JAM CM
Journal of Applied Mathematics and Computational Mechanics 2026, 25(3), 20-46
www.amcm.pcz.pl p-ISSN 2299-9965
DOI: 10.17512/jamcm.2026.3.02 e-ISSN 2353-0588

CPSOA: A HYBRID METAHEURISTIC FOR SOLVING OPTIMIZATION PROBLEMS – A CASE STUDY ON SYSTEMS OF NONLINEAR EQUATIONS

Mohammed A. El-Shorbagy[1]*, Abeer A. Ahmed[2]

[1] Department of Mathematics, College of Science and Humanities in Al-Kharj
Prince Sattam bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[2] Department of Mathematics, Faculty of Science
Aswan University, Aswan 81528, Egypt
mohammed_shorbagy@yahoo.com

Received: 21 December 2025; Accepted: 1 July 2026

Abstract. Many scientific and engineering models require solving systems of nonlinear equations (SNLEs), and significant advances in these fields depend on solving such systems accurately and efficiently. Our research proposes a hybrid intelligent algorithm to tackle SNLEs. The chaotic particle swarm optimization algorithm (CPSOA) combines the particle swarm optimization algorithm (PSOA) with the chaotic search technique (CST). Its goal is to improve performance, increase solution diversity, accelerate convergence, avoid premature convergence to local optima, and optimize the search process. Reformulating SNLEs as an optimization problem is the initial step. Within the CPSOA framework, CST updates infeasible solutions while PSOA updates feasible ones. Since infeasible solutions are often located near the optimal solution, CPSOA optimizes them rather than dismissing them, thereby improving search efficiency and achieving an effective exploitation-exploration balance. Fifty benchmark functions were used to assess convergence speed and solution quality for the proposed algorithm. Four benchmark systems of nonlinear equations with different dimensions were also used to evaluate the accuracy, robustness, and stability of CPSOA. The study validated performance using the non-parametric Friedman and Wilcoxon tests. Experimental results show that CPSOA outperforms existing techniques in solving a wide range of optimization problems and systems of nonlinear equations.

MSC 2010: *65K10, 68T20*
Keywords: *particle swarm optimization algorithm, chaotic search technique, system of nonlinear equations, hybrid intelligent algorithm, optimization*

1. Introduction

Systems of nonlinear equations (SNLEs) arise frequently in real-world applications across various fields, including economics, chemistry, and physics [1, 2], where precise and effective solutions are crucial for modeling, forecasting, and analysis.

*Corresponding author

However, solving SNLEs remains a fundamentally challenging task, as nonlinear systems are often highly sensitive, computationally demanding, and may admit multiple or no solutions.

Historically, researchers have employed derivative-based approaches, especially Newton-type algorithms, to solve SNLEs. These approaches are widely used; however, they have several major drawbacks. The quality of the initial guess significantly affects their performance, and they typically fail when derivatives are difficult to compute, do not exist, or are computationally expensive. Moreover, poor initialization can cause the algorithm to diverge or converge to incorrect solutions. Bader [3] points out that the Newton-Raphson approach also requires substantial memory, which limits its applicability to large systems. To overcome these constraints, a tensor-based methodology that utilizes Krylov subspace techniques is introduced for the resolution of nonlinear systems. Additional strategies have been examined, including interval methods [4, 5] and continuation methods [6]. Interval methods are recognized for their robustness and guaranteed convergence; yet, they tend to be inefficient and less applicable for large-scale problems. Conversely, continuation approaches tend to be effective primarily for low-dimensional problems [7].

Despite extensive research, developing efficient, scalable, and robust methods for solving SNLEs remains challenging. Each existing method possesses distinct advantages and disadvantages regarding accuracy, cost, scalability, and resilience. Consequently, substantial opportunities remain for developing algorithms that more effectively balance exploration and exploitation, accommodate high-dimensional systems, and deliver fast, reliable convergence. This growing demand has prompted the exploration of intelligent and hybrid metaheuristic strategies, designed to rectify the deficiencies of conventional methods while broadening their applicability in complex, real-world domains.

Metaheuristic algorithms have emerged as efficient techniques for solving systems of nonlinear equations (SNLEs). Their flexibility and adaptability allow them to tackle complex, nonlinear, and multidimensional problems more effectively than traditional numerical methods. Unlike conventional deterministic approaches, metaheuristics exhibit a lower propensity to become trapped in local optima and can provide near-optimal solutions within a reasonable computational time. A significant trend in this domain is the integration of conventional numerical methods with metaheuristic procedures through hybridization. This combination aims to capitalize on the strengths of both methodologies: classical approaches provide strong local search capabilities and rapid exploitation, while metaheuristics enhance global exploration. Hybrid algorithms effectively address major challenges such as premature convergence, sensitivity to initial conditions, and slow convergence rates by achieving a better balance between exploration and exploitation.

Hybrid metaheuristic methods have proven highly effective in improving solution quality, enhancing robustness, and increasing reliability in solving SNLEs across a wide range of application domains. For example, Karr et al. [8] introduced a hybrid Newton-genetic algorithm (GA) approach, in which GA is employed to generate high-quality initial values that guide Newton's method during the local search process. Wang et al. [9] introduced an enhanced particle swarm optimization (PSO) integrated with a controller, aimed at improving the search dynamics and convergence characteristics of the traditional PSO. Ouyang et al. [10] combined PSO with the Nelder-Mead method, employing PSO to determine suitable initial values while the Nelder-Mead method performs local optimization. Nevertheless, this approach remains susceptible to convergence to local minima.

Subsequent developments include the work of Jia and He [11], who combined PSO with an artificial bee colony (ABC) to enhance global search capability, and the work of Yang et al. [12], who integrated the Hooke-Jeeves method with glowworm swarm optimization (GSO) to accelerate the local search process. Hirsch et al. [13] presented the continuous greedy randomized adaptive search procedure (C-GRASP), which proved effective in solving nonlinear systems through adaptive search strategies. In addition, Mo et al. [14] investigated the challenges inherent in high-dimensional optimization problems and observed that conventional PSO often suffers from slow convergence and premature stagnation. The conjugate direction approach was incorporated into the PSO framework to improve scalability and convergence efficiency. This integration resulted in the conjugate direction PSO (CDPSO), which combines the global search capability of PSO with the systematic methodology of conjugate directions to enhance solution quality. Furthermore, Toutounian et al. [15] proposed a hybrid optimization technique that integrates electromagnetic metaheuristics (EM) with the Newton-GMRES algorithm, effectively combining stochastic exploration with deterministic refinement. Finally, Sacco and Henderson [16] introduced a two-phase hybrid approach to address optimization problems characterized by complex landscapes and numerous local optima.

On the other hand, various standalone metaheuristic algorithms have been employed to solve SNLEs, including the genetic algorithm (GA) [17], evolutionary algorithm (EA) [18], firefly algorithm (FA) [19], artificial bee colony (ABC) [20], invasive weed optimization (IWO) [21], imperialist competitive algorithm (ICA) [22], and PSO [23].

The chaotic search technique (CST) is a mathematical method commonly employed to improve the performance of various optimization algorithms. This approach is based on simulating the dynamic behavior of nonlinear systems [24]. The versatility and efficiency of CST have attracted considerable attention, leading to its application in diverse areas such as chaos control and optimization research [25].

Numerous studies have demonstrated the effectiveness of incorporating chaos theory into metaheuristic algorithms [26], and the resulting approaches differ significantly across the literature. However, none of the existing methodologies employs the same structural mechanism as the proposed CPSOA. The hybrid chaos and quasi-Newton approach [27] is one of the earliest efforts in this field, wherein chaotic optimization functions solely as a preliminary global search mechanism, succeeded by a quasi-Newton method for local convergence. This method is devoid of swarm intelligence and fails to differentiate between feasible and non-feasible solutions.

Recently, chaotic noise-based PSOA (CN-BPSOA) [28] incorporated chaotic noise – derived from the logistic map – directly into the PSO update equations. This noise is consistently applied to all particles, irrespective of feasibility, mainly to minimize solution duplication and expedite convergence.

In addition to conventional equation solving, chaotic maps have been integrated into multiple optimization frameworks for varied applications. The chaotic vortex search algorithm [29] employs various chaotic maps, notably the Tent map, throughout all search operations to enhance feature selection; however, it does not incorporate a method for addressing infeasible solutions. The chaotic evolution optimization algorithm [30] utilises a two-dimensional discrete memristive map to incorporate chaotic dynamics into a differential evolution framework. Nonetheless, it neither utilises PSO nor addresses nonlinear equation systems.

The binary chaos-based olympiad optimization algorithm [31] utilises chaos in a binary context for software feature selection, a challenge distinctly separate from systems of nonlinear equations (SNLEs). The Levy chaotic particle swarm optimization-based cluster routing protocol (LCPSO-CRP) [32] integrates Levy flight with chaotic techniques to enhance energy-efficient cluster routing in wireless sensor networks, a domain distinct from mathematical problem resolution. Moreover, several chaos-based clustering algorithms [33] limit the use of the logistic map to the initialisation phase, resulting in chaos being dormant during the real search operation. In educational technology, chaotic particle swarm optimization (C-PSO) [34] integrates the Henon map without differentiating between feasible and infeasible particles. Likewise, the dynamic self optimization chaotic particle swarm optimization (DSCPSO) method [35] integrates Tent chaos into the adaptive updates of the inertia weight and learning factors, introducing uniform perturbations to enhance motor parameter estimation. Finally, the opposition-based learning adaptive chaotic PSO (LCPSO) [36] employs chaotic elite opposition-based learning exclusively during initialisation to augment population variety.

The main feature of CPSOA, compared with existing approaches, is its structured dual-path strategy. In the proposed algorithm, standard PSO is employed to update feasible solutions, whereas CST is selectively used to refine infeasible ones, rather than applying chaotic perturbations uniformly to all particles or restricting chaos to the initialization stage. This deliberate separation enables the algorithm to preserve diversity, escape local optima, and effectively exploit high-quality feasible regions. This distinctive infeasible-solution handling mechanism constitutes the principal novelty and core contribution of the proposed CPSOA algorithm for solving systems of nonlinear equations.

Research findings suggest that hybridization with chaotic maps decreases computational time and improves convergence toward global optima. Numerous real-world optimization problems involve constraints, making the administration of constraints a fundamental aspect of algorithm development [37]. Our proposed methodology entails the proactive refinement of infeasible solutions by CST rather than their elimination, thereby enhancing the overall efficacy of the search process.

