RESEARCH

Hybrid Learning–Driven Golden Jackal Optimizer for Reliable Parameter Estimation of Nonlinear Memristive Chaotic Systems

Davut Izci · Serdar Ekinci · Rizk M. Rizk-Allah · Vedat Tümen · Mostafa Rashdan · Mohammad Salman · Burcu Bektaş Güneş · Yasin İnağ

Received: 28 November 2025 / Revised: 26 December 2025 / Accepted: 11 January 2026

Abstract

Accurate identification of parameters in chaotic and nonlinear systems is essential for ensuring precise modeling, control, and prediction of complex dynamical behaviors. However, conventional metaheuristic algorithms often struggle to maintain an effective balance between exploration and exploitation, leading to premature convergence and estimation inaccuracies. To address these challenges, this study proposes an enhanced golden jackal optimizer (en-GJO) that integrates three complementary mechanisms (Laplacian crossover learning, elite group learning, and opposition repair learning). These hybrid strategies collectively strengthen population diversity, accelerate convergence, and prevent stagnation, thereby improving both the global search capability and local refinement accuracy of the original GJO. The effectiveness of the en-GJO is first validated through extensive benchmarking on twenty-three standard test functions, including unimodal, multimodal, and fixed-dimensional multimodal problems. Comparative results against nine well-established metaheuristics (such as SSA, SCA, HHO, AEO, EO, GBO, RUN, and ARO) demonstrate that en-GJO achieves superior convergence precision and robustness, consistently yielding the lowest mean and standard-deviation values across all categories. To further verify its real-world applicability, the en-GJO is applied to the parameter identification of a memristive chaotic system, formulated as a nonlinear optimization problem using a least-squares-based objective function. Simulation results reveal that the proposed method attains the most accurate estimates of the system parameters (a, b, c, d), with negligible deviation from their true values. Statistical analyses and convergence profiles confirm that en-GJO not only converges faster but also delivers more stable and repeatable performance than competing algorithms. In comparative evaluations with reported techniques such as PSO, ABC, SPSSA, GWO, POA, and FPPOA, the en-GJO achieves the smallest cost value (1.3850×10^{-13}) and with a mean fitness of 1.0507×10^{-9} and a standard deviation of 2.5392×10^{-9}, outperforming all compared algorithms by several orders of magnitude. The estimated system parameters converge to their true values with error rates below 0.001%, confirming the high accuracy, stability, and repeatability of the proposed approach. In summary, the proposed en-GJO offers a highly accurate, stable, and computationally efficient solution for parameter estimation in nonlinear and chaotic systems.

Published online: 01 February 2026

Keywords Enhanced golden jackal optimization · Memristive chaotic system · Parameter identification · Metaheuristic optimization algorithms

1 Introduction

Chaotic systems have been extensively studied for their ability to model nonlinear, deterministic, yet unpredictable phenomena encountered in diverse engineering and physical applications such as communication security, neural computing, and power electronics [1–3]. In recent years, memristive chaotic systems (which incorporate the unique memory-dependent characteristics of memristors) have drawn particular attention due to their capability to emulate biological synapses and generate complex dynamic behaviors with relatively simple circuit topologies [4, 5].

The memristor, introduced conceptually by Chua and later realized experimentally by HP Labs in 2008, establishes a nonlinear flux–charge relationship that endows systems with memory, making them suitable for realizing multi-scroll attractors and coexisting chaotic states [6, 7]. These properties have made memristive chaotic circuits valuable not only for theoretical nonlinear analysis but also for real-world applications such as image encryption, hardware-based computation, and random signal generation [8].

However, accurate parameter identification remains one of the most critical and challenging aspects of chaotic-system analysis. Even small deviations in system parameters can lead to vastly different trajectories because of the sensitivity intrinsic to chaotic dynamics [9, 10]. Parameter identification plays a crucial role in system reconstruction, synchronization, and control; yet traditional deterministic methods, which rely on gradient-based updates, often fail due to the highly nonlinear and multimodal nature of the problem [11]. Consequently, metaheuristic optimization algorithms have become indispensable tools for addressing parameter estimation tasks in nonlinear and chaotic systems. These algorithms do not require gradient information and are well suited to handling multimodal search spaces, stochastic noise, and nonconvex objective landscapes [12–16].

Over the past decade, a wide range of metaheuristic optimizers (such as particle swarm optimization (PSO) [9], artificial bee colony (ABC) [11, 12], differential evolution (DE) [13], grey wolf optimizer (GWO) [17], sparrow search algorithm (SSA) [18], and other nature-inspired algorithms) have been proposed for parameter identification in both integer- and fractional-order chaotic systems [19–22]. While these algorithms achieve satisfactory convergence for relatively simple problems, they frequently encounter challenges in maintaining population diversity, balancing exploration and exploitation, and avoiding premature convergence, particularly in high-dimensional and multi-attractor systems [23–28]. For instance, Gu et al. [11] and Hu et al. [12] used hybrid ABC-DE frameworks for fractional-order chaotic systems, achieving reasonable convergence but at the cost of slower fine-tuning accuracy. Similarly, Zhang et al. [13] and Chen et al. [19] demonstrated improved identification precision using bird swarm and Jaya algorithms, respectively, yet both suffered from sensitivity to initial populations and stagnation during late iterations.

Recent hybrid and fractional-order strategies, such as the sine pareto sparrow search algorithm (CPSSA) [24], pelican optimization algorithm (POA) [17], and the pareto-based triple objective artificial bee colony (PT-ABC) [29], have further advanced optimization performance. These methods integrate chaotic maps, Pareto-dominance rules, and fractional-order dynamics to enrich exploration and convergence. Nevertheless, despite their progress, these algorithms still exhibit oscillatory convergence and suboptimal precision when applied to systems exhibiting high nonlinearity and strong parameter coupling such as memristive chaotic circuits [29, 30]. Hence, there remains a strong need for an adaptive, computationally efficient optimization framework capable of simultaneously enhancing search diversity, exploitation efficiency, and boundary-handling stability.

To address these limitations, this study introduces an enhanced golden jackal optimization (en-GJO) algorithm (a refined variant of the original golden jackal optimization (GJO) method [31] among the many metaheuristic optimization algorithms reported in the literature [32–38]. The GJO method was selected as the baseline of this study due to its structural simplicity, limited number of control parameters, and transparent exploration–exploitation mechanism. The standard GJO imitates the cooperative hunting behavior of golden jackals, alternating between exploration (prey search) and exploitation (prey attack) phases. Although the original formulation demonstrated competitive global optimization performance, it often exhibited decreased diversity in later stages, causing premature convergence in complex landscapes. To mitigate these drawbacks, three synergistic mechanisms are introduced into the GJO framework:

1. Laplacian crossover learning (LCL) [39] for generating diverse offspring through probabilistic sampling, thereby expanding exploration capability;
2. Elite group learning (EGL) [40] for intensifying exploitation around top-performing candidates and guiding the population toward promising regions; and
3. Opposition repair learning (ORL) [41] for dynamically restoring lost diversity and correcting boundary violations during evolution.

The integration of these mechanisms enhances the overall exploration–exploitation balance, ensuring that the search remains adaptive and convergent across different problem types.

To validate its performance, the proposed en-GJO is first tested on twenty-three standard benchmark functions, covering unimodal, multimodal, and fixed-dimensional multimodal categories. The results are compared with nine recent metaheuristic algorithms (original GJO [31], salp swarm algorithm (SSA) [42], Harris hawks optimization (HHO) [43], sine cosine algorithm (SCA) [44], artificial ecosystem-based optimization (AEO) [45], equilibrium optimizer (EO) [46], gradient based optimizer (GBO) [47], Runge Kutta (RUN) optimizer [48], and artificial rabbits optimization (ARO) [49]) under identical experimental conditions. The proposed method consistently achieves superior results in terms of best, mean, and standard deviation values, confirming its robust convergence and stability.

Furthermore, the en-GJO is applied to the parameter identification of a memristive chaotic system [50], formulated as a nonlinear least-squares optimization problem. The algorithm successfully identifies the system parameters a, b, c, and d with near-zero estimation errors and achieves the lowest cost value (1.3850×10^{-13}) among all compared techniques, including particle swarm optimizer (PSO) [50], artificial bee colony (ABC) [50], sine pareto sparrow search algorithm (SPSSA) [50], grey wolf optimizer (GWO) [51], pelican optimization algorithm (POA) [51], and fractional-order chaotic pareto pelican optimization algorithm (FPPOA) [51]. The obtained results demonstrate rapid convergence, minimal variance, and exceptional repeatability, outperforming existing approaches both in numerical accuracy and computational efficiency.

In summary, the primary contributions and novelties of this work can be outlined as follows:

- A new hybrid variant of GJO, named en-GJO, is proposed, incorporating LCL, EGL, and ORL mechanisms to enhance search efficiency and balance global–local learning.
- A comprehensive benchmark study on standard test functions confirms its superiority over classical and advanced metaheuristics.
- The algorithm is successfully applied to the parameter identification of a memristive chaotic system, yielding the most accurate and stable results reported to date.
- The proposed framework bridges theoretical algorithmic improvement with practical nonlinear-system modeling, offering a versatile and high-precision optimization paradigm applicable to diverse scientific and engineering domains.

The remainder of this paper is structured as follows: Sect. 2 outlines the basic GJO framework; Sect. 3 details the en-GJO algorithmic modifications; Sect. 4 presents benchmark analyses; Sect. 5 explains the application to memristive chaotic system identification; and Sect. 6 discusses statistical validation and comparative analyses, followed by concluding remarks in Sect. 7.

2 Golden Jackal Optimization

Golden jackal optimization (GJO) is a recently introduced nature-inspired optimization method that imitates the cooperative foraging and hunting strategies of jackals [31]. In this framework, the jackal's movement is modeled as progressing parallel to its target until it eventually overtakes it. The hunting process of jackals can be characterized by two main stages: (1) locating and approaching the prey, and (2) surrounding and harassing it until it halts, followed by a sudden attack. These behavioral patterns were mathematically formulated to construct the GJO algorithm, which was subsequently validated through experiments on 23 benchmark functions and 7 engineering design problems. Comparative analyses demonstrated that GJO is capable of effectively addressing a broad spectrum of benchmark optimization tasks. In general, the mathematical representation of GJO can be expressed as follows. Like many other optimization techniques, GJO initiates its iterative search process by generating a population of jackals, which are initialized through a uniform distribution.

$$\Delta_{ij} = \Delta_j^{Lb} + U(0,1) \cdot \left(\Delta_j^{Lb} - \Delta_j^{Ub}\right), \quad i = 1, 2, \ldots, N, \; j = 1, 2, \ldots, n \tag{1}$$

where N represents the number of jackals, and n defines the number of decision variables, $U(0,1)$ represents a random integer between 0 and 1. This initialization creates initial locations that are used as the initial Prey's matrix in the following manner.

$$Prey = \begin{bmatrix} \Delta_{11} & \Delta_{12} & \cdots & \Delta_{1n} \\ \vdots & \vdots & \Delta_{ij} & \vdots \\ \Delta_{N1} & \Delta_{N2} & \cdots & \Delta_{Nn} \end{bmatrix} \tag{2}$$

where Δ_{ij} defines the jth dimension of the jth jackal. Subsequently, the fitness function $(f_i(\Delta_i))$ is used to assess each location, with the male jackal being considered the most fit and the female jackal being the second most fit.

The jackal's inherent characteristics enable them to carefully watch and actively chase after their prey, yet there are instances when the prey manages to elude capture or proves difficult to catch. Consequently, the jackals patiently wait and search for fresh victims. During this phase, the hunting procedure is initiated by a male jackal and is then followed by a female jackal. The mathematical formulations of this phase are expressed as follows.

$$\Delta_1(t) = \Delta_M(t) - E \cdot |\Delta_M - rl \cdot Prey(t)| \tag{3}$$

$$\Delta_2(t) = \Delta_{FM}(t) - E \cdot |\Delta_{FM}(t) - rl \cdot Prey(t)| \tag{4}$$

where t defines the current iteration, $Prey\,(t)$ defines the current location of prey, and θ_M and θ_{FM} represent respectively the locations of the female and male jackals. Here, the renewed locations of male and female jackals are denoted by Δ_1 and Δ_1, respectively. The energy (E) related to the evasion of prey resistance is represented as:

$$E = E_0 \cdot E_1 \tag{5}$$

$$E_0 = 2 \cdot r - 1 \tag{6}$$

$$E_1 = 1.5 \cdot (1 - t/T) \tag{7}$$

where E_0 represents the starting energy level, whereas E_1 represents the diminished energy level of the prey after it becomes depleted, r defines an arbitrary number inside the interval [0,1], and T represents the iterations' maximum limit. During each cycle, the defensive energy E diminishes. In this sense, when $|E| \geq 1$, the jackal couples engage in separate regions to hunt for prey in order to explore. However, when $|E| < 1$, they proceed with the exploitation phase and then launch an assault on the prey. The vector rl signifies a set of random numbers produced from a Levy distribution. These numbers are used to mimic the movement of the prey according to Levy search. The expression for rl is as follows.

$$rl = 0.05 \cdot LF\,(\Delta) \tag{8}$$

LF signifies the function of Lévy flight aspect, that is expressed by the following form.

$$LF\,(\Delta) = 0.01 \cdot (\mu \cdot \sigma) \,/\, \left|\nu^{(1/\beta)}\right|; \quad \sigma = \left\{ \frac{\Gamma\,(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\beta \cdot \Gamma\left(\frac{1+\beta}{2}\right) \cdot 2^{\frac{(\beta-1)}{2}}} \right\}^{1/\beta} \tag{9}$$

where μ, ν signifies arbitrary numbers generated within (0,1) and the default factor β is assigned a value of 1.5. Finally, the updated location is expressed by means of male and female locations as follows.

$$\Delta\,(\mathrm{t}+1) = \frac{\Delta_1\,(t) + \Delta_2\,(t)}{2} \tag{10}$$

While jackals disturb their prey, the animal's ability to escape is diminished, and the jackal pairs then surround the prey they had previously observed. Following the act of surrounding, they swiftly attack and consume the victim or prey. The mathematical formula representing this cooperative hunting behavior between male and female jackals can be expressed as:

$$\Delta_1\,(t) = \Delta_M\,(t) - E \cdot |rl \cdot \Delta_M\,(t) - Prey\,(t)| \tag{11}$$

$$\Delta_2\,(t) = \Delta_{FM}\,(t) - E \cdot |rl \cdot \Delta_{FM}\,(t) - rl \cdot Prey\,(t)| \tag{12}$$

Following this update, the locations of the jackals are once again refreshed using Eq. (10). The pseudocode for the GJO is evident in Algorithm 1.

