mathematics

MDPI

Article

Employing Quantum Fruit Fly Optimization Algorithm for Solving Three-Dimensional Chaotic Equations

Qasim M. Zainel [1], Saad M. Darwish [2,*] and Murad B. Khorsheed [3]

[1] College of Physical Education and Sports Sciences, University of Kirkuk, Kirkuk 36001, Iraq
[2] Department of Information Technology, Institute of Graduate Studies and Research, Alexandria University, Alexandria 21526, Egypt
[3] College of Administration & Economics, University of Kirkuk, Kirkuk 36001, Iraq
* Correspondence: saad.darwish@alexu.edu.eg; Tel.: +2-01222632369

Abstract: In a chaotic system, deterministic, nonlinear, irregular, and initial-condition-sensitive features are desired. Due to its chaotic nature, it is difficult to quantify a chaotic system's parameters. Parameter estimation is a major issue because it depends on the stability analysis of a chaotic system, and communication systems that are based on chaos make it difficult to give accurate estimates or a fast rate of convergence. Several nature-inspired metaheuristic algorithms have been used to estimate chaotic system parameters; however, many are unable to balance exploration and exploitation. The fruit fly optimization algorithm (FOA) is not only efficient in solving difficult optimization problems, but also simpler and easier to construct than other currently available population-based algorithms. In this study, the quantum fruit fly optimization algorithm (QFOA) was suggested to find the optimum values for chaotic parameters that would help algorithms converge faster and avoid the local optimum. The recommended technique used quantum theory probability and uncertainty to overcome the classic FA's premature convergence and local optimum trapping. QFOA modifies the basic Newtonian-based search technique of FA by including a quantum behavior-based searching mechanism used to pinpoint the position of the fruit fly swarm. The suggested model has been assessed using a well-known Lorenz system with a specified set of parameter values and benchmarked signals. The results showed a considerable improvement in the accuracy of parameter estimates and better estimation power than state-of-the art parameter estimation approaches.

Keywords: chaotic system; fruit fly optimization algorithm; quantum-inspired computation; parameter estimation

MSC: 68T20

Citation: Zainel, Q.M.; Darwish, S.M.; Khorsheed, M.B. Employing Quantum Fruit Fly Optimization Algorithm for Solving Three-Dimensional Chaotic Equations. *Mathematics* **2022**, *10*, 4147. https://doi.org/10.3390/math10214147

Academic Editor: Tao Zhou

Received: 21 September 2022
Accepted: 13 October 2022
Published: 6 November 2022

1. Introduction

Chaos theory studies nonlinear dynamic systems. Chaos is the interaction between regularity and probability-based unpredictability [1]. Weather and climate, biological and ecological processes, the economy, social structures, and other natural phenomena all exhibit chaotic regimes. The primary feature of chaos is its ability to generate a wide range of complex patterns. For use as cryptographic secret keys, relevant mathematical models may produce a vast amount of data. Confusion and diffusion are two key features of cryptography, and chaos theory has the unique quality of having a direct connection to both features. Furthermore, the deterministic but unexpected dynamics of chaotic systems may be a powerful tool in the development of a superior cryptosystem [2,3].

The fundamental benefit of chaos is that unauthorized users see chaotic signals as noise [2]. Chaotic-based encryption techniques are utilized for military, mobile, and private data [3]. These applications demand real-time, rapid, secure, and reliable monitoring. Most chaos-based secure communication systems use chaos synchronization [4]. Chaos synchronization is vital for achieving security after information has been transferred [5].

Therefore, many cryptographic algorithms have adopted popular chaotic models that depict chaos by employing mathematical models, such as a logistic map.

Chaos-based secure communication has issues. Due to the limitations of chaos theory and techniques for creating chaos, attackers may sometimes determine the chaotic system employed in encryption through state reconstruction. Second, transmission and sampling delays make chaotic synchronization difficult. Due to the limits of digital computer accuracy, computer chaotic maps are always periodic. Therefore, chaos-based public-key cryptography has collisions [6]. Finally, picking the input parameters limits chaos theory. The techniques used to determine these characteristics rely on the data dynamics and the desired analysis, which is often complicated and inaccurate. Due to a chaotic system's complicated nature, many practical characteristics are unknown and difficult to quantify [7]. Parameter estimation is a major issue.

Two parameter estimation methods exist. One is the synchronization method [3,8], which proposes updating parameter estimation based on chaotic system stability. Its methodologies and sensitivities rely on the considered system; hence, updating may be challenging due to the complexity of the chaotic system. Another method is through metaheuristic algorithms. Metaheuristic algorithms are intelligent optimization algorithms [9,10]. It translates parameter estimation into a multidimensional optimization problem using sample data from the original system. It is easier to implement than synchronization. Metaheuristic algorithms are popular for estimating chaotic system parameters [11,12]. Metaheuristic techniques require starting system settings. In many circumstances, the original values cannot be retrieved, making reconstruction and management of the chaotic system difficult. Most of these approaches are also used to estimate chaotic system parameters. Few apply to complex chaotic systems [13].

The fruit fly optimization algorithm (FOA) is simple and easy to comprehend compared with other sophisticated algorithms. FOA only requires adjusting the population size and maximum generation number. Traditional intelligent algorithms need at least three parameters. The influence of numerous factors on algorithm performance is hard to examine; hence, they are generally determined via several tests. An incorrect parameter will impair algorithm performance and complexity [14]. However, there is still a lot of potential for development of FOA variations to obtain greater performance, particularly for complicated practical issues related to convergence speed or avoiding being trapped into the local optimum.

When it comes to population-based optimization methods, variability in the population and unpredictability in the search process are two factors that often play a pivotal role. By using quantum mechanics instead of Newtonian dynamics, the quantum-behaved particle swarm optimization (QPSO) increases the particles' capacity to escape the local optimum. Classical quantum mechanics is the theoretical underpinnings of quantum theory, which aims to appropriate some of the mysteriousness of quantum behavior processes. Integrating quantum theory into the original FA, the quantum firefly algorithm (QFA) is able to combat the loss of variety [15]. Quantum mechanics may be used to explain how fruit flies navigate the environment in search of food; their actions are characterized by a wave function of uncertainty. A quantum-behaved approach can avoid premature convergence and help escape from the local optimum.

1.1. Problem Statement and Motivation

Chaotic systems are very sensitive to initial parameter choices. Long-term system behavior prediction is difficult. Synchronization and chaos control in nonlinear systems depend on exact parameter values in chaotic systems; if one of these values is uncertain, the system will not perform as intended. Some parameters are unknown or difficult to quantify due to the complexity of chaotic systems (such as secure communication). If we wish to control or synchronize chaotic systems, we must estimate unknown system parameters. Too many factors may cause the parameter estimation algorithm for 3D chaotic systems to become more complex, which in turn increases the amount of effort required

to calculate the results. This is why most algorithms struggle to find the global optimum. As a result of its effectiveness, FA has been used to tackle a wide range of optimization issues, leading to significant progress in a short period of time. The motivation is to take insights from quantum theory to improve upon the FA for estimating the parameters of a 3D chaotic system.

1.2. Contribution and Methodology

The work presented in this paper is an extension of the work introduced in Ref. [16], where quantum mechanics was used in the fruit fly optimization algorithm to make it easier for particles to get out of the local optimum, so that the chaotic system parameters could be estimated. In this paper, the QFOA was adopted to solve the parameter estimation problem of the Lorenz chaotic system to achieve the synchronization with the aim of transmitting data correctly. Fitness function based on the mean square error was utilized to find the minimum error between the original and estimated ones in different directions. To achieve high performance in terms of time and accuracy, the suggested model selected only some samples from the received signal to check the synchronization early. QFOA variables were tuned to estimate the unknown chaotic system parameters. Then, these estimated parameters were used later, inside the well-known fourth-order Runge–Kutta algorithm, to build the estimated original signal (a chaotic signal with a known structure) to yield synchronization.

The rest of this paper is organized as follows: Section 2 provides a background and literature review of some studies related to estimating the parameters of the chaotic system; Section 3 presents the proposed methodology based on the analysis of the previous techniques; Section 4 reports a complete evaluation of the proposed methodology, along with the results and the discussion; and the final section contains the conclusion based on the previous sections and future directions for research.

2. Background and Related Work

This section offers some important background related to the proposed model and includes a literature review on parameters estimation of the chaotic system as one of the most important techniques to achieve chaotic synchronization concerns on wireless communication networks.

2.1. Preliminaries

2.1.1. Chaos Theory

Chaos theory is an alternative description and explanation of the behavior of nonlinear dynamical systems [17]. In mathematical language, a dynamical system is classified as a chaotic system [18–21] if it has the following properties:

- Sensitive to initial conditions—each point in a system is arbitrarily near other points with drastically different behavior. Qualitatively, two paths with a starting separation δX_0 diverge.

$$|\delta X(t)| \approx e^{\lambda t}|\delta X_0| \tag{1}$$

λ is the Lyapunov exponent. One positive Lyapunov exponent indicates chaotic behavior, whereas more than one indicates hyperchaotic behavior.

