symmetry

MDPI

Article

A New Pelican Optimization Algorithm for the Parameter Identification of Memristive Chaotic System

Qi Xiong [1,2,*], Jincheng She [1] and Jinkun Xiong [3]

[1] International College, Hunan University of Arts and Sciences, Changde 415000, China
[2] School of Automation Science and Engineering, Faculty of Electronic and Information Engineering, Xi'an Jiao Tong University, Xi'an 710049, China
[3] College of Geography and Environmental Science, Northwest Normal University, Lanzhou 730070, China
* Correspondence: xiongqi@huas.edu.cn

Abstract: A memristor is a kind of nonlinear electronic component. Parameter identification for memristive chaotic systems is a multi-dimensional variable optimization problem. It is one of the key issues in chaotic control and synchronization. To identify the unknown parameters accurately and quickly, we introduce, in this paper, a modified Pelican Optimization Algorithm (POA) called the fractional-order chaotic Pareto Pelican Optimization Algorithm (FPPOA). First, the pelican population's diversity is augmented with the integration of a fractional chaotic sequence. Next, the utilization of the Pareto distribution is incorporated to alter the hunting strategy of pelicans in the POA. These measures are effective in hastening the speed of finding an optimal solution and circumventing local optimization issues. Thirdly, the FPPOA is used to determine the values of the parameters of the simplest memristive chaotic system, which has a property of conditional symmetry. The proposed algorithm was evaluated during simulations, where it was utilized to solve six objective functions of varying unimodal and multimodal types. The performance of the FPPOA exceeds three traditional swarm intelligence optimization algorithms. In the parameter identification experiment, the results for the parameters with the FPPOA had error rates all within a 1% range. Extensive testing shows that our new strategy has a faster rate of convergence and better optimization performance than some other traditional swarm algorithms.

Keywords: memristive chaotic system; parameter identification; Pelican Optimization Algorithm; fractional-order difference

Citation: Xiong, Q.; She, J.; Xiong, J. A New Pelican Optimization Algorithm for the Parameter Identification of Memristive Chaotic System. *Symmetry* **2023**, *15*, 1279. https://doi.org/10.3390/sym15061279

Academic Editor: Alexander Zaslavski

Received: 20 May 2023
Revised: 10 June 2023
Accepted: 15 June 2023
Published: 19 June 2023

1. Introduction

A memristor is a type of nonlinear electronic component capable of retaining the quantity of a charge that has previously passed through it. It establishes a connection between the magnetic flux and electric charge within the circuit [1]. Since the successful design of the genuine memristor by Strukov et al. [2] in 2008, research on memristors has emerged as a popular subject. It has garnered significant attention in various fields, such as AI [3,4], materials [5], and circuits [6,7], among others.

The memristor sets itself apart from conventional nonlinear electronic devices owing to its unique nonlinearity, switching mechanism, and memristance. As a result of these distinctive features, it exhibits a diverse range of complex behaviors when utilized in the construction of oscillation circuits [8]. Hence, the memristor proves to be highly appropriate for the creation of chaotic circuits that can yield a multitude of distinct behaviors, making it an ideal candidate for various engineering applications. The utilization of a system exhibiting chaotic behavior with memristive properties is becoming another area of active research [9,10].

In a memristive chaotic system, small variations in the initial conditions or system parameters can lead to drastically different outcomes, making parameter identification an essential task for understanding and controlling the system's behavior [11,12]. Due

to the absence of appropriate parameter optimization, the chaotic efficiency of a system may remain restricted based on chaos measurements. Setting parameters solely based on experience may result in inferior performance for real-time applications. On the contrary, a newly formed chaotic system that is set up with optimal parameters may exhibit the greater potential to be utilized in more real applications [13–15].

The process of identifying parameters in a chaotic system is a challenging nonlinear optimization problem. It involves searching through multiple dimensions. A convincing solution to this issue involves employing optimization algorithms based on swarm intelligence (SI). Over the past few decades, there has been remarkable progress in optimal engineering designs thanks to SI optimization algorithms [16]. SI offers advantages over traditional optimization methods. By converting the parameter identification procedure into a conventional multi-dimensional optimization challenge, SI algorithms have proven to be effective in the application to various chaotic systems, including the Lorenz, Chen, Rössler, and other renowned continuous chaotic systems. The identification of system parameters is achieved through various algorithms, including particle swarm optimization [17,18], differential evolution [19], artificial bee colony optimization [20], the bird swarm algorithm [21], and the Jaya algorithm [22]. These algorithms shed light on the task at hand.

On the other hand, there are only a limited number of studies that concentrate on identifying parameters for discrete chaotic systems [23–25]. A challenge is posed in identifying the parameters of discrete chaotic maps due to the increased sensitivity of the parameters in discrete nonlinear systems. This motivated us to create a novel SI optimization algorithm to precisely determine the parameters of a discrete memristive chaotic system.

In 2022, Trojovský P. et al. [26] proposed a novel SI optimization algorithm named the Pelican Optimization Algorithm (POA). It was designed by imitating the behaviors exhibited by pelicans when searching for and acquiring food. In comparison to eight renowned SI optimization algorithms, the POA attains exceptionally comparative performance through the maintenance of a harmonious equilibrium between exploration and exploitation. Therefore, the POA is used to solve some real-world applications. Despite its usefulness, the conventional POA is also prone to succumbing to local optimization. To circumvent this issue, several scholars have suggested alternative, enhanced techniques. Tuerxun et al. [27] introduced Tent chaos to augment the population diversity and incorporated a dynamic weight factor to facilitate the pelican's continual position update. These methods outperform traditional POAs and yield superior results in ten benchmark functions. However, the execution times of different algorithms are not compared in [27]. In this study, a modified POA called the FPPOA is proposed. The main contributions of this paper can be outlined as follows.

(1) We used fractional-order chaotics to initialize the population members of the POA instead of random initialization;
(2) We employed the Pareto distribution to accelerate global convergence and prevent it from being trapped in a local optimum;
(3) The improved POA was tested with benchmark functions;
(4) The FPPOA was employed to determine the parameters of an exemplary chaotic system.