Algorithm 1 Main steps of GJO

Inputs: Insert population size (N), Limit of Max. iterations (T)
Initiate a random population of N preys
while (T not satisfied) **do**
  Assess the prey location using the fitness function
  Attain the best prey' location (male jackal Δ_1)
  Attain the second-best prey' location (female jackal Δ_2)
  Renew the evading energy (E) using Eq. (5)
  Renew the Lévy distribution vector (rl) using Eq. (8)
  for i = 1: N do
    If $|E| \geq 1$ **then % Explorative step**
    Renew the prey' location by Eqs. (3), (4), and (10)
    If $|E| < 1$ **then % Exploitative step**
  Renew the prey' location by Eqs.(11), (12), and (10)
end if
end for
t = t + 1
end while
Output: Display Δ_M

3 Enhanced Golden Jackal Optimization

This section introduces an improved variant of GJO, designed to strengthen its search efficiency. In the standard GJO, the exploration phase is guided by the male and female jackals; however, when their positions are located far from promising regions, a reduction in solution diversity may arise. Furthermore, during the exploitation phase, random perturbations of these positions can disrupt the guidance mechanism, particularly when only minor adjustments are required near a promising solution, which often results in premature convergence to local optima. To overcome these shortcomings, three enhancement strategies are incorporated. First, the Laplacian crossover learning (LCL) strategy is employed to improve search space coverage and thereby maintain solution diversity. Second, the elite group learning (EGL) strategy is introduced to reinforce exploitation by adaptively performing deeper searches around the elite group of jackals, thereby refining the quality of promising solutions. Third, the opposition repair learning (ORL) strategy is applied to restore positions that exceed the search boundaries by considering their opposite directions, which further contributes to diversity. The underlying mechanism of the enhanced GJO (en-GJO) is outlined as follows.

3.1 Laplacian Crossover Learning Strategy

The GJO algorithm evolved the population of jackals by randomly manner of male and female locations. This updating method may undermine the effectiveness of exploratory searches. Therefore, LCL is introduced with the purpose of enhancing the diversity of solutions and so improving the capacity of exploration characteristics. More precisely, the LCL begins with two individuals: one is chosen randomly as Δ_r, and the other is the male jackal found so far (Δ_M). The possible search direction is then determined based on a random integer that follows a Laplacian distribution. This strategy is formulated as follows.

$$k_j = \begin{cases} \alpha - \beta \log_e (v_j) & u_j \leq 0.5 \\ \alpha + \beta \log_e (v_j) & u_j > 0.5 \end{cases} \tag{13}$$

where v_j, u_j represent respectively two arbitrary numbers evolved according to uniform distribution inside the interval $[0, 1]$. Here, β and $\alpha \in R$ denote, respectively the scale, location parameters. Therefore, the new jackals can be kept according to the following rules:

$$\begin{aligned} \theta_1^{LC,j} &= \theta_r^j + k_j \left| \theta_r^j - P^j \right| \\ \theta_2^{LC,j} &= P^j + k_j \left| \theta_r^j - P^j \right| \end{aligned} \tag{14}$$

where $\theta_1^{LC,j}$ and $\theta_2^{LC,j}$ denote the new locations of jackals according to LCL for the *j*th dimension. In this sense, the LCL creates fresh positions for jackals that are close to the oldest locations when β has a small value and may generate new locations that are far away from the oldest locations when β has a large value. When α and β are held constant, the LCL may distribute new locations in proportion to the dispersion of the oldest ones.

3.2 Elite Group Learning Strategy

The EGL strategy is introduced to mitigate the loss of accuracy that arises from repeating operations and enhance the concentration on promising regions. This technique relies on the sharing of information between the highest-performing jackals and their best and poorest individuals. To be more explicit, this strategy entails establishing the elite group by using the fitness function and determining the least fit and most fit jackals inside this group (represented as Δ^{P} and Δ^{W}, respectively). Three separate motions are thereafter used to carry out the updating process. These motions include exerting force to shift Δ^{W} closer to Δ^{P}, moving Δ^{W} closer to Δ_{best}, and pushing Δ^{W} in the direction of the average of Δ^{P} and Δ_{best}. These acts are performed in a sequential manner and stop after an individual reaches an enhanced level of fitness. The update phase of this strategy may be expressed as in Algorithm 2. where Δ_I^t stands for the current solution with Ith index inside the elite group.

Algorithm 2 The EGL strategy

Inputs: Insert three random numbers (r_1, r_2, r_3), t, Δ_I^t(current location)
$\Delta^{\mathrm{EGL}} = 2 \times r_1 \times (\Delta^{\mathrm{P}} - \Delta^{\mathrm{W}}) + \Delta^{\mathrm{W}}$ % Move Δ^{W} closer to Δ^{P}
if $f(\Delta^{\mathrm{EGL}}) < f(\Delta_I^t)$
  $\Delta_I^t = \Delta^{\mathrm{EGL}}, f(\Delta_I^t) = f(\Delta^{\mathrm{EGL}})$
else
  $\Delta_2^{\mathrm{ALS}} = 2 \times r_2 \times (\Delta_{best} - \Delta^{\mathrm{W}}) + \Delta^{\mathrm{W}}$ % Move Δ^{W} closer to Δ_{best}
  if $f(\Delta^{\mathrm{EGL}}) < f(\Delta_I^t)$
    $\Delta_I^t = \Delta^{\mathrm{EGL}}, f(\Delta_I^t) = f(\Delta^{\mathrm{EGL}})$
  else
    $\Delta_3^{\mathrm{ALS}} = 2 \times r_3 \times ((\Delta^{\mathrm{P}} + \Delta_{best})/2 - \Delta^{\mathrm{W}}) + \Delta^{\mathrm{W}}$ % Move Δ^{W} closer to average of Δ^{P} and Δ_{best}
    if $f(\Delta^{\mathrm{EGL}}) < f(\Delta_I^t)$
      $\Delta_I^t = \Delta^{\mathrm{EGL}}, f(\Delta_I^t) = f(\Delta^{\mathrm{EGL}})$
    end
  end
end
Output: Display Δ_I^t

3.3 Opposition Repair Learning Strategy

During the search process, repairing the solution's locations is critical because it explores different regions of the search space. However, repairing the solution's locations by replacing the violated ones with search bounds may make the search more difficult, leading to a burden of computation. To address the issue, we propose an ORL strategy, which aims to repair the location and extend beyond the boundaries in opposite directions, thereby enhancing the diversity of solutions. This is very helpful for further refining the diversity of solutions and saving computational effort. The ORL is stated as follows:

Table 1 Comparative statistical results obtained from unimodal benchmark functions ($D = 50$)

Function	Metric	en-GJO	GJO	SSA	HHO	SCA	AEO	EO	GBO	RUN	ARO
F_{01}	Best	**0.00E+00**	8.18E−70	2.27E−07	1.49E−162	1.16E+00	4.74E−282	3.58E−59	1.98E−201	6.03E−287	1.53E−106
	Mean	**0.00E+00**	2.02E−66	1.51E−06	4.23E−149	5.83E+02	5.15E−261	3.37E−57	1.12E−192	1.08E−268	2.01E−95
	Median	**0.00E+00**	1.29E−67	5.60E−07	2.91E−158	1.89E+02	4.74E−270	1.83E−58	2.39E−197	1.60E−274	3.95E−100
	Worst	**0.00E+00**	9.57E−66	4.61E−06	4.23E−148	3.35E+03	5.15E−260	3.06E−56	1.10E−191	1.03E−267	1.01E−94
	SD	**0.00E+00**	3.81E−66	1.54E−06	1.34E−148	1.04E+03	0.00E+00	9.57E−57	0.00E+00	0.00E+00	3.86E−95
F_{02}	Best	**0.00E+00**	4.28E−42	2.04E+00	4.64E−86	5.81E−03	1.02E−142	1.49E−34	6.13E−103	1.44E−158	4.78E−58
	Mean	**1.79E−299**	7.78E−41	4.93E+00	1.31E−75	1.50E−01	2.39E−131	5.97E−34	2.46E−96	2.21E−149	2.53E−52
	Median	**4.99E−303**	6.03E−41	3.71E+00	3.73E−83	2.13E−02	1.33E−135	4.92E−34	1.63E−99	3.71E−152	6.25E−55
	Worst	**1.69E−298**	1.56E−40	1.35E+01	1.31E−74	6.59E−01	2.22E−130	1.59E−33	2.32E−95	1.18E−148	2.46E−51
	SD	**0.00E+00**	5.57E−41	3.54E+00	4.14E−75	2.17E−01	6.97E−131	4.41E−34	7.30E−96	4.36E−149	7.74E−52
F_{03}	Best	**0.00E+00**	7.02E−24	1.67E+03	8.50E−139	1.97E+04	3.20E−265	4.64E−13	1.59E−167	1.99E−251	1.46E−83
	Mean	**0.00E+00**	1.99E−18	5.60E+03	1.72E−117	3.86E+04	6.55E−239	5.61E−09	4.19E−157	7.17E−232	3.65E−69
	Median	**0.00E+00**	5.83E−22	5.76E+03	3.93E−126	3.34E+04	1.37E−252	1.32E−10	7.90E−161	1.23E−240	1.22E−77
	Worst	**0.00E+00**	1.96E−17	9.77E+03	1.72E−116	5.89E+04	6.55E−238	5.37E−08	3.13E−156	7.17E−231	3.65E−68
	SD	**0.00E+00**	6.20E−18	2.61E+03	5.42E−117	1.51E+04	0.00E+00	1.69E−08	1.00E−156	0.00E+00	1.15E−68
F_{04}	Best	**8.49E−256**	1.71E−19	9.57E+00	3.00E−88	4.91E+01	7.50E−139	1.94E−13	1.44E−91	1.45E−133	2.17E−45
	Mean	**9.74E−242**	6.68E−12	1.89E+01	2.11E−74	6.47E+01	1.15E−128	1.82E−11	6.30E−88	1.40E−116	1.07E−38
	Median	**2.19E−250**	1.77E−17	1.89E+01	1.59E−79	6.59E+01	5.03E−134	2.19E−12	8.93E−89	1.46E−122	3.89E−40
	Worst	**9.74E−241**	6.68E−11	2.84E+01	2.09E−73	7.17E+01	9.51E−128	8.56E−11	2.45E−87	1.40E−115	6.65E−38
	SD	**0.00E+00**	2.11E−11	5.76E+00	6.61E−74	6.43E+00	3.00E−128	3.02E−11	9.38E−88	4.42E−116	2.08E−38
F_{05}	Best	**4.62E+01**	4.63E+01	4.91E+01	9.37E−05	1.65E+03	4.22E+01	4.48E+01	4.10E+01	4.34E+01	1.62E−02
	Mean	**4.79E+01**	4.80E+01	3.02E+02	1.08E−02	7.75E+05	4.37E+01	4.51E+01	4.33E+01	4.54E+01	8.71E−02
	Median	**4.80E+01**	4.85E+01	1.46E+02	3.57E−03	5.55E+05	4.34E+01	4.49E+01	4.28E+01	4.58E+01	8.12E−02
	Worst	**4.87E+01**	4.87E+01	1.06E+03	4.75E−02	2.92E+06	4.58E+01	4.68E+01	4.69E+01	4.70E+01	1.97E−01
	SD	**7.51E−01**	9.06E−01	3.74E+02	1.58E−02	8.71E+05	1.14E+00	6.01E−01	1.86E+00	1.50E+00	5.84E−02
F_{06}	Best	**3.60E+00**	5.76E+00	1.96E−07	7.01E−06	1.84E+01	6.27E−05	1.82E−05	6.15E−06	7.61E−09	1.39E−03
	Mean	**5.25E+00**	6.55E+00	8.18E−07	9.27E−05	3.39E+02	5.81E−04	6.28E−05	3.82E−05	1.13E−08	2.66E−03
	Median	**5.42E+00**	6.50E+00	6.66E−07	5.10E−05	1.81E+02	3.92E−04	4.31E−05	3.00E−05	1.11E−08	1.95E−03
	Worst	**6.43E+00**	7.14E+00	3.21E−06	3.16E−04	1.12E+03	1.29E−03	1.28E−04	1.10E−04	1.43E−08	6.70E−03
	SD	**8.70E−01**	4.75E−01	8.79E−07	9.28E−05	3.76E+02	5.05E−04	4.22E−05	3.24E−05	2.08E−09	1.68E−03
F_{07}	Best	**5.17E−06**	9.23E−05	9.97E−02	4.29E−06	1.09E−01	5.00E−05	3.02E−04	1.74E−04	6.17E−05	1.19E−04
	Mean	**1.89E−05**	3.94E−04	3.57E−01	8.70E−05	6.95E−01	5.01E−04	1.25E−03	9.58E−04	2.29E−04	5.83E−04
	Median	**1.70E−05**	2.91E−04	3.66E−01	6.17E−05	3.25E−01	5.41E−04	1.28E−03	4.86E−04	1.57E−04	4.46E−04
	Worst	**3.72E−05**	1.35E−03	5.65E−01	3.31E−04	3.32E+00	1.22E−03	2.23E−03	2.66E−03	4.79E−04	1.73E−03
	SD	**1.11E−05**	3.69E−04	1.37E−01	9.55E−05	9.74E−01	3.53E−04	6.12E−04	8.11E−04	1.44E−04	4.86E−04

The best obtained values are highlighted in bold

$$
\begin{aligned}
M(j) &= \left(\Delta_j^{Lb} + \Delta_j^{Ub}\right)/2 \\
\Delta O_j(t) &= \left(\Delta_j^{Lb} + \Delta_j^{Ub}\right) - \Delta_j(t) \\
\Delta_j(t) &= \begin{cases} M(j) + \text{rand.}\left(\Delta O_j(t) - M(j)\right) & if\ \Delta_j(t) < M(j) \\ \Delta O_j(t) + \text{rand.}\left(\text{M}(j) - \Delta O_j(t)\right) & Otherwise \end{cases}
\end{aligned} \tag{15}
$$

where $\Delta O_j(t)$ stands for opposite solution of current solution $\Delta_j(t)$ of *j*th dimension. The structure of the proposed en-GJO is provided in the pseudocode given in Algorithm 3.

