RESEARCH ARTICLE

A Comparison of Evolutionary Computation Techniques for Parameter Estimation of Chaotic Systems

Jorge Gálvez [1], **Gustavo R. Hernández** [1], **Omar Avalos** [1], **Angel Casas-Ordaz** [1,*], **Marco Perez-Cinseros** [1] and **Diego Oliva** [1]

[1] Departamento de Ingeniería Electro-Fotónica, Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI), Universidad de Guadalajara, Guadalajara, Jalisco 44430, México

Abstract

In recent years, Parameter Estimation (PE) has become a topic of growing interest due to its broad applications in science and engineering. An important application is the identification of Chaotic Systems (CS), which enables synchronization and control of chaotic behavior. However, the parameter estimation of CS is a highly nonlinear and multidimensional optimization problem where traditional approaches are often unsuitable. To overcome these limitations, Evolutionary Computation Techniques (ECT) have been widely adopted to tackle complex nonlinear optimization tasks. Recently, classical and modern ECT methods have been proposed for estimating the parameters of chaotic systems. However, most reported studies rely exclusively on cost function values, overlooking the quality and consistency of the solutions obtained. This paper presents a comparative study of representative evolutionary techniques for estimating the parameters of chaotic systems. The study assesses the performance of the techniques and the homogeneity of the solutions through statistical analysis. Experimental results on the Lorenz and Chen systems are examined and validated using nonparametric tests.

Keywords: evolutionary computation, chaotic systems, lorenz system, chen system.

1 Introduction

Parameter estimation (PE) is a fundamental research area in many branches of engineering since it is closely related to modeling and controlling nonlinear and highly complex systems. PE plays a significant role in various applications, including control theory, telecommunications, and signal processing [1–3]. Recently, PE has become particularly relevant in identifying and tuning Chaotic Systems (CS), which are characterized by unstable and highly sensitive dynamics dependent on initial conditions. In adaptive control, these systems require synchronization and stabilization, which are typically achieved through feedback-based approaches. Two of the most recognized examples in the literature are the Lorenz system [4] and the Chen system [5], which are frequently adopted as benchmarks for trajectory regulation. The central difficulty in this field is identifying optimal parameter values that enable accurate control. This task can be posed as a

Submitted: 21 August 2025
Accepted: 28 November 2025
Published: 20 December 2025

Vol. 1, **No.** 2, 2025.
10.62762/TSEL.2025.913117

***Corresponding author:**
✉ Angel Casas-Ordaz
angel.casas@academicos.udg.mx

Citation
Gálvez, J., Hernández, G. R., Avalos, O., Casas-Ordaz, A., Perez-Cinseros, M., & Oliva, D. (2025). A Comparison of Evolutionary Computation Techniques for Parameter Estimation of Chaotic Systems. *ICCK Transactions on Swarm and Evolutionary Learning*, 1(2), 83–93.

multidimensional, nonlinear optimization problem.

In practice, PE involves designing strategies that can automatically adjust the parameters of an adaptive model so that the estimated system behaves as closely as possible to the original chaotic system. This process is generally expressed as minimizing an error function that measures the difference between the output of the real system and the estimated model's output. The smaller the error, the higher the quality of the parameter estimation and, consequently, the fidelity of the resulting model.

Historically, gradient-based methods, such as the least mean square algorithm and its improved variants, have been extensively used to solve this problem. However, these methods are only effective when the system is linear in its parameters. The nonlinear nature of CS introduces additional challenges, including large search spaces, sensitivity to parameter variations, and numerous local minima. These limitations often lead to suboptimal outcomes when conventional techniques are used.

In response to these challenges, evolutionary computation techniques (ECT) have emerged as a powerful alternative for solving PE in chaotic systems. Inspired by natural or collective processes, these approaches operate on populations of candidate solutions that exchange information iteratively to approximate the global optimum. Popular methods in this category include particle swarm optimization (PSO) [6], artificial bee colony (ABC) [7], harmony search (HS) [8], differential evolution (DE) [9], cuckoo search algorithm (CSA) [10], and gravitational search algorithm (GSA) [11]. The application of these algorithms to CS parameter estimation has proven highly effective, with studies reporting robust identification and reliable synchronization. PSO and DE stand out among these techniques as representative methods that have been frequently explored in the literature.

The applicability of evolutionary algorithms depends heavily on the problem at hand, as no single optimization method performs optimally in every scenario. Therefore, whenever new algorithms are proposed, it is crucial to evaluate their efficiency. These evaluations are typically carried out using synthetic benchmark functions to verify whether an algorithm properly passes statistical tests. However, there are only a few comparative studies of evolutionary algorithms in the literature, such as the work presented in [12]. This underscores the importance of analyzing and comparing their performance from a practical application perspective [13].

This paper presents a comparative analysis of several evolutionary algorithms for parameter estimation in chaotic systems. The study emphasizes recently developed methods, such as GSA and CSA, while also considering well-established approaches, including PSO, ABC, HS, and DE. The experimental results obtained on two different chaotic systems are examined within a statistically rigorous framework.

The remainder of the paper is structured as follows: Section 2 introduces Evolutionary Computation Techniques. Section 3 outlines parameter estimation for chaotic systems. Section 4 discusses the experimental findings and their statistical validation, and Section 5 presents the conclusions.