2. Methods

2.1. Theory of Parameter Identification

Suppose an n-dimensional chaotic system at its starting point has m parameters. This system can be described with Equation (1):

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

where the system's state vector is denoted by $X = (x_1, x_2, \ldots, x_n)^T \in R^n$, $\theta = (\theta_1, \theta_2, \ldots, \theta_m)^T \in R^m$ is system parameter vector, X_0 is the initial state, and $F : R^n * R^m \rightarrow R^n$ is a given nonlinear vector function.

The identified system can be characterized as long as the system structure is known, as in Equation (2):

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

where $Y = (y_1, y_2, \ldots, y_n)^T \in R^n$ is the vector of the state, and $\hat{\theta} = \left(\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_m\right)^T \in R^m$ represents the specified system parameter of the designated system. We can use Equation (3) to describe the parameter identification issue:

$$\hat{\theta} = arg \min_{\theta} J(\theta) = arg \min_{\theta} \frac{1}{D_n} \sum_{i=1}^{D_n} \|x_i - y_i\|^2 \tag{3}$$

where, for state variables, D_n represents the data's length, x_i is the actual value, and y_i denotes the estimated value of the system in its corresponding state. By utilizing the aforementioned methodology, we can transform the parameter identification problem of the chaotic system into a multivariable optimization problem, and the pertinent variables can then be changed to reduce the desired value of J. A new Pelican Optimization Algorithm for the parameter identification of memristive chaotic systems is shown in Figure 1.

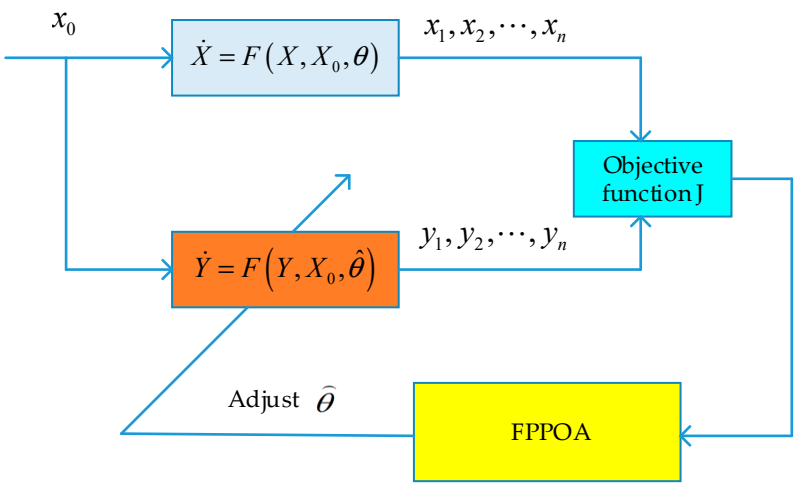

Figure 1. Optimization principle of parameter identification [25].

Initially, the sum of squared errors in the parameter estimation is computed by comparing the output values of the original system with those of the system to be estimated. Subsequently, the feedback obtained is fed into the FPPOA to generate new estimates. Finally, the estimated value is adjusted through multiple iterations to minimize the optimization function, J.

Determining the parameters of chaotic systems effectively and accurately is a challenge, given their dynamic instability and sensitivity to initial parameters. It is a multi-dimensional optimization issue, which is ideal for solving by swarm intelligence algorithms. The Pelican Optimization Algorithm search algorithm is easy to implement when solving multi-dimensional problems and can be effective in reducing the error rate, so this paper takes Equation (3) as the objective function to propose a modified POA.

2.2. Principle of Pelican Optimization Algorithm

The pelican is a part of the population in the population-based algorithm known as the POA, and pelicans frequently cooperate while pursuing prey. When pelicans spot their target, they plunge into it from an altitude of 10 to 20 m. They subsequently extend their wings to coerce the fish towards the shallows, making it easier for pelicans to catch their prey. The pelican's beak takes in a significant quantity of water while catching fish, causing the bird to thrust its head forward to eject the surplus liquid before engulfing the fish. The modeling of the aforementioned strategy served as the primary source of

inspiration for the design of the planned POA. It has been divided into two phases—Phase 1: Moving towards Prey (Exploration Phase) and Phase 2: Winging on the Water Surface (Exploitation Phase) [26].

2.2.1. Initialization

The POA is an algorithm based on populations in which each member of the pelican population is considered a candidate solution. The random initialization of each member in the population marks the beginning of the optimization procedure, employing Equation (4):

$$x_{i,j} = l_j + rand \cdot \left(u_j - l_j\right), i = 1, 2, \ldots, N, j = 1, 2, \ldots, m \tag{4}$$

where the total number of individuals in the population is denoted by N, $x_{i,j}$ is the candidate solution's value, and the vector, *rand*, is random, with values between [0, 1]. l_j represents the lower bound of the problem variables at index jth and the jth upper bound by u_j.

2.2.2. Moving towards Food Source (Exploration Phase)

The pelican locates its prey during this phase and swiftly descends upon it from a great height. The stochastic arrangement of the prey within the search area enhances the pelican's exploration capability. Equation (5) mathematically simulates the pelican's location update during each iteration, along with a depiction of its location.

$$x_{i,j}^{P_1} = \begin{cases} x_{i,j} + rand \cdot \left(p_j - I \cdot X_{\Gamma,j}\right), F_p < F_i; \\ x_{i,j} + rand \cdot \left(x_{i,j} - p_j\right), \quad else, \end{cases} \tag{5}$$

where $x_{i,j}^{P_1}$ represents the updated status of the ith pelican in the jth dimension based on Phase 1, p_j represents the prey's location in the jth dimension, and F_p represents the fitness function value. The value of parameter I is randomly assigned as either 1 or 2. When I equals 2, it facilitates increased member displacement, thereby enabling the exploration of uncharted territories within the search space. Consequently, parameter I influences the exploration capability of the Pelican Optimization Algorithm (POA) to efficiently scan the search space.