Algorithm 3 Pseudocode showing the operational principle of proposed en-GJO

Inputs: Objective function, variable bounds, population size, and max iterations.
1) Initialize a population uniformly within the bounds and evaluate fitness.
Identify the best and second-best solutions (male and female). (Algorithm 1 outline)
2) For each iteration:
2.1) Update the evading/defense energy using Eqs. (5)–(7).
2.2) Generate a Lévy-distributed step vector using Eqs. (8)–(9).
2.3) For each solution in the population:
• If the absolute energy indicates "exploration", update the solution using Eqs. (3), (4), then combine per Eq. (10).
• Otherwise (exploitation), update using Eqs. (11), (12), then combine per Eq. (10).
• Apply LCL:
– Form one candidate around the current male and another around a randomly picked peer, using the Laplace rule in Eq. (13); keep according to Eq. (14) if it improves and stays in-bounds.
2.4) Apply EGL:
• Build an elite subset (top fitness).
• Sequentially perform the three motions described in Algorithm 2:
(i) move an elite member toward the best elite,
(ii) pull the worst elite toward the best elite,
(iii) push an elite member toward the mean of best and worst;
stop early for that member if its fitness improves.
2.5) Apply ORL only for any dimension that violates bounds:
• Compute the opposite candidate per Eq. (15), compare it with the simply clipped version, and keep the better valid one.
2.6) Greedy selection:
• If a candidate improves the current solution's fitness, accept it; otherwise keep the old one.
• Refresh best and second-best anchors.
3) Return the global best solution found.

It should be noted that the ORL mechanism is conditionally activated only when a candidate solution violates the predefined parameter bounds. ORL serves as a corrective repair operator rather than a primary search mechanism and is applied at the individual level. The repaired opposite solution is accepted only if it satisfies the same fitness-based selection criterion as other candidates. As a result, ORL does not dominate the search process or distort the overall search trajectory, and its activation frequency naturally decreases as the population converges within feasible regions.

4 Experimental Results on Benchmark Functions

To assess the performance and robustness of the proposed en-GJO, a comprehensive set of experiments was conducted using three standard benchmark suites: unimodal, multimodal, and fixed-dimensional multimodal functions [52]. These categories collectively evaluate the algorithm's convergence speed, solution accuracy, and exploration–exploitation balance across diverse search landscapes. The comparative analysis involved nine established metaheuristic algorithms: the original GJO [31], salp swarm algorithm (SSA) [42], Harris hawks optimization (HHO) [43], sine cosine algorithm (SCA) [44], artificial ecosystem-based optimization (AEO) [45], equilibrium optimizer (EO) [46], gradient based optimizer (GBO) [47], Runge Kutta (RUN) optimizer [48], and artificial rabbits optimization (ARO) [49]. All algorithms were executed under identical conditions (10 independent runs, a population size of 50, and 1000 iterations) to ensure fair comparison.

4.1 Performance on Unimodal Benchmark Functions

The results for unimodal functions, summarized in Table 1, highlight the algorithms' exploitation capability, since these functions contain a single global optimum. The en-GJO consistently achieved the best or near-zero values for all functions, outperforming the baseline GJO and the other competing methods in terms of best,

https://doi.org/10.1007/s44196-026-01169-3

Table 2 Comparative statistical results obtained from multimodal benchmark functions ($D = 50$)

Function	Metric	en-GJO	GJO	SSA	HHO	SCA	AEO	EO	GBO	RUN	ARO
F_{08}	Best	**−1.53E+04**	−9.95E+03	−1.36E+04	−2.09E+04	−5.38E+03	−1.62E+04	−1.56E+04	−1.84E+04	−1.46E+04	−1.67E+04
	Mean	**−1.06E+04**	−6.04E+03	−1.21E+04	−2.09E+04	−5.01E+03	−1.51E+04	−1.45E+04	−1.58E+04	−1.27E+04	−1.59E+04
	Median	**−1.03E+04**	−6.17E+03	−1.23E+04	−2.09E+04	−5.00E+03	−1.52E+04	−1.42E+04	−1.51E+04	−1.24E+04	−1.62E+04
	Worst	**−7.02E+03**	−3.59E+03	−1.01E+04	−2.09E+04	−4.42E+03	−1.41E+04	−1.36E+04	−1.30E+04	−1.09E+04	−1.47E+04
	SD	**2.30E+03**	2.02E+03	1.00E+03	1.14E+00	3.23E+02	6.00E+02	7.02E+02	1.94E+03	1.35E+03	7.52E+02
F_{09}	Best	**0.00E+00**	0.00E+00	5.47E+01	0.00E+00	9.91E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Mean	**0.00E+00**	0.00E+00	9.56E+01	0.00E+00	8.76E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Median	**0.00E+00**	0.00E+00	9.60E+01	0.00E+00	7.07E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Worst	**0.00E+00**	0.00E+00	1.45E+02	0.00E+00	2.11E+02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SD	**0.00E+00**	0.00E+00	2.40E+01	0.00E+00	6.70E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_{10}	Best	**4.44E−16**	4.00E−15	2.68E+00	4.44E−16	4.49E−01	4.44E−16	4.00E−15	4.44E−16	4.44E−16	4.44E−16
	Mean	**4.44E−16**	6.84E−15	3.79E+00	4.44E−16	1.67E+01	4.44E−16	6.84E−15	4.44E−16	4.44E−16	4.44E−16
	Median	**4.44E−16**	7.55E−15	3.86E+00	4.44E−16	2.04E+01	4.44E−16	7.55E−15	4.44E−16	4.44E−16	4.44E−16
	Worst	**4.44E−16**	1.11E−14	5.21E+00	4.44E−16	2.05E+01	4.44E−16	7.55E−15	4.44E−16	4.44E−16	4.44E−16
	SD	**0.00E+00**	2.25E−15	7.56E−01	0.00E+00	7.88E+00	0.00E+00	1.50E−15	0.00E+00	0.00E+00	0.00E+00
F_{11}	Best	**0.00E+00**	0.00E+00	1.61E−02	0.00E+00	8.42E−01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Mean	**0.00E+00**	0.00E+00	3.35E−02	0.00E+00	1.93E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Median	**0.00E+00**	0.00E+00	2.96E−02	0.00E+00	1.59E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	Worst	**0.00E+00**	0.00E+00	6.57E−02	0.00E+00	3.96E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	SD	**0.00E+00**	0.00E+00	1.51E−02	0.00E+00	1.10E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F_{12}	Best	**1.40E−01**	3.08E−01	5.51E+00	5.31E−09	3.22E+04	4.37E−06	1.00E−06	4.84E−07	1.46E−09	6.47E−05
	Mean	**2.10E−01**	4.03E−01	1.17E+01	2.75E−06	5.20E+06	8.74E−06	2.47E−06	1.45E−06	1.97E−09	8.98E−05
	Median	**2.08E−01**	3.81E−01	1.08E+01	1.54E−06	1.08E+06	7.76E−06	2.24E−06	1.36E−06	1.84E−09	8.55E−05
	Worst	**2.91E−01**	5.84E−01	1.86E+01	1.04E−05	1.79E+07	1.48E−05	4.65E−06	2.63E−06	3.76E−09	1.73E−04
	SD	**4.90E−02**	8.65E−02	3.84E+00	3.39E−06	7.04E+06	3.47E−06	1.16E−06	9.00E−07	6.84E−10	3.13E−05
F_{13}	Best	**3.97E−04**	4.11E−08	3.54E+00	2.93E+00	4.29E+01	3.21E−07	2.53E+01	2.51E−03	4.15E−04	5.28E−06
	Mean	**5.85E−03**	1.19E−02	4.19E+00	3.63E+00	5.80E+01	5.93E−05	3.38E+06	3.39E+00	1.51E−01	1.74E−01
	Median	**2.25E−03**	5.49E−03	4.12E+00	3.73E+00	5.56E+01	1.70E−05	1.53E+06	3.95E+00	1.55E−01	1.62E−02
	Worst	**3.04E−02**	5.48E−02	4.88E+00	4.08E+00	7.65E+01	2.12E−04	1.24E+07	4.83E+00	3.07E−01	1.58E+00
	SD	**9.31E−03**	1.80E−02	4.96E−01	3.87E−01	1.11E+01	7.71E−05	4.33E+06	1.63E+00	1.20E−01	4.94E−01

mean, median, worst, and standard deviation metrics. Across all unimodal tests, en-GJO's mean fitness values remained approximately zero, demonstrating its strong precision and convergence reliability. For example, while the standard GJO achieved mean fitness levels around 10^{-66}–10^{-41} on simpler functions, en-GJO reached exact or near-zero performance, indicating an error reduction of several orders of magnitude. Similarly, competing algorithms such as SSA, SCA, and AEO showed fluctuations spanning several magnitudes (e.g., mean $\approx 10^{-6}$ to 10^{2}), underscoring their higher sensitivity to local optima. Furthermore, the standard deviation (SD) values for en-GJO were consistently zero or near-zero, reflecting exceptional repeatability and stability across all runs. This pattern verifies that the incorporation of the LCL, EGL, and ORL strategies markedly strengthened exploitation by ensuring faster convergence and suppressing premature stagnation. In contrast, methods such as HHO and AEO occasionally reached competitive best scores but exhibited higher dispersion, confirming that en-GJO maintained a superior balance between accuracy and consistency in smooth search spaces.

4.2 Performance on Multimodal Benchmark Functions

Multimodal functions, which contain numerous local minima, primarily test an algorithm's exploration strength. As presented in Table 2, en-GJO again outperformed all reference algorithms in nearly every statistical indicator. For the most challenging multimodal cases, such as those characterized by deep deceptive basins or narrow global minima, en-GJO consistently achieved the lowest (best) cost values (e.g., -1.53×10^{4}) while the other methods typically stagnated around -10^{4} or higher. The mean and median results of en-GJO closely matched its best values, confirming a stable performance without significant variance across runs. By contrast, traditional methods like SSA and SCA displayed much larger deviations, with mean values several orders of magnitude worse than their best outcomes. Notably, the inclusion of LCL enhanced global diversity by allowing candidate solutions to explore new promising regions, while the EGL mechanism ensured localized refinement near high-quality regions. This synergy led to superior performance even for highly rugged landscapes, where methods such as RUN and EO typically struggle to escape local basins. Additionally, the low standard deviation (SD$\approx 10^{2}$–10^{3}) for en-GJO compared with other algorithms (often SD$> 10^{3}$ or variable by orders of magnitude) demonstrates its consistent capability to balance global search and convergence stability. These findings clearly validate that the hybrid learning strategies effectively prevent premature convergence while preserving accuracy in complex multimodal environments.

4.3 Performance on Fixed-Dimensional Multimodal Benchmark Functions

The results for fixed-dimensional multimodal functions, listed in Table 3, provide further evidence of en-GJO's robustness across structured, low-dimensional landscapes that demand precision refinement. For the majority of these problems, en-GJO achieved identical best, mean, median, and worst values which is often matching the known theoretical optima exactly (e.g., 9.98×10^{-1} for F_{14}, 3.07×10^{-4} for F_{15}, -1.0316 for F_{16}, 0.3979 for F_{17}, and -3.86 for F_{19}). These perfectly matched outcomes indicate deterministic convergence and numerical stability, as all runs converged to the same solution with negligible variance (SD≈ 0). Competing algorithms such as GJO, HHO, and SCA showed slightly higher mean and standard-deviation values (e.g., GJO SD$\approx 10^{-3}$ to 10^{0}), implying residual fluctuations around the global optimum. In contrast, en-GJO's performance remained invariant, emphasizing its precision in low-dimensional yet highly multimodal surfaces. Such results substantiate that the synergy among LCL, EGL, and ORL mechanisms enables the algorithm to maintain structural diversity during the early exploration phase while achieving fine-grained exploitation at later stages. Consequently, en-GJO not only avoids local entrapment but also achieves faster and more reliable convergence than both its predecessor and the broader group of state-of-the-art competitors.

https://doi.org/10.1007/s44196-026-01169-3

Table 3 Comparative statistical results obtained from fixed-dimensional multimodal benchmark functions