2 Evolutionary computation techniques

Optimization in real-world environments is frequently treated as a black-box problem, where only limited information is available. Under these circumstances, the absence of knowledge about the objective or fitness function makes parameter estimation and model analysis difficult. Evolutionary Computation Techniques (ECTs) have shown to be effective in such scenarios, since their operators can be easily adapted to complex search spaces. Most ECTs follow a population-based scheme: a group of candidate solutions is initialized (commonly at random) and iteratively refined to move towards better regions of the search space. Each algorithm applies its own evolutionary operators independently to update the population until convergence. The flexibility of these methods has enabled their successful application across several branches of engineering, particularly in sophisticated optimization settings such as multimodal, dynamic, noisy, or multi-objective problems [14–16].

2.1 Particle swarm optimization (PSO)

Proposed by Kennedy et al. [6], PSO is a stochastic search technique inspired by collective behaviors observed in birds and fish. A swarm of N individuals, referred to as particles, represents possible solutions to the problem. At iteration k ($k = 0 \ldots gen$), each particle $\mathbf{p}_i^k$ is a d-dimensional vector associated with the decision variables of the system. Their quality is assessed through the evaluation function $f(\mathbf{p}_i^k)$, which provides the fitness value. During the iterative process, each particle retains both its best historical position $\mathbf{p}_i^*$ and the best solution found so far in the swarm $\mathbf{g}$.

In this work, we implement the PSO modification introduced by Lin et al. [17]. The particle velocity and position are updated as:

$$v_{i,j}^{k+1} = \omega v_{i,j}^{k} + c_1 r_1 (p_{i,j}^{*} - p_{i,j}^{k}) + c_2 r_2 (g_j - p_{i,j}^{k}), \\ p_{i,j}^{k+1} = p_{i,j}^{k} + v_{i,j}^{k+1} \quad (1)$$

where ω is the inertia weight, c_1, c_2 are acceleration coefficients, and r_1, r_2 are uniform random numbers in $[0, 1]$.

2.2 Artificial bee colony (ABC)

The Artificial Bee Colony algorithm, introduced by Karaboga [7], simulates the foraging behavior of honeybees. A population of N possible food locations evolves across iterations. Each location $\mathbf{l}_i^k$ is a d-dimensional vector where fitness corresponds to nectar quality. Based on this fitness evaluation, locations are modified through neighborhood-based exploration. The main transformation is:

$$\mathbf{t}_i = \mathbf{l}_i^k + \phi(\mathbf{l}_i^k - \mathbf{l}_r^k) \quad (2)$$

where $r \neq i$ and $\phi \in [-1, 1]$ is a random factor. A greedy selection process then decides whether $\mathbf{t}_i$ replaces the current location.

2.3 Cuckoo search (CS)

Yang et al. [10] presented the Cuckoo Search algorithm, inspired by brood parasitism and incorporating Lévy flights [18] for solution exploration. A set of eggs $\{\mathbf{e}_i^k\}$ evolves through three operators: a) Lévy flights for new candidate generation, b) probabilistic nest replacement, and c) elitist selection.

The Lévy flight step uses random walks based on Mantegna's method [19], producing large but rare exploration steps, while elitist selection ensures only solutions with better fitness survive.

2.4 Harmony search (HS)

Harmony Search, proposed by Geem [8], models the improvisation process of musicians striving for harmony. The algorithm generates new solutions by combining three strategies: a) selecting elements from memory (controlled by HMCR), b) pitch adjustment (PAR and bandwidth factors), and c) generating random candidates to guarantee diversity. This combination balances exploration and exploitation of the search space.

2.5 Differential evolution (DE)

Differential Evolution [9] is a population-based algorithm built on three operators: mutation, crossover, and selection. In mutation, three solutions are picked to create a donor vector $\mathbf{v}_i$. This vector undergoes crossover with probability C_r, producing a trial vector $\mathbf{u}_i$. Selection ensures that only the solution with better fitness between $\mathbf{x}_i$ and $\mathbf{u}_i$ survives to the next generation [20].

2.6 Gravitational search algorithm (GSA)

The Gravitational Search Algorithm, proposed by Rashedi in 2009 [11], is based on Newton's laws of gravity and motion. Agents represent candidate solutions and move under the influence of gravitational forces exerted by others. Heavier masses (corresponding to better fitness values) exert stronger attraction, guiding the system toward high-quality solutions. The dynamics are expressed as force, acceleration, velocity, and position update equations, which together drive the search process.

3 Parameter estimation for chaotic systems

Chaotic systems (CS) are inherently nonlinear, which makes them suitable for analysis through nonlinear control and synchronization techniques, such as those reported in Hegazi [21] and Huang [22]. However, when system parameters are unknown, conventional control strategies are no longer applicable. This challenge has motivated extensive research on parameter estimation (PE) methods for chaotic systems.

To formulate the problem, consider the following n-dimensional nonlinear model:

$$\dot{X} = F(\mathbf{X}, \mathbf{X}_0, \boldsymbol{\theta}) \quad (3)$$

where $\mathbf{X} = [x_1, x_2, \ldots, x_n]^T$ is the state vector, $\mathbf{X}_0$ the initial conditions, and $\boldsymbol{\theta} = [\theta_1, \theta_2, \ldots, \theta_n]^T$ the parameter vector. The nonlinear mapping F governs the system dynamics.

For estimation purposes, a model with estimated states and parameters is considered:

$$\hat{X} = F(\hat{\mathbf{X}}, \mathbf{X}_0, \hat{\boldsymbol{\theta}}) \quad (4)$$

where $\hat{\mathbf{X}} = [\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n]^T$ is the estimated state vector and $\hat{\boldsymbol{\theta}} = [\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_n]^T$ represents the estimated parameters.