- Topological mixing—implies system evolution, so that every area or open set of its phase space will overlap. This assumption has profound implications for one-dimensional systems.
- Periodic orbit density—each space point is arbitrarily near periodic orbits and is regular. Not meeting this requirement may prevent topological mixing systems from becoming chaotic. In chaos theory, the butterfly effect is the sensitivity of a system to starting conditions. Small changes in a dynamical system's starting state may have huge long-term effects. Time makes such systems unpredictable.

2.1.2. Lyapunov Exponents

The Lyapunov exponents help investigate chaotic or hyperchaotic dynamical systems. Lyapunov exponents categorize dynamical systems so that we can see their behavior. A dynamical system is chaotic if it has one positive Lyapunov exponent, and hyperchaotic if it has more [22–24]. Consider two locations in space, X_0 and $X_0 + \Delta X_0$, which form orbits using an equation or set of equations. Sensitive dependency may only occur in particular parts of a system; hence, this separation depends on the beginning value, $\Delta x(X_0,t)$. For chaotic points, $\Delta x(X_0,t)$ acts unpredictably. The mean exponential rate of divergence of two near orbits is defined as [25].

$$\lambda = \lim_{\substack{t \to \infty \\ |\Delta X_0| \to 0}} \frac{1}{t} \ln \left| \frac{\Delta x(X_0,t)}{\Delta X_0} \right| \tag{2}$$

The Lyapunov exponent, λ, is used to differentiate orbits. If $\lambda < 0$, the orbit attracts a stable fixed point or periodic orbit. The more negative the exponent, the better the stability. If $\lambda = 0$, the system is steady state. A conservative system has this exponent and are Lyapunov stable. In this case, orbits would stay apart. For $\lambda > 0$, the orbit is chaotic. Nearby points diverge to any arbitrary separation.

To define sphere trajectories, we require linearized systems or variational equations. $\vec{x} = \vec{F}(\vec{x})$, where $\vec{x} = (x_1, x_2, \ldots\ldots, x_n)$ and $\vec{F} = (f_1, f_2, \ldots\ldots, f_n)$. Any ordinary numerical differential equation solution may create $\varnothing(\vec{x_0})$. Formally, partial derivatives explain how these perturbations respond. Consider the Lorenz system [26–28]:

$$\begin{cases} \dot{x} = \theta_1(y - x) \\ \dot{y} = \theta_2 x - y - xz \\ \dot{z} = -\theta_3 z + xy \end{cases} \tag{3}$$

θ_1, θ_2, and θ_3 are Lorenz parameters. To set up the linearized system for the above equations, the right-hand Jacobian is needed.

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} \end{bmatrix} \tag{4}$$

$$J = \begin{bmatrix} -\theta_1 & \theta_1 & 0 \\ \theta_2 - Z & -1 & -x \\ y & x & -\theta_3 \end{bmatrix} \tag{5}$$

$$J = \begin{bmatrix} \delta_{x1} & \delta_{y1} & \delta_{z1} \\ \delta_{x2} & \delta_{y2} & \delta_{z2} \\ \delta_{x3} & \delta_{y3} & \delta_{z3} \end{bmatrix} \tag{6}$$

The ith equation's x variation component is δ_{xi}. Column sums are the x, y, and z coordinates of the evolving variant. The rows represent the vector coordinates of the original x, y, and z variations. Linear equations:

$$\begin{bmatrix} \dot{\delta}_{x1} & \dot{\delta}_{y1} & \dot{\delta}_{z1} \\ \dot{\delta}_{x2} & \dot{\delta}_{y2} & \dot{\delta}_{z2} \\ \dot{\delta}_{x3} & \dot{\delta}_{y3} & \dot{\delta}_{z3} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial z} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial z} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial z} \end{bmatrix} \begin{bmatrix} \delta_{x1} & \delta_{y1} & \delta_{z1} \\ \delta_{x2} & \delta_{y2} & \delta_{z2} \\ \delta_{x3} & \delta_{y3} & \delta_{z3} \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} \dot{\delta}_{x1} & \dot{\delta}_{y1} & \dot{\delta}_{z1} \\ \dot{\delta}_{x2} & \dot{\delta}_{y2} & \dot{\delta}_{z2} \\ \dot{\delta}_{x3} & \dot{\delta}_{y3} & \dot{\delta}_{z3} \end{bmatrix} = \begin{bmatrix} -\theta_1 & \theta_1 & 0 \\ \theta_2 - Z & -1 & -x \\ y & x & -\theta_3 \end{bmatrix} \begin{bmatrix} \delta_{x1} & \delta_{y1} & \delta_{z1} \\ \delta_{x2} & \delta_{y2} & \delta_{z2} \\ \delta_{x3} & \delta_{y3} & \delta_{z3} \end{bmatrix} \tag{8}$$

2.1.3. Chaos Synchronization

Chaos synchronization occurs when two (or more) chaotic systems (identical or non-identical) adapt a characteristic of their motion to the same behavior, owed to force or coupling. This includes trajectories and phase locking. Complete, projective, and antiphase synchronization have been explored [29]. These three synchronization types are usually of interest for master-slave configurations, i.e., two connected systems. However, for a more general case of networks, the less regular synchronization regimes such as multi-clustering and synchronization of groups of nodes are of relevance. See [30,31] for more details.

Complete synchronization means having equivalent state variables over time. Generalized synchronization for master-slave systems implies a functional relation between connected chaotic oscillators, $x_2(t) = F[x_1(t)]$.:

1. Complete Synchronization

Considering the following master and slave systems:

$$\dot{x} = \theta(x), \tag{9}$$

$$\dot{y} = \psi(y) + u(x, y), \tag{10}$$

State vectors x, $y \in \mathbb{R}$ is the vector controller for f, g: $\mathbb{R}^n \rightarrow \mathbb{R}^n$. The system error dynamics are:

$$e(t) = y(x) - x(t), \tag{11}$$

The systems are said to be in complete synchronization if:

$$\lim_{t \to \infty} \|e(t)\| = 0 \tag{12}$$

2. Anti-Phase Synchronization

In this type, given the same master-slave systems, the error dynamics for the systems are defined as:

$$e(t) = y(x) + x(t) \tag{13}$$

The systems are said to be in anti-synchronization if Equation (12) is satisfied.

3. Projective Synchronization

In this type, given the same master-slave systems, the error dynamics for the systems are defined as:

$$e(t) = y(x) - \alpha x(t) \tag{14}$$

where $\alpha \neq 0$ is the constant, called a scaling factor. The systems are said to be in projective synchronization if Equation (12) is satisfied. By setting appropriate values for α, synchronized systems may be scaled to desired levels and proportionally grow. Complete synchronization and anti-synchronization are specific examples of projective synchronization where $\alpha = 1$ and $\alpha = -1$. Greater mathematical complexity and chaos characterize the Lorenz map because of its higher dimension. As one-dimensional chaotic maps need fewer computing processes, they are better suited for applications that need to run with minimal latency. More basic chaotic maps, however, have serious security flaws. This shortcoming arises because of the restricted chaotic range, reduced chaotic complexity, and accelerated rate of degradation of dynamic behavior [32,33].

Several approaches for chaotic synchronization have been presented. Active nonlinear control and adaptive mode control have been widely employed for synchronization in recent literature [29]. Based on the Lyapunov stability theory, active nonlinear control has gained popularity in recent years. Adaptive control assumes that there is a controller

with a fixed structure and complexity for each potential plant parameter value, which can achieve the required performance with suitable controller parameter values. All these strategies are not applicable if the parameters of the chaotic system are unknown. Chaos control and synchronization focus on estimating the unknown parameters of chaotic dynamical systems. Parameter identification may be transformed into a multi-dimensional optimization problem using an objective function [34–36].