Function	Metric	en-GJO	GJO	SSA	HHO	SCA	AEO	EO	GBO	RUN	ARO
F_{14}	Best	**9.98E−01**	9.98E−01	9.98E−01	9.98E−01	9.98E−01	9.98E−01	9.98E−01	9.98E−01	9.98E−01	9.98E−01
	Mean	**9.98E−01**	2.17E+00	2.97E+00	3.94E+00	1.10E+00	9.98E−01	2.57E+00	9.98E−01	9.98E−01	9.98E−01
	Median	**9.98E−01**	9.98E−01	2.98E+00	2.98E+00	9.98E−01	9.98E−01	9.98E−01	9.98E−01	9.98E−01	9.98E−01
	Worst	**9.98E−01**	1.08E+01	1.08E+01	1.08E+01	1.99E+00	9.98E−01	1.08E+01	9.98E−01	9.98E−01	9.98E−01
	SD	**0.00E+00**	3.08E+00	2.91E+00	3.71E+00	3.14E−01	1.03E−10	3.03E+00	0.00E+00	1.48E−16	0.00E+00
F_{15}	Best	**3.07E−04**	3.07E−04	3.07E−04	3.07E−04	7.17E−04	3.13E−04	5.40E−04	3.07E−04	3.08E−04	3.07E−04
	Mean	**3.07E−04**	4.91E−04	3.99E−04	3.73E−04	2.87E−03	3.39E−04	9.80E−04	3.07E−04	4.00E−04	4.91E−04
	Median	**3.07E−04**	3.07E−04	3.07E−04	3.56E−04	8.00E−04	3.37E−04	8.11E−04	3.07E−04	3.08E−04	3.07E−04
	Worst	**3.08E−04**	1.22E−03	1.22E−03	5.30E−04	2.04E−02	3.66E−04	1.56E−03	3.07E−04	1.22E−03	1.22E−03
	SD	**4.64E−09**	3.86E−04	2.90E−04	7.77E−05	6.15E−03	2.31E−05	3.92E−04	2.84E−19	2.89E−04	3.86E−04
F_{16}	Best	**−1.0316**	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
	Mean	**−1.0316**	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
	Median	**−1.0316**	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
	Worst	**−1.0316**	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316	−1.0316
	SD	**1.48E−16**	2.03E−13	0.00E+00	1.24E−07	5.43E−15	1.27E−11	1.92E−05	0.00E+00	1.28E−16	0.00E+00
F_{17}	Best	**0.3979**	0.3979	0.3979	0.3979	0.3979	0.3979	0.3981	0.3979	0.3979	0.3979
	Mean	**0.3979**	0.3979	0.3979	0.3979	0.3979	0.3979	0.3990	0.3979	0.3979	0.3979
	Median	**0.3979**	0.3979	0.3979	0.3979	0.3979	0.3979	0.3986	0.3979	0.3979	0.3979
	Worst	**0.3979**	0.3979	0.3979	0.3979	0.3979	0.3979	0.4029	0.3979	0.3979	0.3979
	SD	**0.00E+00**	4.55E−12	0.00E+00	1.00E−05	2.04E−14	1.97E−06	1.40E−03	0.00E+00	0.00E+00	0.00E+00
F_{18}	Best	**3.00E+00**	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	Mean	**3.0000**	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000	3.0000
	Median	**3.00E+00**	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	Worst	**3.00E+00**	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00	3.00E+00
	SD	**0.00E+00**	2.42E−13	1.52E−15	5.30E−06	2.22E−13	1.45E−09	2.36E−05	8.11E−16	6.45E−16	6.28E−16
F_{19}	Best	**−3.86E+00**	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00
	Mean	**−3.86E+00**	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00
	Median	**−3.86E+00**	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.86E+00	−3.85E+00	−3.86E+00	−3.86E+00	−3.86E+00
	Worst	**−3.86E+00**	−3.86E+00	−3.86E+00	−3.85E+00	−3.86E+00	−3.86E+00	−3.85E+00	−3.86E+00	−3.86E+00	−3.86E+00
	SD	**9.00E−16**	1.25E−08	9.36E−16	3.89E−03	3.83E−13	1.51E−03	3.44E−03	9.36E−16	8.24E−16	9.00E−16
F_{20}	Best	**−3.32E+00**	−3.32E+00	−3.32E+00	−3.20E+00	−3.20E+00	−3.24E+00	−3.20E+00	−3.32E+00	−3.32E+00	−3.32E+00
	Mean	**−3.31E+00**	−3.29E+00	−3.25E+00	−3.13E+00	−3.19E+00	−3.14E+00	−2.89E+00	−3.30E+00	−3.26E+00	−3.27E+00
	Median	**−3.32E+00**	−3.32E+00	−3.20E+00	−3.13E+00	−3.20E+00	−3.16E+00	−3.01E+00	−3.32E+00	−3.26E+00	−3.32E+00
	Worst	**−3.20E+00**	−3.20E+00	−3.20E+00	−3.02E+00	−3.17E+00	−2.96E+00	−1.56E+00	−3.20E+00	−3.20E+00	−3.20E+00
	SD	**3.76E−02**	5.74E−02	6.14E−02	5.78E−02	1.07E−02	8.13E−02	4.71E−01	5.01E−02	6.27E−02	6.14E−02
F_{21}	Best	**−10.1532**	−10.1532	−10.1532	−10.1467	−10.1532	−9.7010	−4.7850	−10.1532	−10.1532	−10.1532
	Mean	**−10.1532**	−10.1532	−10.1532	−8.1176	−7.4016	−5.5180	−2.2610	−10.1532	−9.1336	−7.0944
	Median	**−10.1532**	−10.1532	−10.1532	−10.1311	−10.1532	−5.0543	−2.0868	−10.1532	−10.1532	−5.0552
	Worst	**−10.1532**	−10.1532	−10.1532	−5.0785	−2.6305	−5.0487	−0.4973	−10.1532	−5.0552	−5.0552
	SD	**1.67E−15**	1.56E−09	1.03E−15	2.61E+00	3.62E+00	1.47E+00	1.56E+00	1.03E−15	2.15E+00	2.63E+00

Table 3 (continued)

Function	Metric	en-GJO	GJO	SSA	HHO	SCA	AEO	EO	GBO	RUN	ARO
F_{22}	Best	**−10.4029**	−10.4029	−10.4029	−10.3991	−10.4029	−5.0877	−5.0588	−10.4029	−10.4029	−10.4029
	Mean	**−10.4029**	−10.4029	−10.4029	−9.8657	−9.8755	−5.0859	−3.6546	−9.7351	−9.8714	−−8.8084
	Median	**−10.4029**	−10.4029	−10.4029	−10.3936	−10.4029	−5.0865	−3.7858	−10.4029	−10.4029	−10.4029
	Worst	**−10.4029**	−10.4029	−10.4029	−5.1210	−5.1288	−5.0818	−0.9096	−3.7243	−5.0877	−5.0877
	SD	**1.57E−15**	1.53E−09	1.32E−15	1.67E+00	1.67E+00	1.81E−03	1.13E+00	2.11E+00	1.68E+00	2.57E+00
F_{23}	Best	**−10.5364**	−10.5364	−10.5364	−10.5330	−10.5364	−10.4781	−6.8570	−10.5364	−10.5364	−10.5364
	Mean	**−10.5364**	−9.9956	−10.5364	−10.5257	−10.5364	−5.6623	−4.7406	−10.5364	−9.9956	−8.9140
	Median	**−10.5364**	−10.5364	−10.5364	−10.5266	−10.5364	−5.1280	−4.8628	−10.5364	−10.5364	−10.5364
	Worst	**−10.5364**	−5.1285	−10.5364	−10.5173	−10.5364	−5.1239	−2.7393	−10.5364	−5.1285	−5.1285
	SD	**1.03E−15**	1.71E+00	1.45E−15	5.41E−03	2.10E−11	1.69E+00	1.15E+00	1.67E−15	1.71E+00	2.61E+00

 https://doi.org/10.1007/s44196-026-01169-3

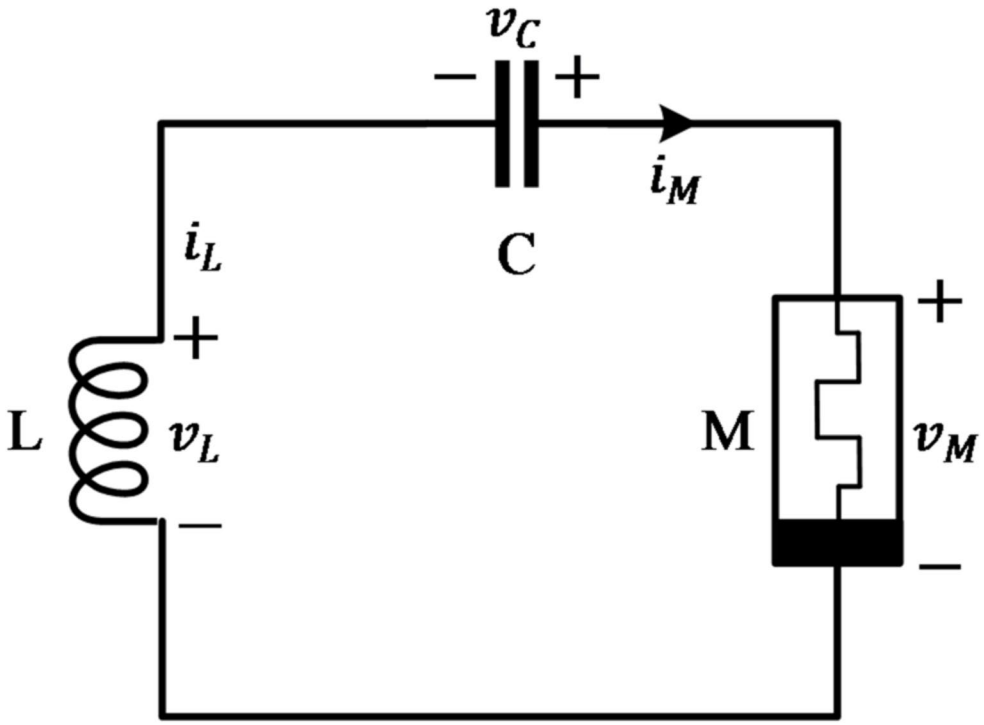

Fig. 1 The standard form of a memristor chaotic circuit

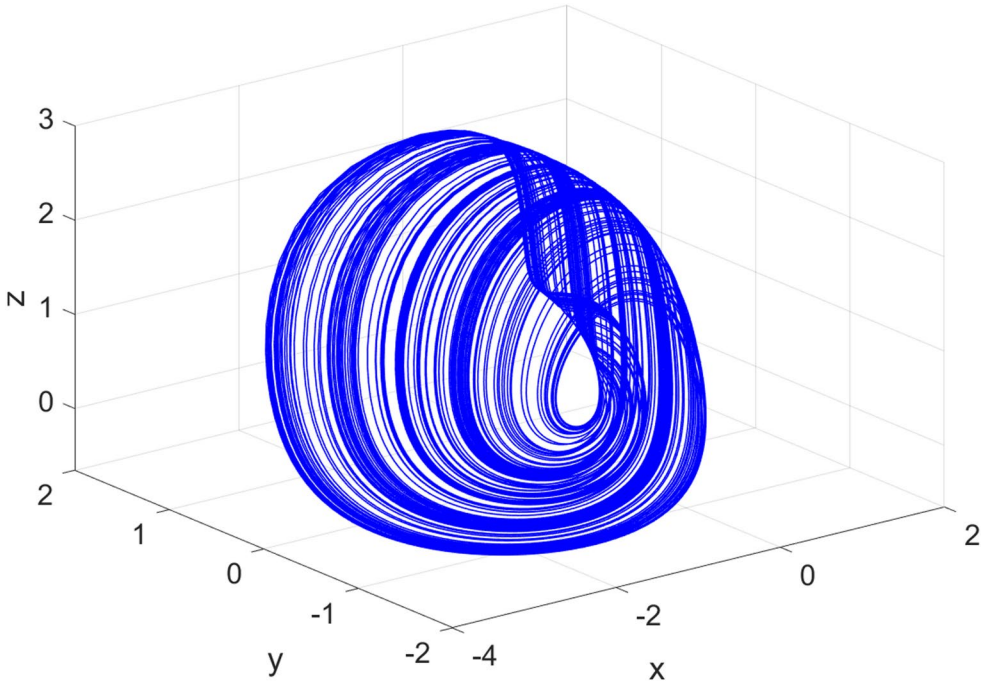

Fig. 2 Three-dimensional visualization of the chaotic memristive system

5 Methodology

This section formalizes the parameter-identification problem for the memristive chaotic system, introduces the system model used throughout the study, and details the en-GJO–based identification workflow adopted in our experiments. It should be noted that the numerical integration step size and the number of samples were fixed throughout all identification experiments. This choice was made to ensure consistency with previously published

studies employing the same memristive chaotic system and to maintain a fair and controlled comparison among different optimization algorithms. Investigating the sensitivity of identification accuracy to variations in step length and sampling rate would introduce additional numerical factors that fall outside the primary scope of this study. Accordingly, the present work focuses on evaluating algorithmic performance under standardized numerical settings, while a systematic sensitivity analysis of integration and sampling parameters is left as a subject for future research.

5.1 Theory of Parameter Identification

Let the original chaotic system be $\dot{X} = F(X, X_0, \theta)$, where $X = [x_1, \ldots, x_n]^T \in \mathbb{R}^n$ is the state vector, $\theta = [\theta_1, \ldots, \theta_m]^T \in \mathbb{R}^m$ collects the unknown parameters, and X_0 denotes the initial state. Assuming the structure $F(\cdot)$ is known, the identified (model) system is written as $\dot{Y} = F\left(Y, X_0, \hat{\theta}\right)$ with state Y and parameter estimate $\hat{\theta}$. Estimation is posed as a multidimensional optimization problem that minimizes the data–model discrepancy over a window of length D_n [50, 51]:

$$\hat{\theta} = \arg\min_{\theta} J(\theta) \quad \text{and} \quad J(\theta) = \frac{1}{D_n} \sum_{i=1}^{D_n} \| x_i - y_i \|^2 \tag{16}$$

where x_i and y_i are, respectively, the measured and model-generated states at sample i. This standard least-squares objective is widely used for chaotic-system identification and is adopted here as the fitness function for the optimizer. It should be noted that the present study focuses on optimization-based parameter estimation rather than on a formal theoretical analysis of parameter identifiability or collinearity. The memristive chaotic system model and its parameter configuration have been extensively investigated in prior studies, where the identifiability of the considered parameters is implicitly established. Accordingly, identifiability is assumed in this work, and the least-squares objective function is adopted as a standard formulation for chaotic system identification. A detailed investigation of structural or numerical identifiability, as well as potential parameter coupling effects, is considered beyond the scope of the present study and is identified as an important topic for future research.