2.2.3. Winging on the Water Surface (Exploitation Phase)

Upon reaching the water's surface, the pelicans extend their wings to drive the fish towards shallower waters, making it easier for pelicans to catch their prey. This behavior could be mathematically modeled as Equation (6):

$$x_{i,j}^{P_2} = x_{i,j} + R \cdot \left(1 - \frac{1}{T}\right) \cdot (2 \cdot rand - 1) \cdot x_{i,j}, \tag{6}$$

where $x_{i,j}^{P_2}$ represents the updated status of the ith pelican in the jth dimension, considering Phase 2, R is a constant with a value of 0.2, and the radius of $x_{i,j}$ is represented by $R \cdot (1 - t/T)$, where t represents the iteration counter, and T represents the maximum number of iterations. The "$R \cdot (1 - t/T)$" coefficient represents the radius of the population members' neighborhood, enabling local searches near each member to attain convergence towards an enhanced solution. The exploitation power of the POA is greatly influenced by the coefficient, which allows it to get closer to the optimal global solution. In the early iterations, the coefficient has a high value, leading to a broader range being taken into account for each member in the whole population of pelicans. The "$R \cdot (1 - t/T)$" coefficient decreases as the number of iterations grows, resulting in a reduced search range for each member. These treatments can help the POA find the solution closer to the global optimal.

2.3. Fractional-Order Chaotic Cauchy Pelican Optimization Algorithm

2.3.1. Population Initialization with a Chaotic Fractional Sequence

Nature exhibits frequent occurrences of chaos, which is a prevailing nonlinear phenomenon. It finds extensive applications in optimizing search problems due to the ergodic and random features. Chaotic variables can not only preserve population diversity but also enable the algorithm to move beyond local optimization and enhance its capacity for global searches [28]. Fractional order chaos can generate sequences characterized by enhanced ergodicity and accelerated convergence speed, thereby leading to a reduction in chaotic fluctuations [29]. Equation (7) presents a 2D-SIMM chaotic system of fractional-order, as proposed by He et al. [30].

$$\begin{cases} {}^{c}\Delta_{t_0}^{q} x_1(t) = a\sin(\omega x_2(t+q-1))\sin(b/x_1(t+q-1)) - x_1(t+q-1) \\ {}^{c}\Delta_{t_0}^{q} x_2(t) = a\sin(\omega x_1(t+q))\sin(b/x_2(t+q-1)) - x_2(t+q-1) \end{cases} \tag{7}$$

where a, b, and q are the system parameters, and $a, b, \omega \in (0, +\infty)$, $t \in R$, and $c_{\Delta_{t_0}^{q}}$ are the Caputo-like fractional differences.

As a result, the numerical solution of this system is given as [29]

$$\begin{cases} x_1(i+1) = x_1(0) + \frac{1}{\Gamma(q)} \sum_{i+1}^{j=1} \frac{\Gamma(i-j+q)}{\Gamma(i-j+1)} \left[a\sin(\omega x_2(j-1))\sin\left(\frac{b}{x_1(j-1)}\right) - x_1(j-1) \right] \\ x_2(i+1) = x_2(0) + \frac{1}{\Gamma(q)} \sum_{i+1}^{j=1} \frac{\Gamma(i-j+q)}{\Gamma(i-j+1)} \left[a\sin(\omega x_1(j-1))\sin\left(\frac{b}{x_2(j-1)}\right) - x_2(j-1) \right], \end{cases} \tag{8}$$

where $q \in (0, 1)$ and $\Gamma(\cdot)$ are gamma functions. Figure 2 displays the phase diagrams when $a = 2$, $b = 3$, and $\omega = \pi$.

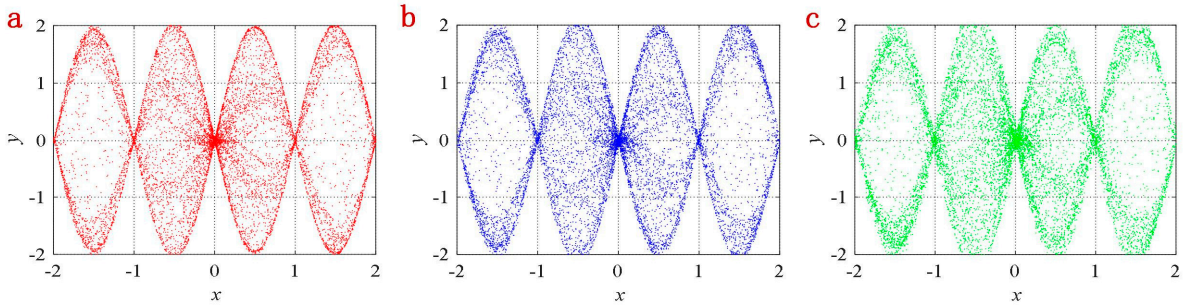

Figure 2. Diagrams of Equation (8) with various derivative orders. (**a**) $q = 0.98$; (**b**) $q = 0.95$; (**c**) $q = 0.9$.

We can use the following steps to initialize the population of the POA by using Equation (8):

Step 1: Let $q = 0.9$, $a = 2$, $b = 3$, and $\omega = \pi$;

Step 2: Use Equation (8) to produce two chaotic sequences, x_1 or x_2;

Step 3: Select a chaotic sequence from Step 2, such as x_2, and transform it to the solution space of the problem that needs to be resolved.

Therefore, Equation (9) assigns the recently produced chaotic sequence, Z, as the starting point for the sparrow population.

$$x_{i,j} = l_j + x_2 \cdot (u_j - l_j), i = 1, 2, \ldots, N, j = 1, 2, \ldots, m \tag{9}$$

where u_j represents the maximum value, and l_j represents the minimum value of the problem.