Particle swarm optimization (PSO) [38], a prominent population-based metaheuristic, has demonstrated robust performance in tackling complex optimization tasks. Nevertheless, its limitations have motivated the development of numerous improved or hybrid PSO variants. Accordingly, a novel hybrid method known as the chaotic PSOA (CPSOA) is introduced in this paper for addressing systems of nonlinear equations (SNLEs), specifically developed to augment solution diversity by incorporating chaotic search mechanisms, attain an effective equilibrium between exploration and exploitation, and enhance overall performance by strategically retaining and employing infeasible solutions throughout the search process. Moreover, CPSOA achieves enhanced solution quality and expedited convergence rates compared to current methodologies. The study includes thorough benchmarking against recognized metaheuristic algorithms, stringent statistical validation of outcomes, and numerical evaluation of the algorithm's efficacy across benchmark optimization functions and systems of nonlinear equations.

The remainder of this paper is organized as follows. Section 2 presents the formulation of systems of nonlinear equations and discusses the associated solution challenges. Section 3 reviews the fundamentals of the particle swarm optimization algorithm (PSOA) and the chaotic search technique (CST). Section 4 introduces the proposed chaotic PSOA (CPSOA). Section 5 reports the computational results. Finally, Section 6 concludes the paper and outlines directions for future research.

2. System of nonlinear equations (SNLEs)

A system of nonlinear equations can be mathematically represented as [39]:

$$\begin{aligned} f_1(X) &= 0, \\ f_2(X) &= 0, \\ &\vdots \\ f_m(X) &= 0, \end{aligned} \tag{1}$$

where $X = [x_1, x_2, \ldots, x_n]^T$ represents the vector of unknown variables, and $f_i(X)$ $(i = 1, 2, \ldots, m)$ are nonlinear functions.

Main characteristics of SNLEs include multiple solutions, nonlinearity, sensitivity to initial conditions, and computational complexity [40].

SNLEs can be reformulated as optimization problems, thereby facilitating the application of metaheuristic and evolutionary algorithms. The primary objective is to define a function that quantifies the extent to which a candidate solution satisfies all equations. The system in Eq. (1) can be transformed into the following optimization problem by minimizing the sum of squared residuals [41]:

$$\min F(X) = \sum_{i=1}^{m} [f_i(X)]^2 \tag{2}$$

Important aspects of this formulation are as follows: (1) Objective Function: $F(X) \geq 0$, and the global minimum $F(X^*) = 0$ corresponds to a solution of the original SNLE, (2) Feasible Solutions: Any X that satisfies $F(X) = 0$ is a feasible solution to SNLE, (3) Infeasible Solutions: Candidate solutions for which $F(X) > 0$ are infeasible, but they can still provide valuable guidance toward the optimum, and (4) Advantages: This formulation facilitates the application of metaheuristic optimization algorithms to solve SNLEs, particularly in situations where traditional derivative--based methods are ineffective or computationally expensive, thereby enabling efficient solution of high-dimensional, nonlinear, and multimodal systems.

3. Preliminaries

This section summarizes the PSOA algorithm and CST, emphasizing their core concepts and characteristics that make them suited for the hybrid strategy.

3.1. The main characteristic of PSOA

Bird flocking inspired the population-based optimization method PSOA. PSOA, like evolutionary algorithms (EAs), can solve complex optimization problems faster than EAs. PSO's simplicity allows for easy implementation, requiring only minor parameter adjustments [42]. Within the optimization problem's search space, PSO randomly initializes particles. Each particle evaluates the objective function at its current location and determines its movement direction by combining three main influences: its personal best position, the swarm's optimal position, and a stochastic perturbation that guarantees search diversity. After modifying particle positions, the swarm converges on the fitness function global optimum, like a flock of birds feeding.

Formally, the i-th particle is represented by an n-dimensional position vector as $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, with an associated velocity vector $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$. The best position previously visited by particle i is denoted as $p_i^{\text{best}} = (p_{i1}, p_{i2}, \ldots, p_{in})$, while the best position discovered by the swarm is expressed as $g^{\text{best}} = (g_1, g_2, \ldots, g_n)$. The PSO method's core idea is represented in the following equations, which change the velocity and position of the swarm's i-th particle in each iteration. The particle's movement is controlled by the velocity update equation, which is composed of three parts: the particle's previous velocity, its best-known position, and the best position found by the swarm. In mathematical terms, this can be expressed as follows:

$$v_i^{(t+1)} = wv_i^t + c_1 r_1 \left(p_i^{(\text{best},t)} - x_i^t\right) + c_2 r_2 \left(g^{\text{best}} - x_i^t\right), \tag{3}$$

$$x_i^{(t+1)} = x_i^t + v_i^{(t+1)}; \tag{4}$$

where t is the number of iterations, w is the inertia weight, c_1 and c_2 are acceleration coefficients, and $r_1, r_2 \sim U(0,1)$ are random numbers uniformly distributed between 0 and 1.

3.2. Chaotic search technique (CST)

CST studies nonlinear dynamical systems. Despite deterministic laws, these systems often exhibit stochastic behavior. CST is effective for modeling and enhancing diverse systems in physics, robotics, microbiology, and computer science due to its inherent randomness [43]. The effectiveness of CST stems from three chaos theory principles: (1) Sensitivity to initial conditions, where small changes can lead to diverse system trajectories, increasing search process diversity; (2) quasi-stochastic behavior allows chaotic sequences to mimic randomness, enabling the use of chaotic variables in optimization while maintaining deterministic control; and (3) ergodicity ensures non-repetitive navigation of all attainable states within a specific range, boosting global solution space exploration. These three characteristics improve meta-heuristic optimization algorithms by balancing exploration and exploitation. CST integration enhances algorithms by preventing premature convergence, avoiding local optima, and accelerating global convergence [44].

Chaotic maps are deterministic mathematical functions that exhibit sensitive dependence on initial conditions, producing sequences that appear random yet remain bounded and non-repeating. Due to their ergodicity, unpredictability, and ability to escape local optima, chaotic maps have been widely employed in optimization algorithms to enhance exploration and avoid premature convergence [42]. In this work, the logistic map is adopted as the underlying chaotic mechanism within the CST. The logistic map is defined as:

$$x_{t+1} = cx_t\,(1 - x_t)\,; \tag{5}$$

where $x_0 \in (0, 1)$, $x_0 \notin \{0.0,\ 0.25,\ 0.50,\ 0.75,\ 1.0\}$. At $c = 4.0$, the logistic map operates in a fully chaotic regime, generating sequences with strong chaotic properties that effectively guide the search process across the solution space.

Research has shown that chaotic maps can improve optimization algorithms' efficiency. These maps improve search, prevent premature convergence, and guide algorithms to global solutions. Hybrid intelligence algorithms use chaotic maps to facilitate exploration and exploitation due to their mathematical simplicity and strong ergodic features.

4. Chaotic particle swarm optimization algorithm (CPSOA)

The CPSOA solves systems of nonlinear equations (SNLEs) by integrating PSOA and CST. In CPSOA, PSOA updates candidate solutions (agents) using Eqs. (3) and (4). PSOA is effective, but premature convergence can trap it in local optima.

The conventional PSOA struggles to balance exploitation and exploration, resulting in suboptimal search performance. PSOA leverages CST as an auxiliary search technique to overcome these limitations. PSOA and CST, together with infeasible solutions, improve search quality, solution diversity, local optima avoidance, convergence speed, and overall performance.

CPSOA operates in two phases. First, SNLEs are transformed into optimization problems. PSOA updates feasible solutions, while CST updates infeasible solutions. Allowing infeasible solutions to remain in the population significantly shortens the search process and reduces the number of function evaluations required. Previous studies have used the Logistic map for generating chaotic sequences due to its simplicity and effectiveness [45].

4.1. Procedure of the proposed algorithm

Here is a detailed explanation of the method that has been suggested:

Step 1: Initialization. Initialize parameters for PSOA, initialize randomly NP particles $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, $\forall i = 1, \ldots, NP$ with initial positions $x_i^{(t+1)}$ and velocities $v_i^{(t+1)}$, and $t = 0$.

Step 2: Check the feasibility of particles. Check the feasibility of the particles' positions. Feasible particles are added to the feasible list (FL). Infeasible ones (outside search space) go to infeasible list (IFL).

Step 3: Fitness evaluation. The optimization fitness function with n variables is evaluated for every particle in FL.

Step 4: Setting $\boldsymbol{p}^{(\mathbf{best,t+1})}$ and $\boldsymbol{g}^{\mathbf{best}}$. For each particle in the FL, set $p_i^{(\mathrm{best},t+1)}$ as the current position, and use the best particle from initialization to determine g^{best} and its objective value.

Step 5: Update the particles' positions.

- Update the particle locations in the feasible list (FL) according to Eqs. (3) and (4).
- Generate the chaotic numbers using the following logistic map:

$$\gamma_{j+1} = 4\gamma_j(1-\gamma_j), \tag{6}$$

  where $\gamma_0 \in (0,1)$, $\gamma_0 \notin \{0.25, 0.5, 0.75, 1\}$, $j = 1, 2, \ldots, L_{in}$, and L_{in} denotes the number of infeasible particles.
- Update the positions of particles in the infeasible list (IFL) using the Chaotic Search Technique (CST) according to the following equation:

$$X_u^j = \gamma_j X_{\inf}^j + (1-\gamma_j) g^{\text{best}}, \ \forall j = 1,2,\ldots,L_{in}, \tag{7}$$

where X_u^j represents the updated position of the infeasible particle, $X_{\inf}^j$ denotes the position of the infeasible particle, and g^{best} is the best position among the feasible particles.

Step 6: Feasibility Check of Solutions. The position update reassesses each solution's feasibility. A particle is feasible and in the feasible list (FL) if its position is in the search space. If not, it goes on the infeasible list (IFL).

Step 7: Fitness Evaluation.

Step 8: Updating $p^{(\text{best},t+1)}$ and g^{best}.

- For every particle in the FL, update its $p_i^{(\text{best},t+1)}$, if the current position provides a better fitness value than the previous $p_i^{(\text{best},t)}$, $\quad \forall i = 1,\ldots,L_f$; where L_f denotes the number of feasible particles.
- Among all the particles in the swarm, find the global ideal position, g^{best}.

Step 9: Termination Check. After reaching the stopping condition, which may be convergence tolerance, maximum iterations T, or fitness value, the algorithm stops. If not, repeat Step 5 and continue iterating.

Step 10: Output results. Present the algorithm's results, including the optimal position and highest fitness value. See Figure 1 for the CPSOA flow diagram.

4.2. Computational complexity analysis

The computational complexity of CPSOA mostly relies on the population size NP, the dimension n of the problem, and the maximum number of iterations T. In each iteration, the algorithm modifies particle positions and evaluates the objective function for all particles (feasible and nonfeasible). Consequently, the total computational complexity may be estimated as $O(T \times NP \times n)$, which is analogous to the complexity of conventional population-based metaheuristic algorithms. The incorporation of the CST entails supplementary computations for managing infeasible solutions; nonetheless, this added burden is rather minimal in relation to the total cost of fitness evaluation.

