Library Progress International
Vol.44 No. 3, Jul-Dec 2024: P. 11297-11308

Print version ISSN 0970 1052
Online version ISSN 2320 317X

Original Article

Available online at www.bpasjournals.com

Chaotic Particle Swarm Optimization for Parameter Estimation in Nonlinear Dynamic Systems

Bhanu Sekhar Obbu[1*], Zamrooda Jabeen[2]

[1,2]Department of Mathematics, National Institute of Technology, Srinagar, Jammu and Kashmir, India
[1]o.bhanusekhar@gmail.com, [2]zjabeen19@nitsri.net

How to cite this article: Bhanu Sekhar Obbu, Zamrooda Jabeen (2024) Chaotic Particle Swarm Optimization for Parameter Estimation in Nonlinear Dynamic Systems. *Library Progress International,* 44(3), 11297-11308.

ABSTRACT

The estimation of parameters for chaotic systems is an important area of study in nonlinear dynamics, and it has recently attracted attention from a number of other fields. A multi-dimensional optimization problem is the simplest way to characterize the current scenario. Chaotic particle swarm optimization (CPSO) is a novel technique in evolutionary computation that has attracted much attention and is extensively employed due to its simplicity, ease of implementation, and rapid convergence. To our knowledge, no published study exists on applying CPSO for parameter estimation in chaotic systems. This research employs a CPSO methodology to precisely ascertain the Lorenz system's parameters. The effectiveness and durability of CPSO are showcased through numerical simulations and comparisons. In addition, the study also investigates the impact of population size on optimization performance.

Keywords :Evolutionary algorithms, particle swarm optimization, linear dynamic systems, chaotic theory

Introduction

Nonlinear dynamic systems are characterized by a non-proportional interaction between variables, resulting in a phenomenon where minor changes in input can cause significantly larger changes in output [1]. These systems are known for their complex and unpredictable behavior, often displaying chaos, bifurcations, and strange attractors. Here's a deeper exploration of the fundamental concepts, characteristics, and applications of nonlinear dynamic systems:- Nonlinearity: The system's equations are nonlinear, meaning they cannot be represented as a simple sum of the effects of each input. Initial condition sensitivity: Minute disparities in the starting state can result in significantly divergent consequences, commonly known as the "butterfly effect." Bifurcations: As a parameter within the system changes, the system can experience sudden, qualitative behavioral shifts, known as bifurcations. Chaos: Chaotic systems exhibit behavior that appears random and unpredictable, even though they are deterministic. Strange Attractors: In the phase space of a chaotic system, trajectories can converge to a set with a fractal structure, known as a strange attractor. Nonlinear dynamics are essential for designing and controlling various engineering systems, including mechanical, aerospace, and electrical systems. Nonlinear dynamics provide insights into complex biological systems, such as neural networks, heart rhythms, and ecosystem dynamics. Economic systems often display nonlinear behaviors, like boom-and-bust cycles, which can be modeled using nonlinear dynamic systems. Nonlinear dynamics are utilized to model physiological processes and disease progression, including the dynamics of cardiac arrhythmias and the spread of infectious diseases. Nonlinear models can describe social phenomena, such as the spread of information or diseases in populations, social networks, and crowd behavior.

Algorithms that mimic the process of natural selection are known as evolutionary algorithms (EAs). Unlike more conventional optimization techniques, they excel at resolving complicated issues involving nonlinear dynamics [2]. Nonlinear dynamics involves systems where changes in the system's state are not proportional to the initial conditions, resulting in behaviors such as chaos and bifurcations. Here are some key applications of evolutionary algorithms in this field: Parameter Estimation: EAs can be employed to estimate parameters in nonlinear dynamic models, a challenging task due to these systems' sensitivity to initial conditions and parameter values. Control of Chaos: EAs can design controllers to stabilize chaotic systems by finding control parameters that suppress or manage chaotic behavior in nonlinear dynamic systems. Time Series Prediction: EAs evolve models to predict future states of a nonlinear dynamic system from historical data, which is particularly useful in fields like meteorology, finance, and biology. System Identification: In nonlinear dynamics, accurately identifying the underlying system from observed data is crucial. EAs can evolve mathematical models that describe the system's behavior. Optimization of Nonlinear Systems: EAs optimize the performance of systems described by nonlinear dynamics, including finding optimal operational conditions, configurations, or designs that maximize or minimize a specific objective. Modeling and Simulation: EAs generate models that simulate the behavior of complex nonlinear systems, aiding in understanding system behavior, testing hypotheses, and conducting impractical experiments in the real world [3].

Evolutionary algorithms are used to improve control algorithms and the physical design of robots, which frequently exhibit nonlinear dynamics due to the intricate interactions between their components and the environment. In financial markets, which are highly nonlinear and chaotic, EAs are employed to model and predict market behavior, optimize trading strategies, and manage risks. These are applied to model and understand complex biological systems, including population dynamics, the spread of diseases, and ecological interactions. In engineering fields, EAs are used to design and control systems such as aerospace vehicles, chemical reactors, and power systems, where nonlinear dynamics are prevalent.