2.1.4. Chaotic Maps

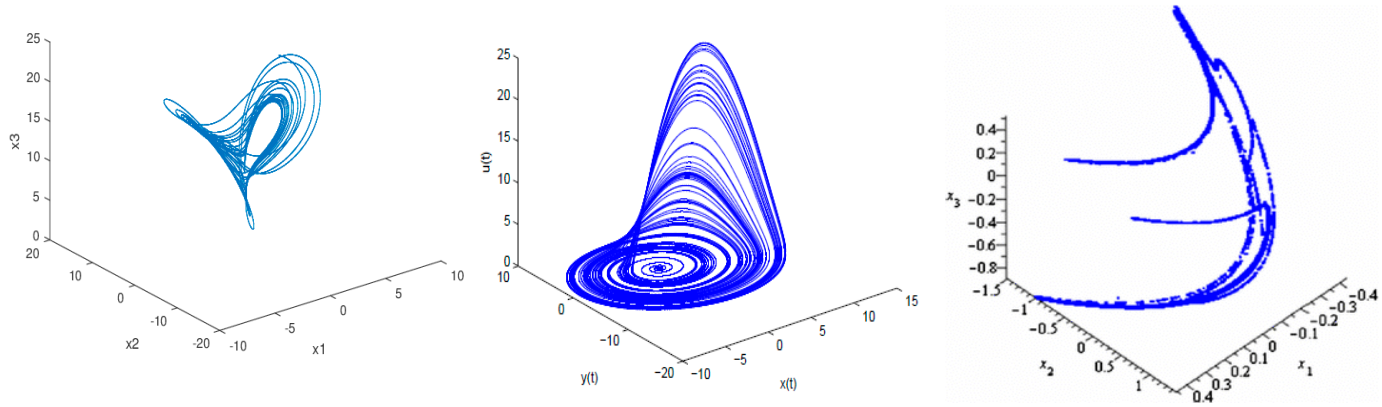
Chaotic maps are differential equations that describe chaotic discrete dynamics [18]. Chaos can only be detected in deterministic, continuous systems with a three-dimensional phase space or more. Low-dimensional chaotic systems are resource-efficient. The logistic map is a typical low-dimensional system [37]. Chaos is degenerative in these systems. It is hard to give the output sequence a long period. High-dimensional chaotic systems are more nonlinear. However, they have the drawbacks of excessive resource consumption and low-speed performance. Therefore, a large-period, high-dimensional, digital chaotic system with high speed and minimal resources is needed. Chen, Rossler, and Henon are 3D chaotic systems utilized in wireless communication [38].

1. Chen Chaotic System

Chen identified a classical chaotic attractor in a basic 3D system [38]:

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = (c - a)x - xz - cy \\ \dot{z} = xy + bz \end{cases} \tag{15}$$

x, y, and z are state variables, whereas a, b, and c are parameters. Chen chaotic-based encryption relies on secret keys. An invader cannot guess the wireless key. As Chen chaotic systems are sensitive to beginning circumstances and system characteristics, two near-initial conditions lead to diverse paths, as shown in Figure 1a.



(**a**) (**b**) (**c**)

Figure 1. 3D view of (**a**) Chen chaotic, (**b**) Rossler chaotic, and (**c**) Henon chaotic map.

2. Rossler Chaotic System

Rossler is a basic chaotic dynamical system with one non-linear term with standard system equations [39]:

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = x + ay \\ \dot{z} = b + xz - cz \end{cases} \tag{16}$$

x, y, and z are state variables; a and b are fixed; and c is the control parameter. Rossler attractor parameters are a = 0.2, b = 0.2, and c = 5.7. Figure 1b shows the Rossler chaotic attractor. This system is the minimum for continuous chaos for at least three reasons: (1) Its

phase space has minimal dimensions, (2) Nonlinearity is minimal because there is a single quadratic term, and (3) It generates a chaotic attractor with a single lobe, unlike the Lorenz attractor, which has two.

3. Henon Chaotic System

The Henon chaotic map is a chaotic discrete-time dynamical system. The map simplifies the Lorenz model's Poincare portion. The plane will either approach the Henon odd attractor or diverge to infinity.

$$\begin{cases} \dot{x} = a - (y^2 + bz) \\ \dot{y} = x \\ \dot{z} = y \end{cases}, \tag{17}$$

The Henon chaotic map parameters are a = 1.4 and b = 0.3. The conventional Henon map is chaotic (Figure 1c).

2.1.5. Quantum Fruit Fly Optimization Algorithm

Optimizing means picking the best element (based on some criteria) from a group of options or finding the least or maximum output for an experiment [34]. Heuristic methods are intelligent search strategies that speed up the process of obtaining a satisfying or near-optimal solution in bio-inspired procedures. A heuristic approach is simpler than an analytical one. However, precision is lost. Metaheuristics are iterative processes that help identify near-optimal solutions. Metaheuristics combine heuristic approaches to improve their performance [10,40]. Recent metaheuristic algorithms include the FOA [41]. FOA is inspired by fruit fly foraging. FOA has fewer adjusting parameters, less computational quantity, and offers great global search and convergence abilities. FOA is two-phased. The first step is smelling. In this phase, flies travel toward food by smelling it. Second phase begins when they are closer to the food supply: the vision stage. The fruit flies utilize their eyesight to come closer to the food. This phase repeats until the fruit fly eats the food. The steps of FOA include [42,43]:

(1) The random initial position of a fruit fly. *Init X_axis*; *Init Y_axis*.
(2) A fruit fly's sense of smell searches randomly for food.

$$\begin{cases} X_i = X_{axis} + Random\ Value\ R_1 \\ Y_i = Y_{axis} + Random\ Value\ R_2 \end{cases} \tag{18}$$

(3) As the food's location is unknown, the distance (*Dist*) to the origin is inferred before calculating the decision value of smell concentration (S).

$$\begin{cases} Dist_i = \sqrt{X_i^2 + Y_i^2} \\ S_i = \frac{1}{Dist_i} \end{cases} \tag{19}$$

(4) The smell concentration decision value (S) is inserted in the Fitness function to calculate the fruit fly's $Smell_i$.

$$Smell_i = Function(S_i) \tag{20}$$

(5) Determine the fruit fly swarm's strongest smell (seek for the maximum value)

$$[bestSmell \quad bestIndex] = \max(Smell) \tag{21}$$

(6) Using the best smell concentration and x, y coordinates, the fruit fly swarm flies to the position.

$$\begin{cases} Smellbest = bestSmell \\ X\_axis = X(bestindex) \\ Y\_axis = Y(bestindex) \end{cases} \tag{22}$$

(7) If the smell concentration is better than the previous iteration of smell concentration, execute Step 6.

Quantum theory assigns the fruit fly swarm to move in quantum space. The delta potential well model increases the uncertainty that fruit flies recognize and migrate to food. All quantum objects have wave-like features and may be in several locations at once; hence, they are characterized in quantum theory by the wave function (x, t), rather than by their position x and velocity v. A location's likelihood of hosting the item in quantum space is determined by the strength of the wave function at that location, as shown below in module form [15].

$$|\psi(x,t)|^2 dxdydz = Qdxdydz \tag{23}$$

$Qdxdydz$ is the object's probability of appearing at (x, y, z) at time t. Thus, $|\psi(x,t)|^2$ is the probability density function meeting the equation:

$$\int_{-\infty}^{+\infty} |\psi|^2 dxdydz = \int_{-\infty}^{+\infty} Qdxdydz = 1 \tag{24}$$

Schrödinger's equation describes object motion in quantum physics.

$$i\hbar \frac{\partial}{\partial t}\psi(X,t) = \hat{\mathrm{H}}\psi(X,t) \tag{25}$$

$$\hat{\mathrm{H}} = -\frac{\hbar}{2m}\nabla^2 + V(t) \tag{26}$$

$\hbar$ is the Planck Constant, $\hat{\mathrm{H}}$ is the Hamiltonian operator, m is the object mass, and $V(t)$ denotes the potential field of the object. Fruit flies search for food in the delta potential well, where they move in quantum space. Quantum behavior replaces fruit fly foraging and random search in quantum space. Both fruit fly smell and vision become more uncertain, increasing population diversity. One-dimensional space was used for simplicity. If food source location is x, its potential energy in the one-dimensional delta potential well is:

$$V(x) = -\gamma\delta(x - \rho_{axis}) = -\gamma\delta(y) \tag{27}$$

where the location of the fruit fly swarm, $\rho_{axis,}$ is in the center of the delta potential well. According to Schrödinger's equation, the following normalized wave function can be obtained:

$$\psi(y) = \frac{1}{\sqrt{L}} e^{-|y|/L} \tag{28}$$

L is the delta potential well length. Thus, the probability density function is:

$$Q(y) = |\psi(y)|^2 = \frac{1}{L} e^{-|y|/L} \tag{29}$$

This equals

$$y = \pm\frac{L}{2}\ln\frac{1}{u} \tag{30}$$

u is a random number (0, 1). Thus, we can determine the fruit fly's food source location:

$$x = \rho_{axis} \pm \frac{L}{2}\ln\frac{1}{u} \tag{31}$$

The model assumes that a 1D delta potential well is on each dimension at the swarm center attractor point, and osphresis-based search has quantum properties. The fruit fly's quantum-behaved foraging is shown by the wave function, not randomly. The employed QFOA model included swarm location initialization, osphresis-based search, and vision-based search. The employed QFOA model used quantum-behaved searching instead of random osphresis-based searching. In the osphresis-based search process, M_{osp}, new food

source locations (X_{axis}, Y_{axis}) were generated in the delta potential well. FOA's quantum-behaved searching mechanism is:

$$\begin{cases} X_i = X_{axis} \pm \frac{L_{x,i}}{2} \ln \frac{1}{r_x} \\ Y_i = Y_{axis} \pm \frac{L_{y,i}}{2} \ln \frac{1}{r_y} \end{cases} \tag{32}$$

where $i = 1, 2, \ldots, M_{osp}$, r_x and r_y are random [0, 1] values. $L_{x,i}$ and $L_{y,i}$ are delta potential well characteristic lengths of the corresponding dimension, determined by the fruit fly's last search location, based on their olfactory senses.

$$\begin{cases} L_{x,i} = 2b|X_{axis} - X_i| \\ L_{y,i} = 2b|Y_{axis} - Y_i| \end{cases} \tag{33}$$

i is the iteration number and b controls the quantum searching range.

$$b = b_1 \, logsig\left(10 \cdot \left(0.5 - \frac{g}{G_{max}}\right)\right) + b_2 \tag{34}$$

b_1 and b_2 restrict the value range to $b \in [b_2, b_1 + b_2]$.