5. Computational results

This section evaluates the efficacy and robustness of the CPSOA in two stages. In the first stage, 50 benchmark functions are used to evaluate the global optimization, convergence dynamics, and stability of metaheuristic algorithms [46] for CPSOA and four swarm intelligence algorithms: standard PSOA, GWO [47], MVO [48], and WOA [49]. This analysis assesses CPSOA's strengths and weaknesses in comparison to its competitors. The focus is on global exploration, local exploitation, convergence dynamics, and solution quality. In the second phase, we assess the effectiveness of CPSOA by applying it to a set of four systems of nonlinear equations (SNLEs). These benchmarks serve as representative test cases to assess the algorithm's capacity to handle nonlinear and complex mathematical structures. The results of this phase demonstrate that CPSOA is effective in both theoretical benchmark optimization and the resolution of practical engineering and mathematical problems.

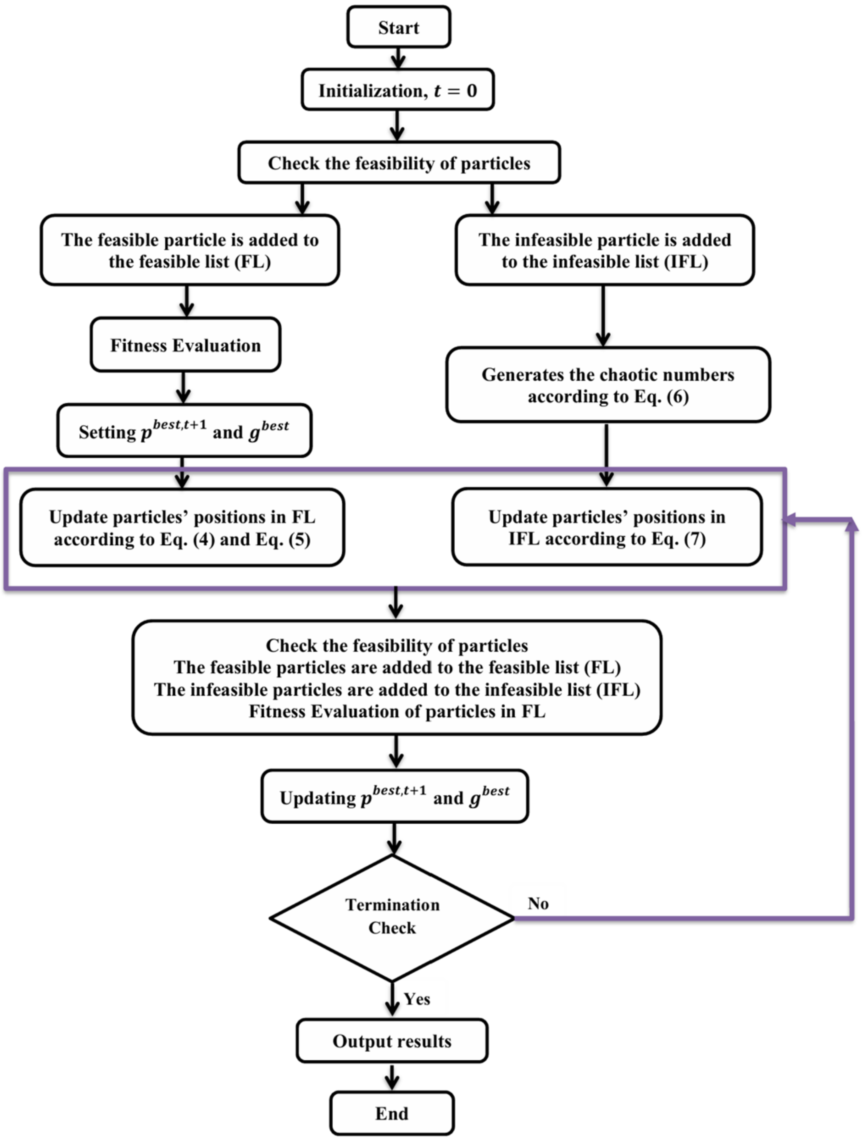

Fig. 1. Diagram illustrating the workflow of the proposed CPSOA algorithm

Table 1 delineates the parameter configurations for all optimization problems to maintain consistency. The parameter configurations of CPSOA were determined based on widely adopted values in the PSOA literature [50], supplemented by preliminary empirical studies aimed at attaining stable performance. The experiments utilize 50 particles and 300 iterations. We execute each algorithm, including CPSOA and the four comparative swarm intelligence (SI) methods, 30 times for each problem instance to enhance statistical reliability and mitigate stochasticity. Statistical metrics, such as the best solution, worst solution, average solution, median, standard deviation, and overall rank, are used to evaluate algorithm performance, consistency, and robustness. Computations were carried out in MATLAB R2014a on a Windows 7 PC with an AMD E1-2100 APU (1 GHz) and 4 GB RAM for all algorithms. This computational framework ensures fairness and reproducibility. CPSOA necessitates additional computations relative to standard PSOA owing to the integration of the CST for the adjustment of infeasible solutions. Nonetheless, this additional computational cost remains minor due to the mathematically straightforward nature of the chaotic update process, which does not necessitate intricate operations. Furthermore, the superior convergence characteristics and solution quality attained by CPSOA might diminish redundant search iterations and offset the extra computing burden.

Table 1. Parameter settings of the employed metaheuristic algorithms

Algorithm	Parameters	Values
GWO	Parameter of convergence – decreased through interval	[2 0]
WOA	Likelihood of entangling mechanism – helical factor	$0.5, 1$
MVO	Wormhole existence probability (WEP)	$\in [0, 2]$
CPSOA	Cognitive parameter c_1	2.8
	Social parameter c_2	1.3
	Inertia weight w	0.6

5.1. Empirical and simulation findings for 50 benchmark functions

CPSOA was experimentally tested across 50 benchmark functions, including both unimodal and multimodal types. Unimodal functions (F1-F25) are used to measure convergence efficiency and exploitation speed, as they possess a single global optimum. Optimization becomes more challenging for multimodal functions (F26-F50) due to the presence of numerous local optima. These complexities grow in severity with dimensionality; thus, multimodal functions are crucial for testing an algorithm's exploration capabilities. The algorithm effectively balances local optima avoidance and global optimization.

Tables 2-7 detail the statistical analysis of objective function values, with BEST, WORST, and STD indicating the minimum, maximum, and standard deviation of fitness values, respectively. The best results are bolded for easy comparison of the examined algorithms. Except for F17, F19, F20, F23, F24, and F25, CPSOA solved all unimodal benchmark functions optimally.

Table 2. Statistical comparison of CPSOA and competing algorithms on 50 benchmark functions

F	Metric	CPSOA	PSOA	GWO	MVO	WOA
F1	STD	0	1.05E-157	4.26E-13	3.292854	1.31E-50
	MEAN	0	8.23E-158	6.33E-13	17.27928	9.27E-51
	MEDIAN	0	2.56E-158	4.97E-13	16.78804	2.47E-51
	WORST	0	2.42E-157	1.12E-12	22.64764	3.06E-50
	BEST	0	1.42E-159	1.43E-13	13.59967	1.60E-54
	RANK	**1**	2	4	5	3
F2	STD	7.03E-05	1.57E-04	1.56E-03	0.04816	2.93E-03
	MEAN	7.10E-05	2.33E-04	4.51E-03	0.13408	2.48E-03
	MEDIAN	3.67E-05	2.77E-04	4.52E-03	0.13594	8.10E-04
	WORST	1.87E-04	4.43E-04	6.98E-03	0.19650	7.16E-03
	BEST	1.86E-05	5.95E-05	3.09E-03	0.06835	1.74E-04
	RANK	**1**	2	4	5	3
F3	STD	0	0	2.99E-62	7.76E-07	1.71E-80
	MEAN	0	2.66E-165	1.35E-62	1.33E-06	7.70E-81
	MEDIAN	0	7.91E-172	1.30E-66	9.68E-07	5.40E-83
	WORST	0	1.32E-164	6.71E-62	2.35E-06	3.83E-80
	BEST	0	1.06E-173	2.87E-69	6.01E-07	5.93E-87
	RANK	**1**	2	4	5	3
F4	STD	0	5.85E-79	1.27E-07	1.61E+01	1.37E-31
	MEAN	0	4.99E-79	2.89E-07	5.06E+01	1.64E-31
	MEDIAN	0	2.19E-79	2.43E-07	4.79E+01	2.58E-31
	WORST	0	1.26E-78	4.69E-07	7.59E+01	2.74E-31
	BEST	0	9.14E-81	1.55E-07	3.41E+01	4.09E-33
	RANK	**1**	2	4	5	3
F5	STD	0	7.56E-77	8.02E-03	6.44	1.67E+01
	MEAN	0	6.23E-77	1.39E-02	2.03E+01	6.90E+01
	MEDIAN	0	4.64E-77	1.36E-02	1.91E+01	7.66E+01
	WORST	0	1.84E-76	2.33E-02	3.07E+01	8.30E+01
	BEST	0	1.20E-78	5.11E-03	1.30E+01	4.16E+01
	RANK	**1**	2	3	4	5
F6	STD	1.96E-10	4.55E-10	5.20E-01	3.36	8.49E-01
	MEAN	1.90E-10	1.52E-09	2.60	1.54E+01	1.01
	MEDIAN	1.06E-10	1.58E-09	2.91	1.41E+01	6.47E-01
	WORST	4.66E-10	2.06E-09	2.97	2.10E+01	2.52
	BEST	1.45E-11	8.18E-10	1.74	1.28E+01	5.14E-01
	RANK	**1**	2	4	5	3
F7	STD	0	0	1.01E+01	1.16E+01	0
	MEAN	–275	–275	–2.13E+02	–2.37E+02	–275
	MEDIAN	–275	–275	–2.11E+02	–2.39E+02	–275
	WORST	–275	–275	–2.04E+02	–2.22E+02	–275
	BEST	–275	–275	–2.29E+02	–2.53E+02	–275
	RANK	**1**	**1**	5	4	**1**
F8	STD	0	2.68E-147	6.05	1.50E+03	3.41E+04
	MEAN	0	1.66E-147	5.78	7.21E+03	2.05E+05
	MEDIAN	0	5.48E-149	2.49	7.26E+03	2.15E+05
	WORST	0	6.20E-147	1.59E+01	9.45E+03	2.52E+05
	BEST	0	7.43E-150	1.23	5.61E+03	1.68E+05
	RANK	**1**	2	3	4	5
F9	STD	0	7.72E-79	1.37E-07	5.77E+48	2.81E-30
	MEAN	0	5.50E-79	5.16E-07	3.57E+48	1.30E-30
	MEDIAN	0	1.36E-79	4.91E-07	3.96E+44	3.71E-32
	WORST	0	1.79E-78	7.46E-07	1.33E+49	6.33E-30
	BEST	0	1.40E-81	3.84E-07	2.65E+37	7.60E-33
	RANK	**1**	2	4	5	3