Nonlinear systems are characterized by chaotic behavior, which is characterized by both boundedness and instability. Its sensitivity to initial conditions characterizes it and includes an endless number of unstable periodic motions. Chaotic systems have been extensively researched and analyzed in numerous disciplines to understand and manage their control and synchronization [4]. Several proposed methodologies depend on the presupposition that the parameters of chaotic systems are pre-established. Nevertheless, establishing these characteristics can be challenging due to the complexities of chaotic systems.

Consequently, estimating parameters for chaotic systems has gained significant attention in the last ten years [5]. Several studies have concentrated on utilizing synchronization-based techniques for parameter estimation. The literature [6] presents a new adaptive differential evolution algorithm that can reliably, quickly, and precisely extract photovoltaic parameters. For each individual, the suggested method uses their fitness value to determine an adjusted crossover rate value; this increases the likelihood that beneficial characteristics will be passed down to subsequent generations. Also, to make things even between exploration and exploitation, we employ a dynamic population reduction method to speed up convergence. The PV parameters identification in [7] utilized the simulated annealing approach. The genetic algorithm [8] was utilized to identify the parameters of PV solar cells and modules. A common application of adaptive control systems in the field is the estimation of parameters for non-linear dynamic systems [9].

In the literature [10], these two approaches and their variations have been utilized to address fifteen distinct parameter estimate issues of varied levels of complexity. The estimation results are examined using nonparametric statistical techniques to ascertain whether any algorithm has statistical superiority over others across the evaluated problems. As determined by the accuracy of parameter estimates, the results demonstrate significant variations among the algorithms, with more recent and advanced algorithms surpassing their conventional counterparts. Regarding the estimation of the covariance matrices of the process and measurement noise in the Kalman filter, an evolutionary technique is presented in the literature [11].

The performance of the sub-optimal filter is going to be improved as the goal of this experiment. An implementation of a fitness function is carried out in order to accomplish multi-dimensional optimization. During the verification process, the results for both the linear and nonlinear dynamics of the quadrotor attitude are examined. Optimal Kalman filter, covariance-matrix adaptation evolution technique, and simulated annealing are some of the optimization strategies that are compared to the proposed method. The proposed method is superior to the alternatives when it comes to fine-tuning the state estimator, as demonstrated by the Monte Carlo analysis. Ho Pham Huy Anh and colleagues [12] introduced an adaptive differential evolution (ADE) approach to identify the parameters of the uncertain pendulous. The utilization of diverse benchmark test functions validates the efficacy of the ADE algorithm. Tao Luo et al. [13] introduce a novel LCPSO-CRP protocol to enhance cluster routing in IWSNs. The researchers propose a cluster routing model that is designed to be energy-efficient.

Additionally, they introduce a new objective function that is unique and innovative. A combination of PSO's exploration and exploitation capabilities with Elite Learning, better Parameter updating, and an exponential mutation operator (PSO-ELPM) is recommended in the literature [14] for a more balanced approach. This algorithm guides optimization by looking at the most successful particles in the population, called elites. The elitism method facilitates the acquisition of significant insights into the solution area. There is a new approach to estimating permanent magnet synchronous motor (PMSM) parameters that has been suggested in the literature [15]. For parameter estimation, this approach employs dynamic self-optimization (DSCPSO), which stands for chaotic particle swarm optimization with dynamic self-optimization.

It also achieves real-time compensation of VSI nonlinearity by simultaneously estimating its nonlinearity. Roberto F. R. et al. [16] The Extended Kalman Filter is suggested as a substitute method for precisely determining the states of the transmission line terminals and the parameter vectorized matrix. The method's performance, meanwhile, is quite sensitive to the starting points. These starting points are usually retrieved by hand, which can be a tedious and labor-intensive process. As a consequence, optimizing with Particle Swarm Optimization speeds up the EKF's convergence, which means less time spent adjusting hyper-parameters and better expected results.

This research uses a CPSO approach to handle the multi-dimensional optimization problem of parameter estimation for chaotic systems. CPSO's effectiveness, efficiency, and resilience are demonstrated by numerical simulations utilizing the Lorenz system and comparisons to outcomes achieved using PSO and genetic algorithms (GAs).

The remaining part of this work is structured as follows. Section 2 presents parameter estimation as a problem when optimizing many dimensions. Section 3 offers a concise overview and execution of CPSO. Section 4 contains numerical simulations and comparisons. Section 5 ultimately ends by providing a concise summary of the findings.