Since evolutionary algorithms rely on the definition of a fitness function to guide the search, an appropriate performance measure must be selected. In this work, the Mean Squared Error (MSE) is employed, which compares real system outputs with estimated responses over M samples:

$$J(\theta) = \frac{1}{M} \sum_{k=1}^{M} (X(k) - \hat{X}(k))^2 \tag{5}$$

where $X(k)$ denotes the observed state values and $\hat{X}(k)$ the estimated ones at time k. The objective is to minimize the cost function $J(\theta)$ by adjusting the parameter vector. The optimal solution θ^* satisfies:

$$\theta^* = \arg\min(J(\theta)) \tag{6}$$

Figure 1 illustrates this optimization framework. The computational block representing the evolutionary techniques is replaced by each of the algorithms detailed in Section 2, while the chaotic system block is instantiated either by the Chen or Lorenz systems, which are described in the following section.

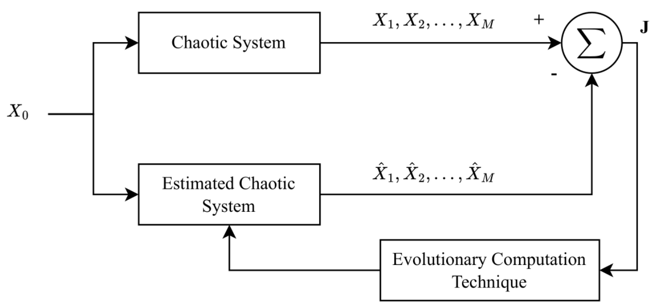


Figure 1. Parameter estimation of chaotic systems using evolutionary computation techniques.

4 Experimental Results

In this study, a set of comparative experiments is conducted to assess the performance of different evolutionary computation techniques. For the parameter estimation task, two chaotic systems are considered: the Lorenz and Chen systems.

The Lorenz system is defined as:

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y - x), \\ \frac{dy}{dt} &= rx - y - xz, \\ \frac{dz}{dt} &= xy - bz \end{aligned} \tag{7}$$

where the original parameters are $\sigma = 10$, $r = 28$, and $b = 8/3$.

Similarly, the Chen system is formulated as:

$$\begin{aligned} \frac{dx}{dt} &= a(y - x), \\ \frac{dy}{dt} &= (b - a)x + by - xz, \\ \frac{dz}{dx} &= -cz + xy \end{aligned} \tag{8}$$

with its original parameters given by $a = 35$, $b = 28$, and $c = 3$. For both systems, the initial conditions are set to $[1, 1, 1]$. The number of states, for both the original and estimated systems, is fixed at 100.

The parameter configuration for each evolutionary computation algorithm employed in the comparison is summarized in Table 1. A common population size of 25 individuals and a maximum of 100 iterations are used for all methods.

Table 1. Parameter settings of the evolutionary computation algorithms.

Algorithm	Parameters
PSO	$c_1 = 2$, $c_2 = 2$; inertia weight decreases linearly from 0.9 to 0.2 [23]
ABC	Implemented following reference guidelines [7]
DE	Variant: DE/rand/bin; $CR = 0.5$, differential weight $= 0.2$ [9]
HS	$HMCR = 0.95$, $PAR = 0.3$, implemented according to [8]
CSA	$p_a = 0.25$, generations $= 100$, based on [10, 24]
GSA	Implemented following reference guidelines [11]

The experimental results are reported in two parts. The first section evaluates the ability of the algorithms to estimate the parameters of the Lorenz and Chen systems, both individually (one parameter at a time) and jointly (two or three parameters simultaneously). The second section provides a statistical analysis of the results using the Wilcoxon Rank Test, complemented with the Bonferroni correction for validation.

4.1 Chaotic system parameter estimation

To assess the effectiveness of the evolutionary computation techniques (ECTs), we designed six experiments that progressively increase in difficulty. The tasks involve estimating one, two, or three parameters of the Lorenz and Chen chaotic systems. This staged design first examines behavior in simple, low-dimensional settings and then moves to more challenging multi-parameter configurations.

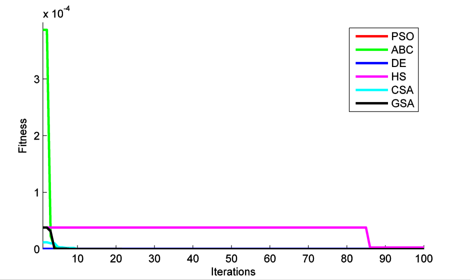
4.1.1 *One-dimensional parameter estimation*

We begin with single-parameter estimation while keeping the remaining parameters fixed at their nominal values. For the Lorenz system, the parameter of interest is either r or b; for the Chen system, one among a, b, or c is tuned. This reduces the search to a one-dimensional space and helps isolate how each method behaves under minimal complexity. All algorithms were run for 100 iterations, and performance was evaluated with the objective function J. The results for the Lorenz system are summarized in Table 2.

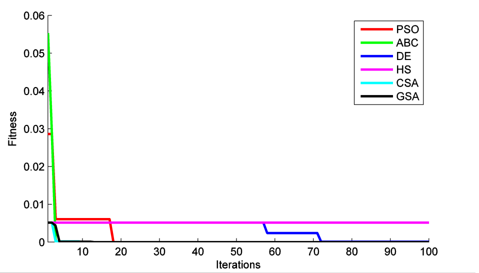
As seen in Table 2, the Gravitational Search Algorithm (GSA) achieves the most accurate estimates, reaching near-zero objective values with consistent stability. CSA and ABC are also competitive, though their final precision is slightly below that of GSA.

An analogous setup was used for the Chen system; results appear in Table 3.

The Chen results follow the same pattern: GSA attains the smallest error values. Figures 2(a)–(b) show the trajectory of J along iterations for both systems. These curves indicate that GSA not only produces more precise solutions but also converges in fewer iterations.