2.3.2. Update Position in the Exploitation Phase

The Pareto distribution is frequently employed in swarm intelligence algorithms to enhance global convergence and prevent getting stuck in local optima. If the cumulative

distribution function of a random variable is given by the subsequent equation, it is illustrated as conforming to the Pareto distribution [31].

$$F(x_3) = \begin{cases} 1 - \left(\frac{k}{x_3}\right)^h, & x_3 \geq k \\ 0, & x_3 < k \end{cases} \tag{10}$$

where h and k are the shape parameter and scale parameter, respectively. They are both greater than 0.

We can use Equation (10) to modify Equation (5), as follows:

$$x_{i,j}^{P_2} = x_{i,j} + \alpha \otimes Pareto(k,h) \otimes (x_{i,j} - x_{best}) \tag{11}$$

where x_{best} is the best solution at tth iteration, and a is a constant, which is equal to 0.01.

2.4. A Chaotic System with Memristive Properties

Like other electronic devices and circuits [32–35], a memristor chaotic circuit model can be implemented using a circuit. By utilizing a linear passive inductor, a linear passive capacitor, and a nonlinear active charge-driven memristor, we can build the most basic chaotic circuit employing a memristor [36]. The circuit diagram is shown in Figure 3 [36]. Its dimensions equation is Equation (12).

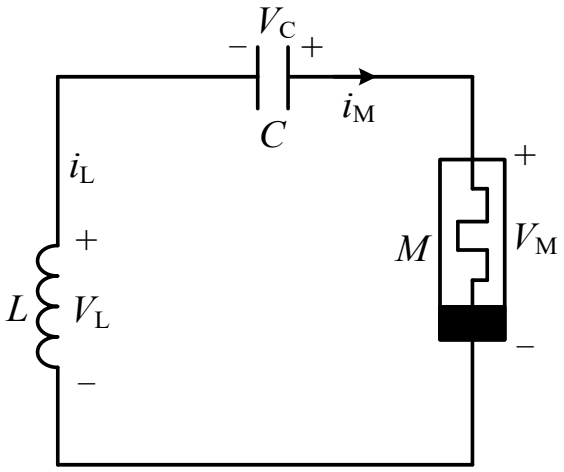

Figure 3. The simplest memristor chaotic circuit [36].

$$\begin{cases} \dot{x} = ay \\ \dot{y} = -b(x + d(z^2 - 1)y) \\ \dot{z} = y - cz + yz \end{cases} \tag{12}$$

where $\dot{x} = \frac{dx}{dt}, \dot{y} = \frac{dy}{dt}, \dot{z} = \frac{dz}{dt}$. The values a, b, c, and d represent the system parameters. When the system parameters are assigned specific values (a = 1, b = 1/3, c = 0.6, d = 1.5), system (12) demonstrates hyperchaotic and symmetrical behavior. By setting the initial state values as $x(0)$ = 0.1, $y(0)$ = 0, and $z(0)$ = 0, we can use the fourth-order Runge Kutta method to solve the differential, Equation (12). In Figure 4, the phase diagrams of the hyperchaotic attractor are illustrated, with a step length of h = 0.1 and sampling number of Sn = 20,000.

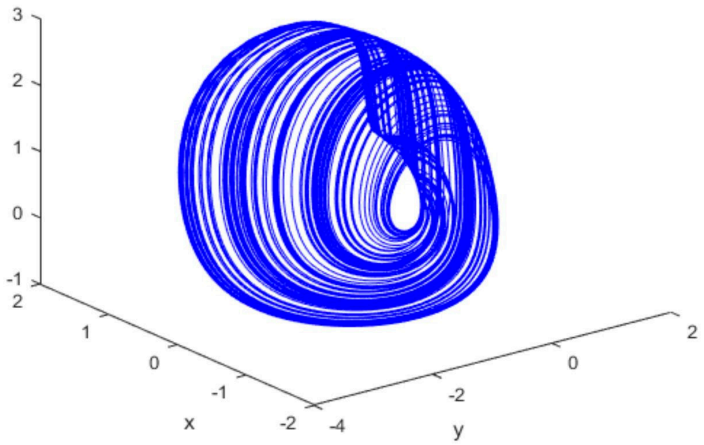

Figure 4. A diagram depicting the chaotic memristive system in three dimensions.

2.5. Parameter Identification with FPPOA

Algorithm 1 outlines the pseudo code for ascertaining the FPPOA-based parameters of a memristive system.

Algorithm 1 Pseudo code of FPPOA in parameter identification

Input: the maximum iterations as the *Tmax*, the population size as *N*, and the range of parameters.
Output: the values of the parameters.
1. Initialize the members using the fractional chaotic sequence;
2. Calculate the objective function value of each individual, and find the Gbest;
3. while ($t < T_{max}$);
4. Update the location of the pelican by Equations (5) and (11);
5. Compute the value of the individual's objective function;
6. Sort the objective function value;
7. Obtain the best solution (Gbest);
8. Update the best solution if it is better than before;
9. $t = t + 1$;
End while
10. Output the values of parameters.

3. Experiments Results

To evaluate the effectiveness of the FPPOA, we conducted experiments to compare the benchmark functions and identify respective parameters. The experiments for comparing the benchmark functions were intended to showcase the general improvement in the performance achieved by the FPPOA, while the parameter identification tests aimed to confirm the applicability of the FPPOA to practical projects. These experiments were conducted on a desktop PC with a 3.20 GHz CPU and 32 GB memory, using software such as Windows 10 and MATLAB 2022b.

3.1. Benchmark Function Comparison Experiment

To evaluate the performance of the FPPOA algorithm, we carried out 30 independent experiments on six distinct benchmark test functions [37]. These functions included three high-dimensional unimodal functions (F1–F3) and three high-dimensional multimodal functions (F4–F6), which are listed in Table 1.

Table 1. Benchmark test functions.