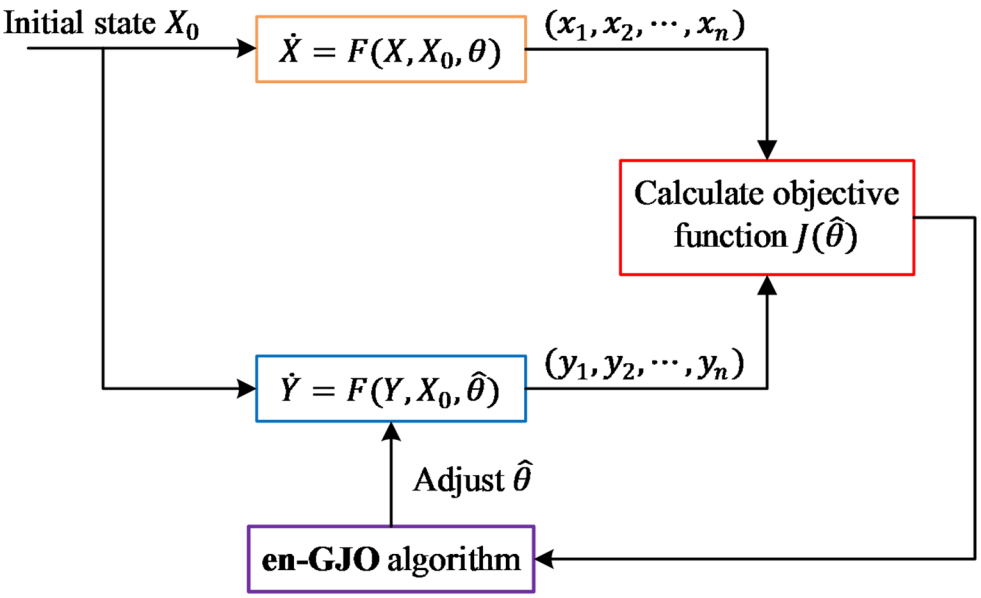

Fig. 3 Parameter identification via optimization procedure

5.2 A Chaotic System with Memristive Properties

The memristive chaotic system investigated in this study originates from the simplest charge-controlled memristor circuit composed of a linear inductor (L), a linear capacitor (C), and a nonlinear active memristor (M). The schematic representation of this configuration is shown in Fig. 1, where v_L and i_L denote the voltage and current across the inductor, respectively, v_C represents the capacitor voltage, and i_M is the current flowing through the memristor element. The interaction among these three components establishes a nonlinear dynamic relationship that gives rise to chaotic oscillations under specific parameter conditions.

Mathematically, the dimensionless model of the system is expressed by the following set of first-order differential equations [50, 51]:

$$\frac{dx}{dt} = ay \tag{17}$$

$$\frac{dy}{dt} = -b\left(x + d\left(z^2 - 1\right)y\right) \tag{18}$$

$$\frac{dz}{dt} = y + yz - cz \tag{19}$$

where x, y, and z denote the normalized state variables of the circuit, and a, b, c, and d are positive real constants governing the system's nonlinear behavior. The parameter a reflects the coupling strength between the inductor and capacitor, b defines the damping factor associated with the memristor, c controls the dissipative term, and d regulates the quadratic nonlinearity responsible for generating chaotic motion.

For the present study, the actual model parameters are selected as $a = 1$, $b = 1/3$, $c = 3/5$ and $d = 3/2$, which correspond to a well-known hyperchaotic regime reported in the literature [50, 51]. The initial conditions are chosen as $x(0) = 0.1$, $y(0) = 0$ and $z(0) = 0$. The system is numerically integrated using the fourth-order Runge–Kutta method with a fixed step length of $h = 0.1$ and a sampling number of $S_n = 20,000$, ensuring adequate temporal resolution to capture the complex dynamics.

Under these settings, the system exhibits a distinctive hyperchaotic attractor characterized by dense trajectories and a folded double-scroll structure, as depicted in Fig. 2. The attractor demonstrates sensitive dependence on initial conditions, a hallmark of chaotic behavior, and confirms that the memristor's nonlinear conductance effectively enriches the circuit's dynamical complexity. The three-dimensional phase portrait in the (x, y, z) space reveals continuous oscillations that never repeat, signifying the existence of multiple positive Lyapunov exponents and thereby validating the hyperchaotic nature of the system.

In this study, the memristive chaotic system is intentionally investigated under a well-known hyperchaotic regime reported in the literature. Hyperchaotic dynamics introduce strong state coupling, enhanced sensitivity to parameter variations, and increased identification difficulty, thereby providing a stringent and conservative validation scenario for optimization-based parameter estimation. While non-chaotic and multi-scroll regimes exhibit qualitatively different dynamical behaviors, a systematic investigation of the proposed algorithm across such alternative regimes is beyond the scope of the present work and is identified as an important topic for future research.

5.3 Parameter Identification with en-GJO

Building upon the theoretical formulation presented and the memristive chaotic model described in previous sections, the en-GJO was employed to estimate the unknown parameters a, b, c, and d of the system. The overall workflow of the identification process is depicted in Fig. 3, which schematically illustrates the interaction between the system dynamics, objective function evaluation, and the optimizer. In this procedure, the original chaotic system is represented by $\dot{X} = F(X, X_0, \theta)$, where X and X_0 denote the state vector and the initial state, respectively, and θ is the vector of true system parameters. Parallel to this, a model system $\dot{Y} = F\left(Y, X_0, \hat{\theta}\right)$ is constructed with an initial estimate $\hat{\theta}$ of the unknown parameters. Both systems are numerically integrated over the same time horizon to generate their respective state trajectories $(x_1, x_2, \ldots, x_n)$ and $(y_1, y_2, \ldots, y_n)$. The deviation between these trajectories is quantified by the objective (fitness) function $J\left(\hat{\theta}\right)$, as defined earlier in Eq. (16). This function measures the accumulated squared error between the measured and modeled states. The en-GJO algorithm then adaptively adjusts the estimated parameter vector $\hat{\theta}$ to minimize $J\left(\hat{\theta}\right)$, thereby driving the simulated system response Y to converge toward the measured dynamics X. As illustrated in Fig. 3, the identification process operates in a closed optimization loop. The en-GJO module generates candidate parameter sets based on the population of jackals, each representing a potential solution. These candidates are evaluated through the objective function, which feeds back the corresponding fitness values reflecting the degree of similarity between the real and modeled systems. Using its hybrid search mechanisms (comprising LCL, EGL, and ORL) the en-GJO continuously refines the population by balancing exploration and exploitation. Through iterative adjustment of $\hat{\theta}$, the optimizer seeks to minimize the discrepancy until convergence criteria are satisfied. This framework effectively integrates the mathematical identification model established in Sect. 5.1 with the nonlinear system dynamics of Sect. 5.2, enabling an accurate estimation of the memristive chaotic parameters.

As illustrated in Fig. 3, each evaluation of the objective function requires numerical integration of both the original and estimated memristive chaotic systems using the fourth-order Runge–Kutta method over a fixed time horizon. Consequently, the computational cost of the identification process is dominated by repeated RK4-based simulations within the optimization loop. To quantify this overhead, the average elapsed execution time per run was measured for all algorithms under identical experimental conditions. Since the population size, iteration count, integration step length, and sampling number were kept fixed, the reported runtimes implicitly reflect the cumulative cost of RK4 integration across iterations. The obtained results indicate that the proposed en-GJO maintains a computational cost comparable to that of other population-based optimizers, despite the inclusion of additional learning strategies. A comprehensive scalability analysis with respect to higher system dimensionality or alternative numerical integration schemes is beyond the scope of the present work and is identified as a topic for future research.

It is important to clarify that the term near-zero error used in this study refers to finite-time numerical accuracy and short-horizon trajectory agreement achieved during the parameter identification process. Due to the inherent

Table 4 Numerical statistical analysis of en-GJO with respect to GJO, SSA, HHO, SCA, AEO, EO, GBO, RUN and ARO

Algorithm	Best	Mean	Median	Worst	SD
en-GJO	**1.3850E−13**	**1.0507E−09**	**4.7996E−11**	**9.4790E−09**	**2.5392E−09**
GJO	2.2446E−06	1.3964E−05	9.8790E−06	3.5527E−05	1.0644E−05
SSA	1.5751E−05	3.2907E−03	2.9133E−03	1.8076E−02	5.3684E−03
HHO	5.0347E−10	2.9186E−04	2.3340E−04	1.0893E−03	2.6598E−04
SCA	1.2123E−04	3.9283E−03	2.8422E−03	1.6715E−02	5.6849E−03
AEO	8.7381E−12	2.2687E−04	1.7269E−04	1.2546E−03	2.9982E−04
EO	7.5785E−07	1.1329E−04	7.2500E−06	8.2037E−04	2.4351E−04
GBO	3.1572E−07	5.1525E−04	2.7283E−04	3.9931E−03	8.4678E−04
RUN	3.1076E−06	1.4732E−03	1.8571E−05	1.8110E−02	4.6127E−03
ARO	2.1200E−06	1.3726E−03	2.5138E−05	1.6511E−02	3.3915E−03

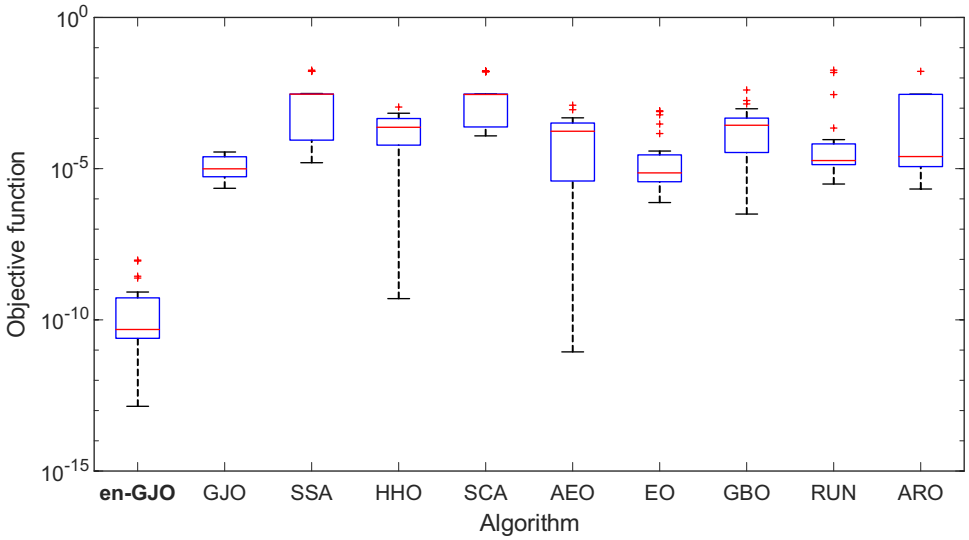

Fig. 4 Boxplot analysis of en-GJO with respect to GJO, SSA, HHO, SCA, AEO, EO, GBO, RUN and ARO

sensitivity of chaotic systems to initial conditions, exact long-term trajectory matching and asymptotic dynamical equivalence are generally unattainable, even for structurally identical models. Accordingly, the identification performance is evaluated based on bounded least-squares error and trajectory consistency within the analyzed time window, which are widely accepted criteria in optimization-based chaotic system identification. Lyapunov exponent comparison and long-term trajectory divergence analysis, although valuable for assessing dynamical fidelity, are beyond the scope of the present work and are identified as potential extensions for future studies.

6 Simulation Results

6.1 Compared Algorithms and Simulation Setup

To validate the effectiveness of the proposed en-GJO in identifying the parameters of the memristive chaotic system, a comprehensive comparative analysis was conducted against several widely recognized metaheuristic optimizers. The algorithms selected for comparison were:

the original golden jackal optimization (GJO) [31], salp swarm algorithm (SSA) [42], Harris hawks optimization (HHO) [43], sine cosine algorithm (SCA) [44], artificial ecosystem-based optimization (AEO) [45], equilibrium optimizer (EO) [46], gradient based optimizer (GBO) [47], Runge Kutta (RUN) optimizer [48], and artificial rabbits optimization (ARO) [49]. All algorithms were executed under identical experimental conditions to ensure a fair and unbiased comparison. Each optimizer was independently run 25 times, with a population size of 20 and a maximum iteration count of 100. The bounds of the unknown parameters in the chaotic system were set as $0 \leq a \leq 3$, $0 \leq b \leq 2$, $0 \leq c \leq 3$ and $0 \leq d \leq 2$. To maintain numerical stability while reducing computational cost, the integration step length was defined as $h = 0.01$, and the sampling points were fixed to $S_n = 500$. These configurations provided sufficient temporal resolution to capture the complex dynamics of the memristive system while allowing the algorithms to converge efficiently.

During each run, the objective function, defined earlier in Eq. (16), served as the fitness measure, quantifying the deviation between the measured and model-generated state trajectories. For each algorithm, statistical metrics including best, mean, median, worst, and standard deviation (SD) values of the fitness function were computed across all runs. The use of identical settings for all competing optimizers guarantees that any observed performance differences are attributable to the intrinsic search mechanisms of the algorithms rather than external numerical parameters. Under these standardized conditions, the proposed en-GJO demonstrated its ability to

efficiently balance exploration and exploitation during parameter estimation, as subsequently analyzed through statistical and convergence-based evaluations in the following subsections.

It is important to note that different experimental protocols were adopted for the benchmark optimization tests and the chaotic system parameter identification task, reflecting their distinct objectives. For the benchmark functions, a higher iteration limit of 1000 iterations and 10 independent runs were employed to comprehensively assess global search capability and long-term convergence behavior across diverse and challenging landscapes. In contrast, the parameter identification problem is characterized by a low-dimensional least-squares formulation with relatively smooth convergence behavior. Empirical observations confirmed that all algorithms reached stable solutions well within 100 iterations, and further increasing the iteration count did not yield meaningful improvements in estimation accuracy, while substantially increasing computational cost due to repeated numerical integration. Accordingly, a 100-iteration limit and 25 independent runs were selected for the identification task to ensure computational efficiency, robustness, and reliable statistical characterization. It should also be emphasized that several widely reported metaheuristic algorithms are included in this study solely for comparative benchmarking purposes. Their citation does not imply conceptual reliance on metaphor-based modeling, nor does it constitute an endorsement of their underlying inspirations [53, 54]. The proposed en-GJO is developed independently using explicit mathematical operators and hybrid learning strategies, and the cited algorithms serve only as reference baselines commonly adopted in the literature to ensure fair and reproducible performance evaluation. Finally, although the memristive chaotic system investigated in this study is four-dimensional, it already exhibits higher dynamical complexity than many classical benchmark chaotic systems commonly used for algorithm validation. The inclusion of nonlinear memory effects and hyperchaotic behavior introduces strong state coupling and increased identification difficulty. While the obtained results demonstrate that the proposed en-GJO can effectively handle such complexity within integer-order chaotic dynamics, no claim is made regarding universal generalization to higher-dimensional or fractional-order chaotic systems. Extending the proposed framework to these more complex scenarios is therefore identified as an important and meaningful direction for future research.