2. Problem formulation

Here is a chaotic system [17] with n dimensions:

$$\dot{X} = F(X, X_0, \theta_0)\dot{X} = F(X, X_0, \theta_0) \tag{1}$$

where $X = (x_1, x_2, \ldots, x_n)^{\mathrm{T}} \in R^n X = (x_1, x_2, \ldots, x_n)^{\mathrm{T}} \in R^n$ denotes the state vector, and $\theta_0 = (\theta_{10}, \theta_{20}, \ldots, \theta_{d0})^{\mathrm{T}}\theta_0 = (\theta_{10}, \theta_{20}, \ldots, \theta_{d0})^{\mathrm{T}}$ is a set of original parameters and $X_0 X_0$ denotes the initial state.

When estimating the parameters, it is important to make the assumption that the structure of the system is already known. Consequently, the estimated system may be stated in the following manner:

$$\dot{Y} = F(Y, X_0, \theta)\dot{Y} = F(Y, X_0, \theta) \tag{2}$$

where $Y = (y_1, y_2, \ldots, y_n)^{\mathrm{T}} \in R^n Y = (y_1, y_2, \ldots, y_n)^{\mathrm{T}} \in R^n$ denotes the state vector, and $\theta = (\theta_1, \theta_2, \ldots, \theta_d)^{\mathrm{T}}\theta = (\theta_1, \theta_2, \ldots, \theta_d)^{\mathrm{T}}$ is a set of estimated parameters.

In light of this, the problem of parameter estimation can be transformed into the optimization problem that is shown below:

$$\min J = \frac{1}{M}\sum_{k=1}^{M} \|X_k - Y_k\|^2 \text{ by searching suitable } \theta^*$$

$$\min J = \frac{1}{M}\sum_{k=1}^{M} \|X_k - Y_k\|^2 \text{ by searching suitable } \theta^*$$

(3)

Here J represents the error cost function

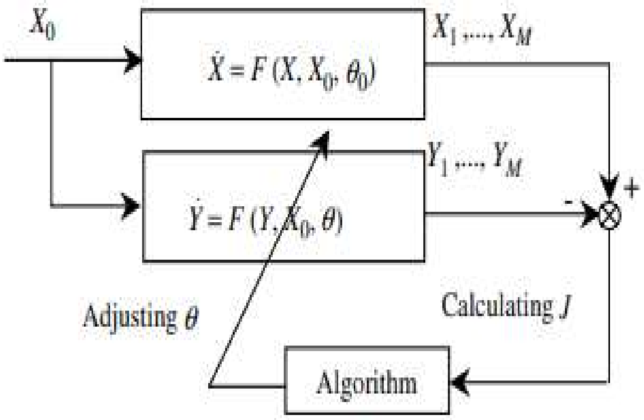


Figure1. The concept of parameter estimation for chaotic systems

The data length that is used for parameter estimate is denoted by the letter 'M'. The state vectors of the original system and the estimated system at time k are denoted by the symbols X_k and Y_k, respectively, where k can range from 1 to as many as M. The process of estimating parameters for chaotic systems is a difficult endeavour that requires continual optimization and involves multiple dimensions. The decision vector, denoted as h, is optimized to minimize the objective function J. The concept of parameter estimates for chaotic systems within the optimization framework can be elucidated by referring to Figure 1.

Obtaining the parameters of chaotic systems is problematic due to their unstable dynamic behavior. Moreover, the issue frequently entails numerous variables and multiple local optima within the J terrain. Consequently, conventional optimization techniques are susceptible to becoming stuck in local optima, which hinders the attainment of the globally optimal parameters.

3. **Chaotic particle swarm optimization**

To improve the swarm's capabilities in exploration and exploitation, the CPSO algorithm, a unique form of the standard PSO algorithm, incorporates concepts from chaos theory. The addition of controlled chaotic behavior into the equations used to update the velocity of particles in CPSO results in the particles exhibiting behaviors that are more diverse and unexpected within the search space. This feature facilitates improved global convergence and the ability to escape local optima. The infusion of chaos into CPSO balances exploration and exploitation, enabling the algorithm to navigate intricate and multi-modal optimization landscapes adeptly. CPSO has exhibited promising outcomes in addressing demanding optimization problems, particularly those characterized by high dimensionality or nonlinearity. The original PSO velocity and position updating equations of particles are given as

$$\boldsymbol{V}_i^{k+1} = w\boldsymbol{V}_i^k + c_1 * \text{rand}_1 * \left(\boldsymbol{P}_{besti} - \boldsymbol{P}_i^k\right) + c_2 * \text{rand}_2 * \left(\boldsymbol{G}_{best} - \boldsymbol{P}_i^k\right)$$

$$\boldsymbol{V}_i^{k+1} = w\boldsymbol{V}_i^k + c_1 * \text{rand}_1 * \left(\boldsymbol{P}_{besti} - \boldsymbol{P}_i^k\right) + c_2 * \text{rand}_2 * \left(\boldsymbol{G}_{best} - \boldsymbol{P}_i^k\right)$$

