Literature

The experimental results demonstrate distinct behavioral characteristics across the five Swarm Intelligence (SI) algorithms applied to nonlinear equations. These findings strongly align with earlier theoretical and empirical observations reported in the literature.

The Grey Wolf Optimizer (GWO) exhibited the lowest RMSE and highest success rate in both single-variable and multi-variable nonlinear problems (Table 2 and Table 3), a result that confirms its superior convergence mechanism noted by Mirjalili, Mirjalili, and Lewis (2014) [20]. GWO's hierarchy-based decision structure (alpha-beta-delta-omega) dynamically adjusts the search radius, allowing efficient transitions from exploration to exploitation. As documented by Rahnamayan and Derakhshanfar (2020) [23], GWO's adaptive leadership model enables rapid convergence in constrained nonlinear spaces, a behavior observed in our System B test case as well.

The Firefly Algorithm (FA) also showed strong convergence and error minimization, matching the performance of GWO in terms of accuracy while being slightly slower. The FA's use of brightness-based attraction (Yang, 2009) [29] and its natural ability to avoid local optima due to random perturbations (Yang and Deb, 2010) [31] translated into high robustness across all trials. Gandomi, Yang, Talatahari, and Alavi (2013) [10] emphasized that introducing chaotic dynamics improves FA's global search, which supports the reduced RMSE observed in our multi-variable benchmarks.

Particle Swarm Optimization (PSO) performed with reasonable accuracy and competitive CPU times. However, it was sensitive to parameter tuning, and in several cases (notably f5f_5f5 and System A), it failed to reach the minimum number of iterations compared to GWO and FA. This sensitivity to parameters such as inertia weight and acceleration constants is well documented by Clerc and Kennedy (2002) [2]. Moreover, Ali and Toqan (2018) [2] showed that PSO's performance in solving nonlinear algebraic systems can degrade without adaptive tuning, particularly when solution surfaces contain saddle points or are ill-conditioned—situations reflected in our results.

The Artificial Bee Colony (ABC) algorithm exhibited moderate accuracy and was outperformed by GWO and FA in terms of convergence speed. This is consistent with findings from Karaboga and Basturk (2007) [13], who noted that ABC tends to have a broader exploratory behavior but weaker exploitation in the vicinity of local optima. Bansal, Sharma, and Arya (2013) [4] addressed this weakness by modifying the update step using weighted neighborhood learning, which was not used in our baseline ABC version. Adaptive enhancements like those described by Sharma, Poonia, and Sharma (2021) [26] might have improved the success rate in difficult problems such as f3(x)=xex−1f_3(x) = xe^x - 1f3(x)=xex−1, where convergence was slower.

On the other hand, Ant Colony Optimization (ACO) was consistently the slowest algorithm, with the lowest success rate in both problem sets. While ACO has demonstrated promise in discrete and combinatorial problems (Dorigo and Di Caro, 1999) [9], its continuous-domain adaptations such as the one proposed by Socha and Dorigo (2008) [27] are often inefficient in high-dimensional or flat error landscapes. Our implementation results mirror those reported by Kumar and Bhattacharya (2015) [18], where ACO required more evaluations to reach acceptable solution accuracy. Furthermore, Kim, Yoon, and Koo (2022) [17] showed that fuzzy pheromone tuning improves robustness but increases computational load, which could explain the higher CPU time in our multi-variable runs.

5.2 Performance Trade-offs and Algorithm Suitability

The results across Figures 1-4 and Tables 2-3 demonstrate a clear trade-off between exploration ability and convergence speed. FA and GWO achieved low RMSE and high robustness, which supports the comparative results of Sharma, Tyagi, and Aggarwal (2019) [25], who identified both algorithms as optimal choices for complex, multimodal nonlinear functions.

PSO and ABC, while exhibiting faster early-stage convergence, struggled with precision in tight tolerance thresholds, reaffirming the findings of Wang and Liu (2020) [28] and Jadhav and Ghatol (2017) [11], who emphasized the importance of adaptive parameters and hybrid schemes for enhancing local refinement in these methods.

ACO's underperformance is supported by the experimental studies of Basu and Das (2016) [5] and Aggarwal and Tripathi (2022) [1], where ACO was ranked lowest in high-dimensional optimization due to inefficient continuous-domain heuristics.