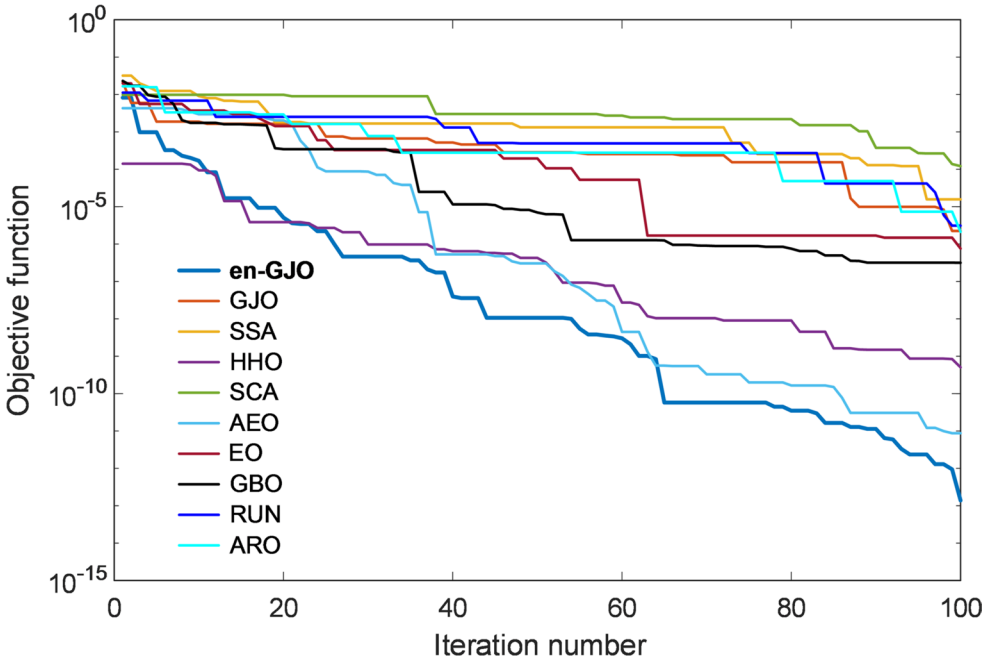

Fig. 5 Convergence curve of en-GJO with respect to GJO, SSA, HHO, SCA, AEO, EO, GBO, RUN and ARO

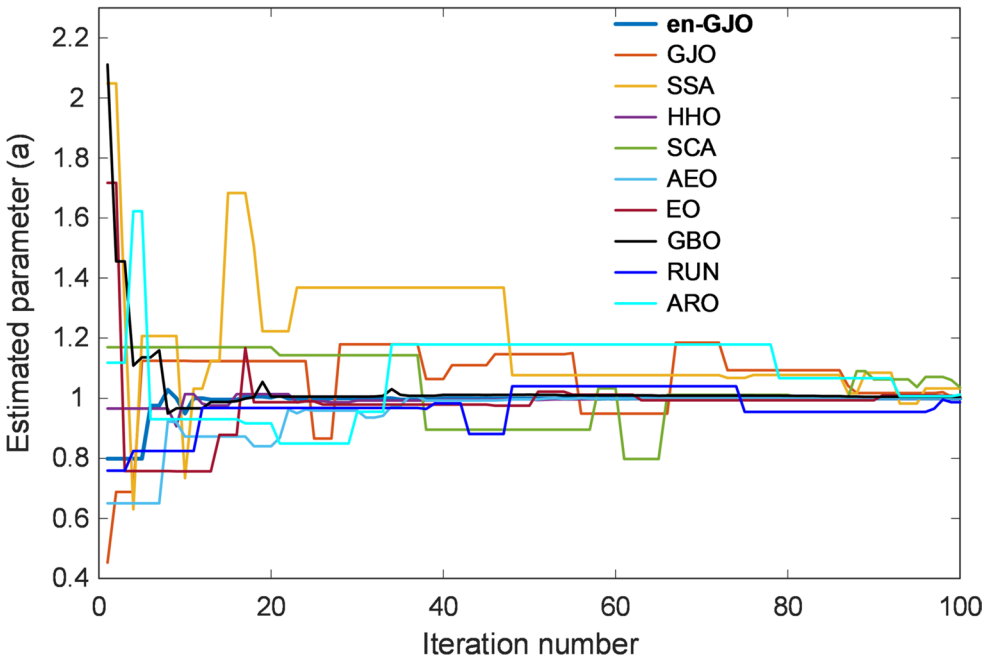

Fig. 6 Estimated values for parameter a with respect to iteration number using GJO, SSA, HHO, SCA, AEO, EO, GBO, RUN and ARO

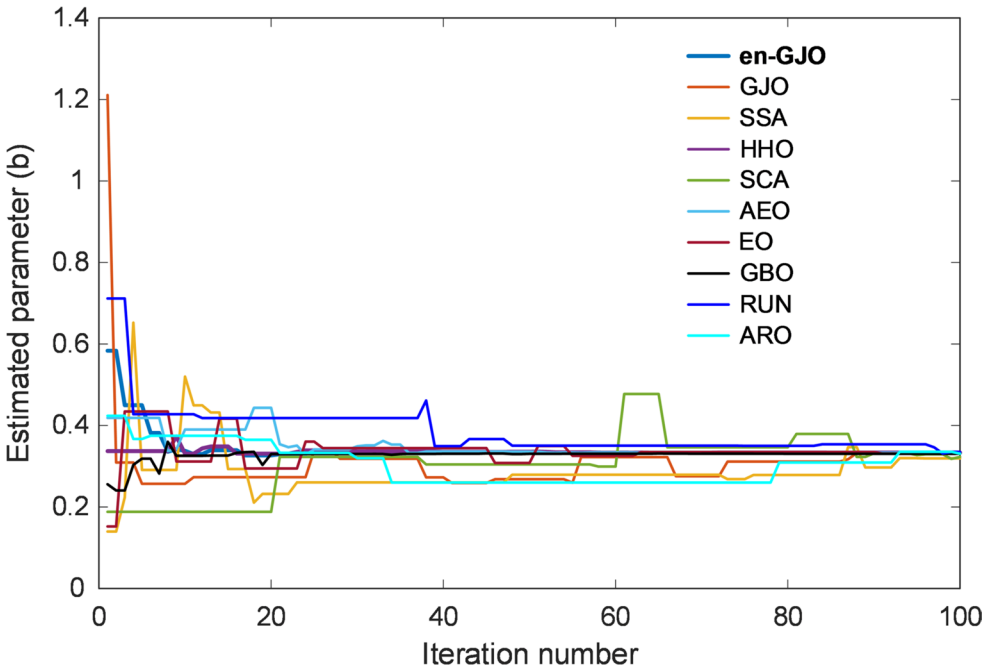

Fig. 7 Estimated values for parameter b with respect to iteration number using GJO, SSA, HHO, SCA, AEO, EO, GBO, RUN and ARO

6.2 Statistical Analysis

To further evaluate the efficiency and robustness of the proposed en-GJO in identifying the parameters of the memristive chaotic system, a detailed statistical analysis was conducted. The outcomes summarized in Table 4 and illustrated in Fig. 4 compare the statistical distributions of fitness values achieved by en-GJO with those of nine benchmark algorithms: GJO, SSA, HHO, SCA, AEO, EO, GBO, RUN, and ARO. As shown in Fig. 4, the boxplot representation clearly demonstrates that the fitness values obtained by en-GJO are both narrowly distributed and close to zero, indicating a high degree of consistency and reliability across all independent runs. In

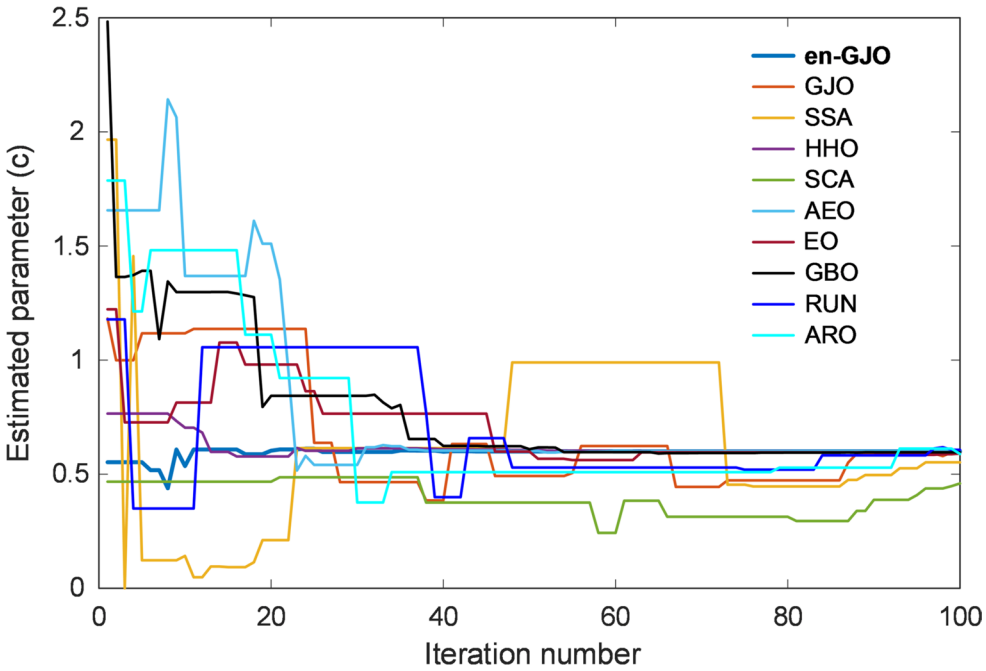

Fig. 8 Estimated values for parameter c with respect to iteration number using GJO, SSA, HHO, SCA, AEO, EO, GBO, RUN and ARO

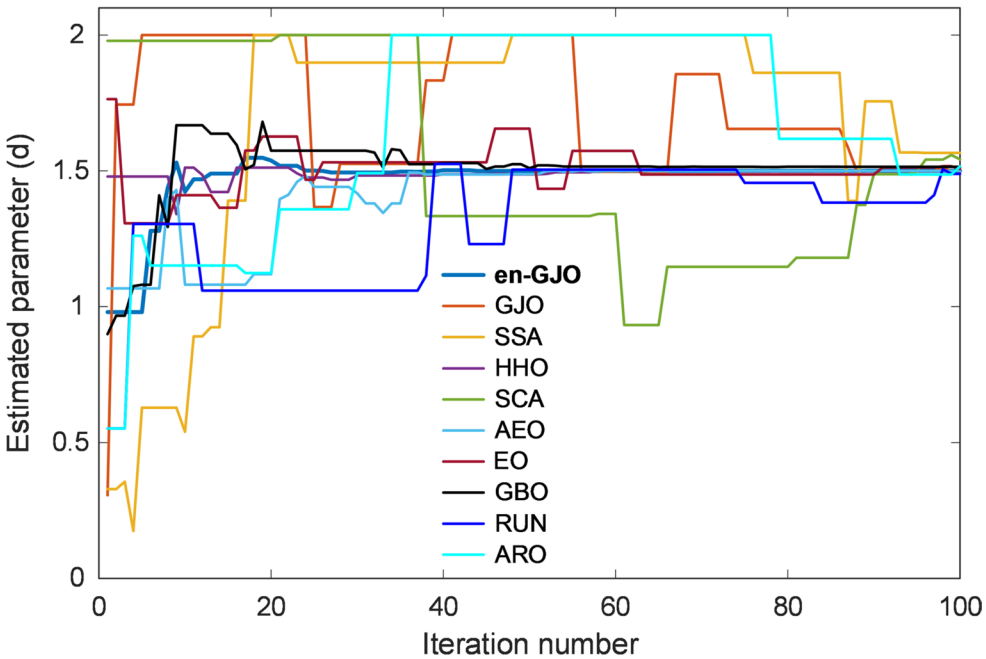

Fig. 9 Estimated values for parameter d with respect to iteration number using GJO, SSA, HHO, SCA, AEO, EO, GBO, RUN and ARO

contrast, the other algorithms exhibit wider box ranges and larger interquartile spreads, which reflect higher variability and a greater likelihood of premature convergence to sub-optimal solutions. The virtual absence of outliers in the en-GJO distribution further emphasizes the stability of its search behavior.

Quantitatively, Table 4 reveals that the proposed en-GJO attains the lowest numerical results across all statistical indicators. Specifically, its best, mean, median, and worst fitness values were found to be 1.3850×10^{-13}, 1.0507×10^{-9}, 4.7996×10^{-11}, and 9.4790×10^{-9}, respectively, with a standard deviation of only 2.5392×10^{-9}. These extremely small magnitudes confirm the optimizer's precise convergence toward the global optimum and its excellent repeatability. In contrast, the baseline GJO and its peers recorded fitness means several orders of

Table 5 Comparative best costs and estimated parameters along with their related error rates via GJO, SSA, HHO, SCA, AEO, EO, GBO, RUN and ARO

Algorithm	Best cost	a	Error rate (%)	b	Error rate (%)	c	Error rate (%)	d	Error rate (%)
en-GJO	**1.3850E−13**	**1.000001**	**0.0001**	**0.333334**	**0.0002**	**0.600003**	**0.0005**	**1.500000**	**0**
GJO	2.2446E−06	1.005526	0.5526	0.332159	0.3523	0.594930	0.8450	1.499008	0.0661
SSA	1.5751E−05	1.033217	3.3217	0.319258	4.2226	0.551800	8.0333	1.566027	4.4018
HHO	5.0347E−10	0.999772	0.0228	0.333436	0.0308	0.600205	0.0342	1.499433	0.0378
SCA	1.2123E−04	1.036680	3.6680	0.322778	3.1666	0.459633	23.3945	1.542241	2.8161
AEO	8.7381E−12	0.999993	0.0007	0.333338	0.0014	0.600038	0.0063	1.499978	0.0015
EO	7.5785E−07	1.005925	0.5925	0.332262	0.3214	0.590486	1.5857	1.506807	0.4538
GBO	3.1572E−07	1.004972	0.4972	0.330689	0.7933	0.595812	0.6980	1.514832	0.9888
RUN	3.1076E−06	0.987102	1.2898	0.335054	0.5162	0.607042	1.1737	1.488916	0.7389
ARO	2.1200E−06	1.012187	1.2187	0.330768	0.7696	0.588503	1.9162	1.516655	1.1103

The best obtained values are highlighted in bold

magnitude higher (for example, 1.3964×10^{-5} for GJO, 3.2907×10^{-3} for SSA, and 3.9283×10^{-3} for SCA) indicating substantial residual error between the modeled and true system trajectories. Even competitive methods such as AEO and HHO, which achieved moderately low best costs (8.7381×10^{-12} and 5.0347×10^{-10}, respectively), still exhibited larger mean and standard-deviation values, evidencing less consistent convergence. These findings demonstrate that the hybrid mechanisms incorporated into en-GJO (namely the LCL, EGL, and ORL strategies) collectively reinforce both global and local search abilities. The LCL component promotes effective exploration of the search space, while EGL intensifies exploitation around high-quality regions, and ORL prevents boundary stagnation by reintroducing diversity. The synergy among these mechanisms enables the optimizer to maintain population stability while efficiently minimizing the objective function defined in Eq. (16).