Table 3. Statistical comparison of CPSOA and competing algorithms on 50 benchmark functions

F	Metric	CPSOA	PSOA	GWO	MVO	WOA
F10	STD	0	0	4.21E-44	1.06E-04	6.17E-148
	MEAN	0	0	1.92E-44	5.73E-05	4.48E-148
	MEDIAN	0	0	1.15E-46	1.27E-05	6.27E-166
	WORST	0	0	9.45E-44	2.46E-04	1.21E-147
	BEST	0	0	4.57E-47	3.67E-08	1.23E-172
	RANK	**1**	**1**	4	5	3
F11	STD	4.27E-05	5.70E-04	8.34E-01	5.69E+02	4.22E-01
	MEAN	4.63E-05	4.44E-04	4.77E+01	7.68E+02	4.82E+01
	MEDIAN	3.44E-05	1.18E-04	4.79E+01	4.72E+02	4.83E+01
	WORST	1.17E-04	1.42E-03	4.84E+01	1.72E+03	4.86E+01
	BEST	4.59E-06	9.88E-05	4.62E+01	3.66E+02	4.76E+01
	RANK	**1**	2	3	5	4
F12	STD	0	8.83E-159	1.92E-15	1.09E-02	2.69E-55
	MEAN	0	3.98E-159	2.01E-15	3.95E-02	1.71E-55
	MEDIAN	0	2.01E-163	1.71E-15	3.90E-02	7.20E-57
	WORST	0	1.98E-158	5.26E-15	5.56E-02	6.20E-55
	BEST	0	1.09E-166	2.95E-16	2.76E-02	6.90E-59
	RANK	**1**	2	4	5	3
F13	STD	8.34E-04	1.11E-04	1.21E-05	2.68E+01	1.05E-03
	MEAN	2.49E-01	2.50E-01	6.67E-01	4.29E+01	6.68E-01
	MEDIAN	2.50E-01	2.50E-01	6.67E-01	4.09E+01	6.67E-01
	WORST	2.50E-01	2.50E-01	6.67E-01	8.42E+01	6.69E-01
	BEST	2.48E-01	2.49E-01	6.67E-01	1.01E+01	6.67E-01
	RANK	**1**	2	3	5	4
F14	STD	0	1.66E-154	6.71E-05	1.22E+01	4.94E-25
	MEAN	0	1.03E-154	5.96E-05	3.11E+01	2.26E-25
	MEDIAN	0	1.59E-155	2.83E-05	2.90E+01	3.22E-41
	WORST	0	3.89E-154	1.78E-04	5.14E+01	1.11E-24
	BEST	0	6.36E-161	1.83E-05	1.91E+01	3.34E-53
	RANK	**1**	2	4	5	3
F15	STD	0	1.59E-135	1.06E-01	1.00E+01	1.95E+02
	MEAN	0	1.19E-135	1.40E-01	5.09E+01	8.91E+02
	MEDIAN	0	2.04E-136	1.35E-01	5.63E+01	8.52E+02
	WORST	0	3.60E-135	3.06E-01	5.82E+01	1.18E+03
	BEST	0	8.46E-141	4.72E-02	3.52E+01	6.47E+02
	RANK	**1**	2	3	4	5
F16	STD	0	4.47E-01	0	0	0
	MEAN	–1	–0.2	1.26E-179	3.90E-180	0
	MEDIAN	–1	0	5.54E-185	4.87E-229	0
	WORST	–1	0	6.26E-179	1.95E-179	0
	BEST	–1	–1	3.71E-201	5.70E-273	0
	RANK	**1**	**1**	5	4	3
F17	STD	9.56	1.46E+01	1.77E+01	5.83	1.63E+03
	MEAN	1.33E+01	5.69E+01	2.35E+01	4.43	1.41E+03
	MEDIAN	1.51E+01	5.86E+01	2.15E+01	2.51	5.90E+02
	WORST	2.41E+01	7.31E+01	4.59E+01	1.46E+01	3.74E+03
	BEST	2.04	3.46E+01	5.12	0.651	3.79E+01
	RANK	2	4	3	**1**	5
F18	STD	0	0	3.17E-143	1.35E-07	1.32E-62
	MEAN	0	3.40E-163	1.42E-143	1.44E-07	5.97E-63
	MEDIAN	0	2.29E-165	7.05E-151	1.23E-07	5.38E-66
	WORST	0	1.69E-162	7.08E-143	3.19E-07	2.96E-62
	BEST	0	8.14E-166	1.86E-183	2.97E-09	6.97E-71
	RANK	**1**	3	2	5	4

Table 4. Statistical comparison of CPSOA and competing algorithms on 50 benchmark functions

F	Metric	CPSOA	PSOA	GWO	MVO	WOA
F19	STD	2.20E-02	2.43E-03	1.79E-07	3.41E-01	3.41E-01
	MEAN	1.86E-02	3.87E-03	1.57E-07	1.52E-01	1.52E-01
	MEDIAN	1.52E-02	4.25E-03	8.12E-08	1.77E-07	2.54E-11
	WORST	5.54E-02	6.37E-03	4.47E-07	7.62E-01	7.62E-01
	BEST	8.25E-04	3.05E-04	1.26E-08	4.35E-10	7.75E-14
	RANK	5	4	3	2	**1**
F20	STD	4.86E-03	1.24E-02	8.72E-07	9.29E-07	1.69E-03
	MEAN	8.98E-03	1.43E-02	8.94E-07	2.42E-06	8.97E-04
	MEDIAN	8.18E-03	1.02E-02	5.77E-07	2.52E-06	2.51E-04
	WORST	1.40E-02	3.03E-02	2.26E-06	3.77E-06	3.91E-03
	BEST	2.38E-03	2.34E-03	9.44E-08	1.32E-06	1.60E-05
	RANK	5	4	**1**	2	3
F21	STD	0	0	0	0	0
	MEAN	1.38E-87	1.38E-87	1.38E-87	1.38E-87	1.38E-87
	MEDIAN	1.38E-87	1.38E-87	1.38E-87	1.38E-87	1.38E-87
	WORST	1.38E-87	1.38E-87	1.38E-87	1.38E-87	1.38E-87
	BEST	1.38E-87	1.38E-87	1.38E-87	1.38E-87	1.38E-87
	RANK	**1**	**1**	**1**	**1**	**1**
F22	STD	0	3.95E-156	1.88E-85	5.10E-08	1.39E-149
	MEAN	0	3.51E-156	8.40E-86	4.59E-08	6.36E-150
	MEDIAN	0	2.07E-156	7.21E-92	1.93E-08	6.50E-154
	WORST	0	7.76E-156	4.20E-85	1.18E-07	3.12E-149
	BEST	0	4.83E-163	1.35E-99	2.96E-09	1.69E-158
	RANK	**1**	2	4	5	3
F23	STD	6.41E-07	4.54E-06	5.34E-07	7.29E-06	3.15E-05
	MEAN	2.93E-01	2.93E-01	2.93E-01	2.93E-01	2.93E-01
	MEDIAN	2.93E-01	2.93E-01	2.93E-01	2.93E-01	2.93E-01
	WORST	2.93E-01	2.93E-01	2.93E-01	2.93E-01	2.93E-01
	BEST	2.93E-01	2.93E-01	2.93E-01	2.93E-01	2.93E-01
	RANK	3	2	**1**	4	5
F24	STD	6.37E-02	5.95E-02	4.85E-05	7.05E-03	5.24E-03
	MEAN	1.92E+01	1.92E+01	1.91E+01	1.91E+01	1.91E+01
	MEDIAN	1.92E+01	1.91E+01	1.91E+01	1.91E+01	1.91E+01
	WORST	1.93E+01	1.92E+01	1.91E+01	1.91E+01	1.91E+01
	BEST	1.91E+01	1.91E+01	1.91E+01	1.91E+01	1.91E+01
	RANK	3	5	**1**	4	2
25	STD	1.08E-04	9.96E-04	2.44E-06	5.48E-07	1.72E-05
	MEAN	9.19E-05	7.68E-04	2.78E-06	4.06E-07	1.02E-05
	MEDIAN	3.74E-05	4.01E-04	2.21E-06	2.78E-07	3.11E-06
	WORST	2.13E-04	2.46E-03	5.55E-06	1.36E-06	4.09E-05
	BEST	2.31E-06	2.57E-05	2.21E-08	1.62E-08	3.27E-10
	RANK	4	5	3	2	**1**
F26	STD	4.49E-06	1.28E-06	1.40E+01	4.2959	6.26E+01
	MEAN	1.64E-05	1.44E-05	2.40E+02	178.1659	8.41E+01
	MEDIAN	1.52E-05	1.38E-05	2.36E+02	176.0467	1.22E+02
	WORST	2.40E-05	1.64E-05	2.63E+02	183.1839	1.40E+02
	BEST	1.31E-05	1.34E-05	2.27E+02	173.3471	5.26
	RANK	**1**	2	5	4	3
F27	STD	0	0	2.07	47.6277	0
	MEAN	0	0	5.61	275.0878	0
	MEDIAN	0	0	6.12	282.1144	0
	WORST	0	0	7.78	325.7988	0
	BEST	0	0	2.22	202.2181	0
	RANK	**1**	**1**	4	5	**1**

Table 5. Statistical comparison of CPSOA and competing algorithms on 50 benchmark functions