(4)

$$\boldsymbol{P}_i^{k+1} = \boldsymbol{P}_i^k + \boldsymbol{V}_i^{k+1}\boldsymbol{P}_i^{k+1} = \boldsymbol{P}_i^k + \boldsymbol{V}_i^{k+1} \quad (5)$$

The term "chaos" characterizes the seemingly unpredictable behaviour of a nonlinear, bounded, and non-converging dynamical system with only a few independent variables. Chaotic sequences can be efficiently stored, demonstrating easily and rapidly generated patterns. Among the various maps illustrating chaotic behaviour, logistic maps find widespread use. The following equations can describe the chaotic sequences and random variables produced by employing logistic maps.

$$rand_1\ (k) = \lambda * rand_1\ (k-1) * [1 - rand_1\ (k-1)]$$

$$rand_1\ (k) = \lambda * rand_1\ (k-1) * [1 - rand_1\ (k-1)] \quad (6)$$

$$rand_2\ (k) = \lambda * rand_2\ (k-1) * [1 - rand_2\ (k-1)]$$

$$rand_2\ (k) = \lambda * rand_2\ (k-1) * [1 - rand_2\ (k-1)] \quad (7)$$

When considering the logistic map, the chaotic sequence is determined by the equation:

$$c_r(k) = \lambda * c_r(k-1) * [1 - c_r(k-1)]c_r(k) = \lambda * c_r(k-1) * [1 - c_r(k-1)] \quad (8)$$

$$\text{rand}_1\ (0), \text{rand}_2\ (0) \text{ and } c_r(0) \notin \{0, 0.25, 0.5, 0.75, 1\}$$

When λ = 4, the logistic map displays argotic behaviour within the interval (0, 1). However, with a given value of k, the distribution of the logistic map deviates from uniformity. Specifically, values within the intervals [0, 0.1] and [0.9, 1] occur more frequently than across the rest of the range [0, 1]. In the context of CPSO, the velocity equation undergoes modification as follows:

$$\boldsymbol{V}_i^{k+1} = w * \boldsymbol{V}_i^k + C_1 * C_r * \left(\boldsymbol{P}_{\text{best}} - \boldsymbol{P}_i^k\right) + C_2 * (1 - C_r) * \left(\boldsymbol{G}_{\text{best}} - \boldsymbol{P}_i^k\right)$$

$$\boldsymbol{V}_i^{k+1} = w * \boldsymbol{V}_i^k + C_1 * C_r * \left(\boldsymbol{P}_{\text{best}} - \boldsymbol{P}_i^k\right) + C_2 * (1 - C_r) * \left(\boldsymbol{G}_{\text{best}} - \boldsymbol{P}_i^k\right) \quad (9)$$

Where

rand1, rand2	Random numbers between 0 and 1
w	Inertia weight
w_{max}	The initial inertia weight value is equal to 0.9.
w_{min}	The inertia weight's final value is 0.4.
$iter_{max}$	Maximum amount of allowable iterations
C_r	Deterministic displaying chaotic dynamics
λλ	The driving parameter, which ranges from 0 to 4, governs the behavior of the chaotic sequence.
C_r, x_i^k	i^{th} chaotic variable for k^{th} iteration, which has been distributed in range [0, 1]

Optimization algorithms incorporating chaos produce diverse outcomes owing to their extreme sensitivity to initial conditions. Chaotic optimization algorithms demonstrate proficiency in locating global optima due to their distinctive motion patterns. Their capacity to escape local optima enhances global optimization performance, effectively addressing the original PSO algorithm's tendency to become trapped in local extremes and exhibit slow convergence in later stages. Figure 2 depicts the flow chart of the CPSO algorithm.