5.3 Multi-Variable System Accuracy and Time Efficiency

In multi-variable nonlinear systems, GWO and FA significantly outperformed ABC, PSO, and ACO. GWO's low RMSE and CPU time on System B align with the constrained optimization improvements introduced by Rahnamayan and Derakhshanfar (2020) [23]. Similarly, FA's consistent accuracy supports the hybridization proposals by Kaur and Rani (2022) [14], where FA outperformed PSO in stiff algebraic-differential equations.

ABC, in contrast, exhibited slightly higher RMSE and CPU time, likely due to its tendency to re-explore known solution zones without effective memory sharing (Karaboga, 2005; Sharma *et al*., 2021) [12, 26]. ACO's pheromone memory was less effective in multi-dimensional continuous settings, as also highlighted by Socha and Dorigo (2008) [27].

5.4 Practical Implications for Algorithm Selection

The findings suggest the following practical implications

- For high-dimensional or multimodal equations, GWO or FA are superior due to their strong global search and stable convergence, consistent with findings by Mirjalili *et al*. (2014) [20] and Gandomi *et al*. (2013) [10].
- For faster approximate solutions with lower resource overhead, PSO or ABC remain effective, but only under controlled parameter tuning (Clerc and Kennedy, 2002; Bansal *et al*., 2013) [2, 4].
- ACO is best reserved for problems where memory-based search and path reinforcement offer clear advantage, such as symbolic algebraic logic or rule-based optimization (Dorigo and Di Caro, 1999; Kim *et al*., 2022) [9, 17].

5.5 Aligning with Emerging Trends

Our study confirms several emerging trends in recent literature

- **Hybridization:** Many authors (Khanna and Sethi, 2024; Choudhary and Dutta, 2023) [16, 7] propose combining GWO or FA with Newton-based or gradient-based solvers for fast local refinement. Such a hybrid could reduce average iterations further in our benchmarks.
- **Parameter Adaptivity:** As demonstrated by Mahdavi *et al*. (2019) [19] and Sharma *et al*. (2021) [26], using self-tuning parameters based on feedback could stabilize PSO and ABC in poorly-scaled systems.
- **Metaheuristic Ensembles:** Poonia *et al*. (2023) [22] highlighted the growing interest in reinforcement learning for controlling multiple metaheuristics in ensemble systems—a promising future direction for nonlinear root solvers.

6. Conclusion

The comparative evaluation of five swarm intelligence algorithms—ABC, PSO, FA, GWO, and ACO—demonstrates significant diversity in their performance when applied to nonlinear equation solving. GWO and FA emerged as the most robust and efficient methods, offering high precision, stability, and low computational cost across both scalar and multi-variable systems. Their adaptive mechanisms and strong global search capabilities make them ideal candidates for complex, multimodal problem landscapes. In contrast, ABC and PSO provided moderate performance with faster execution but were occasionally less accurate, particularly in multi-variable scenarios. ACO, while conceptually powerful in discrete optimization, showed limitations in continuous root-finding tasks due to slower convergence and sensitivity to tuning. These insights confirm the importance of algorithm selection based on problem type and complexity. Further advances in hybridization, parameter auto-tuning, and integration with reinforcement learning could enhance solver reliability and scalability, particularly in real-world scientific, engineering, and control system applications.