2.2. Related Work

Several works on estimating chaotic system parameters have been recently published [13,44,45]. Real-world estimation is difficult for parameters of a complex 3D chaotic system. Most gradient-based methods are sensitive to initial conditions, trapping them in local minima. Estimating 3D chaotic parameters using soft computing techniques at a suitable cost function, is one solution, such as global optimization algorithms. Several cases of chaotic system parameter estimation using optimization algorithms have been reported [44,46,47]. The following section summarizes these algorithms.

Li et al. [35] combined the artificial bee colony algorithm (ABC) and differential evolution (DE) to estimate chaotic system parameters. Gao et al. [48] proposed chaos firefly optimization (CFA) for identifying Lorenz chaotic system parameters. Using chaotic search to update the standard Firefly algorithm improved optimization accuracy and speed. Recent pioneering work has combined the cuckoo search (CS) algorithm and orthogonal learning to estimate Lorenz and Chen chaotic system parameters [49]. He et al. also used particle swarm optimization (PSO) to estimate Lorenz system parameters [50]. This technique does not sufficiently explore the solution space. Small populations produce poor results. Li et al. [51] introduced the chaotic ant swarm (CAS) algorithm to determine chaotic system parameters.

Gholipour et al. [52] estimated chaotic system parameters with the artificial bee colony algorithm. Wei and Yu [53] presented a hybrid cuckoo search (HCS) algorithm inspired by differential evolution. The presented HCS offers two novel mutation strategies to fully exploit the neighborhood. Three chaotic systems with and without time delays were simulated and compared to other optimization methods to test HCS. Experimental results showed HCS's superiority in chaotic system parameter estimation due to its high calculation accuracy, fast convergence speed, and strong robustness. In [54], the authors introduced a two-stage estimation technique that combined the guaranteed approach and swarm intelligence.

Zhuang et al. [55] presented a new hybrid Jaya–Powell method for estimating the parameters of a Lorenz chaotic system. The proposed Jaya–Powell algorithm combines the Jaya algorithm, which seeks the relatively global optimum, with the Powell algorithm, which seeks the relatively local optimum, to provide a more precise and efficient estimate. This algorithm's searching technique makes it easier to strike a middle ground between exploration and exploitation throughout the optimization process. The suggested Jaya–Powell algorithm does not need the careful adjustment of appropriate parameters as it does not rely on any algorithm-specific parameters. Compared with seven benchmark

methods, the proposed hybrid Jaya–Powell algorithm provided more precise estimates and converged more quickly.

The work presented in [56] explored how to use several metaheuristic algorithms for the recognition of parameters in a fractional-order financial chaotic system. The algorithms that have been put into place are the ant colony optimizer, grey wolf optimizer, whale optimization algorithm, and artificial bee colony optimizer. As an objective function, mean square error was used to estimate the system's parameters. Zhang et al. [57] offered a novel method of parameter estimation that made use of numerical differentiation to streamline the preparation of observational data. Given the noisy observations on a subset of dependent variables, numerical differentiation may be used to approximately determine the values of the dependent variables and their derivatives. The parameter estimation issue may be simplified by substituting these approximations into the original system. The precision and efficiency of their technology are shown by numerical examples.

Encouraged by recent developments in data assimilation, Carlson et al. [58] built a dynamic learning technique to estimate missing parameters of a chaotic system using just a subset of available data. The authors convincingly proved, under plausible assumptions that this approach converged to the right parameters when the system under issue was the standard three-dimensional Lorenz system. They computationally showed the effectiveness of this technique on the Lorenz system by recovering any correct subset of the three non-dimensional parameters of the system, provided that an appropriate subset of the state was observable. Over the last two decades, studies on how to synchronize a Lorenz chaotic system have been more prominent. Model reference adaptive control (MRAC) synchronization scheme design has been the primary focus of the majority of the research. For this problem, C. Peng, and Y. Li [59] suggested two system identification strategies. The observer–Kalman filter identification method was the first method used. The second kind of discretization was the bilinear transform. The new approach significantly improved the accuracy of the discovered parameters, which were therefore already very near to actual values.

Rizk-Allah et al. [60] presented a unique approach to parameter estimation for the chaotic Lorenz system, using a modified form of particle swarm optimization (PSO). The suggested technique, a memory-based particle swarm optimization (MbPSO) algorithm, modeled the parameter estimation of the Lorenz system as a multidimensional issue. To change the population's orientation and improve search efficiency, MbPSO added two additional variables to the classic PSO. The results showed that the suggested algorithm performed much better than the original PSO, when particle memories were linked to those of other particles. The primary goal of the study [61] was to apply a deep learning technique to the problem of estimating the parameters of chaotic systems, such as the Lorenz system. In this research, the authors used the k-means technique to build out the workflow of a deep neural network (DNN)-based approach. The DNN approach works well for difficult, nonlinear problems. Using the proposed approach, 98% of correct training data and 73% of test data were predicted.

The parameter identification for the discrete memristive chaotic map was the primary topic of the research presented by Peng et al. [62], in which a novel intelligent optimization technique called the adaptive differential evolution algorithm was suggested. To handle the hyperchaotic and attractors that coexist in the investigated discrete memristive chaotic maps, the identification objective function had two unique components: time sequences and return maps. It was shown via numerical simulations that the suggested approach outperformed the other six existing algorithms and maintained the ability to correctly identify the original system's properties, even when subjected to noise interference.

Although chaotic system parameter estimation has been studied for decades, it can still be improved. According to the review, past studies focused on: (1) Estimating a single chaotic system parameter and (2) Not addressing the best optimization technique for exploration and exploitation in a unified framework. Most bio-inspired optimization techniques for chaotic system parameter estimation combine two or more algorithms to

improve exploration and exploitation. To the best of our knowledge, little attention has been paid to developing a bio-inspired parameter estimation technique for a chaotic system with few training samples.

3. Materials and Methods

Let $\dot{X} = F(X, X_0, \theta_0)$ be a continuous nonlinear chaotic system, where $X = (x_1, x_2, \ldots, x_N)' \in \mathbb{R}^n$ is the chaotic system's state vector, $\dot{X}$ is X's derivative, the resulting solution is parameterized by the initial value X_0, and $\theta_0 = (\theta_{1,0}, \theta_{2,0}, \ldots, \theta_{d,0})'$ are the original parameters. If the system's structure is known, the estimated system may be expressed as $\dot{\widetilde{X}} = F(\widetilde{X}, X_0, \widetilde{\theta})$, where $\widetilde{X} = (\widetilde{x}_1, \widetilde{x}_2, \ldots, \widetilde{x}_N)' \in \mathbb{R}^n$ is the state vector and $\widetilde{\theta} = \left(\widetilde{\theta}_1, \widetilde{\theta}_2, \ldots, \widetilde{\theta}_d\right)'$ is a collection of estimated parameters. Based on X, the fitness function is [49,51]:

$$f\left(\widetilde{\theta}_i^n\right) = \sum_{i=0}^{W}\left[(x_1(t) - \widetilde{x}_{i,t}^n(t))^2 + \ldots + (X_N(t) - \widetilde{x}_{i,N}^n(t))^2\right], \quad (35)$$

where $t = 0, 1, \ldots \ldots W$ and i is the ith state vector. Estimating chaotic system parameters aim to reduce fitness function by minimizing $\widetilde{\theta}_i^n$. Dynamic instability makes chaotic systems difficult to estimate. Due to the problem's many variables and various local search optima, typical optimization in the local optima is difficult [63,64].