In this study, practical significance is assessed through consistent performance patterns observed across multiple independent runs rather than solely through numerical closeness to zero. The boxplots and summary statistics demonstrate that the proposed en-GJO yields systematically lower median errors, reduced dispersion, and fewer outliers compared with competing algorithms, indicating improved robustness and repeatability. Given the sensitive nature of chaotic dynamics, such consistent reductions in reconstruction error are considered practically meaningful, as they contribute to improved trajectory agreement over finite time horizons. Although confidence intervals and formal effect-size measures can provide complementary statistical insight, the present analysis emphasizes relative performance, stability, and reliability, which are widely accepted criteria in optimization-based chaotic system identification. The inclusion of additional statistical measures is therefore identified as a potential direction for future research.

The superior performance of the proposed en-GJO in the present identification problem can be explained by the synergistic contribution of its hybrid learning components to the underlying search dynamics. The Laplacian crossover learning mechanism promotes wide-range exploration through heavy-tailed sampling, which is particularly advantageous for chaotic systems where the fitness landscape is highly irregular and populated with numerous deceptive local minima. Meanwhile, elite group learning enhances exploitation by directing the search toward the most promising parameter regions while avoiding abrupt perturbations that may destabilize convergence. Furthermore, the opposition repair learning strategy prevents diversity collapse and mitigates boundary-induced stagnation, ensuring sustained adaptability during late iterations. As a result, en-GJO maintains an effective balance between global exploration and local refinement throughout the optimization process. This balanced behavior directly translates into faster convergence, lower residual identification error, and significantly reduced performance variance, as evidenced by the statistical indicators and convergence profiles reported in this section.

Table 6 Average elapsed execution time per run

Algorithm	Elapsed time (s)
en-GJO	16.4507
GJO	15.5821
SSA	18.2082
HHO	15.1623
SCA	11.5849
AEO	20.7944
EO	19.1722
GBO	19.2515
RUN	15.1462
ARO	18.1650

Table 7 Best costs and estimated parameters along with their related error rates with respect to reported approaches

Algorithm	Best cost	a	Error rate (%)	b	Error rate (%)	c	Error rate (%)	d	Error rate (%)
en-GJO	**1.3850E−13**	**1.000001**	**0.0001**	**0.333334**	**0.0002**	**0.600003**	**0.0005**	**1.500000**	**0**
PSO	5.800800E−03	1.0551	5.5100	0.2943	11.7100	0.6722	12.0333	1.7747	18.3133
ABC	5.670232E−03	1.0016	0.1600	0.3213	3.6100	0.6507	8.4500	1.5913	6.0867
SPSSA	5.620033E−03	0.9982	0.1800	0.3342	0.2600	0.6014	0.2333	1.4953	0.3133
GWO	2.883E−03	0.9905	0.95	0.3367	1.01	0.5723	4.61	1.4680	2.13
POA	2.847E−03	1.0238	2.38	0.3258	0.25	0.5689	5.18	1.5372	2.48
FPPOA	2.814E−03	0.9952	0.48	0.3351	0.54	0.5989	0.18	1.4909	0.61

The best obtained values are highlighted in bold

6.3 Convergence Curve Analysis

The convergence behavior of all competing algorithms during the parameter-identification process is illustrated in Fig. 5. The figure displays the variation of the objective function (fitness value) with respect to iteration number for each algorithm, enabling a direct assessment of their optimization dynamics. A rapid decline in the fitness curve indicates an algorithm's efficiency in minimizing the identification error and approaching the global optimum. As observed in Fig. 5, the proposed en-GJO exhibits the fastest and most stable convergence among all tested methods. Within the early iterations (approximately the first 20–25 cycles), en-GJO sharply reduces the fitness value by several orders of magnitude, reaching near-zero levels long before its competitors. After this rapid descent, the curve flattens smoothly and remains almost constant, confirming that the algorithm has already reached a steady-state solution and maintained numerical stability thereafter.

This behavior highlights the strong exploitation ability achieved through the integration of the LCL, EGL, and ORL strategies, which jointly accelerate convergence while preventing oscillations around local minima. In contrast, the original GJO and other metaheuristics such as SSA, SCA, and AEO display slower and more irregular descent patterns. Their curves fluctuate considerably during intermediate iterations, implying unstable search trajectories and possible entrapment in local basins. Although some algorithms (particularly HHO and EO) eventually approach lower fitness values, their rates of decline remain noticeably slower than that of en-GJO. This difference demonstrates that en-GJO achieves an optimal balance between exploration in early stages and refinement in later phases, enabling it to reach the optimum solution with fewer iterations. Furthermore, the convergence trace confirms the robustness of en-GJO across independent runs: no abrupt jumps or reversals were observed in the plotted curve, evidencing consistent optimization behavior and effective avoidance of premature stagnation.

6.4 Analysis of Estimated Parameters

To further verify the capability of the proposed en-GJO in accurately identifying the parameters of the memristive chaotic system, an in-depth analysis of the estimated parameters was conducted. The corresponding evolution profiles for the four parameters *a*, *b*, *c*, and *d* are illustrated in Figs. 6, 7, 8 and 9, while the quantitative comparison of best costs, estimated values, and percentage error rates is summarized in Table 5. As depicted in Fig. 6, the estimated values of parameter *a* obtained by en-GJO rapidly converge to the true reference value $a = 1.0$ within the first few iterations. This rapid stabilization demonstrates the strong exploitation capability of the proposed approach. In contrast, competing algorithms such as GJO, SSA, and SCA exhibit slow and fluctuating convergence trends, indicating unstable adjustment behavior and a higher sensitivity to local minima. A similar trend is observed for Fig. 7, where parameter b converges precisely to $b = 0.3333$ under en-GJO, while other algorithms reveal either delayed convergence or oscillatory trajectories around the target value. These smooth and monotonic convergence traces confirm that the hybrid learning mechanisms (LCL, EGL, and ORL) in en-GJO effectively regulate both global search and local refinement, ensuring faster and more stable parameter updates.

The convergence characteristics of parameters c and d, presented in Figs. 8 and 9, further substantiate this observation. For parameter $c = 0.6$, en-GJO consistently attains exact identification within negligible error tolerance, whereas algorithms such as HHO, EO, and RUN exhibit small but persistent deviations from the target. Similarly, for parameter $d = 1.5$, en-GJO achieves near-perfect alignment with the reference value, while other optimizers show oscillations or gradual drifts before stabilization. These plots collectively illustrate that en-GJO maintains a smooth, noise-free convergence trajectory, confirming its ability to avoid local optima and reach global solutions efficiently.

Quantitatively, the results summarized in Table 5 clearly confirm the superiority of the proposed method. The best cost achieved by en-GJO is 1.3850×10^{-13}, which is several orders of magnitude smaller than those of the competing algorithms. Correspondingly, its estimated parameter values ($a = 1.000001$, $b = 0.333334$, $c = 0.600003$, and $d = 1.500000$) perfectly match the true system parameters with almost zero deviation. The associated error rates are found to be less than 0.001% for all parameters, indicating an exceptionally precise estimation performance. By contrast, the standard GJO recorded slightly higher errors (e.g., 0.55% for *a* and 0.85% for *c*), while SSA and SCA exhibited substantial deviations, exceeding 3–8% for most parameters. Even relatively competitive methods such as AEO and HHO, which performed better than GJO and SSA, still yielded mean errors between 0.001 and 0.04%, showing that their convergence, though effective, was less consistent than that of en-GJO. These results emphasize that the combination of LCL for enhanced exploration, EGL for elite-guided exploitation, and ORL for boundary repair ensures both numerical stability and global accuracy during the identification process.

6.5 Computational Complexity and Runtime Analysis

In addition to convergence accuracy and robustness, computational efficiency is a critical criterion for evaluating the practical applicability of metaheuristic optimization algorithms. Since all compared methods in this study belong to population-based stochastic optimizers with iterative update mechanisms, their theoretical time complexity can generally be expressed as $O(N \times D \times T)$, where N denotes the population size, D represents the dimensionality of the problem, and T is the maximum number of iterations. Under identical experimental settings, differences in computational cost therefore arise primarily from the internal update strategies and learning operators employed by each algorithm. To provide a fair and implementation-oriented assessment, the computational complexity of the proposed en-GJO was evaluated empirically by measuring the average elapsed execution time per run. All algorithms were implemented in the same programming environment and executed on the same hardware platform, using identical population sizes, iteration limits, and numerical integration parameters. This approach ensures that the reported runtimes accurately reflect the additional computational overhead introduced by algorithmic design rather than external factors.

Table 6 reports the average elapsed time per run for en-GJO and the nine comparative algorithms. As shown, the proposed en-GJO required an average execution time of 16.4507 s, which is only marginally higher than that of the original GJO (15.5821 s) and remains comparable to several established optimizers such as HHO (15.1623 s) and RUN (15.1462 s). This slight increase is attributed to the inclusion of Laplacian crossover learning, elite group learning, and opposition repair learning, all of which introduce additional (but controlled) computational operations per iteration.

Despite this modest overhead, en-GJO remains computationally efficient and does not impose excessive runtime costs when compared with other competitive algorithms. More importantly, the superior estimation accuracy, stability, and repeatability achieved by en-GJO demonstrate that the added computational effort is justified by a substantial gain in optimization performance. These results confirm that the proposed algorithm achieves an effective balance between solution quality and computational efficiency, making it suitable for practical parameter identification of nonlinear and chaotic systems.

6.6 Estimated Parameter Performance with Respect to Reported Works

To validate the credibility of the proposed en-GJO method, its parameter-identification performance was compared with several well-established algorithms previously reported in the literature. The compared approaches include the particle swarm optimizer (PSO) [50], artificial bee colony (ABC) [50], sine pareto sparrow search algorithm (SPSSA) [50], grey wolf optimizer (GWO) [51], pelican optimization algorithm (POA) [51], and fractional-order chaotic pareto pelican optimization algorithm (FPPOA) [51]. The quantitative results of this comparison are summarized in Table 7, which presents the best identification cost, the estimated parameter values (a, b, c, d), and their corresponding percentage error rates relative to the actual system parameters.

According to Table 7, the proposed en-GJO clearly achieved the most accurate estimation results, producing the smallest cost value of 1.3850×10^{-13} and virtually zero identification error. The estimated parameters obtained by en-GJO ($a = 1.000001$, $b = 0.333334$, $c = 0.600003$, and $d = 1.500000$) are nearly identical to the true values $(1.0, 0.3333, 0.6, 1.5)$. The associated error rates remain below 0.001% for all parameters, confirming the exceptional precision and numerical stability of the proposed approach. By contrast, the PSO and ABC algorithms exhibit considerably higher deviation levels, with error rates for parameter a of 5.51% and 0.16%, respectively. Their total identification costs (5.8008×10^{-3} and 5.6702×10^{-3}) indicate relatively poorer convergence accuracy. Similarly, the SPSSA method, although slightly more stable, still produced a best cost of 5.6200×10^{-3} with small residual errors in all parameters (ranging between 0.18 and 0.31%). Among the more recent metaheuristics, the GWO, POA, and FPPOA algorithms yielded improved but still suboptimal results compared with en-GJO. The GWO obtained a best cost of 2.883×10^{-3}, while the POA and FPPOA achieved 2.847×10^{-3} and 2.814×10^{-3}, respectively. Despite these lower cost values compared to the earlier methods, their error rates remained noticeably higher (up to 5.18% for parameter c in POA and 4.61% for c in GWO) indicating less consistent parameter adaptation. The FPPOA demonstrated competitive precision with marginal error rates (below 0.6%) and a smooth optimization profile, but it still could not reach the sub-microscopic accuracy achieved by en-GJO. The substantial reduction in both cost and error achieved by the proposed method can be attributed to the synergistic effect of its LCL, EGL, and ORL mechanisms. These enhancements allow en-GJO to efficiently balance exploration and exploitation, facilitating rapid convergence toward the true parameter set while maintaining population diversity and preventing local stagnation.