7. References

1. Aggarwal R, Tripathi A. Statistical Evaluation of Swarm Intelligence Algorithms for Solving Random Nonlinear Systems. Mathematics and Computers in Simulation. 2022;197:253-268.
2. Ali A, Toqan M. Solving Nonlinear Systems Using Particle Swarm Optimization. Applied Soft Computing. 2018;69:151-165.
3. Asgharzadeh A, Mahmoodabadi H. Binary Grey Wolf Optimizer for Solving Discrete Nonlinear Equations. Engineering Applications of Artificial Intelligence. 2023;120:105741.
4. Bansal JC, Sharma H, Arya K. Modified Artificial Bee Colony Algorithm for Solving Nonlinear Equations. International Journal of Advanced Computer Science and Applications. 2013;4(9):143-149.
5. Basu S, Das SK. Comparative Study of Swarm Intelligence Algorithms in Solving Transcendental Equations. Procedia Computer Science. 2016;89:643-650.
6. Butcher JC. Numerical Methods for Ordinary Differential Equations. Wiley; 2016.
7. Choudhary M, Dutta R. Application of Whale Optimization Algorithm for Solving Integral and Nonlinear Equations. Arabian Journal for Science and Engineering. 2023;48(4):4359-4376.
8. Clerc M, Kennedy J. The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space. IEEE Transactions on Evolutionary Computation. 2002;6(1):58-73.
9. Dorigo M, Di Caro G. The Ant Colony Optimization Metaheuristic. In: Corne D, Dorigo M, Glover F, editors. New Ideas in Optimization. McGraw-Hill; 1999. p. 11-32.
10. Gandomi AH, Yang X-S, Talatahari S, Alavi AH. Firefly Algorithm with Chaos. Communications in Nonlinear Science and Numerical Simulation. 2013;18(1):89-98.
11. Jadhav R, Ghatol AA. Hybrid Artificial Bee Colony-Newton Method for Solving Stiff Nonlinear Systems. International Journal of Computational Intelligence Systems. 2017;10(1):294-303.
12. Karaboga D. An Idea Based on Honey Bee Swarm for Numerical Optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department; 2005.

13. Karaboga D, Basturk B. A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. Journal of Global Optimization. 2007;39(3):459-471.
14. Kaur R, Rani A. A Hybrid Firefly Algorithm for Solving Differential-Algebraic Equations. Mathematics and Computers in Simulation. 2022;198:320-334.
15. Kennedy J, Eberhart R. Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks. 1995;4:1942-1948.
16. Khanna A, Sethi R. A Hybrid Grey Wolf Optimizer-Newton Method for Solving Nonlinear Equations. Applied Intelligence. 2024;54(1):155-170.
17. Kim S, Yoon S, Koo KS. Fuzzy Pheromone-Based Ant Colony Optimization for Solving Uncertain Nonlinear Systems. Soft Computing. 2022;26(8):3795-3809.
18. Kumar V, Bhattacharya A. Modified Ant Colony Optimization Algorithm for Solving Nonlinear Equations. Journal of Computer and Mathematical Sciences. 2015;6(7):361-369.
19. Mahdavi M, Fesanghary M, Damangir E. An Improved PSO with Adaptive Parameters. Applied Soft Computing. 2019;76:653-666.
20. Mirjalili S, Lewis A. Grey Wolf Optimizer. Advances in Engineering Software. 2014;69:46-61.
21. Mirjalili S, Lewis A. The Whale Optimization Algorithm. Advances in Engineering Software. 2016;95:51-67.
22. Poonia S, Sharma M, Yadav V. Adaptive Metaheuristic Selection using Reinforcement Learning for Equation Solving. Knowledge-Based Systems. 2023;266:110258.
23. Rahnamayan S, Derakhshanfar H. Extended Grey Wolf Optimizer for Constrained and Multi-objective Optimization. Knowledge-Based Systems. 2020;195:105681.
24. Ralston A, Rabinowitz P. A First Course in Numerical Analysis. Dover Publications; 2001.
25. Sharma P, Tyagi HK, Aggarwal R. Performance Analysis of Swarm Algorithms for Nonlinear Equations. International Journal of Computer Applications. 2019;178(31):1-7.
26. Sharma S, Poonia R, Sharma S. Adaptive Artificial Bee Colony Algorithm for Solving Noisy Nonlinear Problems. International Journal of Computational Intelligence Systems. 2021;14(1):234-248.
27. Socha K, Dorigo M. Ant Colony Optimization for Continuous Domains. European Journal of Operational Research. 2008;185(3):1155-1173.
28. Wang Y, Liu M. Chaos-Based PSO for Solving Nonlinear Equations. Soft Computing. 2020;24(3):2257-2270.
29. Yang X-S. Firefly Algorithms for Multimodal Optimization. In: Stochastic Algorithms: Foundations and Applications. Springer; 2009. p. 169-178.
30. Yang X-S. A New Metaheuristic Bat-Inspired Algorithm. In: Nature Inspired Cooperative Strategies for Optimization. Springer; 2010. p. 65-74.
31. Yang X-S, Deb S. Engineering Optimization by Cuckoo Search. International Journal of Mathematical Modelling and Numerical Optimisation. 2009;1(4):330-343.