A chaos communication system comprises of transmitter, receiver, and channel (noise) performance. In the transmitter, the modulation methods utilized to combine the message signal and chaotic carrier are crucial for system security. As a signal must be sent to the receiver, there is a possibility that intruders may receive the signal. Even if intruders do not know the structure or parameters of a chaotic system, they may use signal processing or sophisticated algorithms to extract the message from the transmitter signal. In chaotic masking, the signal is directly added to the chaotic signal; thus, the fluctuation may be recognized by non-linear dynamic forecasting techniques or power spectrum analysis, if the message amplitude/frequency is high enough. Mixing the message should remove any pattern or information from the sent signal. The carrier chaotic signal will be distorted by channel noise before reaching the receiver. Message recovery requires chaotic synchronization at the receiver. Demodulation is an issue in chaotic communication systems. The recommended solution uses a few signal samples instead of large samples that need more calculation. The communication channel is assumed to be free noise, as the emphasis is on estimating the chaotic system's unknown parameters, not channel attacks.

As discussed later, in a quantum model of FOA, each fruit fly represents a particle that has a state depicted by a wave function, instead of position and velocity. The dynamic behavior of the fruit fly is different from that of the fruit fly in standard FOA algorithms; that is, the accurate values of x and v cannot be simultaneously calculated. Its searching performance is better than the original particle swarm optimization algorithm. The quantum particle swarm optimization algorithm is a global convergence guarantee algorithm. The capabilities of a QFOA algorithm to enhance convergence speed and low optimization accuracy were achieved through: (1) A mutation operator to increase the diversity of particles in a population (the delta potential well concept to speed up the convergence speed); (2) An operator based on evolutionary generations to update a contraction expansion coefficient (objective or fitness function for global optimization); (3) An elitist strategy to remain the strong particles.

3.1. At the Transmitter Side

The original signal was hidden using a known 3D Lorenz chaotic signal. Lorenz used $\theta_1 = 10$, $\theta_2 = 28$, and $\theta_3 = 8/3$. This system shows chaotic behavior [65]. Three phases applied chaotic masking. First, we used the fourth-order Runge–Kutta (RK4) to solve the 3D Lorenz chaotic system equation to create the chaotic signal. RK4 examines iterative

steps in four places [66,67]. Runge–Kutta was run three times for each point in phase space with $h = 0.01$ [49–52].

$$\vec{k_2} = h\vec{f}\left(\vec{x}_c + \frac{1}{2}\vec{k_1}\right), \tag{36}$$

$$\vec{k_3} = h\vec{f}\left(\vec{x}_c + \frac{1}{2}\vec{k_2}\right), \tag{37}$$

$$\vec{k_4} = h\vec{f}\left(\vec{x}_c + \frac{1}{2}\vec{k_3}\right), \tag{38}$$

$$\vec{x}_c(t_0 + h) = \vec{x}_c(t_0) + (\vec{k_1} + 2\vec{k_2} + 2\vec{k_3} + \vec{k_4}), \tag{39}$$

$$\vec{k_1} = h\vec{f}\left(\vec{x}_c\right), \tag{40}$$

$$\vec{x}_c(t_0 + h) = \vec{x}_c(t_0) + (\vec{k_1} + 2\vec{k_2} + 2\vec{k_3} + \vec{k_4}), \tag{41}$$

$$\vec{k_1} = \begin{bmatrix} k_{1x} \\ k_{1y} \\ k_{1z} \end{bmatrix} = h\vec{f}\left(\vec{x}_c\right) = h\vec{f}\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = h\vec{f}\begin{bmatrix} x_{c0} \\ y_{c0} \\ z_{c0} \end{bmatrix} = h\begin{bmatrix} \theta_1(y_{c0} - x_{c0}) \\ \theta_2 x_{c0} - y_{c0} - x_{c0}z_{c0} \\ -\theta_3 z_{c0} + x_{c0}y_{c0} \end{bmatrix} \tag{42}$$

$$\vec{k_2} = \begin{bmatrix} k_{2x} \\ k_{2y} \\ k_{2z} \end{bmatrix} = h\vec{f}\left(\vec{x}_c + \frac{1}{2}\vec{k_1}\right) = h\vec{f}\left(\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}_{t=0} + \frac{1}{2}\begin{bmatrix} x_{1x} \\ y_{1y} \\ z_{1z} \end{bmatrix}\right) \tag{43}$$

$$\vec{k_3} = \begin{bmatrix} k_{3x} \\ k_{3y} \\ k_{3z} \end{bmatrix} = h\vec{f}\left(\vec{x}_c + \frac{1}{2}\vec{k_2}\right) = h\vec{f}\left(\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}_{t=0} + \frac{1}{2}\begin{bmatrix} x_{2x} \\ y_{2y} \\ z_{2z} \end{bmatrix}\right) \tag{44}$$

$$\vec{k_4} = \begin{bmatrix} k_{4x} \\ k_{4y} \\ k_{4z} \end{bmatrix} = h\vec{f}\left(\vec{x}_c + \frac{1}{2}\vec{k_3}\right) = h\vec{f}\left(\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix}_{t=0} + \frac{1}{2}\begin{bmatrix} x_{3x} \\ y_{3y} \\ z_{3z} \end{bmatrix}\right) \tag{45}$$

$$\vec{c}\,(t) = \begin{bmatrix} k_{4x} \\ k_{4y} \\ k_{4z} \end{bmatrix} = \begin{bmatrix} x_{c0} \\ y_{c0} \\ z_{c0} \end{bmatrix} + \frac{1}{6}(\vec{k_1} + 2\vec{k_2} + 2\vec{k_3} + \vec{k_4}) \tag{46}$$

The second stage involved sampling the original input to create a discrete signal or accumulating an analogue or continuous signal [47]. Sampling is described by the following arithmetic statement, where $\delta(t)$ represents the impulse train of period Ts [68]:

$$Sampled\ Signal\ x_s(t) = x(t) \cdot \delta(t) \tag{47}$$

$$\delta(t) = a_0 + \sum_{n=1}^{\infty}(a_n \cos(nw_s t) + b_n \sin(nw_s t)) \tag{48}$$

$$a_0 = \frac{1}{T_s}\int_{\frac{-T}{2}}^{\frac{T}{2}} \delta(t)dt = \frac{1}{T_s}\delta(0) = \frac{1}{T_s} \tag{49}$$

$$a_n = \frac{2}{T_s}\int_{\frac{-T}{2}}^{\frac{T}{2}} \delta(t)\cos(nw_s)dt = \frac{2}{T_s}\delta(0) = \frac{1}{T_s}\cos(nw_s 0) = \frac{2}{T_s} \tag{50}$$

$$b_n = \frac{2}{T_s}\int_{\frac{-T}{2}}^{\frac{T}{2}} \delta(t)\sin(nw_s)dt = \frac{2}{T_s}\delta(0) = \frac{1}{T_s}\sin(nw_s 0) = 0 \tag{51}$$

$$\delta(t) = \frac{1}{T_s}\sum_{n=1}^{\infty}\frac{2}{T_s}\delta(t)\cos(nw_s t) + 0) \tag{52}$$

$$x_s(t) = x(t)[\frac{1}{T_s} + \sum_{n=1}^{\infty}\left(\frac{2}{T_s}\cos(n_s t)\right) + 0)] = \frac{1}{T_s}[x(t) + 2\sum_{n=1}^{\infty}(\cos(nw_s t))x(t)] \tag{53}$$

$$x_s(t) = \frac{1}{T_s}[x(t) + 2\cos(\mathrm{n}w_s t).x(t) + 2\cos(n2w_s t).x(t) + 2\cos(n3w_s t).x(t) + \ldots] \quad (54)$$

After sampling the original signal, downsampling reduced the signal's sampling rate by M. When a signal is downsampled, only every Mth sample is taken and all others are discarded. Downsampling balances a dataset by matching the majority class (3D original signal) with minority class samples (3D chaotic signal). In the third stage, the downsampled original signal $\vec{x}_d(t)$ was added, or masked, to the chaotic oscillator output at the transmitter before transmission. The transmitter is represented as follows:

$$\vec{c}(t) = K(\vec{x}_c(t)), \quad (55)$$

$\vec{c}(t)$ is the chaotic system's output after applying RK4. $\vec{x}_m$ is formed by adding $\vec{c}(t)$ to $\vec{x}_d(t)$

$$\vec{x_m}(t) = \vec{c}(t) + \vec{x_d}(t) \quad (56)$$

$$\vec{x}_m = \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix}, \quad \vec{x}_m = \begin{bmatrix} x_d \\ y_d \\ z_d \end{bmatrix} \quad (57)$$

Before transmitting the signal via the channel, upsampling and interpolation were used to rebuild it. The upsampling procedure increases the sampling rate by an integer factor M (interpolation factor) by adding M-1 evenly spaced zeroes between each pair of samples. Mathematically, upsampling is provided by the following equations, where l = 0, $\pm 1, \pm 2, \ldots$. The impulse train $[n]$ represents the sampling function.