Type	Benchmark Functions	Dim	Range	Fmin
Unimodal benchmark functions	$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]$	0
	$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10, 10]$	0
	$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} x_j \right)^2$	30	$[-100, 100]$	0
Multimodal benchmark functions	$F_4(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	$[-32, 32]$	0
	$F_5(x) = \frac{\pi}{n}\left\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$, $y_i = 1 + \frac{x_i + 1}{4}$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	$[-50, 50]$	0
	$F_6(x) = 0.1\left\{ \begin{array}{l} \sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2\left[1 + \sin^2(3\pi x_i + 1)\right] \\ +(x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right] \end{array} \right\} = + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	$[-50, 50]$	0

During the simulation test, we set the number of pelicans to N = 30 and the iteration number to $Tmax$ = 500. The range and dimension were chosen based on the benchmark functions listed in Table 1. We conducted independent experiments on each function 30 times and recorded the best values, average values, standard deviation, and computation time of each algorithm, as shown in Table 2. The best value indicates the algorithm's ability to search for the global optimal solution by exploring a wide range of potential solutions in the search space. The convergence accuracy is represented by the average value, while the stability of the FPPOA under identical benchmark test functions is represented by the standard variance.

Table 2. Experiment results of test functions.

F	SI	Best	Average	Std.	Computation Time (s)
F1	PSO	7.213	1.885×10^{2}	87.91	1.5
	GWO	1.8448×10^{-29}	1.362×10^{-27}	2.4093×10^{-27}	3.6
	POA	1.5245×10^{-118}	1.7523×10^{-101}	9.5379×10^{-101}	3.51
	FPPOA	6.4028×10^{-245}	1.4332×10^{-226}	0	4.52
F2	PSO	8.160150	3.057447×10	20.0995	1.23973
	GWO	2.7059×10^{-17}	9.9414×10^{-17}	9.3072×10^{-17}	3.6
	POA	2.6592×10^{-59}	1.6988×10^{-52}	4.7363×10^{-52}	3.6
	FPPOA	1.0471×10^{-138}	3.3491×10^{-123}	1.8154×10^{-122}	4.67
F3	PSO	2.908686×10^{3}	8.612808×10^{3}	5414.9	6.26
	GWO	7.451×10^{-10}	4.3214×10^{-5}	0.0001	6.6
	POA	4.7259×10^{-115}	3.9312×10^{-97}	2.1532×10^{-96}	9.6
	FPPOA	2.1391×10^{-246}	2.5878×10^{-228}	0	10.49

Table 2. *Cont.*

F	SI	Best	Average	Std.	Computation Time (s)
F4	PSO	9.002	10.423	0.897	2.01
	GWO	6.7946×10^{-14}	9.8973×10^{-14}	1.5666×10^{-14}	9.82
	POA	4.4409×10^{-16}	3.8784×10^{-15}	6.4863×10^{-16}	4.06
	FPPOA	4.4409×10^{-16}	4.4409×10^{-16}	0	4.89
F5	PSO	1.1014	4.680	2.562	8.92
	GWO	0.0133	0.044	0.023	9.38
	POA	0.085	0.165	0.051	14.65
	FPPOA	1.5705×10^{-32}	1.5705×10^{-32}	5.5674×10^{-48}	15.49
F6	PSO	4.334	15.261	9.078	9.03
	GWO	0.101	0.583	0.299	9.15
	POA	1.955	2.763	0.332	14.15
	FPPOA	1.349784×10^{-32}	1.349784×10^{-32}	5.5674×10^{-48}	15.34

Table 2 reveals that, for unimodal benchmark functions, although the optimal value of the FPPOA does not converge to the optimal value, its best and average values are closest to the optimal value compared to the other three algorithms. This indicates that the FPPOA has the highest convergence accuracy. Furthermore, all the standard deviations of the FPPOA are minimal, exhibiting the highest level of stability, indicating its superiority. Although the PSO is the fastest, its accuracy and stability are the poorest among the algorithms. For the multimodal benchmark function, F4, both the POA and FPPOA have the same best value, but the standard deviation of the FPPOA is zero, which is much better than the POA, showing its good stability. For functions F5 and F6, the FPPOA outperforms the other three algorithms in terms of the best or average value, demonstrating its good accuracy. On the other hand, its variance is also the smallest among the four algorithms, also indicating its good stability. Figure 5 presents the convergence curves of six benchmark functions under four optimization algorithms, demonstrating the dynamic convergence characteristics of the FPPOA.