(a)



(b)

Figure 2. Convergence of the objective function in the one-dimensional case for (a) Lorenz and (b) Chen systems.

4.1.2 *Two-dimensional parameter estimation*

We then increase the difficulty by estimating two parameters simultaneously. For Lorenz, we consider the pairs $[\sigma, r]$, $[\sigma, b]$, and $[b, r]$ (Eq. 9); the corresponding outcomes are listed in Table 4.

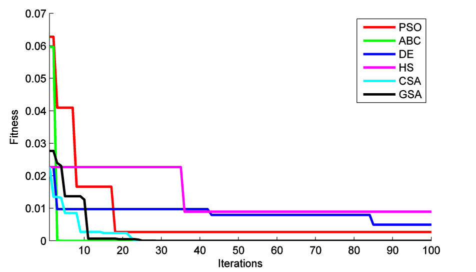
$$[\sigma, r], \ [\sigma, b], \ [b, r] \tag{9}$$

Moving from one to two parameters expands the search space and makes local minima more likely. Even so, GSA maintains high accuracy and typically outperforms the remaining methods across the tested pairs.

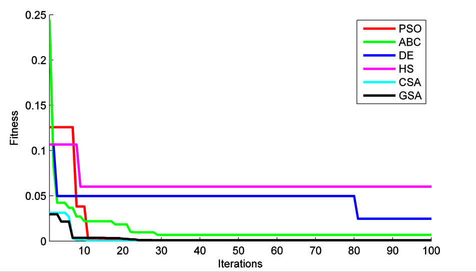
For Chen, the pairs considered are $[a, b]$, $[b, c]$, and $[a, c]$ (Eq. 10). Results are summarized in Table 5.

$$[a, b], \ [b, c], \ [a, c] \tag{10}$$

In Chen, CSA performs particularly well when estimating (a, b), whereas GSA dominates the remaining combinations. Figures 3(a)–(b) depict the convergence patterns. While ABC or CSA may reach low J values quickly in specific pairs, GSA generally attains the most accurate final fitness.



(a)



(b)

Figure 3. Convergence of the objective function in two-dimensional problems for (a) Lorenz and (b) Chen systems.

4.1.3 *Three-dimensional parameter estimation*

The final experiment addresses the full-parameter case, where all three parameters are tuned jointly. This setting captures the complete nonlinearity of both models and pushes the algorithms to explore a comparatively rugged three-dimensional landscape.

Table 2. One-dimensional parameter estimation results for the Lorenz system.

	PSO	ABC	DE	HS	CSA	GSA
σ	10.3415	10.0002	9.9708	10.0082	10.0000	10.0000
J	2.0373e-10	4.5666e-12	2.6443e-07	1.2465e-08	2.6767e-19	***4.2689e-23***
r	28.7378	27.9946	27.9520	28.0135	28.0000	28.0000
J	2.0972e-07	3.2128e-08	2.0873e-06	3.0226e-07	1.3071e-19	***7.7107e-22***
b	3.0000	2.6636	2.6551	2.6705	2.6667	2.6667
J	2.0277e-08	1.6602e-09	4.4504e-08	1.7030e-08	7.0267e-20	***1.3411e-22***

Table 3. One-dimensional parameter estimation results for the Chen system.

	PSO	ABC	DE	HS	CSA	GSA
a	35.3506	35.0053	34.9924	34.9935	35.0000	35.0000
J	3.6998e-09	3.7128e-09	2.0310e-05	4.1511e-08	2.9510e-19	***4.1632e-22***
b	28.3041	28.0000	28.0129	27.9989	28.0000	28.0000
J	2.7571e-06	1.6724e-08	9.9146e-06	1.1411e-08	1.9136e-17	***3.0693e-21***
c	3.1174	2.9990	2.9998	2.9951	3.0000	3.0000
J	1.0765e-09	2.7945e-10	1.9080e-08	6.7298e-10	1.0451e-19	***1.9135e-23***

Table 4. Two-dimensional parameter estimation results for the Lorenz system.

	PSO	ABC	DE	HS	CSA	GSA
σ	10.2293	10.0397	9.9343	10.1521	10.0000	10.0000
r	28.3408	28.0341	28.0136	27.9398	28.0000	28.0000
J	1.4585e-04	9.4979e-05	1.1927e-04	2.4674e-05	7.3093e-12	***2.6085e-21***
σ	10.6089	9.8557	10.0307	9.9680	10.0000	10.0178
b	3.0000	2.7073	2.6575	2.6792	2.6667	2.6609
J	6.6248e-07	1.7635e-05	1.2946e-05	2.0935e-05	1.5621e-12	***1.2385e-19***
b	3.0000	2.4929	2.6851	2.6434	2.6666	2.6667
r	28.3513	28.1535	27.8856	28.0001	28.0001	28.0000
J	0.0011	6.3016e-05	9.5844e-05	1.5472e-04	3.7434e-12	***9.3895e-21***

Table 5. Two-dimensional parameter estimation results for the Chen system.