F	Metric	CPSOA	PSOA	GWO	MVO	WOA
F28	STD	0	0	7.22E-01	0.02696	3.28E-01
	MEAN	0.9	0.9	2.39	1.15315	1.24
	MEDIAN	0.9	0.9	2.14	1.14925	1.31
	WORST	0.9	0.9	3.39	1.18741	1.63
	BEST	0.9	0.9	1.68	1.12465	0.9
	RANK	**1**	**1**	5	4	**1**
F29	STD	2.63E+02	6.58E+02	2.69E+03	2931.246	1.55E+03
	MEAN	8.82E+03	7.81E+03	1.01E+04	12379.42	8.43E+03
	MEDIAN	8.86E+03	7.60E+03	1.08E+04	12510.36	8.38E+03
	WORST	9.15E+03	8.72E+03	1.32E+04	15444.12	1.05E+04
	BEST	8.55E+03	7.03E+03	6.21E+03	8719.596	6.77E+03
	RANK	4	3	**1**	5	2
F30	STD	0	2.24E-80	1.68E-03	3.66715	4.20E-32
	MEAN	0	1.15E-80	2.53E-03	20.09401	1.97E-32
	MEDIAN	0	1.31E-81	2.94E-03	22.01992	2.93E-34
	WORST	0	5.16E-80	4.60E-03	23.16143	9.48E-32
	BEST	0	4.84E-82	4.37E-04	15.87628	3.36E-35
	RANK	**1**	2	4	5	3
F31	STD	4.08E-153	2.37E-36	1.38E-20	76552.23231	1.73E-03
	MEAN	1.82E-153	1.07E-36	6.36E-21	53293.55029	1.23E-03
	MEDIAN	3.82E-229	1.80E-39	2.28E-23	5773.85693	2.54E-10
	WORST	9.12E-153	5.32E-36	3.10E-20	174076.1115	3.63E-03
	BEST	1.13E-270	2.41E-42	3.24E-24	28.91060034	9.54E-20
	RANK	**1**	2	3	5	4
F32	STD	0	0	4.16E-08	0.818515897	3.18E-15
	MEAN	–8.88E-16	–8.88E-16	1.06E-07	3.339007489	4.80E-15
	MEDIAN	–8.88E-16	–8.88E-16	1.06E-07	3.181359589	6.22E-15
	WORST	–8.88E-16	–8.88E-16	1.53E-07	4.750526346	6.22E-15
	BEST	–8.88E-16	–8.88E-16	6.04E-08	2.673440524	–8.88E-16
	RANK	**1**	**1**	4	5	**1**
F33	STD	3.01E-06	7.62E-09	1.51E+01	131.2832586	1.05E+01
	MEAN	1	1	7.15E+01	1025.553987	1.40E+02
	MEDIAN	1	1	6.49E+01	1041.257117	1.37E+02
	WORST	1	1	8.79E+01	1155.922524	1.54E+02
	BEST	1	1	5.51E+01	850.6813034	1.30E+02
	RANK	**1**	2	3	5	4
F34	STD	0	2.10E-66	5.48E-02	0.187082899	8.94E-02
	MEAN	0	9.40E-67	2.60E-01	1.699874384	1.60E-01
	MEDIAN	0	1.26E-70	3.00E-01	1.699876986	9.99E-02
	WORST	0	4.70E-66	3.00E-01	1.899873534	3.00E-01
	BEST	0	2.10E-74	2.00E-01	1.399873398	9.99E-02
	RANK	**1**	2	4	5	3
F35	STD	8.44E-09	2.36E-07	7.84E+01	25.42049527	2.17E+02
	MEAN	–1.96E+03	–1.96E+03	–1.39E+03	–1677.759895	–1.80E+03
	MEDIAN	–1.96E+03	–1.96E+03	–1.35E+03	–1674.986601	–1.96E+03
	WORST	–1.96E+03	–1.96E+03	–1.31E+03	–1646.358783	–1.51E+03
	BEST	–1.96E+03	–1.96E+03	–1.49E+03	–1717.456643	–1.96E+03
	RANK	**1**	2	5	4	3
F36	STD	0	0	9.36E-03	0.09513203	0
	MEAN	0	0	4.18E-03	0.488470868	0
	MEDIAN	0	0	2.24E-14	0.469926869	0
	WORST	0	0	2.09E-02	0.606119909	0
	BEST	0	0	1.72E-14	0.356788135	0
	RANK	**1**	**1**	4	5	**1**

Table 6. Statistical comparison of CPSOA and competing algorithms on 50 benchmark functions

F	Metric	CPSOA	PSOA	GWO	MVO	WOA
F37	STD	0	0	4.84E-22	7.59E-22	4.47E-01
	MEAN	–1	–1	2.68E-22	9.04E-22	–0.2
	MEDIAN	–1	–1	5.64E-23	5.58E-22	9.48E-21
	WORST	–1	–1	1.13E-21	2.24E-21	2.19E-20
	BEST	–1	–1	4.10E-23	4.58E-22	–1
	RANK	**1**	**1**	4	5	**1**
F38	STD	3.82E-29	3.04E-23	4.21E-11	7.07E-18	4.75E-22
	MEAN	1.21E-20	1.21E-20	2.02E-11	7.47E-18	1.26E-20
	MEDIAN	1.21E-20	1.21E-20	8.64E-15	5.84E-18	1.23E-20
	WORST	1.21E-20	1.21E-20	9.54E-11	1.92E-17	1.32E-20
	BEST	1.21E-20	1.21E-20	4.33E-17	9.50E-19	1.21E-20
	RANK	**1**	2	5	4	3
F39	STD	8.53E-12	4.67E-10	4.43E-02	2.02	8.09E-03
	MEAN	1.52E-11	4.51E-10	9.52E-02	5.27	2.29E-02
	MEDIAN	1.48E-11	1.84E-10	8.26E-02	5.94	2.65E-02
	WORST	2.63E-11	1.02E-09	1.68E-01	7.29	2.89E-02
	BEST	2.60E-12	3.35E-11	5.18E-02	2.13	9.21E-03
	RANK	**1**	2	4	5	3
F40	STD	1.97E-11	2.22E-10	3.66E-02	2.07	2.07E-02
	MEAN	1.30E-11	2.17E-10	1.09E-01	5.05	2.30E-02
	MEDIAN	3.43E-12	1.70E-10	1.05E-01	4.71	1.69E-02
	WORST	4.76E-11	5.95E-10	1.62E-01	7.22	5.96E-02
	BEST	6.28E-13	2.37E-11	5.92E-02	2.05	1.04E-02
	RANK	**1**	2	4	5	3
F41	STD	0	1.59E-160	0	1.62E-06	3.58E-89
	MEAN	0	7.15E-161	5.03E-167	2.74E-06	2.55E-89
	MEDIAN	0	2.48E-163	2.60E-184	2.92E-06	1.31E-93
	WORST	0	3.56E-160	2.52E-166	4.54E-06	7.49E-89
	BEST	0	7.48E-166	6.09E-188	1.83E-07	2.34E-101
	RANK	**1**	3	2	5	4
F42	STD	4.32E-03	2.33E-03	9.52E-08	2.18E-05	3.36E-11
	MEAN	–1.96E+02	–1.96E+02	–1.96E+02	–1.96E+02	–1.96E+02
	MEDIAN	–1.96E+02	–1.96E+02	–1.96E+02	–1.96E+02	–1.96E+02
	WORST	–1.96E+02	–1.96E+02	–1.96E+02	–1.96E+02	–1.96E+02
	BEST	–1.96E+02	–1.96E+02	–1.96E+02	–1.96E+02	–1.96E+02
	RANK	5	4	2	3	**1**
F43	STD	9.02E-06	2.32E-06	1.18E-11	1.11E-10	0
	MEAN	–2.02	–2.02	–2.02	–2.02	–2.02
	MEDIAN	–2.02	–2.02	–2.02	–2.02	–2.02
	WORST	–2.02	–2.02	–2.02	–2.02	–2.02
	BEST	–2.02	–2.02	–2.02	–2.02	–2.02
	RANK	4	5	2	3	**1**
F44	STD	4.52	3.94E-01	9.78E-05	1.57E-05	5.20E-06
	MEAN	–1.04E+02	–1.06E+02	–1.07E+02	–1.07E+02	–1.07E+02
	MEDIAN	–1.06E+02	–1.07E+02	–1.07E+02	–1.07E+02	–1.07E+02
	WORST	–9.63E+01	–1.06E+02	–1.07E+02	–1.07E+02	–1.07E+02
	BEST	–1.07E+02	–1.07E+02	–1.07E+02	–1.07E+02	–1.07E+02
	RANK	4	5	3	2	**1**
F45	STD	9.74E-04	8.62E-05	2.62E-08	5.72E-07	2.25E-10
	MEAN	–1.03	–1.03	–1.03	–1.03	–1.03
	MEDIAN	–1.03	–1.03	–1.03	–1.03	–1.03
	WORST	–1.03	–1.03	–1.03	–1.03	–1.03
	BEST	–1.03	–1.03	–1.03	–1.03	–1.03
	RANK	5	4	2	3	**1**

Table 7. Statistical comparison of CPSOA and competing algorithms on 50 benchmark functions

F	Metric	CPSOA	PSOA	GWO	MVO	WOA
F46	STD	3.29E-03	2.32E-03	3.76E-06	1.63E-07	5.54E-06
	MEAN	4.02E-01	4.00E-01	3.98E-01	3.98E-01	3.98E-01
	MEDIAN	4.02E-01	4.00E-01	3.98E-01	3.98E-01	3.98E-01
	WORST	4.07E-01	4.04E-01	3.98E-01	3.98E-01	3.98E-01
	BEST	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01
	RANK	5	4	**1**	3	2
F47	STD	3.09E-03	3.34E-03	2.74E-03	2.32E-06	5.36E-03
	MEAN	–3.85	–3.86	–3.86	–3.86	–3.86
	MEDIAN	–3.85	–3.86	–3.86	–3.86	–3.86
	WORST	–3.85	–3.85	–3.86	–3.86	–3.85
	BEST	–3.86	–3.86	–3.86	–3.86	–3.86
	RANK	5	4	3	**1**	2
F48	STD	5.92E-02	6.31E-03	9.65E-02	6.80E-02	1.36E-01
	MEAN	–3.26	–3.28	–3.25	–3.25	–3.21
	MEDIAN	–3.30	–3.28	–3.32	–3.20	–3.29
	WORST	–3.19	–3.27	–3.14	–3.19	–3.01
	BEST	–3.31	–3.29	–3.32	–3.32	–3.32
	RANK	4	5	2	**1**	3
F49	STD	1.93E-09	5.32E-09	2.64E-08	4.61E-08	5.26E-10
	MEAN	–2.06	–2.06	–2.06	–2.06	–2.06
	MEDIAN	–2.06	–2.06	–2.06	–2.06	–2.06
	WORST	–2.06	–2.06	–2.06	–2.06	–2.06
	BEST	–2.06	–2.06	–2.06	–2.06	–2.06
	RANK	**1**	3	4	5	2
F50	STD	0	0	0	7.14E-03	0
	MEAN	1	1	1	1.01	1
	MEDIAN	1	1	1	1.01	1
	WORST	1	1	1	1.02	1
	BEST	1	1	1	1	1
	RANK	**1**	**1**	**1**	5	**1**

CPSOA performed well in multimodal test functions, approaching theoretical optimum values for most benchmark functions. This demonstrates the algorithm's diversity preservation and avoidance of premature convergence.

To illustrate the optimization process, Figures 2 and 3 show the convergence curves of selected benchmark functions and compare the convergence characteristics of CPSOA to other algorithms. These illustrations demonstrate that CPSOA generally converges faster and reaches better final objective values than the competing algorithms over a wide range of benchmark functions. These convergence patterns confirm that the integration of the CST effectively enhances the balance between exploration and exploitation, leading to improved solution quality and reduced vulnerability to premature convergence.