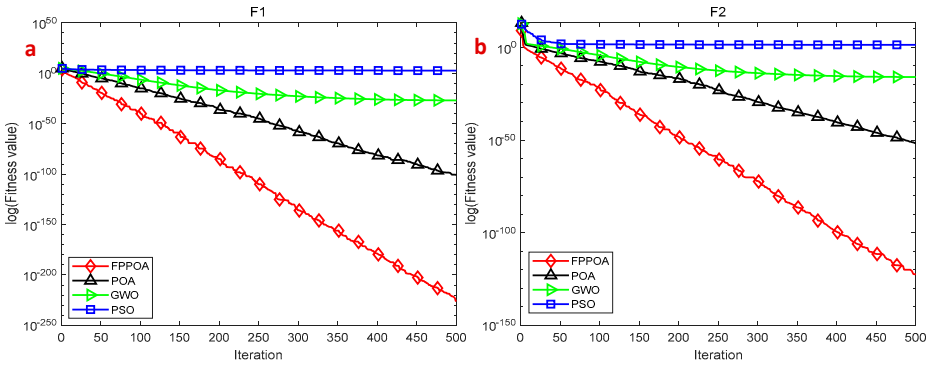

Figure 5. *Cont.*

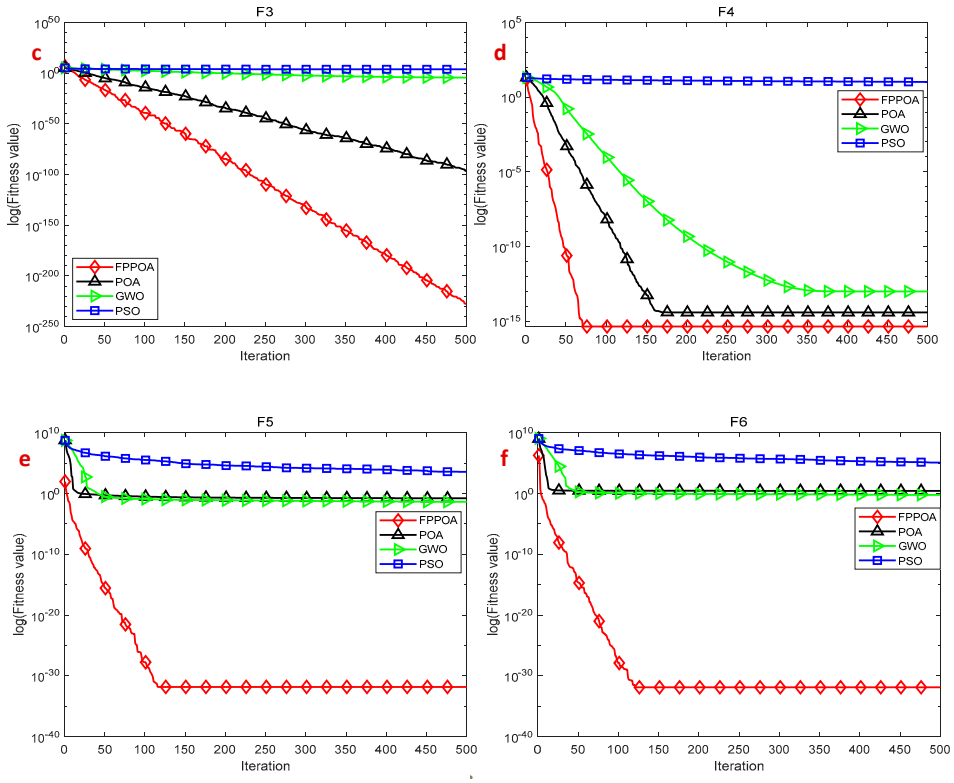

Figure 5. Curves of four algorithms on six test functions. (**a**) F1; (**b**) F2; (**c**) F3; (**d**) F4; (**e**) F5; (**f**) F6.

Examining Figure 5, we can observe that the FPPOA demonstrates a swifter downward trend with a more pronounced slope, demanding less iteration in comparison to the remaining three algorithms, specifically as depicted in Figure 5d. As a result, in terms of convergence speed and optimization impact, the FPPOA surpasses the PSO, GWO, and POA. The feasibility and superiority of the FPPOA are demonstrated by these findings.

3.2. Simulation Test on Parameter Identification

In the following simulation test, we used the FPPOA to estimate the parameter value in Equation (12), in which $x(0) = 0.1$, $y(0) = 0$, and $z(0) = 0$ are the initial values. The range of the parameters was set to $0 \leq a \leq 3, 0 \leq b \leq 2, 0 \leq c \leq 3, 0 \leq d \leq 2$. To reduce the computation time, the step length, h, was set to 0.01, and the sampling points, $Sn = 1000$. The number of pelicans was $N = 20$, and the iteration number was $Tmax = 100$.

The convergence curve of the simulation test is shown in Figure 6. The final identification results of the four different algorithms are provided in Table 3. Figure 7 displays the curve of parameter identification convergence.

Table 3. Results of parameter identification by four different algorithms.

SI	*a*	Error Rate%	*b*	Error Rate%	*c*	Error Rate%*d*		Error Rate%	Best *J* Value
PSO	1.5408	54	0.2138	35.86	0.2196	63.4	2.5068	67.12	1.848×10^{-2}
GWO	0.9905	0.95	0.3367	1.01	0.5723	4.61	1.4680	2.13	2.883×10^{-3}
POA	1.0238	2.38	0.3258	0.25	0.5689	5.18	1.5372	2.48	2.847×10^{-2}
FPPOA	0.9952	0.48	0.3351	0.54	0.5989	0.18	1.4909	0.61	2.814×10^{-2}

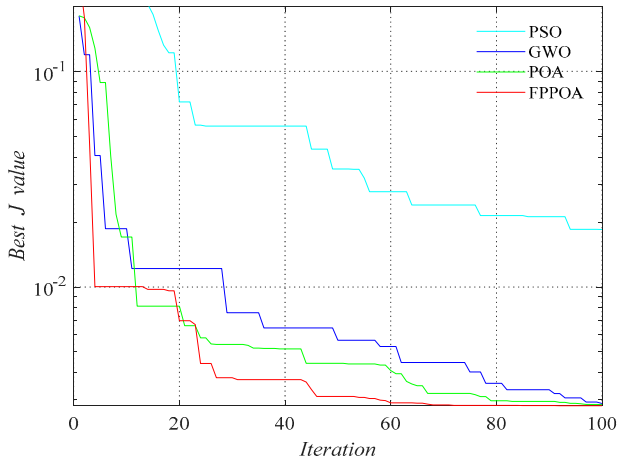

Figure 6. The convergence curve of a memristive chaotic system.