	PSO	ABC	DE	HS	CSA	GSA
a	35.9623	36.0448	34.8281	36.3564	35.0003	35.2291
b	29.4769	28.5689	27.9031	28.6719	28.9992	28.1116
J	5.6641e-04	6.0294e-04	2.3821e-04	3.0392e-04	***6.0028e-12***	1.2149e-06
b	28.0709	28.0301	28.1287	28.0202	28.0000	27.9983
c	3.8663	3.0461	2.7461	2.9995	3.0000	3.0024
J	4.5480e-04	8.6206e-05	1.1307e-04	1.9767e-04	3.7080e-11	***1.6683e-19***
a	35.3818	34.9870	34.8585	34.8587	35.0000	35.0000
c	3.7622	3.0047	3.3146	3.3168	3.0461	3.0000
J	5.8252e-04	1.7365e-05	8.8813e-04	3.6050e-05	4.1869e-11	***8.6711e-20***

For Lorenz, we optimize $[\sigma, r, b]$ (Eq. 11); results are listed in Table 6.

$$[\sigma, r, b] \tag{11}$$

For Chen, the set $[a, b, c]$ (Eq. 12) is optimized; the results are shown in Table 7.

$$[a, b, c] \tag{12}$$

Across the three-dimensional tests, CSA and GSA stand out. ABC often descends quickly at the beginning, but CSA and GSA usually deliver more accurate final solutions. Figure 4 plots the evolution of J and illustrates these trends for both systems.

4.2 Statistical Analysis

To rigorously validate the performance differences among the six algorithms, we apply a non-parametric test for independent samples: the Wilcoxon rank-sum test (also known as the Mann–Whitney U test) [25, 26]. Each algorithm is executed 35 times to provide enough samples for reliable pairwise comparisons.

Because multiple Wilcoxon tests are carried out across algorithms and settings, the family-wise error rate would otherwise inflate, increasing the chance of Type I errors (i.e., rejecting a true null hypothesis). To control this, we adjust the p-values using the Bonferroni method [27]. The combination of Wilcoxon with Bonferroni-adjusted p-values allows us to determine, for every algorithm pair, whether the observed differences are statistically significant.

The analysis is organized by system (Lorenz and Chen) and, within each system, by dimensionality of

Table 6. Three-dimensional parameter estimation results for the Lorenz system.

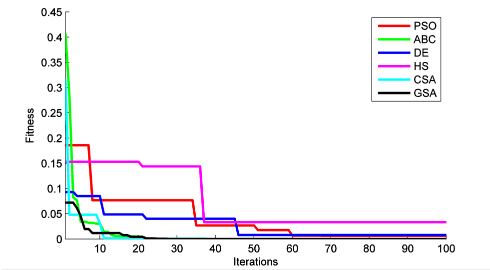
	PSO	ABC	DE	HS	CSA	GSA
σ	11.0000	9.7413	10.0568	10.1445	9.9994	10.0100
r	28.9779	27.9440	27.9183	28.1141	28.0002	28.0002
b	3.0000	2.7191	2.6113	2.6112	2.6670	2.6631
J	4.0458e-04	5.2850e-04	0.0029	1.1198e-04	2.2072e-09	***1.3663e-16***

Table 7. Three-dimensional parameter estimation results for the Chen system.

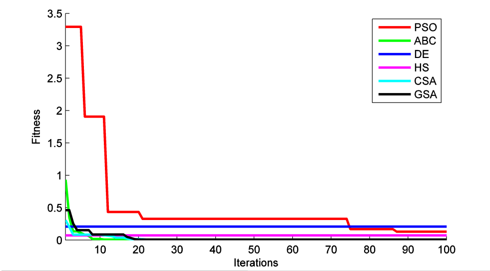
	PSO	ABC	DE	HS	CSA	GSA
a	30.7156	37.2663	31.9672	35.2433	35.0002	34.8441
b	26.1472	29.1574	26.3746	28.2414	28.0001	27.9170
c	4.0171	2.4549	3.3811	3.2948	2.9999	3.0189
J	0.0019	0.0049	0.0055	0.0124	***1.3222e-07***	3.1576e-05

Table 8. Statistical results for the one-dimensional estimation of σ, r, and b in the Lorenz system.

Versus	Lorenz CS (σ)		Lorenz CS (r)		Lorenz CS (b)	
	Wilcoxon (p)	Bonferroni (p)	Wilcoxon (p)	Bonferroni (p)	Wilcoxon (p)	Bonferroni (p)
PSO vs ABC	0.5183	0	0.0039	0	1.3529e-09	1
PSO vs CSA	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
PSO vs HS	0.001	1	0.3534	0	2.3362e-06	1
PSO vs GSA	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
PSO vs DE	2.2549e-10	1	0.2799	0	1.1798e-05	1
ABC vs CSA	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
ABC vs HS	4.5838e-05	1	0.0033	1	1.3349e-07	1
ABC vs GSA	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
ABC vs DE	9.5761e-12	1	3.9372e-05	1	1.2204e-08	1
CSA vs HS	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
CSA vs GSA	6.5455e-13	1	7.1329e-13	1	1.8196e-12	1
CSA vs DE	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
HS vs GSA	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
HS vs DE	4.3992e-07	1	0.8234	0	0.0042	0
GSA vs DE	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1



(a)



(b)

Figure 4. Convergence of the objective function in three-dimensional problems for (a) Lorenz and (b) Chen systems.

the estimation task (one-, two-, and three-parameter settings). The corresponding tables report both the raw Wilcoxon p-values and their Bonferroni-adjusted counterparts. In the column labeled "Bonferroni (p)", a value of 1 indicates that the compared pair of algorithms differs significantly under the chosen significance level.

4.2.1 *Statistical Analysis for the Lorenz system*

We assess statistical significance for all experiments described in Section 4.1, partitioned into: (a) one-dimensional, (b) two-dimensional, and (c) three-dimensional analyses.

a) One-dimensional statistical analysis

Each Lorenz parameter (σ, r, b) is estimated independently by every algorithm. The Wilcoxon results and their Bonferroni corrections are summarized in Table 8.

b) Two-dimensional statistical analysis

We now consider the parameter pairs from Eq. 9. Table 9 compiles the Wilcoxon and Bonferroni-adjusted results for $[\sigma, r]$, $[\sigma, b]$, and $[r, b]$.

c) Three-dimensional statistical analysis

Table 9. Statistical results for the two-dimensional estimation of σ, r, and b in the Lorenz system.