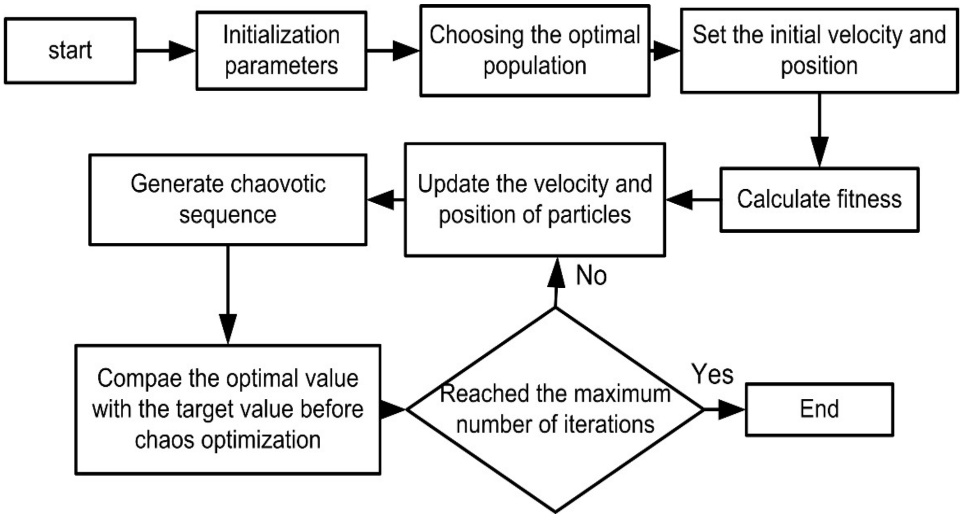


Figure 2. Flow chart of CPSO

4. Simulation and Comparisons

The Lorenz system was discovered by E. N. Lorenz in 1963. The system is a dynamic set of nonlinear ordinary differential equations that can be represented in the following way [15].

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1) \\ \dot{x}_2 = bx_1 - x_1x_3 - x_2 \\ \dot{x}_3 = x_1x_2 - cx_3 \end{cases} \begin{cases} \dot{x}_1 = a(x_2 - x_1) \\ \dot{x}_2 = bx_1 - x_1x_3 - x_2 \\ \dot{x}_3 = x_1x_2 - cx_3 \end{cases} \tag{10}$$

Where a=10, b=28, c=8/3 are the actual parameters.

Within the context of our simulation, the initial Lorenz system can be said to progress independently from a beginning position that is chosen at random. As shown in Figure 1, after a time of transient behavior, a state vector is selected to serve as the initial state X_0 for the purpose of parameter estimation. After that, J is computed by taking M states (M = 300) from both the actual and estimated systems in a sequential manner. CPSO has a termination condition that set the maximum generation number at 100. This number serves as the maximum generation number. It has been estimated that the population size is twenty, forty, and one hundred and twenty when there are one, two, and three unknown factors, respectively. There is a representation of the information in Figure 1. A value of 2.0 is assigned to both the coefficients $c_1$and c_2, while the inertia weight w decreases in a linear fashion from 0.9 to 0.4.

The search parameters are as follows: $20 \leqslant b \leqslant 30$, $2 \leqslant c \leqslant 3$, and $9 \leqslant a \leqslant 11$. Furthermore, a comparison is conducted using a binary-coded genetic algorithm (GA) from [14], with parameter settings that are the same as those mentioned in the associated literature. In order to provide a fair comparison, the PSO and GA algorithms use the same amount of computational power. In particular, the population size, searching range, and maximum generation of the GA are all the same as those of the PSO.

Initially, we focus on estimating a single parameter from a set of three potential parameters, namely a, b, and c, in a one-dimensional parameter estimation scenario. Table 1 presents the statistical outcomes of implementing CPSO, PSO, and GA in three different situations. Each method was separately executed 20 times for each scenario. Table 1 unequivocally illustrates that CPSO, PSO, and GA produce estimated values that closely correspond to the actual values, confirming their robust performance. However, the CPSO method regularly achieves better outcomes than the PSO and GA algorithm in terms of average and worst-case scenarios.

Additionally, parameter estimation in a two-dimensional setting is taken into consideration. More specifically, estimates are needed for the three parameters (a, b, and c) that are currently unknown. The statistical findings obtained from 20 independent runs of the Genetic Algorithm (GA), PSO, and CPSO for three different scenarios are presented in Tables 2-4.

Tables 2–4 indicate that the CPSO consistently achieves better results than the PSO and GA regarding the highest, average, and lowest outputs. In Tables 2 and 3, the CPSO outperforms the PSO and GA, achieving superior outcomes. Next, we will examine the procedure of estimating the parameters in the Lorenz system, which entails calculating the values of all the unidentified parameters in a three-dimensional space. Table 5 displays the statistical results obtained from 20 distinct executions of both the Particle Swarm Optimization (PSO) and the Genetic Algorithm (GA).

Table1. Statistical outcomes of several techniques for estimating parameters in one-dimensional data.

	Average result			Best result			Worst result		
	CPSO	PSO	GA	CPSO	PSO	GA	CPSO	PSO	GA
a	0.10	0.1012	0.1104	0.1000	0.1000	0.10000	0.20025	0.19036	0.500076
J	0.001	0.100000	0.000039	0.001	0.1000	0.00130	0.3000	0.5000	0.700726
b	0.28	0.282	0.282774	0.2800	0.2820	0.2821	0.47998	0.77990	0.28000
J	0.15	0.0076	0.9141	0.00000	0.1500	0.9042	0.9085	0.900152	0.45.557
c	0.2.65	0.266	0.265	2.65321	0.267	0.26667	0.78562	0.2866667	0.2924999
J	0.001	0.030	0.1087	0.000000	0.001	0.0014	0.900002	0.700015	0.307.343

Table 2. Statistical outcomes of several techniques for estimating parameters in a two-dimensional context.