$$x_U[n] = \begin{cases} x_m\left[\frac{n}{M}\right]. & n = 0, \pm M, \pm 2M, \ldots \\ 0 & otherwise \end{cases} \quad (58)$$

$$x_U[n] = x_m[n]p[n] = \sum_{J=-\infty}^{+\infty} x_m[l]\delta[n - lM] \quad (59)$$

$$p[n] = \sum_{\tau=-8}^{+\infty} \delta[n - lM] \quad (60)$$

After upsampling, interpolation was used to create new data points within a specified range. If the sampling instants are near enough, the signal can be accurately recreated by low-pass filter interpolation. Low-pass filtering $x_U[n]$ reconstructs $x_m[n]$. The interpolated signal $x_T[n]$ is calculated as [69]:

$$x_T[N] = x_U[n] * h[n] \quad (61)$$

$h[n]$ denotes the impulse response of the low-pass filter:

$$h[n] = \frac{M\Omega_C}{2\pi} sinc\left(\frac{n\Omega_C}{\pi}\right) \quad (62)$$

Ω_C is the cutoff frequency of the discrete time filter. So, the equivalent interpolation formula can be written as:

$$x_T[n] = \sum_{J=-\infty}^{+\infty} x_m[lM]h_T[n - lM] \quad (63)$$

$$x_T[n - lM] = \frac{M\Omega_C}{2\pi} sinc\left[\frac{\Omega_C}{\pi}(n - lM)\right] \quad (64)$$

$h[n]$ is the impulse response of the interpolating filter. The interpolation using the *sinc* function is commonly referred to as band limited interpolation.

3.2. On the Receiver Side

On the receiver side, the received signal (masked original signal) was downsampled. To use chaotic communications, two identical chaotic oscillators were needed in the trans-

mitter (or master) and receiver (or slave). Unknown receiver-side parameters $(\widetilde{\theta}_1, \widetilde{\theta}_2, \widetilde{\theta}_3)$ needed to be approximated. The quantum fruit fly optimization algorithm (QFOA) estimates the 3D Lorenz chaotic system's unknown parameters. The fundamental QFOA includes a setup step and a cycle of smelling, evaluating, and flocking [15,43,70]. The QFOA control parameters were set, including the maximum number of generations and population size, and the fruit fly swarm's location was randomized. As the original FOA can only solve continuous optimization issues, it was adapted to tackle synchronization in chaos-based communication networks. Each fly picked randomly from the search space group, including $\widetilde{\theta}_1$, $\widetilde{\theta}_2$, and $\widetilde{\theta}_3$. As stated in [71], the search space for unknown chaotic system parameters is [9 11], [20 30], and [2 3]. Given these initial answers, QFOA repeated the following steps [72]:

- These solutions were input to a predefined chaotic receiver system. The RK4 was used in the 3D Lorenz equations to create chaotic signals (one for each fruit fly).
- Each fly determined food concentration using the mean square error between the predicted chaotic signal and the downsampled received signal (smelling process).
- Each fly shared its position with others. The flies compared their solutions to choose the best one.
- Flies migrated to the solution with the lowest fitness value, which became the new best solution (vision process).

This stage outputs the 3D Lorenz chaotic system's ideal parameters. The second stage of synchronization used these characteristics as inputs. RK4 was used again to create the estimated 3D chaotic signal. The third stage received this estimated signal and the downsampled received signal. We then subtracted the two signals to get the sampled signal. The original signal was reconstructed using upsampling and interpolation. Perfect synchronization is key to reconstructing the original signal. Table 1 provides the link between QFOA parameters and the parameters estimation problem of the chaotic system.

Table 1. The link between QFOA parameters and the parameters estimation problem of the chaotic system.

QFOA Parameters	Chaotic Synchronization Problem
Number of iterations	The search process's best solution iteration count.
Number of swarms m	m = 25.
Initial location	The initial solution is randomly selected from each parameter's search space.
Smell concentration	Mean square error (Objective or fitness function).
Vision	Smell concentration-based parameter selection.

4. Results

This section analyses the model's efficiency. Experiments were performed to test the model's reliability in estimating chaotic system parameters. The suggested approach optimized synchronization with the Lorenz chaotic system and speech signal. The 20 to 30 dB weaker speech signal was combined with the chaotic mask signal to create a broadcast signal. Table 2 shows the experiment's algorithm settings. The recommended model was implemented in MATLAB R2017b (9.3.0.713579) 64-bit. The model was constructed using a laptop with an Intel (R), Core (TM) i5-8250U CPU@ 1.60 GHZ @ 1.80 GHz, 8 GB, and 64-bit operating system, with a x64 processor.

In the proposed chaotic parameters estimate model, various statistical parameters were employed to evaluate model performance. These evaluations included [45] the mean (average) of best fitness values and standard deviations. For a robust model, these means (mean of best fitness) needed to be as low as possible, where optimum fitness quantifies the difference between estimated and sent signals. Standard deviations (Std.) shows how measurements for a group are spread apart from the average (mean) or anticipated value. A low standard deviation suggests that most data points are near to the mean (more reliable). A large standard deviation suggests the data points are widely scattered (less reliable).

Table 2. The parameters of the optimization algorithms (Reference parameters collected from previous studies).

Algorithm	Parameters	Values
Fruit fly (FOA)	Number of swarms	25
	Maximum number of iterations	50
Cuckoo search (CS)	Number of swarms	25
	Probability rate	0.20
	Maximum number of iterations	50
Practical swarm (PSO)	Number of swarms	25
	Inertia weight	0.8
	Acceleration coefficient	1.5
	Maximum number of iterations	50
Genetic algorithm (GA)	Number of swarms	25
	Crossover rate	0.7
	Mutation rate	0.3
	Maximum number of iterations	50
Firefly algorithm (FA)	Number of swarm	25
	Initial brightness of each fly	1
	Absorption coefficient of light	1
	Step size (α)	1
	Maximum number of iterations	50

4.1. Experiment 1: Comparison with Existing Methods

The first batch of tests compared the proposed model to comparable techniques [44] that used the GA, PSO, and CS to find the 3D Lorenz chaotic system characteristics solely using chaotic signals. Default swarm parameters were utilized. Table 3 shows that the suggested model is superior to the prior techniques. QFOA's calculated parameters matched the original parameters' real values. According to [49], the 3D Lorenz chaotic system's initial parameters are $\theta_1 = 10$, $\theta_2 = 28$, and $\theta_3 = 8/3$, allowing complete synchronization between the master and slave chaotic systems. The estimated parameters matched the CS-based model, but the QFOA outperformed in terms of the optimal function's mean and standard deviation. Most data points were close to the mean with a low standard deviation (more reliable). QFOA was more effective and resilient than other chaotic system parameter estimation strategies. The model and system responses were synchronized. This gain was due to the proposed model's higher coverage and exploration of the searching space, which improved parameter estimate accuracy and led to the discovery of optimum chaotic parameter values compared with existing techniques.

Table 3. Comparison of statistical results for the Lorenz system, in case of only using chaotic signal.

Models	Means of the Best Fitness	Std. Dev. of the Best Fitness	θ_1	θ_2	θ_3
QFOA	9.53×10^{-9}	5.83×10^{-9}	10.00	28.00	2.6666
CS	1.71×10^{-4}	1.69×10^{-4}	10.00	28.00	2.6664
PSO	0.118	0.269	9.998	27.99	2.6665
GA	1.332	2.784	10.027	28.01	2.6691

4.2. Experiment 2: Effect of QFOA Iteration

The second set of experiments investigated the effect of the QFOA number of iterations on the proposed model to identify the correct parameters of the 3D Lorenz chaotic system using only chaotic signals and masking voice signals with the chaotic signals. QFOA was performed 30 times every iteration, with 50 iterations total and $W = 30$ for data sampling. Default swarms were utilized. After 20 iteration, the parameters θ_1, θ_2, and θ_3 converge to the actual values. QFOA reached stable values in 25 iterations. As the fitness function value

declines rapidly to zero, indicating that QFOA may converge quickly to the global optimum. These few iterations did not require complex calculations. By adjusting the location of the QFOA swarms by modifying the number of iterations, the algorithm could reach an ideal balance between exploitation and exploration. At the same time, elitism in population iteration may have sped up the convergence and assured continual optimization. This highlights the remarkable efficiency of QFOA in accomplishing global optimization.

4.3. Experiment 3: Effect of Number of Swarms

The third set of experiments was implemented to find a suitable number of QFOA swarms that helped to reduce computational effort without sacrificing estimation precision. For the three-dimensional Lorenz system, the proposed model was run by setting the QFOA swarm numbers as 10, 30, and 100, respectively. In general, tiny populations provide poor outcomes. As the population grows, outcomes improve, but more fitness tests are needed. Beyond a certain point, outcomes are not significantly influenced. When there are too few swarms, the solution space is not sufficiently searched, resulting in unsatisfactory outcomes. Considering search quality and computational effort, a population size between 30 and 60 is suggested. A larger population size is suggested for estimating additional parameters. Size 25 performed well. Considering processing costs and estimating accuracy, a large population size is unnecessary.