Versus	Lorenz CS (σ, r)		Lorenz CS (σ, b)		Lorenz CS (r, b)	
	Wilcoxon (p)	Bonferroni (p)	Wilcoxon (p)	Bonferroni (p)	Wilcoxon (p)	Bonferroni (p)
PSO vs ABC	0.0071	0	0.2004	0	1.4622e-05	1
PSO vs CSA	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
PSO vs HS	0.0085	0	0.9813	0	8.5166e-06	1
PSO vs GSA	1.1741e-07	1	6.5455e-13	1	6.5455e-13	1
PSO vs DE	0.0108	0	0.0272	0	3.7218e-04	1
ABC vs CSA	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
ABC vs HS	0.3847	0	0.3238	0	0.8786	0
ABC vs GSA	4.8024e-10	1	6.5455e-13	1	6.5455e-13	1
ABC vs DE	0.2695	0	0.4109	0	0.0821	0
CSA vs HS	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
CSA vs GSA	2.5964e-09	1	6.5455e-13	1	6.5455e-13	1
CSA vs DE	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
HS vs GSA	1.6602e-10	1	6.5455e-13	1	6.5455e-13	1
HS vs DE	0.9158	0	0.1155	0	0.0485	0
GSA vs DE	6.0469e-11	1	6.5455e-13	1	6.5455e-13	1

Table 10. Statistical results for the three-dimensional estimation of σ, r, and b in the Lorenz system.

Versus	Lorenz CS (σ, r, b)	
	Wilcoxon (p)	Bonferroni (p)
PSO vs ABC	0.0136	0
PSO vs CSA	6.5455e-13	1
PSO vs HS	5.6036e-05	1
PSO vs GSA	1.4118e-12	1
PSO vs DE	0.0526	0
ABC vs CSA	6.5455e-13	1
ABC vs HS	0.0345	0
ABC vs GSA	2.7700e-12	1
ABC vs DE	0.3068	0
CSA vs HS	6.5455e-13	1
CSA vs GSA	1.2855e-04	1
CSA vs DE	6.5455e-13	1
HS vs GSA	2.3326e-11	1
HS vs DE	6.5834e-04	1
GSA vs DE	6.5455e-13	1

Table 11. Statistical results for the one-dimensional estimation of a, b, and c in the Chen system.

Versus	Chen CS (a)		Chen CS (b)		Chen CS (c)	
	Wilcoxon (p)	Bonferroni (p)	Wilcoxon (p)	Bonferroni (p)	Wilcoxon (p)	Bonferroni (p)
PSO vs ABC	3.7218e-04	1	0.0026	1	0.0316	0
PSO vs CSA	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
PSO vs HS	0.557	0	0.8234	0	0.1519	0
PSO vs GSA	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
PSO vs DE	6.1187e-06	1	0.4043	0	1.1741e-07	1
ABC vs CSA	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
ABC vs HS	0.0512	0	0.0023	1	5.5367e-04	1
ABC vs GSA	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
ABC vs DE	1.1288e-10	1	2.3643e-04	1	8.0537e-09	1
CSA vs HS	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
CSA vs GSA	7.1329e-13	1	6.5455e-13	1	7.7721e-13	1
CSA vs DE	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
HS vs GSA	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
HS vs DE	0.0062	0	0.2747	0	9.11e-07	1
GSA vs DE	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1

Finally, we evaluate the full set $[\sigma, r, b]$. Table 10 reports the Wilcoxon and Bonferroni-adjusted p-values for the three-parameter case.

4.2.2 Statistical Analysis for the Chen system

We repeat the same protocol for the Chen system, again splitting the analysis into: (a) one-dimensional, (b) two-dimensional, and (c) three-dimensional settings, consistent with Section 4.1.

Table 12. Statistical results for the two-dimensional estimation of a, b, and c in the Chen system.

Versus	Chen CS (a, b)		Chen CS (b, c)		Chen CS (a, c)	
	Wilcoxon (p)	Bonferroni (p)	Wilcoxon (p)	Bonferroni (p)	Wilcoxon (p)	Bonferroni (p)
PSO vs ABC	0.0054	0	0.4313	0	0.6724	0
PSO vs CSA	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
PSO vs HS	0.0101	0	0.6135	0	0.2355	0
PSO vs GSA	1.1569e-06	1	1.9797e-12	1	7.7721e-13	1
PSO vs DE	0.0052	0	0.0022	1	0.0355	0
ABC vs CSA	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
ABC vs HS	0.9251	0	0.769	0	0.557	0
ABC vs GSA	3.4569e-09	1	1.9797e-12	1	1.0046e-12	1
ABC vs DE	0.9345	0	0.0801	0	0.2219	0
CSA vs HS	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
CSA vs GSA	6.5455e-13	1	1.9354e-10	1	1.2228e-11	1
CSA vs DE	6.5455e-13	1	6.5455e-13	1	6.5455e-13	1
HS vs GSA	1.3073e-08	1	2.1535e-12	1	1.0046e-12	1
HS vs DE	1	0	0.0355	0	0.5729	0
GSA vs DE	2.0919e-09	1	1.0046e-12	1	6.5455e-13	1

Table 13. Statistical results for the three-dimensional estimation of a, b, and c in the Chen system.