	Average result			Best result			Worst result		
	CPSO	PSO	GA	CPSO	PSO	GA	CPSO	PSO	GA
a	0.101452	0.103777	0.9863719	0.09952	0.1778	0.030342	0.200569	0.4017997	0.91974
b	0.28230	0.289400	0.280081	0.280001	0.280081	0.277013	27.23654	0.2792305	0.28986
J	0.015326	0.016936	0.373294	0.00112	0.00191	0.231930	0.065530	0.081147	0.15412

Table 3. Statistical outcomes of several techniques for estimating parameters in a two-dimensional context

	Average result			Best result			Worst result		
	CPSO	PSO	GA	CPSO	PSO	GA	CPSO	PSO	GA
a	0.192301	0.29980	0.951140	0.19700	0.19801	0.101275	0.985390	0.992503	0.929427
c	0.265532	0.26653	0.2654561	2.69667	2.66667	0.2667048	0.26656	0.66448	0.624927
J	0.007156	0.7440	0.4553793	0.47000	0.57000	0.206471	0.56125	0.59740	0.386.996

Table 4. Statistical outcomes of several techniques for estimating parameters in a two-dimensional context.

	Average result			Best result			Worst result		
	CPSO	PSO	GA	CPSO	PSO	GA	CPSO	PSO	GA
b	0.028024	0.2004684	0.161623	0.099741	0.299417	0.2014393	0.2202452	0.34171	0.953003
c	0.263210	0.666998	0.616623	0.25432	2.666627	0.267658	0.6684120	0.669051	0.499762
J	0.573261	0.597665	0.76960	0.000781	0.00793	0.48645	0.8423601	0.909609	0.4347

Table 5. Statistical outcomes of several techniques for estimating parameters in three-dimensional space.

	Average result			Best result			Worst result		
	CPSO	PSO	GA	CPSO	PSO	GA	CPSO	PSO	GA
a	0.012321	0.118417	0.139783	0.001321	0.195332	0.067167	0.536214	0.608212	0.929003
b	0.00312	0.293390	0.742735	0.00234	0.27146	0.922058	0.732560	0.704424	0.927605
c	0.00665	0.666281	0.648585	0.00675	0.667013	0.663426	0.663210	0.657231	07562049
J	0.092301	0.18278	0.762894	0.047235	0.048645	0.310715	0.29650	0.406026	0.480057

Table 5 demonstrates that the results achieved by CPSO are superior to those produced by PSO and GA, both in terms of the best, average, and worst outcomes. The average outcome achieved by CPSO surpasses even the optimal outcomes achieved by PSO and GA. Furthermore, the estimated parameters derived using CPSO exhibit a high degree of proximity to the genuine values of the original parameters. Therefore, it can be inferred that CPSO is superior and more resilient than GA and PSO when estimating parameters for chaotic systems.

A case study is carried out with the estimate of two-dimensional parameters (a and c) as an example to investigate the search efficiency of CPSO. The objective function J's usual progression is depicted in Figure 3 with the same regulating factors as previously mentioned. The typical convergence of parameters a and c is shown in Figure 4.Figure 3 shows how the value of J rapidly drops to zero, indicating that CPSO can reach the global optimum rapidly. Furthermore, Figure 4 shows how quickly parameters a and c converge to their true values, underscoring the remarkable effectiveness of CPSO in achieving global optimization.

To assess the influence of swarm size on the efficiency of the Chaotic Particle Swarm Optimization (CPSO) method, tests are performed using the described two-dimensional parameter estimation problem, while maintaining other regulating elements in CPSO unchanged. Figures 5 and 6 depict the impact of population size on the average value of J, based on 20 separate runs and the total number of assessments. According to Figure 5, if the population size is too small, the results will be unsatisfactory since there will not be enough solutions for space exploration. As the population increases, the outcomes improve but require more fitness tests, as seen in Figure 6. However, there is a point at which the outcomes will not be significantly influenced. It is advisable to select a population size ranging from 40 to 60, considering both the search quality and the computing work. To estimate additional parameters, it is advisable to use a larger population size.