4.4. Experiment 4: Influence of the Data Sampling W

The fourth series of experiments tested how data sampling affected model accuracy. To reduce the amount of parameter setting combinations, the model changed one parameter *W* at a time, while leaving other parameters (number of swarms, number of iterations, etc.) at default values. The impact of modifying these variables was also considered. General factors for selecting *W* were minimum fitness mean and highest estimate accuracy. All scenarios were run 30 times for comparison. Table 4 lists the estimation results and the means of the best fitness values for different data sampling *W*. As shown, the estimation accuracy declined as *W* increased. Moving from 30 samples to 100 decreased the mean of fitness values by 36%, whereas moving from 100 samples to 200 decreased the mean of fitness values by 45%. These three groups of input data may have provided a satisfactory estimate, but the 30 samples of data had the least variation. Different inputs impacted the first iteration, but for all instances, it took roughly 25 iterations for the algorithm to converge to zero, indicating these three conditions could all acquire quite accurate anticipated outcomes. As expected, chaotic parameter estimate accuracy falls as *W* rises. The crucial sensitivity of the nonlinear system to starting circumstances and parameters made the fitness function more difficult as *W* increased. To decrease estimate bias in target nonlinear systems, it is vital to sample enough data.

Table 4. Statistical results for the extended Lorenz chaotic system with varied data sampling.

Number of Samples	Means of the Best Fitness	θ_1	θ_2	θ_3
$W = 30$	9.45×10^{-9}	10.00	28.000	2.6667
$W = 100$	1.49×10^{-8}	10.00	27.998	2.6666
$W = 200$	2.18×10^{-8}	9.99	27.997	2.6666

4.5. Experiment 5: Comparison with another Quantum Metaheuristic Algorithm

The fifth series of tests compared the proposed model with a comparable strategy that used the quantum firefly (QFA) algorithm to determine the ideal chaotic parameters of the 3D Lorenz chaotic system exclusively using chaotic signal and masking speech sounds with chaotic signal. Both techniques were performed 30 times to compare fitness means and standard deviations. Default swarms were utilized. Table 5 shows that the estimated chaotic parameters while masking speech signals with chaotic signals are similar to the

QFA-based model. The mean fitness values and standard deviations of QFOA were 37 and 66% lower than in QFA.

Table 5. Statistical results for the Lorenz system.

	Model	Means of the Best Fitness	Std. Dev. of the Best Fitness	θ_1	θ_2	θ_3
Masking voice signal with chaotic signal	QFOA	1.04×10^{-8}	6.27×10^{-9}	10.00	28.00	2.6667
	QFA	1.61×10^{-8}	1.85×10^{-8}	10.00	27.99	2.6666
Chaotic only	QFOA	9.53×10^{-9}	5.79×10^{-9}	10.00	28.00	2.6667
	QFA	1.42×10^{-8}	1.18×10^{-8}	10.00	28.00	2.6667

In general, the quantum-inspired firefly algorithm (QFA) ensured the diversification of firefly-based generated solution sets, using the superstitions quantum states of the quantum computing concept. However, it suffered from premature convergence and stagnation; this was mainly dependent on the ability of the employed potential field to handle movement uncertainty. The suggested QFOA algorithm, inspired by the delta potential field, presented the most balanced computational performance in terms of exploitation (accuracy and precision) and exploration (convergence speed, and acceleration). The advantage of such models, on the one hand, is that they are "exactly solvable", e.g., the spectrum and eigenvectors are explicitly known; on the other hand, many interesting physical features are retained, despite the simplification involved in approximating short-range with zero-range. Thus, QFOA was more effective and resilient than QFA in estimating chaotic parameters.

4.6. Experiment 6: Estimation Accuracy with Different Chaotic Systems

The sixth group of experiments was conducted to determine the efficiency of the proposed model regarding the different chaotic systems, including the 3D Chen and 3D Rossler chaotic systems in cases of only using the chaotic signal. The algorithm was run 30 times and the default parameters of QFOA were used. Table 6 shows that the estimated parameters derived by QFOA were close to the original parameters for chaotic systems. As stated in [44], the original parameters of 3D Chen chaotic system were $\theta_1 = 35$, $\theta_2 = 3$, and $\theta_3 = 28$; whereas, as stated in [73], the original parameters of 3D Rossler chaotic system were $\theta_1 = 0.2$, $\theta_2 = 0.4$, and $\theta_3 = 5.7$, through which perfect synchronization could be obtained between the master and slave chaotic systems. In the search process, fruit flies modified their places based on individual and swarm experiences. This expanded the solution search space and prevented premature convergence. This also improved the algorithm's convergence speed. Generalized synchronization was possible with certain parameters [74].

Table 6. Estimation accuracy for different chaotic system using default QFOA parameters.

Chaotic Systems	θ_1	θ_2	θ_3
Lorenz	10.000	28.0000	2.6667
Chen	35.000	2.9999	27.999
Rossler	0.2000	0.3999	5.6999

Computer simulations of the three 3D chaotic systems and comparisons with other metaheuristic approaches proved the suggested method's efficiency. The impact of data sampling, iterations, and swarms on estimating accuracy was also studied. Theoretical study and computer simulation led to the following conclusions: (1) A shorter data sample length improves estimate accuracy because a longer sample length complicates the objective function. (2) The highest number of iterations improves estimating accuracy by moving

the swarms. Thus, exploitation and exploration balance each other. (3) Many swarms will investigate enough space for study, improving estimate accuracy. These swarms are computationally intensive. To decrease estimate bias in chaotic systems, use the right data sampling, iterations, and swarms.

For our simulations, we used some of the most famous chaotic systems as examples. The number of parameters for these chaotic systems was not large, and the system was not complex. At present, the most studied chaotic neural network systems have many parameters, and the weight of these systems affects the complexity of the network. However, the suggested simpler model may be adapted to deal with chaotic neural network systems and other complicated chaotic systems. In our case, instead of searching for only three chaotic parameters, which represented the final solution picked from the search space based on a quantum-inspired particle's movement, more parameters could be correctly estimated by increasing the number of fruit flies. Therefore, there is a trade-off between computational cost and required best fitness evaluation function that must be balanced.

4.7. Industrial Application Case: Financial Chaotic System

Due to the nonlinear nature of the financial markets, chaos models using nonlinear dynamics have been a popular topic in recent years. Uncertainty in the market environment has a particularly negative impact on the financial system. Therefore, describing the financial chaos model with random elements is more practical. Due to deterministic instability, financial chaos, such as the extreme turbulence of the financial market and the financial crisis, occurs during the functioning of the financial system, which has significant detrimental effects on economic development and social stability. Controlling the financial system from a chaotic to a periodic state is as simple as modifying the controller settings. As a first step, we theoretically obtained a range of values for the controller parameters by analyzing the financial system's dynamic equations and controllers. Later, we investigated the effects of these parameters on the system.

5. Conclusions

Chaotic synchronization is key for chaotic signals in a communication system. On the receiver end, the chaotic system's parameters are unknown; thus, the task is to determine the ideal values to retrieve the message signal. Using the fruit fly optimization technique, this article improved chaotic synchronization in chaos-based wireless networks. In this study, parameter estimation for a three-dimensional Lorenz chaotic system was set up as a multi-dimensional optimization problem and solved using the quantum fruit fly optimization method. Quantum theory was employed by the FOA model and replaced the osphresis-based search of FOA with a quantum behavior-based searching mechanism. The quantum fruit fly optimization technique improved parameter estimation accuracy by carefully exploiting the search space and converging, which suggested that the algorithm could estimate optimum parameter values. Furthermore, it enhanced the exploration of optimal solutions by sharing information regarding parameter values. The difference between the proposed model and existing metaheuristic algorithms was the use of fruit fly optimization to produce better quality solutions and convergence speed, i.e., establishing an optimal trade-off between exploration and exploitation. This model may be extended to other chaotic systems.

The results and discussion of this study led to the following conclusions (important results): (1) Numerical simulations indicate the proposed approach can accurately predict chaotic system parameters. The suggested model is faster and more accurate than current techniques. This outcome is due to balancing exploitation and exploration in the search space. (2) Even with the original signal added to the chaotic signal, the current algorithm can still identify it well, especially for the Lorenz system. (3) As with final estimated results, 30 samples of data has the highest accuracy and least variation, proving that the amount of input data affects algorithm stability.