Versus	Chen CS (a, b, c)	
	Wilcoxon (p)	Bonferroni (p)
PSO vs ABC	0.8694	0
PSO vs CSA	6.5455e-13	1
PSO vs HS	0.1025	0
PSO vs GSA	1.2204e-08	1
PSO vs DE	0.4109	0
ABC vs CSA	6.5455e-13	1
ABC vs HS	0.0761	0
ABC vs GSA	1.4002e-08	1
ABC vs DE	0.4109	0
CSA vs HS	6.5455e-13	1
CSA vs GSA	6.5455e-13	1
CSA vs DE	6.5455e-13	1
HS vs GSA	1.9354e-10	1
HS vs DE	0.3783	0
GSA vs DE	8.0537e-09	1

a) One-dimensional statistical analysis

Each parameter (a, b, c) is tested across all algorithms. Table 11 lists the Wilcoxon p-values and their Bonferroni-adjusted versions.

b) Two-dimensional statistical analysis

We next analyze the parameter pairs in Eq. 10. Table 12 presents the Wilcoxon and Bonferroni outcomes for $[a, b]$, $[b, c]$, and $[a, c]$.

c) Three-dimensional statistical analysis

The full-parameter case $[a, b, c]$ is summarized in Table 13, again reporting Wilcoxon and Bonferroni-adjusted p-values for all algorithm pairs.

Overall, Tables 8–13 cover 15 pairwise comparisons per setting and dimension. Using a 0.05 significance level for Wilcoxon and applying the Bonferroni adjustment for multiple testing, we find strong evidence that not all algorithms perform equivalently. In particular, the solutions produced by GSA and CSA are frequently and significantly different from those of the remaining methods, and they also show significant differences between each other across several problem dimensionalities and systems.

5 Conclusions

This work compared six evolutionary computation techniques for parameter estimation in chaotic systems, framing the task as a continuous optimization problem over the Lorenz and Chen models. Alongside widely used baselines—PSO, ABC, HS, and DE—we evaluated two recent approaches, GSA and CSA, under one-, two-, and three-parameter settings.

Across most scenarios, GSA and CSA achieved the most accurate estimates and did so with strong statistical support. Using the Wilcoxon rank-sum test with Bonferroni-adjusted p-values, we observed significant performance differences that consistently favored GSA and CSA in the one- and two-parameter problems, and again in the full three-parameter case. These gains align with their search operators: both methods balance exploration and exploitation effectively, allowing them to negotiate the highly nonconvex landscapes typical of chaotic dynamics.

The classical methods (PSO, ABC, HS, DE) remain competitive in terms of convergence speed or in specific parameter pairs. Yet, their final accuracy is generally inferior to that of GSA and CSA under our protocol. Overall, the evidence indicates that modern gravitational- and cuckoo-inspired strategies offer practical advantages for chaotic system identification.

Practical implications. For practitioners facing parameter estimation in nonlinear or chaotic models, GSA and CSA are strong defaults when accuracy is paramount. When rapid early descent is desired, ABC or HS may be attractive, but our results suggest switching to GSA/CSA for fine-tuning.

Limitations and future work. Our study fixed population size and iteration budget across algorithms; adaptive budgets or tuned hyperparameters could further shift the rankings. Future work should examine robustness under measurement noise, partial-state observations, and larger-dimensional chaotic systems, as well as hybrid strategies that combine the fast early progress of ABC/HS with the high-precision refinement of GSA/CSA.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Zhou, X., Yang, C., & Gui, W. (2014). Nonlinear system identification and control using state transition algorithm. *Applied mathematics and computation, 226*, 169-179. [CrossRef]

[2] Ljung, L. (1999). System identification: theory for the user. Upper Saddle River, N.

[3] Pai, P. F., Nguyen, B. A., & Sundaresan, M. J. (2013). Nonlinearity identification by time-domain-only signal processing. *International Journal of Non-Linear Mechanics, 54*, 85-98. [CrossRef]

[4] Yang, S. K., Chen, C. L., & Yau, H. T. (2002). Control of chaos in Lorenz system. *Chaos, Solitons & Fractals, 13*(4), 767-780. [CrossRef]

[5] Yassen, M. T. (2003). Chaos control of Chen chaotic dynamical system. *Chaos, Solitons and Fractals, 15*(2), 271–283. [CrossRef]

[6] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE. [CrossRef]

[7] Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization* (Technical Report No. TR06). Department of Computer Engineering, Engineering Faculty, Erciyes University.

[8] Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *simulation, 76*(2), 60-68. [CrossRef]

[9] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization, 11*(4), 341-359. [CrossRef]

[10] Yang, X. S., & Deb, S. (2009, December). Cuckoo search via Lévy flights. In *2009 World congress on nature & biologically inspired computing (NaBIC)* (pp. 210-214). IEEE. [CrossRef]

[11] Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. *Information sciences, 179*(13), 2232-2248. [CrossRef]

[12] Cuevas, E., Gálvez, J., Hinojosa, S., Avalos, O., Zaldívar, D., & Pérez-Cisneros, M. (2014). A comparison of evolutionary computation techniques for IIR model identification. *Journal of Applied Mathematics, 2014*(1), 827206. [CrossRef]

[13] Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation, 1*(1), 67-82. [CrossRef]

[14] Ahn, C. W. (2006). *Advances in evolutionary algorithms: theory, design and practice*. Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[15] Chiong, R., & Weise, T. (2012). *Variants of evolutionary algorithms for real-world applications* (Vol. 2). Z. Michalewicz (Ed.). Berlin: Springer. [CrossRef]

[16] Oltean, M. (2007). Evolving evolutionary algorithms with patterns. *Soft Computing, 11*(6), 503-518. [CrossRef]

[17] Lin, Y. L., Chang, W. D., & Hsieh, J. G. (2008). A particle swarm optimization approach to nonlinear rational filter modeling. *Expert Systems with Applications, 34*(2), 1194-1199. [CrossRef]

[18] Pavlyukevich, I. (2007). Lévy flights, non-local search and simulated annealing. *Journal of Computational Physics, 226*(2), 1830–1844. [CrossRef]

[19] Mantegna, R. N. (1994). Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes. *Physical Review E, 49*(5), 4677. [CrossRef]

[20] De Jong, K. (1988). Learning with genetic algorithms: An overview. *Machine Learning, 3*, 121–138. [CrossRef]

[21] Hegazi, A. S., Ahmed, E., & Matouk, A. E. (2011). The effect of fractional order on synchronization of two fractional order chaotic and hyperchaotic systems. *Journal of Fractional Calculus and Applications, 1*(3), 1-15.