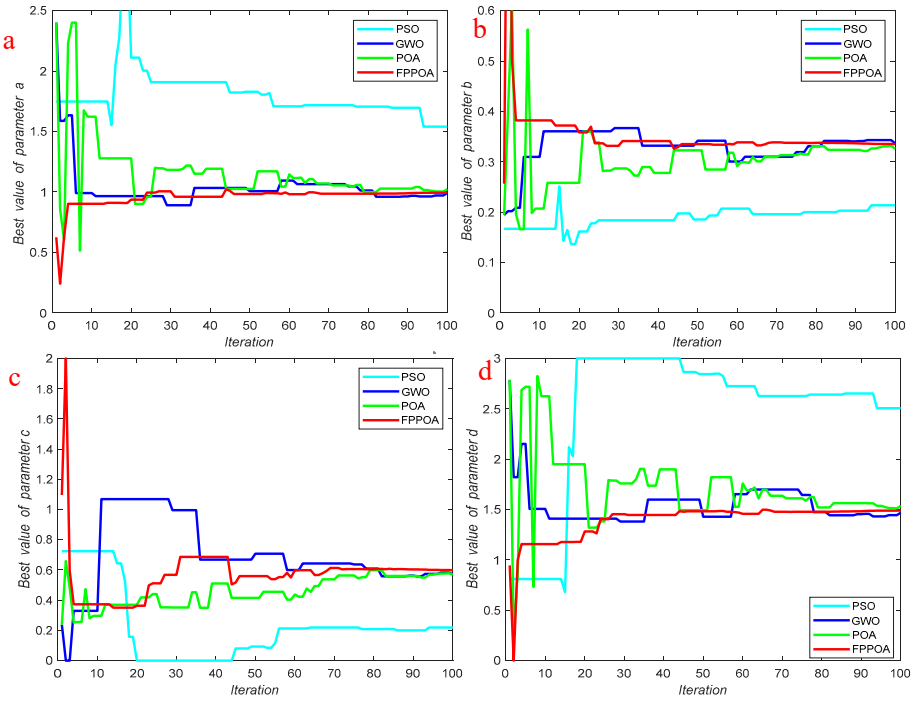

Figure 7. Convergence and optimization curve of parameters. (**a**) Optimization curve of parameter a. (**b**) Optimization curve of parameter b. (**c**) Optimization curve of parameter c. (**d**) Optimization curve of parameter d.

Figure 6 clearly shows that the FPPOA exhibits the swiftest speed of convergence and the tiniest *J* value. While reaching the optimal value, the POA exhibits a considerably slower convergence rate compared to the FPPOA. This indicates that, out of the four algorithms, the FPPOA delivers the most exceptional performance. From Figure 7, it is evident that the FPPOA always converges to the optimal value the fastest among the four algorithms. Additionally, according to Table 3, the parameter values derived from the FPPOA exhibit the closest proximity to the real value. These findings collectively showcase the superior convergence and stability of the FPPOA when compared to the remaining three algorithms.

4. Conclusions

This paper presents the application of the inventive optimization technique, the revised Pelican Search Algorithm, to perform parameter estimation in a memristive chaotic system. To accelerate convergence and avoid local optimum, this algorithm utilizes fractional-order

chaos for population initialization, and the positions of the pelicans in the POA are adjusted by incorporating the Pareto distribution. The accuracy and effectiveness of the proposed algorithm are validated using six standard test functions, which demonstrate that the FPPOA achieves high search accuracy. In addition, the identification results for the four parameters have error rates that are all within a 1% range. It demonstrates that the FPPOA exhibits superior accuracy and stability when compared to alternative algorithms, such as the POA, GWO, and PSO, in determining the parameters of the most basic memristive chaotic system. This confirms the satisfactory effectiveness and robustness of the FPPOA and indicates that this algorithm can also bring advantages to other chaotic memristive systems. If the equation of other systems is known, their parameters can be identified using the methods described in this paper.

Author Contributions: Conceptualization, Q.X. and J.S.; methodology, Q.X. and J.X.; software, Q.X. and J.X.; validation, J.S and J.X.; formal analysis, Q.X.; investigation, J.S.; resources, Q.X. and J.S.; data curation, Q.X.; writing—original draft preparation, Q.X. and J.S.; writing—review and editing, J.S. and J.X.; visualization, J.S; supervision, Q.X.; project administration, Q.X.; funding acquisition, Q.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Hunan Province, grant number 2021JJ50137.

Data Availability Statement: Not Applicable.

Acknowledgments: The authors would like to thank the anonymous reviewers for their constructive comments and insightful suggestions. The authors would also like to thank Jun Shen of the University of Wollongong in Australia for his revision suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mazumder, P.; Kang, S.M.; Waser, R. Memristors: Devices, models, and applications. *Proc. IEEE* **2012**, *100*, 1911–1919. [CrossRef]
2. Strukov, D.B.; Snider, G.S.; Stewart, D.R.; Williams, R.S. The missing memristor found. *Nature* **2008**, *453*, 80–83. [CrossRef] [PubMed]
3. He, S.; Liu, J.; Wang, H.; Sun, K. A discrete memristive neural network and its application for character recognition. *Neurocomputing* **2023**, *523*, 1–8. [CrossRef]
4. Yao, P.; Wu, H.; Gao, B.; Tang, J.; Zhang, Q.; Zhang, W.; Yang, J.J.; Qian, H. Fully hardware-implemented memristor convolutional neural network. *Nature* **2020**, *577*, 641–646. [CrossRef]
5. Zhao, B.; Xiao, M.; Zhou, Y.N. Synaptic learning behavior of a TiO_2 nanowire memristor. *Nanotechnology* **2019**, *30*, 425202. [CrossRef]
6. Lakshmi Priya, G.; Rawat, N.; Abhishek, S.; Venkatesh, M. Modeling and simulation of double gate dielectric stack silicon substrate memristor circuits for low power applications. *Silicon* **2022**, *14*, 12403–12414. [CrossRef]
7. Babacan, Y.; Yesil, A.; Tozlu, O.F.; Kacar, F. Investigation of STDP mechanisms for memristor circuits. *AEU-Int. J. Electron. Commun.* **2022**, *151*, 154230. [CrossRef]
8. Itoh, M.; Chua, L.O. Dynamics of memristor circuits. *Int. J. Bifurc. Chaos* **2014**, *24*, 1430015. [CrossRef]
9. Chen, M.; Wang, Z.; Nazarimehr, F.; Jafari, S. A novel memristive chaotic system without any equilibrium point. *Integration* **2021**, *79*, 133–142. [CrossRef]
10. Ostrovskii, V.; Fedoseev, P.; Bobrova, Y.; Butusov, D. Structural and Parametric Identification of Knowm Memristors. *Nanomaterials* **2021**, *12*, 63. [CrossRef] [PubMed]
11. Yan, S.; Ren, Y.; Song, Z.; Shi, W.; Sun, X. A memristive chaotic system with rich dynamical behavior and circuit implementation. *Integration* **2022**, *85*, 63–75. [CrossRef]
12. Li, Y.; Huang, X.; Song, Y.; Lin, J. A new fourth-order memristive chaotic system and its generation. *Int. J. Bifurc. Chaos* **2015**, *25*, 1550151. [CrossRef]
13. Kumar, C.; Raj, T.D.; Premkumar, M.; Raj, T.D. A new stochastic slime mould optimization algorithm for the estimation of solar photovoltaic cell parameters. *Optik* **2020**, *223*, 165277. [CrossRef]
14. Manoharan, P.; Subramaniam, U.; Babu, T.S.; Padmanaban, S.; Holm-Nielsen, J.B.; Mitolo, M.; Ravichandran, S. Improved perturb and observation maximum power point tracking technique for solar photovoltaic power generation systems. *IEEE Syst. J.* **2020**, *15*, 3024–3035. [CrossRef]
15. Premkumar, M.; Jangir, P.; Sowmya, R.; Elavarasan, R.M.; Kumar, B.S. Enhanced chaotic JAYA algorithm for parameter estimation of photovoltaic cell/modules. *ISA Trans.* **2021**, *116*, 139–166. [CrossRef] [PubMed]
16. Abualigah, L. Group search optimizer: A nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. *Neural Comput. Appl.* **2021**, *33*, 2949–2972. [CrossRef]