The CPSOA design, integrating two complementary mechanisms, accelerates convergence. Unlike numerous optimization techniques, CPSOA maintains and refines infeasible solutions through the CST operator. Infeasible solutions frequently occur in proximity to the global optimum, and their elimination may result in the loss

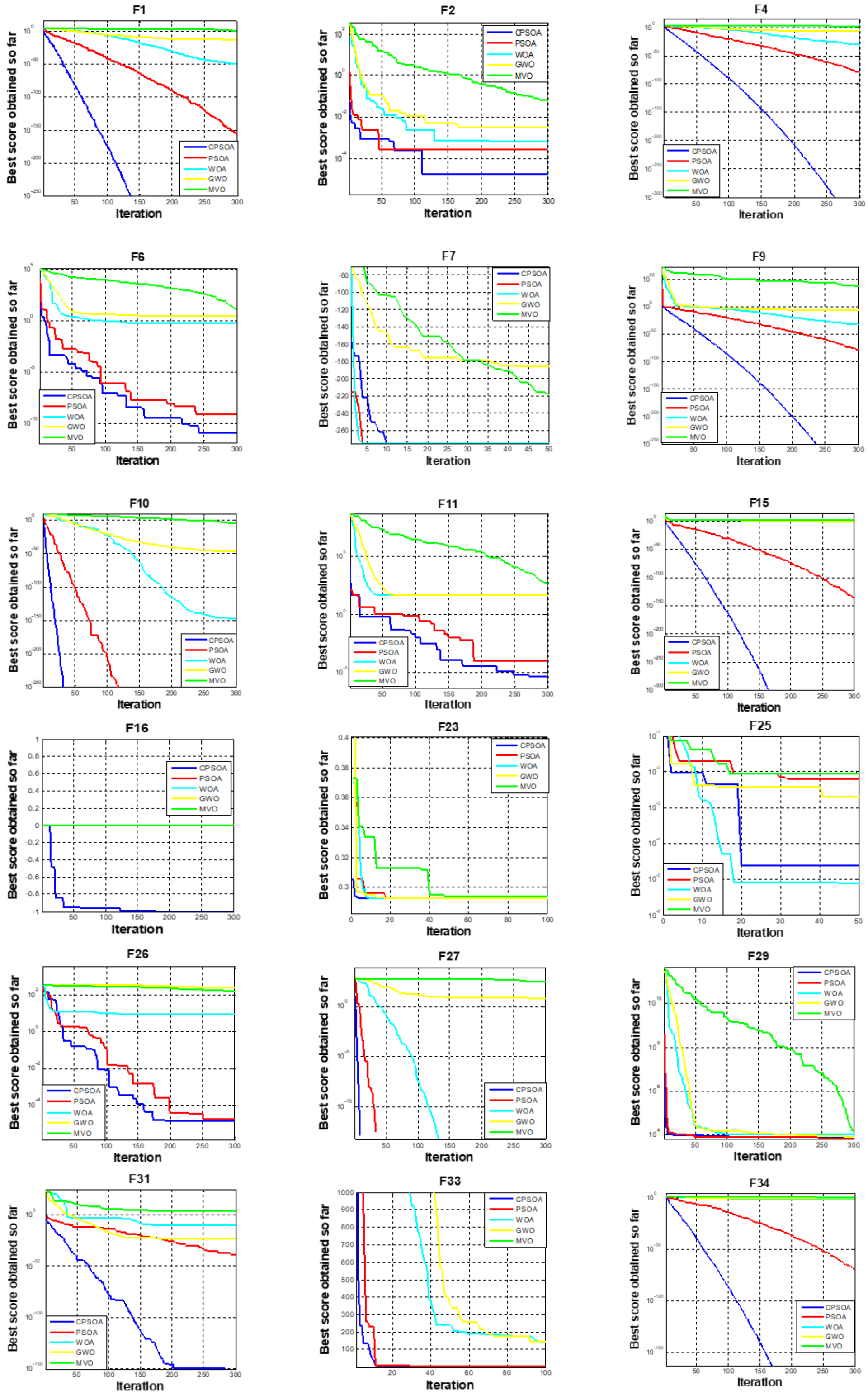

Fig. 2. Convergence curves for F1, F2, F4, F6, F7, F9, F10, F11, F15, F16, F23, F25, F26, F27, F29, F31, F33, and F34

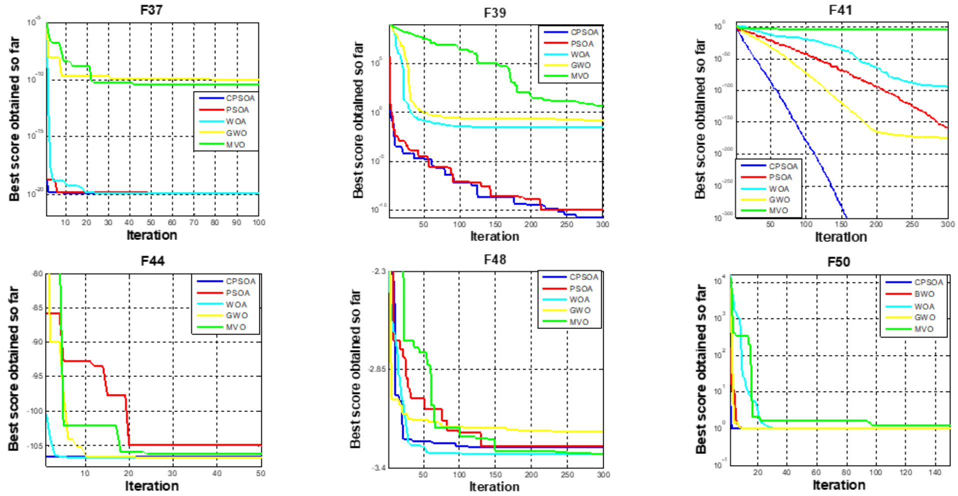

Fig. 3. Convergence curves for F37, F39, F41, F44, F48, and F50

of critical search information. CPSOA balances exploration and exploitation; while PSOA ensures diversity and extensive solution space exploration, the CST enhances exploitation by refining infeasible solutions. This synergistic integration enables CPSOA to navigate the global landscape and prevent premature convergence, resulting in faster and more accurate convergence to optimal solutions.

Simulations on 50 benchmark functions show that CPSOA consistently outperforms or competes with state-of-the-art SI methods. The superior performance in both unimodal and multimodal categories demonstrates its robustness, adaptability, and efficacy in addressing complex optimization challenges.

5.2. Statistical analysis

A non-parametric statistical analysis verified the algorithm performance differences. Analysis of experimental data used the Friedman test, a widely adopted statistical method for ranking algorithms based on performance across several test cases [51]. Table 8 summarizes the Friedman test results, where CPSOA outperforms rival algorithms in average rankings. The p-value ($p < 0.001$) indicates that the observed performance differences are statistically significant and unlikely to have occurred by chance.

To evaluate the pairwise performance between CPSOA and competing algorithms, the Wilcoxon signed-rank test was employed [52]. Refer to Table 9 for the results. This study uses *P-R* to represent positive ranks and *N-R* to represent negative ranks. A higher *P-R* value than *N-R* shows that CPSOA outperforms the competitor in most test cases. Table 9 demonstrates that CPSOA outperforms benchmark algorithms statistically, with *P-R* values consistently exceeding *N-R* values in all pairwise comparisons.

Table 8. Statistical performance analysis using the Friedman test

	Ranks	Statistics
Algorithm	Mean Rank	
CPSOA	2.10	Number of Functions (N) = 50
PSOA	2.67	$\chi^2 = 47.478$
GWO	3.29	Degrees of Freedom (Df) = 4
MVO	4.10	p-value < 0.001
WOA	2.84	

Table 9. Statistical performance analysis using the Wilcoxon test

Comparison	Rank Type	N	Remarks
PSOA – CPSOA	N-R	8	a. PSOA < CPSOA
	P-R	32	b. PSOA > CPSOA
	Ties	10	c. PSOA = CPSOA
	Z-value	–2.177 (based on N-R)	
	Asymp. Sig. (2-tailed)	0.02944	
GWO – CPSOA	N-R	13	d. GWO < CPSOA
	P-R	35	e. GWO > CPSOA
	Ties	2	f. GWO = CPSOA
	Z-value	–2.841 (based on N-R)	
	Asymp. Sig. (2-tailed)	0.00449	
MVO – CPSOA	N-R	11	g. MVO < CPSOA
	P-R	38	h. MVO > CPSOA
	Ties	1	i. MVO = CPSOA
	Z-value	–4.451 (based on N-R)	
	Asymp. Sig. (2-tailed)	8.5E–06	
WOA – CPSOA	N-R	12	j. WOA < CPSOA
	P-R	30	k. WOA > CPSOA
	Ties	8	l. WOA = CPSOA
	Z-value	–2.107 (based on N-R)	
	Asymp. Sig. (2-tailed)	0.0351	

The Friedman and Wilcoxon signed-rank tests confirm that CPSOA outperforms other state-of-the-art algorithms. This suggests that the proposed CPSOA is a robust and competitive optimization strategy that effectively surpasses existing methods.

5.3. Performance of CPSOA on systems of nonlinear equations

Four systems of nonlinear equations (SNLEs) were utilized to assess the proposed CPSOA's ability to solve practical and mathematically rigorous problems. Nonlinear equations are fundamental in science and engineering disciplines such as physics, chemistry, mechanics, and control theory. These systems are inherently difficult to solve due to multiple variables, nonlinear interactions, and the presence of spurious local optima. The following are four SNLEs benchmark problems:

Benchmark 1:

$$\begin{cases} f_1(\mathbf{X}) = (x_1 - 5x_2)^2 + 40\sin^2(10x_3) = 0, \\ f_2(\mathbf{X}) = (x_2 - 2x_3)^2 + 40\sin^2(10x_1) = 0, \\ f_3(\mathbf{X}) = (3x_1 + x_3)^2 + 40\sin^2(10x_2) = 0, \\ \mathbf{X} = [x_1, x_2, x_3], \quad x_i \in [-1, 1],\ i = 1, 2, 3. \end{cases} \tag{8}$$

Benchmark 2:

$$\begin{cases} f_1(\mathbf{X}) = x_1 + e^{x_2} - \cos(x_2) = 0, \\ f_2(\mathbf{X}) = 3x_1 - \sin(x_1) - x_2 = 0, \\ \mathbf{X} = [x_1, x_2], \quad x_i \in [-1, 1],\ i = 1, 2. \end{cases} \tag{9}$$

Benchmark 3:

$$\begin{cases} f_1(\mathbf{X}) = x_1 - \sin(2x_2) = 0, \\ f_2(\mathbf{X}) = x_1 - x_2 = 0, \\ \mathbf{X} = [x_1, x_2], \quad x_i \in [-1, 1],\ i = 1, 2. \end{cases} \tag{10}$$

Benchmark 4:

$$\begin{cases} f_1(\mathbf{X}) = x_2 + 2x_6 + x_9 + 2x_{10} - 10^{-5} = 0, \\ f_2(\mathbf{X}) = x_3 + x_8 - 3.10^{-5} = 0, \\ f_3(\mathbf{X}) = x_1 + x_3 + 2x_5 + 2x_8 + x_9 + x_{10} - 5 \times 10^{-5} = 0, \\ f_4(\mathbf{X}) = x_4 + 2x_7 - 10^{-5} = 0, \\ f_5(\mathbf{X}) = 0.5140473 \times 10^{7} x_5 - x_1^2 = 0, \\ f_6(\mathbf{X}) = 0.1006932 \times 10^{-6} x_6 - 2x_2^2 = 0, \\ f_7(\mathbf{X}) = 0.7816278 \times 10^{-15} x_7 - x_4^2 = 0, \\ f_8(\mathbf{X}) = 0.1496236 \times 10^{-6} x_8 - x_1 x_3 = 0, \\ f_9(\mathbf{X}) = 0.6194411 \times 10^{-7} x_9 - x_1 x_2 = 0, \\ f_{10}(\mathbf{X}) = 0.2089296 \times 10^{-14} x_{10} - x_1 x_2^2 = 0, \\ \mathbf{X} = [x_1, \ldots, x_{10}], \quad -10 \le x_1, \ldots, x_{10} \le 10. \end{cases} \tag{11}$$

Each system was solved by CPSOA alongside several metaheuristic algorithms, including PSOA, GWO, MVO, and WOA. This comparative framework facilitates an equitable and comprehensive assessment of performance among optimization techniques. The fitness function defined in Eq. (2) is employed to compute $F(\mathbf{X})$ for each prospective solution to ensure consistency in evaluation. This strategy establishes a standardized benchmark by assessing all algorithms against identical performance criteria.