[22] Huang, L., Feng, R., & Wang, M. (2004). Synchronization of chaotic systems via nonlinear

control. *Physics Letters A*, *320*(4), 271–275. [CrossRef]

[23] He, Q., Wang, L., & Liu, B. (2007). Parameter estimation for chaotic systems by particle swarm optimization. *Chaos, Solitons and Fractals*, *34*(2), 654–661. [CrossRef]

[24] Patwardhan, A. P., Patidar, R., & George, N. V. (2014). On a cuckoo search optimization approach towards feedback system identification. *Digital Signal Processing*, *32*, 156-163. [CrossRef]

[25] García, S., Molina, D., Lozano, M., & Herrera, F. (2009). A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. *Journal of Heuristics*, *15*(6), 617-644. [CrossRef]

[26] Shilane, D., Martikainen, J., Dudoit, S., & Ovaska, S. J. (2008). A general framework for statistical performance comparison of evolutionary computation algorithms. *Information Sciences*, *178*(14), 2870-2879. [CrossRef]

[27] Bonferroni, C. E. (1935). Il calcolo delle assicurazioni su gruppi di teste. *Studi in Onore del Professor Salvatore Ortu Carboni*, 13–60.

Prof. Jorge Gálvez currently works as a full-time professor assigned to the Department of Innovation Based on Information and Knowledge at the University Center for Exact Sciences and Engineering (CUCEI) of the University of Guadalajara. In 2011, he completed his B.S. degree in Informatics. The M.Sc. degree in Electronics and Computing in 2015. In 2019, he completed his Ph.D. degree in Electronics and Computer Science from Universidad de Guadalajara. Since 2019, he has been a member of the Mexican National Research System (SNII I). His research interests include machine learning, deep learning, evolutionary computation, computer vision, complex systems, and robotics.

Gustavo R. Hernández received a B.S. degree in Mechatronics Engineering from the Centro de Enseñanza Técnica Industrial (CETI) in 2013, and the M.Sc. degree in Electronics and Computer Engineering from the Universidad de Guadalajara in 2019. He is currently a Ph.D. student in Computational Intelligence Sciences at the Universidad de Guadalajara. During his postgraduate research, he specialized in the fields of Optimization, Metaheuristics, Computer Vision, and Intelligent Systems. His current research interests include Predictive Maintenance, Reliability, and Optimization. He is a member of the Automatic Control and Intelligent Systems Research Group.

Prof. Omar Avalos is a Professor in the Department of Electronics at Universidad de Guadalajara. In 2013, he completed the B.S. degree in Communications and Electronics. The M.Sc. and Ph.D. degrees in Electronics and Computer Science from Universidad de Guadalajara in 2015 and 2019, respectively. During his formation, he participated in different projects such as the AutonomusMini in collaboration with the Freie Universität Berlin and the development of a humanoid robot, to mention a few. He is a member of the Mexican National Research System (SNI I). His research interests include various applications of Artificial Intelligence in different engineering fields, such as Biomedical applications, energy consumption optimization, system identification, and image processing.

Angel Casas-Ordaz received a B.S. degree in Electronic Engineering from the Instituto Tecnológico de Ciudad Juárez in 2018, and later the M.Sc. degree in Electronics and Computer Engineering from the Universidad de Guadalajara at the Centro Universitario de Ciencias Exactas e Ingenierías (CUCEI) in 2022; He is currently a Ph.D. student in Electronics and Computer Science at the same campus. During his postgraduate research, he specialized in the field of Evolutionary Computation and Image Segmentation. His current research interests include Evolutionary Computation and Artificial Intelligence, specializing in Metaheuristic Algorithms in the Automatic Control and Intelligent Systems Research Group.

Prof. Marco Perez-Cisneros received the B.S. in Communications and Electronics Engineering from the University of Guadalajara, México, the M.Sc. from ITESO University, México, and the Ph.D. degree from the University of Manchester, UK. He serves as Professor in the Electro-Photonics Department and as Rector of the Exact Sciences and Engineering Centre of the University of Guadalajara. He is a member of the National Research System in México, and since 2018, he has been a member of the Mexican National Science Academy. He is a Senior Member of the IEEE and a regular member of the IET UK. He also serves as the Associated Editor of IEEE Letters.

Prof. Diego Oliva received a B.S. degree in Electronics and Computer Engineering from the Industrial Technical Education Center (CETI) of Guadalajara, México, in 2007 and an M.Sc. degree in Electronic Engineering and Computer Sciences from the University of Guadalajara, México, in 2010. He obtained a Ph. D. in Informatics in 2015 from the Universidad Complutense de Madrid. Currently, he is an Associate Professor at the University of Guadalajara in México. He is a member of the Mexican National Research System (SNII), a Senior member of the IEEE, and a member of the Mexican Academy of Computer Sciences (AMEXCOMP). His research interests include evolutionary and swarm algorithms, hybridization of evolutionary and swarm algorithms, and computational intelligence.