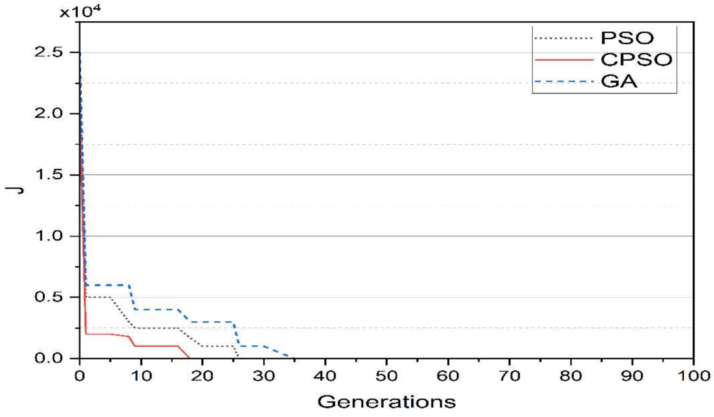
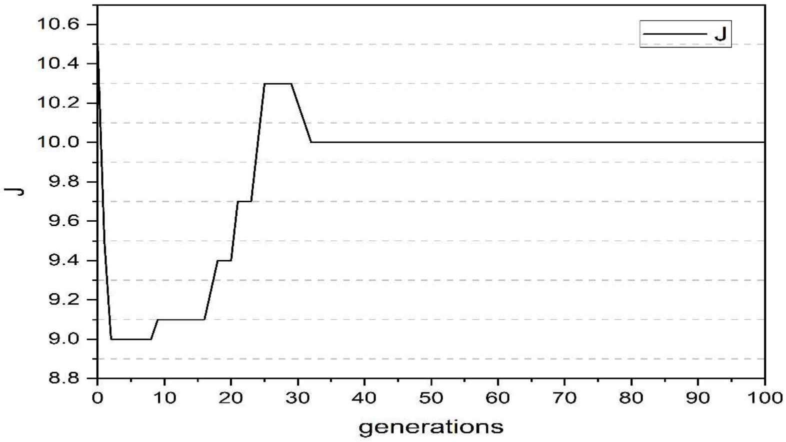
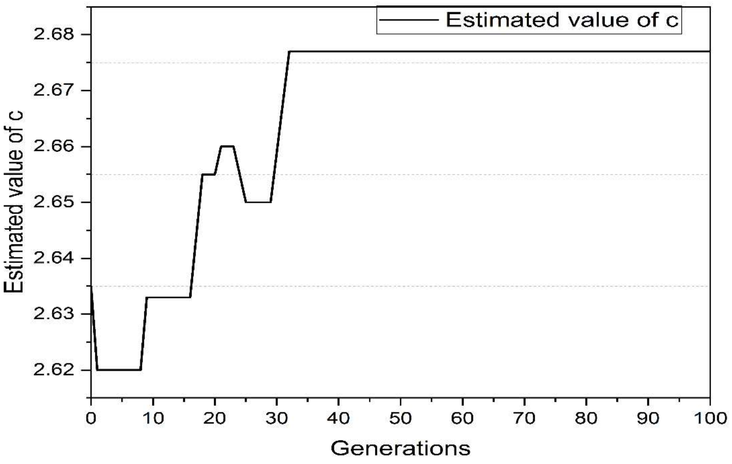


Figure 3. A typical evolution of the objective function value J.



(a)



(b)

Figure 4. A typical search for parameters a and c.

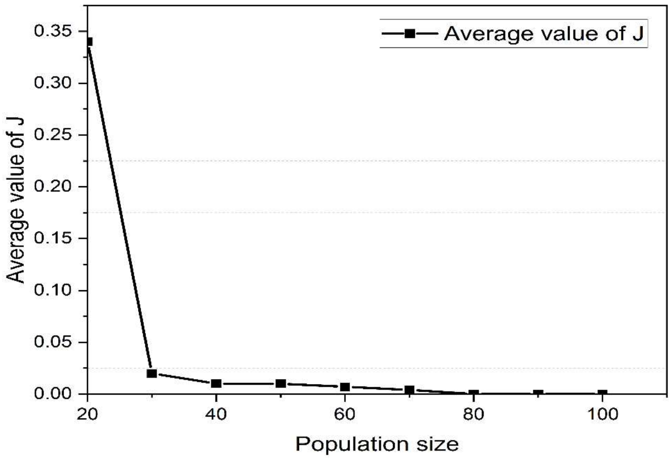


Figure 5. The average objective value of J calculated using CPSO for various population sizes.

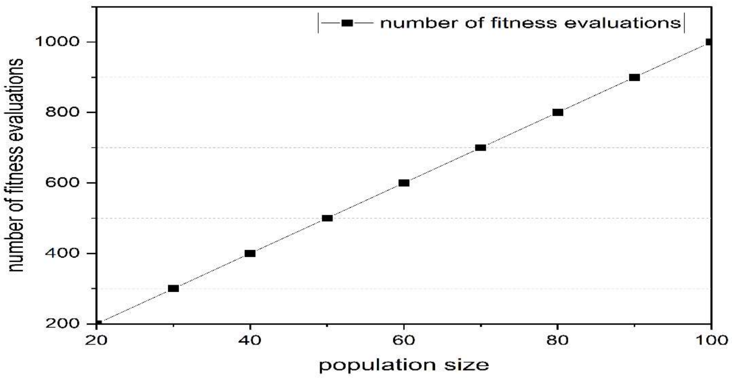


Figure 6. Total number of evaluations in CPSO for different population sizes.