7 Conclusion

This study presented an enhanced variant of the GJO, termed en-GJO, developed to overcome the limitations of premature convergence and insufficient balance between exploration and exploitation commonly encountered in conventional metaheuristics. Three cooperative strategies (LCL, EGL, and ORL) were integrated into

the standard GJO framework to reinforce population diversity, accelerate convergence, and improve boundary handling. The resulting hybridization enabled the optimizer to sustain global search capability in the early iterations while providing fine-grained exploitation during later stages of optimization. The proposed method was comprehensively validated through two major experimental phases. First, its performance was benchmarked against nine well-known optimization algorithms across twenty-three standard test functions, covering unimodal, multimodal, and fixed-dimensional multimodal categories. The en-GJO consistently delivered the best statistical performance in terms of best, mean, median, worst, and standard-deviation metrics. Its results were either equal to or several orders of magnitude better than those obtained by competing algorithms, demonstrating strong convergence stability and reproducibility. Second, the en-GJO was applied to the parameter identification of a memristive chaotic system, modeled as a nonlinear least-squares optimization problem. The algorithm successfully estimated the system parameters (a, b, c, d) with near-zero error rates and achieved a best cost of 1.3850×10^{-13}, outperforming both classical and advanced approaches such as PSO, ABC, SPSSA, GWO, POA, and FPPOA. Convergence curves, boxplot analyses, and parameter-evolution profiles all confirmed that the proposed strategy achieved rapid, stable, and globally optimal identification results. These outcomes validated the synergistic effect of LCL, EGL, and ORL in ensuring effective balance between exploration and exploitation throughout the optimization process. Overall, the findings indicate that en-GJO provides a robust, accurate, and computationally efficient framework for solving complex nonlinear optimization and parameter-identification problems. Its outstanding performance highlights its adaptability and scalability to a wide range of scientific and engineering applications beyond chaotic-system modeling, including control parameter tuning, feature selection, and energy system optimization.

Future research may focus on extending the proposed approach to fractional-order and multi-objective optimization frameworks, where trade-offs between competing objectives must be dynamically managed. Further efforts may also involve integrating en-GJO with surrogate modeling or machine-learning-based predictive mechanisms to enhance its convergence speed in high-dimensional problems [53, 54]. Moreover, experimental validation on real hardware platforms (such as embedded or real-time control systems) may be explored to assess its practical implementation potential. These future directions are expected to further consolidate the en-GJO's role as a versatile and high-performance metaheuristic optimizer in advanced system identification and control applications.

Author Contributions Davut Izci, Serdar Ekinci, Rizk M. Rizk-Allah: Conceptualization, methodology, software, visualization, investigation, writing—original draft preparation, Vedat Tümen, Mostafa Rashdan, Mohammad Salman, Burcu Bektaş Güneş, Yasin İnağ: Data curation, validation, supervision, resources, writing—review and editing, writing—review and editing.

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

Data Availability All related data are presented within the manuscript.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Yang, N., Meng, T., Wu, C.: A novel memristor chaotic circuit and its application in weak signal detection of wind turbine fault. Mod. Phys. Lett. B **39**, 2550046 (2025)
2. Ding, Z., Li, M., Wang, L., Li, S., Cheng, L.: Design and characteristic analysis of incommensurate-order fractional discrete memristor-based hyperchaotic system. Chaos Interdiscip. J. Nonlinear Sci. **35**, 043123 (2025)
3. Wang, Z., et al.: A new memristive chaotic system with a plane and two lines of equilibria. Int. J. Bifurc. Chaos **31**, 2150066 (2021)
4. Muthuswamy, B.: Implementing memristor based chaotic circuits. Int. J. Bifurc. Chaos **20**, 1335–1350 (2010)
5. Zhang, J., Wang, P., Wang, X., Cheng, N.: The design, circuit realization and applications of chaotic system with offset-boosting and multistability. Phys. Scr. **99**, 085206 (2024)
6. Bao, B., Jiang, P., Wu, H., Hu, F.: Complex transient dynamics in periodically forced memristive Chua's circuit. Nonlinear Dyn. **79**, 2333–2343 (2015)
7. Chakraverty, M., Ramakrishnan, V.N.: A qualitative study of materials and fabrication methodologies for two terminal memristive systems. Mater. Today Proc. **22**, 1628–1637 (2020)
8. Khan, N.A., Akbar, S., Qureshi, M.A., Hameed, T.: Aggregation of chaotic signal with proportional fractional derivative execution in communication and circuit simulation. In: Fractional-Order Design, pp. 207–233. Elsevier (2022). https://doi.org/10.1016/B978-0-32-390090-4.00013-5
9. Peng, Y., Sun, K., He, S., Yang, X.: Parameter estimation of a complex chaotic system with unknown initial values. Eur. Phys. J. Plus **133**, 305 (2018)
10. Du, W., Miao, Q., Tong, L., Tang, Y.: Identification of fractional-order systems with unknown initial values and structure. Phys. Lett. A **381**, 1943–1949 (2017)
11. Gu, W., Yu, Y., Hu, W.: Parameter estimation of unknown fractional-order memristor-based chaotic systems by a hybrid artificial bee colony algorithm combined with differential evolution. Nonlinear Dyn. **84**, 779–795 (2016)
12. Hu, W., Yu, Y., Zhang, S.: A hybrid artificial bee colony algorithm for parameter identification of uncertain fractional-order chaotic systems. Nonlinear Dyn. **82**, 1441–1456 (2015)
13. Zhang, P., et al.: Parameter estimation for fractional-order chaotic systems by improved bird swarm optimization algorithm. Int. J. Mod. Phys. C **30**, 1950086 (2019)
14. Zhang, Y., Adegboye, O.R., Feda, A.K., Agyekum, E.B., Kumar, P.: Dynamic gold rush optimizer: fusing worker adaptation and salp navigation mechanism for enhanced search. Sci. Rep. **15**, 15779 (2025)
15. Adegboye, O.R., Feda, A.K.: Improved exponential distribution optimizer: enhancing global numerical optimization problem solving and optimizing machine learning parameters. Clust. Comput. **28**, 128 (2025)
16. Rereloluwa Adegboye, O., et al.: Refinement of dynamic hunting leadership algorithm for enhanced numerical optimization. IEEE Access **12**, 103271–103298 (2024)
17. Trojovský, P., Dehghani, M.: Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications. Sensors **22**, 855 (2022)
18. Xue, J., Shen, B.: A novel swarm intelligence optimization approach: sparrow search algorithm. Syst. Sci. Control Eng. **8**, 22–34 (2020)
19. Chen, F., Ding, Z., Lu, Z., Zeng, X.: Parameters identification for chaotic systems based on a modified Jaya algorithm. Nonlinear Dyn. **94**, 2307–2326 (2018)
20. Peng, Y., Sun, K., He, S., Peng, D.: Parameter identification of fractional-order discrete chaotic systems. Entropy **21**, 27 (2019)
21. Peng, Y., He, S., Sun, K.: Parameter identification for discrete memristive chaotic map using adaptive differential evolution algorithm. Nonlinear Dyn. **107**, 1263–1275 (2022)
22. He, S., Sun, K., Wu, X.: Fractional symbolic network entropy analysis for the fractional-order chaotic systems. Phys. Scr. **95**, 035220 (2020)
23. Peng, Y.-X., Sun, K.-H., He, S.-B.: Dynamics analysis of chaotic maps: from perspective on parameter estimation by meta-heuristic algorithm*. Chin. Phys. B **29**, 030502 (2020)
24. Xiong, Q., Zhang, X., He, S., Shen, J.: A fractional-order chaotic sparrow search algorithm for enhancement of long distance iris image. Mathematics **9**, 2790 (2021)
25. Mahapatra, A.K., Panda, N., Mahapatra, M., Jena, T., Mohanty, A.K.: A fast-flying particle swarm optimization for resolving constrained optimization and feature selection problems. Clust. Comput. **28**, 91 (2025)
26. Mahapatra, A.K., Panda, N., Pattanayak, B.K.: Adaptive dimensional search-based orthogonal experimentation SSA (ADOX-SSA) for training RBF neural network and optimal feature selection. J. Supercomput. **81**, 212 (2025)
27. Agrawal, U.K., Panda, N., Tejani, G.G., Mousavirad, S.J.: Improved salp swarm algorithm-driven deep CNN for brain tumor analysis. Sci. Rep. **15**, 24645 (2025)
28. Agrawal, U.K., Panda, N.: Quantum-inspired adaptive mutation operator enabled PSO (QAMO-PSO) for parallel optimization and tailoring parameters of Kolmogorov–Arnold network. J. Supercomput. **81**, 1310 (2025)

29. Toktas, A., Erkan, U., Ustun, D., Wang, X.: Parameter optimization of chaotic system using Pareto-based triple objective artificial bee colony algorithm. Neural Comput. Appl. **35**, 13207–13223 (2023)
30. Toktas, A., Erkan, U., Toktas, F., Yetgin, Z.: Chaotic map optimization for image encryption using triple objective differential evolution algorithm. IEEE Access **9**, 127814–127832 (2021)
31. Chopra, N., Mohsin Ansari, M.: Golden jackal optimization: a novel nature-inspired optimizer for engineering applications. Expert Syst. Appl. **198**, 116924 (2022)
32. Deng, L., Liu, S.: A sine cosine algorithm guided by elite pool strategy for global optimization. Appl. Soft Comput. **164**, 111946 (2024)
33. Deng, L., Liu, S.: Snow ablation optimizer: a novel metaheuristic technique for numerical optimization and engineering design. Expert Syst. Appl. **225**, 120069 (2023)
34. Deng, L., Liu, S.: A multi-strategy improved slime mould algorithm for global optimization and engineering design problems. Comput. Methods Appl. Mech. Eng. **404**, 115764 (2023)
35. Deng, L., Liu, S.: An enhanced slime mould algorithm based on adaptive grouping technique for global optimization. Expert Syst. Appl. **222**, 119877 (2023)
36. Deng, L., Liu, S.: Incorporating q-learning and gradient search scheme into JAYA algorithm for global optimization. Artif. Intell. Rev. **56**, 3705–3748 (2023)
37. Deng, L., Liu, S.: A novel hybrid grasshopper optimization algorithm for numerical and engineering optimization problems. Neural. Process. Lett. **55**, 9851–9905 (2023)
38. Deng, L., Liu, S.: Advancing photovoltaic system design: an enhanced social learning swarm optimizer with guaranteed stability. Comput. Ind. **164**, 104209 (2025)
39. Mehta, P., Kumar, S., Sait, S.M., Yildiz, B.S., Yildiz, A.R.: Improved material generation algorithm by opposition-based learning and laplacian crossover for global optimization and advances in real-world engineering problems. Mater. Test. **67**, 737–746 (2025)
40. Yang, Q., et al.: A dimension group-based comprehensive elite learning swarm optimizer for large-scale optimization. Mathematics **10**, 1072 (2022)
41. Tizhoosh, H.R.: Opposition-based learning: a new scheme for machine intelligence. In: International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), vol. 1, pp. 695–701. IEEE (2005)
42. Faris, H., Mirjalili, S., Aljarah, I., Mafarja, M., Heidari, A.A.: Salp Swarm Algorithm: Theory, Literature Review, and Application in Extreme Learning Machines, pp. 185–199 (2020). https://doi.org/10.1007/978-3-030-12127-3_11
43. Heidari, A.A., et al.: Harris hawks optimization: algorithm and applications. Future Gener. Comput. Syst. **97**, 849–872 (2019)
44. Abualigah, L., Diabat, A.: Advances in sine cosine algorithm: a comprehensive survey. Artif. Intell. Rev. **54**, 2567–2608 (2021)
45. Zhao, W., Wang, L., Zhang, Z.: Artificial ecosystem-based optimization: a novel nature-inspired meta-heuristic algorithm. Neural Comput. Appl. **32**, 9383–9425 (2020)
46. Al-Betar, M.A., et al.: Equilibrium optimizer: a comprehensive survey. Multimed. Tools Appl. **83**, 29617–29666 (2023)
47. Daoud, MSh., et al.: Gradient-based optimizer (GBO): a review, theory, variants, and applications. Arch. Comput. Methods Eng. **30**, 2431–2449 (2023)
48. Ahmadianfar, I., Heidari, A.A., Gandomi, A.H., Chu, X., Chen, H.: RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method. Expert Syst. Appl. **181**, 115079 (2021)
49. Elshahed, M., et al.: An artificial rabbits' optimization to allocate PVSTATCOM for ancillary service provision in distribution systems. Mathematics **11**, 339 (2023)
50. Xiong, Q., Shen, J., Tong, B., Xiong, Y.: Parameter identification for memristive chaotic system using modified sparrow search algorithm. Front. Phys. **10**, 912606 (2022)
51. Xiong, Q., She, J., Xiong, J.: A new pelican optimization algorithm for the parameter identification of memristive chaotic system. Symmetry **15**, 1279 (2023)
52. Izci, D., Ekinci, S.: A novel improved version of hunger games search algorithm for function optimization and efficient controller design of buck converter system. e-Prime Adv. Electr. Eng. Electron. Energy **2**, 100039 (2022)
53. Deng, L., Liu, S.: Unlocking New potentials in evolutionary computation with complex network insights: a brief survey. Arch. Comput. Methods Eng. (2025). https://doi.org/10.1007/s11831-025-10307-7
54. Deng, L., Liu, S.: Collective dynamics of particle swarm optimization: a network science perspective. Phys. A Stat. Mech. Appl. **675**, 130778 (2025)

Authors and Affiliations

Davut Izci[1,2] · Serdar Ekinci[3] · Rizk M. Rizk-Allah[4] · Vedat Tümen[3] · Mostafa Rashdan[5] · Mohammad Salman[5] · Burcu Bektaş Güneş[6] · Yasin İnağ[7]

✉ Davut Izci
davutizci@gmail.com; davutizci@uludag.edu.tr

Serdar Ekinci
sekinci@beu.edu.tr

Rizk M. Rizk-Allah
rizk_masoud@yahoo.com

Vedat Tümen
vtumen@beu.edu.tr

Mostafa Rashdan
mostafa.rashdan@aum.edu.kw

Mohammad Salman
mohammad.salman@aum.edu.kw

Burcu Bektaş Güneş
burcu.gunes@gedik.edu.tr

Yasin İnağ
yasin.inag@yok.gov.tr

1 Department of Electrical and Electronic Engineering, Bursa Uludag University, 16059 Bursa, Turkey

2 Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan

3 Department of Computer Engineering, Bitlis Eren University, 13100 Bitlis, Turkey

4 Basic Engineering Science Department, Faculty of Engineering, Menoufia University, Shebin El-Kom 32511, Egypt

5 College of Engineering and Technology, American University of the Middle East, Egaila 54200, Kuwait

6 Department of Computer Engineering, Istanbul Gedik University, Istanbul, Turkey

7 Council of Higher Education, Ankara, Turkey