Table 10 displays the computational results of CPSOA and rival algorithms for four systems of nonlinear equations, including the best, worst, mean, median, standard deviation, and ranking values for each benchmark problem.

Table 10. Numerical results for the four benchmark nonlinear equation systems

Benchmark 1					
Measures	CPSOA	PSOA	GWO	MVO	WOA
x_1	–9.00E-164	4.80E-79	–4.00E-60	–2.50E-05	4.35E-38
x_2	8.00E-164	1.41E-78	–4.30E-60	–1.90E-05	–2.80E-38
x_3	5.30E-164	1.97E-79	3.81E-60	8.88E-05	–3.90E-40
std	**0.00E+00**	1.30E-145	3.68E-89	3.53E-01	1.39E-01
F_{avg}	**0.00E+00**	6.10E-146	1.64E-89	2.69E-01	6.21E-02
F_{median}	**0.00E+00**	1.10E-150	1.40E-108	1.64E-01	1.19E-66
F_{worst}	**0.00E+00**	2.90E-145	8.22E-89	8.52E-01	3.10E-01
F_{best}	**0.00E+00**	3.00E-153	6.60E-116	1.19E-05	3.61E-72
Rank	**1**	2	3	5	4
Benchmark 2					
Measures	CPSOA	PSOA	GWO	MVO	WOA
x_1	1.05E-39	–3.60E-19	5.53E-18	–9.70E-05	1.23E-16
x_2	2.10E-39	–9.50E-19	1.11E-17	–7.80E-05	–7.50E-17
std	**4.53E-32**	2.66E-18	3.43E-12	2.26E-04	1.17E-09
F_{avg}	**2.97E-32**	2.20E-18	1.53E-12	4.62E-04	5.38E-10
F_{median}	**3.21E-34**	8.72E-19	2.00E-15	4.76E-04	1.89E-12
F_{worst}	**1.03E-31**	6.36E-18	7.66E-12	7.25E-04	2.63E-09
F_{best}	**3.71E-47**	1.09E-19	3.54E-22	1.46E-04	1.61E-16
Rank	**1**	3	2	5	4
Benchmark 3					
Measures	CPSOA	PSOA	GWO	MVO	WOA
x_1	0.00E+00	4.16E-79	8.77E-43	–9.48E-01	–5.12E-80
x_2	0.00E+00	3.12E-79	4.39E-43	–9.48E-01	–1.71E-79
std	**0.00E+00**	9.78E-76	5.38E-37	2.03E-05	4.38E-06
F_{avg}	**0.00E+00**	4.43E-76	3.03E-37	9.36E-05	1.96E-06
F_{median}	**0.00E+00**	2.58E-78	6.85E-40	8.12E-05	2.30E-75
F_{worst}	**0.00E+00**	2.19E-75	1.24E-36	1.27E-04	9.79E-06
F_{best}	**0.00E+00**	1.56E-79	2.20E-43	7.96E-05	2.06E-79
Rank	**1**	2	4	5	3
Benchmark 4					
Measures	CPSOA	PSOA	GWO	MVO	WOA
x_1	–9.28E-06	–1.42E-07	–7.19E-04	9.59E-04	–2.51E-01
x_2	7.77E-06	–3.53E-07	–3.59E-02	1.32E-02	2.01E-01
x_3	9.40E-06	2.99E-05	3.57E-01	–2.11E+00	–4.11E-01
x_4	1.17E-05	–3.22E-06	–8.01E-02	1.79E-02	–1.74E-01
x_5	1.69E-06	6.60E-06	4.22E-01	–1.52E+00	–8.71E-01
x_6	–2.24E-06	3.37E-06	2.64E-01	–3.76E-01	–5.22E-01
x_7	–8.54E-07	6.56E-06	3.99E-02	–9.36E-03	9.03E-02
x_8	2.02E-05	8.23E-07	–3.57E-01	2.11E+00	3.61E-01
x_9	5.55E-06	7.00E-06	–4.78E-01	1.11E+00	2.55E+00
x_{10}	5.54E-07	–1.66E-06	–7.33E-03	–1.87E-01	–8.71E-01
std	**7.08E-08**	6.24E-08	4.93E-03	3.93E-03	1.99E-01
F_{avg}	**1.03E-07**	1.75E-07	4.91E-03	5.82E-03	1.56E-01
F_{median}	**7.86E-08**	1.73E-07	3.26E-03	6.44E-03	6.39E-02
F_{worst}	**2.24E-07**	2.70E-07	1.05E-02	1.17E-02	4.53E-01
F_{best}	**4.86E-08**	9.47E-08	1.00E-03	1.61E-03	4.26E-02
Rank	**1**	2	3	4	5

The optimal performance metric value is bolded for easy comparison among the evaluated methods. Over 30 independent runs, these statistical measures evaluate solution quality, robustness, and consistency. The results demonstrate that CPSOA effectively solves nonlinear equation systems of various dimensions and degrees of nonlinearity. CPSOA consistently solves all four benchmark problems with high accuracy and usually ranks highest among the algorithms. The algorithm exhibits lower objective function values and standard deviations, indicating high solution accuracy, stability, and reproducibility across runs.

CPSOA performs well due to the complementary interplay between PSOA and the CST. PSOA rapidly explores the feasible regions, whereas the CST refines infeasible solutions and directs them toward promising regions surrounding the global optimum. This hybrid technique balances exploration and exploitation, improves convergence behavior, and prevents premature convergence. The results in Table 10 demonstrate that CPSOA is a reliable optimization framework for systems of nonlinear equations. The method performs effectively on standard benchmark functions and exhibits significant potential for complex mathematical models as well as real-world engineering and scientific applications.

6. Conclusion

This paper presents a hybrid metaheuristic optimization method, the chaotic particle swarm optimization algorithm (CPSOA), for solving systems of nonlinear equations (SNLEs). The proposed approach integrates the robust local search of the chaotic search technique (CST) with the global search capabilities of the particle swarm optimization algorithm (PSOA). The CST optimizes infeasible solutions, whereas PSOA enhances feasible ones. This dual-faceted technique improves the algorithm's exploration-exploitation equilibrium and safeguards critical information contained within infeasible solutions. The performance of CPSOA was evaluated and validated for both theoretical soundness and practical applicability through fifty benchmark functions and four SNLEs. The effectiveness of the proposed technique is supported by comprehensive statistical analysis corroborating the experimental results.

The study highlights the primary benefits of CPSOA, including its simplicity and versatility in optimization tasks, and its derivative-free nature, which is particularly suitable for non-differentiable functions. Key advantages include the retention and refinement of infeasible solutions, outstanding performance across benchmark tests, and rigorous statistical validation. Furthermore, the algorithm exhibits rapid convergence toward optimal solutions and enhances the potential for achieving global optimality through chaotic search mechanisms. These features underscore the significant relevance of CPSOA in diverse engineering and scientific domains.

CPSOA, while offering several advantages, exhibits stochastic characteristics inherent to metaheuristic algorithms. This implies that the nature of the problem, its dimensionality, and its complexity significantly influence computational efficiency and solution quality. Consequently, the robustness and adaptability of CPSOA do not guarantee optimal performance in every optimization scenario, consistent with the ”No Free Lunch” theorem.

Despite CPSOA exhibiting robust empirical performance and statistical superiority across benchmark functions and systems of nonlinear equations, the suggested method continues to serve as a stochastic metaheuristic framework. Consequently, establishing robust mathematical proofs of convergence and stability is challenging due to the nonlinear and probabilistic nature of the search process. This study primarily validates the efficacy of CPSOA by comprehensive computational experiments, convergence behavior analysis, and non-parametric statistical assessments. Establishing a rigorous theoretical convergence analysis for CPSOA constitutes a significant avenue for future research.

Future research will validate and enhance CPSOA in various aspects. This encompasses the application to additional engineering and scientific optimization domains, enhancement of convergence characteristics through adaptive parameter control techniques, exploration of hybridization strategies that integrate metaheuristic principles with machine learning to augment performance and scalability, and the development of efficient distributed and parallel computing architectures for large-scale real-time execution. Also, future research may involve exploring parameter tuning processes and doing thorough sensitivity analyses to enhance the robustness, stability, and generalization capacity of CPSOA in various optimization problems. In this context, future research will also focus on extending the evaluation of CPSOA by testing its performance on highly complex shifted, rotated, and hybrid landscapes, such as the Congress on Evolutionary Computation (CEC) benchmark suites [53,54]. Finally, the relevance of CPSOA to large-scale real-world engineering and scientific optimization challenges, such as those in control systems, power systems, mechanical design, and data science, may be further explored in future research.

Funding

The authors extend their appreciation to Prince Sattam bin Abdulaziz University for funding this research work through the project number (PSAU/2025/01/36283).

Acknowledgments

The authors extend their appreciation to Prince Sattam bin Abdulaziz University for funding this research work through the project number (PSAU/2025/01/36283).

References

[1] Kowalski, K., & Jankowski, K. (1999). Towards complete solutions to systems of nonlinear equations of many-electron theories. *Physical Review Letters*, *81*, 1195-1198.

[2] Barbashov, B.M., Nesterenko, V.V., & Chervyakov, A.M. (1982). General solutions of nonlinear equations in the geometric theory of the relativistic string. *Communications in Mathematical Physics*, *84*, 471-481.

[3] Bader, B.W. (2005). Tensor-Krylov methods for solving large-scale systems of nonlinear equations. *SIAM Journal on Numerical Analysis*, *43*(3), 1321-1347.