Conclusion

Considering the optimization perspective, this study presented the parameter estimation for chaotic systems as a multi-dimensional optimization problem. The problem was addressed using a unique evolutionary algorithm called CPSO. The usefulness, efficiency, and robustness of CPSO were proved by numerical simulation and comparisons using the Lorenz system. This research represents the initial use of chaotic Particle Swarm Optimization (CPSO) for parameter estimation in chaotic systems, as far as our understanding extends. Future research involves the application of Chaotic Particle Swarm Optimization (CPSO) to additional chaotic systems, as well as the development of more efficient and adaptable CPSO-based methods.

References:

[1] Fuchs, A., 2014. *Nonlinear dynamics in complex systems*. Berlin: Springer.

[2] Liao, Zuowen, Wenyin Gong, Xuesong Yan, Ling Wang, and Chengyu Hu. "Solving nonlinear equations system with dynamic repulsion-based evolutionary algorithms." *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50, no. 4 (2018): 1590-1601.

[3] Zelinka, Ivan, and Roman Senkerik. "Chaotic attractors of discrete dynamical systems used in the core of evolutionary algorithms: state of art and perspectives." *Journal of Difference Equations and Applications* 29, no. 9-12 (2023): 1202-1227.

[4] Lu Z, Shieh LS, Chen GR. On robust control of uncertain chaotic systems: a sliding-mode synthesis via chaotic optimization. Chaos, Solitons & Fractals 2003;18(4):819–27.

[5] Fostin HB, Woafo P. Adaptive synchronization of a modified and uncertain chaotic van der Pol–Duffing oscillator based on parameter identification. Chaos, Solitons & Fractals 2005;24:1363–71.

[6] Li, Shuijia, Qiong Gu, Wenyin Gong, and Bin Ning. "An enhanced adaptive differential evolution algorithm for parameter extraction of photovoltaic models." *Energy Conversion and Management* 205 (2020): 112443.

[7] K.M. El-Naggar, M.R. Alrashidi, M.F. Alhajri, A.K. Al-Othman Simulated annealing algorithm for photovoltaic parameters identification Sol. Energy, 86 (1) (2012), pp. 266-274

[8] M. Zagrouba, A. Sellami, M. Bouacha, M. KsouriIdentification of PV solar cells and modules parameters using the genetic algorithms: application to maximum power extraction

[9] Huberman BA, Lumer E. Dynamics of adaptive systems. IEEE Trans Circ Syst 1990;37:547–50.

[10] Banerjee, Amit, and Issam Abu-Mahfouz. "A comparative analysis of particle swarm optimization and differential evolution algorithms for parameter estimation in nonlinear dynamic systems." Chaos, Solitons & Fractals 58 (2014): 65-83.

[11] Sivashankar, M., Sk Abdul Rahman, C. Arvind Kumar, and G. Manohar. "Application of Artificial Intelligence Method for Predicting of Compressive Strength and Materials Required for Self-Compacting

Concrete." In International Conference on Intelligent Manufacturing and Energy Sustainability, pp. 69-80. Singapore: Springer Nature Singapore, 2023.

[12] Anh, Ho Pham Huy, Nguyen Ngoc Son, Cao Van Kien, and Vinh Ho-Huu. "Parameter identification using adaptive differential evolution algorithm applied to robust control of uncertain nonlinear systems." Applied Soft Computing 71 (2018): 672-684.

[13] Luo, Tao, Jianpeng Xie, Baitao Zhang, Yao Zhang, Chaoqun Li, and Jie Zhou. "An improved levy chaotic particle swarm optimization algorithm for energy-efficient cluster routing scheme in industrial wireless sensor networks." Expert Systems with Applications (2023): 122780.

[14] Moazen, Hadi, Sajjad Molaei, Leili Farzinvash, and Masoud Sabaei. "PSO-ELPM: PSO with elite learning, enhanced parameter updating, and exponential mutation operator." Information Sciences 628 (2023): 70-91.

[15] Feng, Wan, Wenjuan Zhang, and Shoudao Huang. "A novel parameter estimation method for PMSM by using chaotic particle swarm optimization with dynamic self-optimization." IEEE Transactions on Vehicular Technology 72, no. 7 (2023): 8424-8432.

[16] Pereira, Ronaldo FR, Felipe P. Albuquerque, Luisa Helena B. Liboni, Eduardo C. Marques Costa, and José Humberto A. Monteiro. "Estimation of the electrical parameters of overhead transmission lines using Kalman Filtering with particle swarm optimization." IET Generation, Transmission & Distribution 17, no. 1 (2023): 27-38.

[17] R. Gao, A novel track control for lorenz system with single state feedback, Chaos Solitons Fractals 122 (2019) 236–244.

[18] Dai D, Ma XK, Li FC, You Y. An approach of parameter estimation for a chaotic system based on genetic algorithm. Acta Phys Sinica 2002;11:2459-62 [in Chinese].