17. Lazzus, J.A.; Vega-Jorquera, P.; Lopez-Caraballo, C.H.; Palma-Chilla, L.; Salfate, I. Parameter estimation of a generalized lotka–volterra system using a modified pso algorithm. *Appl. Soft Comput.* **2020**, *96*, 106606. [CrossRef]
18. Peng, Y.; Sun, K.; He, S.; Yang, X. Parameter estimation of a complex chaotic system with unknown initial values. *Eur. Phys. J. Plus* **2018**, *133*, 1–13. [CrossRef]
19. Toktas, A.; Erkan, U.; Toktas, F.; Yetgın, Z. Chaotic map optimization for image encryption using triple objective differential evolution algorithm. *IEEE Access* **2021**, *9*, 127814–127832. [CrossRef]
20. 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.* **2023**, *35*, 13207–13223. [CrossRef]
21. Zhang, P.; Yang, R.; Yang, R.; Ren, G.; Yang, X.; Xu, C.; Xu, B.; Zhang, H.; Cai, Y.; Lu, Y. Parameter estimation for fractional-order chaotic systems by improved bird swarm optimization algorithm. *Int. J. Mod. Phys. C* **2019**, *30*, 1950086. [CrossRef]
22. Chen, F.; Ding, Z.; Lu, Z.; Zeng, X. Parameters identification for chaotic systems based on a modified Jaya algorithm. *Nonlinear Dyn.* **2018**, *94*, 2307–2326. [CrossRef]
23. Peng, Y.; Sun, K.; He, S.; Peng, D. Parameter identification of fractional-order discrete chaotic systems. *Entropy* **2019**, *21*, 27. [CrossRef] [PubMed]
24. 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* **2020**, *29*, 030502. [CrossRef]
25. Peng, Y.; He, S.; Sun, K. Parameter identification for discrete memristive chaotic map using adaptive differential evolution algorithm. *Nonlinear Dyn.* **2022**, *107*, 1263–1275. [CrossRef]
26. Trojovský, P.; Dehghani, M. Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications. *Sensors* **2022**, *22*, 855. [CrossRef]
27. Tuerxun, W.; Xu, C.; Haderbieke, M.; Guo, L.; Cheng, Z. A wind turbine fault classification model using broad learning system optimized by improved pelican optimization algorithm. *Machines* **2022**, *10*, 407. [CrossRef]
28. Xiong, Q.; Zhang, X.; Xu, X.; He, S. A modified chaotic binary particle swarm optimization scheme and its application in face-iris multimodal biometric identification. *Electronics* **2021**, *10*, 217. [CrossRef]
29. Xiong, Q.; Zhang, X.; He, S.; Shen, J. A fractional-order chaotic sparrow search algorithm for enhancement of long distance iris image. *Mathematics* **2021**, *9*, 2790. [CrossRef]
30. He, S.; Sun, K.; Wu, X. Fractional symbolic network entropy analysis for the fractional-order chaotic systems. *Phys. Scr.* **2020**, *95*, 035220. [CrossRef]
31. Haj Ahmad, H.; Almetwally, E.M. Generating Optimal Discrete Analogue of the Generalized Pareto Distribution under Bayesian Inference with Applications. *Symmetry* **2022**, *14*, 1457. [CrossRef]
32. Alibakhshikenari, M.; Virdee, B.S.; See, C.H.; Abd-Alhameed, R.A.; Falcone, F.; Limiti, E. Super-wide impedance bandwidth planar antenna for microwave and millimeter-wave applications. *Sensors* **2019**, *19*, 2306. [CrossRef]
33. Alibakhshi-Kenari, M.; Andújar, A.; Anguera, J. New compact printed leaky-wave antenna with beam steering. *Microw. Opt. Technol. Lett.* **2016**, *58*, 215–217. [CrossRef]
34. Alibakhshikenari, M.; Virdee, B.S.; Shukla, P.; See, C.H.; Abd-Alhameed, R.A.; Falcone, F.; Limiti, E. Improved adaptive impedance matching for RF front-end systems of wireless transceivers. *Sci. Rep.* **2020**, *10*, 14065. [CrossRef]
35. Awan, W.A.; Naqvi, S.I.; Ali, W.A.E.; Hussain, N.; Iqbal, A.; Tran, H.H.; Alibakhshikenari, M.; Limiti, E. Design and realization of a frequency reconfigurable antenna with wide, dual, and single-band operations for compact sized wireless applications. *Electronics* **2021**, *10*, 1321. [CrossRef]
36. Mou, J.; Xia, B.; Guo, W.; Li, T.; Wang, H. Circuit implementation and application of memristor. *J. Dalian Polytech. Univ.* **2015**, *34*, 214–216.
37. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]