[4] Hansen, E.R., & Sengupta, S. (1981). Bounding solutions of systems of equations using interval analysis. *BIT Numerical Mathematics*, *21*, 203-211.

[5] Neumaier, A. (1990). *Interval Methods for Systems of Equations*. Cambridge: Cambridge University Press.

[6] Morgan, A.P. (1987). *Solving Polynomial Systems Using Continuation for Scientific and Engineering Problems*. Englewood Cliffs: Prentice-Hall.

[7] Van Hentenryck, P., McAllester, D., & Kapur, D. (1997). Solving polynomial systems using a branch and prune approach. *SIAM Journal on Numerical Analysis*, *34*(2), 797-827.

[8] Karr, C.L., Weck, B., & Freeman, L.M. (1998). Solutions to systems of nonlinear equations via a genetic algorithm. *Engineering Applications of Artificial Intelligence*, *11*, 369-375.

[9] Wang, Q., Zeng, J., & Zie, J. (2007). Modified particle swarm optimization for solving systems of equations. In *Proceedings of the 3rd International Conference on Intelligent Computing (ICIC)*. Qingdao, China, 661-667.

[10] Ouyang, A., Zhou, Y., & Luo, Q. (2009). Hybrid particle swarm optimization algorithm for solving systems of nonlinear equations. In *Proceedings of the IEEE International Conference on Granular Computing*, 460-465.

[11] Jia, R., & He, D. (2012). Hybrid artificial bee colony algorithm for solving nonlinear system of equations. In *Proceedings of the 8th International Conference on Computational Intelligence and Security*, 56-60.

[12] Yang, Y., Zhou, Y., & Gong, Q. (2010). Hybrid artificial glow-worm swarm optimization algorithm for solving system of nonlinear equations. *Journal of Computational Information Systems*, *6*(10), 3431-3438.

[13] Hirsch, M.J., Pardalos, P.M., & Resende, M.G.C. (2009). Solving systems of nonlinear equations with continuous GRASP. *Nonlinear Analysis: Real World Applications*, *10*, 2000-2006.

[14] Mo, Y., Liu, H., & Wang, Q. (2009). Conjugate direction particle swarm optimization solving systems of nonlinear equations. *Computers & Mathematics with Applications*, *57*, 1877-1882.

[15] Toutounian, F., Saberi-Nadjafi, J., & Taheri, S.H. (2009). A hybrid of the Newton-GMRES and electromagnetic meta-heuristic methods for solving systems of nonlinear equations. *Journal of Mathematical Modelling and Algorithms*, *8*, 4, 425-443.

[16] Sacco, W.F., & Henderson, N. (2011). Finding all solutions of nonlinear systems using a hybrid metaheuristic with fuzzy clustering means. *Applied Soft Computing*, *11*, 8, 5424-5432.

[17] El-Shorbagy, M.A., & El-Refaey, A.M. (2020). Hybridization of grasshopper optimization algorithm with genetic algorithm for solving system of non-linear equations. *IEEE Access*, *8*, 220944-220961.

[18] Gu, Q., Li, S., & Liao, Z. (2024). Solving nonlinear equation systems based on evolutionary multitasking with neighborhood-based speciation differential evolution. *Expert Systems with Applications*, *238*, 122025.

[19] Ariyaratne, M.K.A., Fernando, T.G.I., & Weerakoon, S. (2020). A self-tuning algorithm to approximate roots of systems of nonlinear equations based on the firefly algorithm. *International Journal of Swarm Intelligence*, *5*, 1, 60-96.

[20] Mansouri, P., Asady, B., & Gupta, N. (2011). A novel iteration method for solving hard problems (nonlinear equations) with artificial bee colony algorithm. *World Academy of Science, Engineering and Technology*, *59*, 594-596.

[21] 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.

[22] Abdollahi, M., Isazadeh, A., & Abdollahi, D. (2013). Imperialist competitive algorithm for solving systems of nonlinear equations. *Computers & Mathematics with Applications*, *65*, 1894-1908.

[23] Jaberipour, M., Khorram, E., & Karimi, B. (2011). Particle swarm algorithm for solving systems of nonlinear equations. *Computers & Mathematics with Applications*, *62*, 566-576.

[24] Coelho, L. dos S., & Mariani, V.C. (2008). Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. *Expert Systems with Applications*, *34*(3), 1905-1913.

[25] Yang, D., Li, G., & Cheng, G. (2007). On the efficiency of chaos optimization algorithms for global optimization. *Chaos, Solitons & Fractals*, *34*(4), 1366-1375.

[26] Saremi, S., Mirjalili, S., & Lewis, A. (2014). Biogeography-based optimisation with chaos. *Neural Computing and Applications*, *25*(5), 1077-1097.

[27] Luo, Y.-Z., Tang, G.-J., & Zhou, L.-N. (2008). Hybrid approach for solving systems of nonlinear equations using chaos optimization and quasi-Newton method. *Applied Soft Computing*, *8*, 2, 1068-1073.

[28] El-Shorbagy, M.A. (2024). Chaotic noise-based particle swarm optimization algorithm for solving system of nonlinear equations. *IEEE Access*, *12*, 118087-118098.

[29] Gharehchopogh, F.S., Maleki, I., & Dizaji, Z.A. (2022). Chaotic vortex search algorithm: metaheuristic algorithm for feature selection. *Evolutionary Intelligence*, *15*, 3, 1777-1808.

[30] Dong, Y., Zhang, S., Zhang, H., Zhou, X., & Jiang, J. (2025). Chaotic evolution optimization: A novel metaheuristic algorithm inspired by chaotic dynamics. *Chaos, Solitons & Fractals*, *192*, 116049.

[31] Arasteh, B., Arasteh, K., Ghaffari, A., & Ghanbarzadeh, R. (2024). A new binary chaos-based metaheuristic algorithm for software defect prediction. *Cluster Computing*, *27*, 7, 10093-10123.

[32] Luo, T., Xie, J., Zhang, B., Zhang, Y., Li, C., & Zhou, J. (2024). An improved levy chaotic particle swarm optimization algorithm for energy-efficient cluster routing scheme in industrial wireless sensor networks. *Expert Systems with Applications*, *241*, 122780.

[33] Arasteh, B. (2023). Clustered design-model generation from a program source code using chaos--based metaheuristic algorithms. *Neural Computing and Applications*, *35*, 4, 3283-3305.

[34] Parkavi, R., Karthikeyan, P., & Abdullah, A.S. (2024). Enhancing personalized learning with explainable AI: A chaotic particle swarm optimization based decision support system. *Applied Soft Computing*, *156*, 111451.

[35] Feng, W., Zhang, W., & Huang, S. (2023). A novel parameter estimation method for PMSM by using chaotic particle swarm optimization with dynamic self-optimization. *IEEE Transactions on Vehicular Technology*, *72*, 7, 8424-8432.

[36] Jiao, C., Yu, K., & Zhou, Q. (2024). An opposition-based learning adaptive chaotic particle swarm optimization algorithm. *Journal of Bionic Engineering*, *21*, 6, 3076-3097.

[37] Singh, H.K., Alam, K., & Ray, T. (2016). Use of infeasible solutions during constrained evolutionary search: A short survey. In *Proceedings of the Australasian Conference on Artificial Life and Computational Intelligence*, 193-205.

[38] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks (ICNN)*, vol. IV, Perth, Australia, 1942-1948.

[39] El-Shorbagy, M.A., & Al-Drees, F.M. (2023). Studying the effect of introducing chaotic search on improving the performance of the sine cosine algorithm to solve optimization problems and nonlinear system of equations. *Mathematics*, *11*(5), 1231.

[40] Omar, H.A., & El-Shorbagy, M.A. (2022). Modified grasshopper optimization algorithm-based genetic algorithm for global optimization problems: The system of nonlinear equations case study. *Soft Computing*, *26*(18), 9229-9245.

[41] El-Shorbagy, M.A., Eldesoky, I.M., Basyouni, M.M., Nassar, I., & El-Refaey, A.M. (2022). Chaotic search-based salp swarm algorithm for dealing with system of nonlinear equations and power system applications. *Mathematics*, *10*(9), 1368.

[42] El-Shorbagy, M.A., Mousa, A.A., & Nasr, S.M. (2016). A chaos-based evolutionary algorithm for general nonlinear programming problems. *Chaos, Solitons & Fractals*, *85*, 8-21.

[43] Skiadas, C.H., & Skiadas, C. (Eds.). (2017). *Handbook of Applications of Chaos Theory*. Boca Raton: CRC Press.

[44] Kaveh, A. (2014). *Advances in Metaheuristic Algorithms for Optimal Design of Structures*. Cham: Springer, 9-40.

[45] El-Shorbagy, M.A. (2024). Trust region based chaotic search for solving multi-objective optimization problems. *Expert Systems*, *41*(12), e13705.

[46] Ayyarao, T.S., Ramakrishna, N.S.S., Elavarasan, R.M., Polumahanthi, N., Rambabu, M., Saini, G., & Alatas, B. (2022). War strategy optimization algorithm: A new effective metaheuristic algorithm for global optimization. *IEEE Access*, *10*, 25073-25105.

[47] Ma, C., Huang, H., Fan, Q., Wei, J., Du, Y., & Gao, W. (2022). Gray wolf optimizer based on aquila exploration method. *Expert Systems with Applications*, *205*, 117629.

[48] Aljarah, I., Mafarja, M., Heidari, A.A., Faris, H., & Mirjalili, S. (2020). Multi-verse optimizer: Theory, literature review, and application in data clustering. *Nature-Inspired Optimizers*, 123-141.

[49] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, *95*, 51-67.

[50] El-Shorbagy, M.A., & Hassanien, A.E. (2018). Particle swarm optimization from theory to applications. *International Journal of Rough Sets and Data Analysis (IJRSDA)*, *5*(2), 1-24.

[51] Sheldon, M.R., Fillyaw, M.J., & Thompson, W.D. (1996). The use and interpretation of the Friedman test in the analysis of ordinal-scale data in repeated measures designs. *Physiotherapy Research International*, *1*(4), 221-228.

[52] Zimmerman, D.W., & Zumbo, B.D. (1993). Relative power of the Wilcoxon test, the Friedman test, and repeated-measures ANOVA on ranks. *Journal of Experimental Education*, *62*(1), 75-86.

[53] Garg, V., Zaheer, H., Tejani, G.G., & Sharma, S. (2026). Comparative study of JAYA algorithm with efficient mutation strategies for solving continuous optimization problems. *Journal of Computational and Applied Mathematics*, 117511.

[54] Garg, V., Deep, K., & Kulkarni, A.J. (2026). Laplacian biogeography-based algorithm using a gaining – sharing knowledge-based strategy for global optimization problems and the Lennard--Jones problem. *Engineering Optimization*, *58*(4), 1000-1031.