For future work, the proposed model should be applied to different chaotic systems, such as in high-dimensional, hyper chaotic systems, and time-delay chaotic systems. Implementation and testing in a real testbed are important in the field of wireless communication. Real deployment tests can bring up issues that did not come up in simulation. To work well in real implementations, changes to the proposed model may be required.

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

Funding: This research received no external funding.

Institutional Review Board Statement: The study did not require ethical approval.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study did not report any data.

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

References

1. Shukla, P.; Khare, A.; Rizvi, M.; Stalin, S.; Kumar, S. Applied Cryptography Using Chaos Function for Fast Digital Logic-Based Systems in Ubiquitous Computing. *Entropy* **2015**, *17*, 1387–1410. [CrossRef]
2. Sadkhan, S.; Al-Sherbaz, A.; Mohammed, R. Chaos based Cryptography for Voice Encryption in Wireless Communication. In Proceedings of the First International Conference of Electrical, Communication, Computer, Power and Control Engineering, Mosul, Iraq, 17–18 December 2013; pp. 191–197.
3. Mondal, B.; Mandal, T. A Multilevel Security Scheme using Chaos based Encryption and Steganography for Secure Audio Communication. *Int. J. Res. Eng. Technol.* **2013**, *2*, 399–403.
4. Fadhel, S.; Shafry, M.; Farook, O. Chaos Image Encryption Methods: A Survey Study. *Bull. Electr. Eng. Inform.* **2017**, *6*, 99–104. [CrossRef]
5. Pecora, L.; Carroll, T. Synchronization of Chaotic Systems Chaos: An Interdisciplinary. *J. Nonlinear Sci.* **2015**, *25*, 097611. [CrossRef]
6. Zhang, H. Chaos Synchronization and Its Application to Secure Communication. Ph.D. Dissertation, Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, USA, 2010.
7. Shewale, G.; Shinde, N.; Shirode, A.; Singh, S.; Solanki, J.; Tajane, M.; Tripathi, G. *Chaos Theory Technical Report-60*; Sardar Patel Institute of Technology: Mumbai, India, 2012; pp. 1–39.
8. Yau, H.; Pu, Y.; Li, S. An FPGA-Based PID Controller Design for Chaos Synchronization by Evolutionary Programming. *Discret. Dyn. Nat. Soc.* **2011**, *2011*, 516031. [CrossRef]
9. Rai, D.; Tyagi, K. Bio-Inspired Optimization Techniques—A Critical Comparative Study. *ACM SIGSOFT Softw. Eng. Notes* **2013**, *38*, 1–7. [CrossRef]
10. Wahab, M.; Meziani, S.; Atyabi, A. A Comprehensive Review of Swarm Optimization Algorithms. *PLoS ONE* **2015**, *10*, e0122827. [CrossRef]
11. Shan, D.; Cao, G.; Dong, H. LGMS-FOA: An Improved Fruit Fly Optimization Algorithm for Solving Optimization Problems. *Math. Probl. Eng.* **2013**, *2013*, 108768. [CrossRef]
12. Ding, S.; Zhang, X.; Yu, J. Twin Support Vector Machines based on Fruit Fly Optimization Algorithm. *Int. J. Mach. Learn. Cybern.* **2016**, *7*, 193–203. [CrossRef]
13. Li, S.; Xu, W.; Li, R.; Zhao, X. A General Method for Chaos Synchronization and Parameters Estimation between Different Systems. *J. Sound Vib.* **2007**, *302*, 777–788. [CrossRef]
14. Niu, J.; Zhong, W.; Liang, Y.; Luo, N.; Qian, F. Fruit Fly Optimization Algorithm based on Differential Evolution and Its Application on Gasification Process Operation Optimization. *Knowl. Based Syst.* **2015**, *88*, 253–263. [CrossRef]
15. Zhang, X.; Xia, S.; Li, X. Quantum Behavior-Based Enhanced Fruit Fly Optimization Algorithm with Application to UAV Path Planning. *Int. J. Comput. Intell. Syst.* **2020**, *13*, 1315–1331. [CrossRef]
16. Abdo, A.H. Optimized Chaotic Parameters Estimation Algorithm to Enhance the Synchronization of Wireless Communication Networks. Master Thesis, Alexandria University, Alexandria, Egypt, 2019.
17. Cattani, M.; Caldas, I.; Souza, S.; Iarosz, K. Deterministic Chaos Theory: Basic Concepts. *Rev. Bras. Ensino Física* **2017**, *39*, 1315. [CrossRef]
18. Fradkov, A.; Evans, R.; Andrievsky, B. Control of Chaos: Methods and Applications in Mechanics Philosophical Transactions of the Royal Society A: Mathematical. *Phys. Eng. Sci.* **2006**, *364*, 2279–2307.
19. Gauthier, Y. The Construction of Chaos Theory. *Found. Sci.* **2009**, *14*, 153–165. [CrossRef]

20. Ren, H.; Baptista, M.; Grebogi, C. Wireless Communication with Chaos. *Phys. Rev. Lett.* **2013**, *110*, 2–5. [CrossRef]
21. Kamil, I.; Fakolujo, O. Lorenz-Based Chaotic Secure Communication Schemes. *Ubiquitous Comput. Commun. J.* **2015**, *7*, 1248–1254.
22. Meador, C. Numerical Calculation of Lyapunov Exponents for Three Dimensional Systems of Ordinary Differential Equations. Master Thesis, Marshall Digital Scholar, Marshal University, Huntington, WV, USA, 2011.
23. Pukdeboon, C. A Review of Fundamentals of Lyapunov Theory. *J. Appl. Sci.* **2011**, *10*, 55–61.
24. Balibrea, F.; Caballero, M. *Examples of Lyapunov Exponents in Two Dimensional Systems Nonlinear Maps and their Applications*; Springer: New York, NY, USA, 2014; pp. 9–15.
25. Bespalov, A.; Polyakhov, N. Determination of the Largest Lyapunov Exponents Based on Time Series. *World Appl. Sci.* **2013**, *26*, 157–164.
26. Leonov, G.; Kuznetsov, N. On Differences and Similarities in the Analysis of Lorenz, Chen, and Lu Systems. *Appl. Math. Comput.* **2015**, *256*, 334–343. [CrossRef]
27. Thanoon, T.; L-Azzawi, S.A. Stability of Lorenz Differential System by Parameters Tikrit. *J. Pure Sci.* **2010**, *15*, 118–222.
28. Chenthittayil, S. Determination of Chaos in Different Dynamical Systems. Master Thesis, the Graduate School of Clemson University, Clemson, SC, USA, 2015.
29. Kose, E.; Akcayoglu, A. Examination of the Eigenvalues Lorenz Chaotic System. *Eur. Sci. J.* **2014**, *10*, 114–121.
30. Feketa, P.; Schaum, A.; Meurer, T. Synchronization and Multicluster Capabilities of Oscillatory Networks with Adaptive Coupling. *IEEE Trans. Autom. Control* **2020**, *66*, 3084–3096. [CrossRef]
31. Gambuzza, L.; Frasca, M.; Latora, V. Distributed Control of Synchronization of a Group of Network Nodes. *IEEE Trans. Autom. Control* **2018**, *64*, 365–372. [CrossRef]
32. Feketa, P.; Schaum, A.; Meurer, T.; Michaelis, D.; Ochs, K. Synchronization of Nonlinearly Coupled Networks of Chua Oscillators. *IFAC-PapersOnLine* **2019**, *52*, 628–633. [CrossRef]
33. Ochs, K.; Michaelis, D.; Solan, E.; Feketa, P.; Schaum, A.; Meurer, T. Synthesis, Design, and Synchronization Analysis of Coupled Linear Electrical Networks. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *67*, 4521–4532. [CrossRef]
34. George, G.; Raimond, K. A Survey on Optimization Algorithms for Optimizing the Numerical Functions. *Int. J. Comput. Appl.* **2013**, *61*, 41–46. [CrossRef]
35. Li, X.; Yin, M. Parameter Estimation for Chaotic Systems by Hybrid Differential Evolution Algorithm and Artificial Bee Colony Algorithm. *Nonlinear Dyn.* **2014**, *77*, 61–71. [CrossRef]
36. Jadon, S.; Tiwari, R.; Sharma, H.; Bansal, J. Hybrid Artificial Bee Colony Algorithm with Differential Evolution. *Appl. Soft Comput.* **2017**, *58*, 11–24. [CrossRef]
37. Jessa, M. Designing Security for Number Sequences Generated by Means of the Sawtooth Chaotic Map. *IEEE Trans. Circuits Syst.* **2006**, *53*, 1140–1150. [CrossRef]
38. Hao, D.; Xin, J.; Meng, H.; Quan, S. A New Three-Dimensional Chaotic System and Its Modified Generalized Projective Synchronization. *Chin. Phys. B* **2011**, *20*, 040507.
39. Ding, Y.; Jiang, W.; Wang, H. Delayed Feedback Control and Bifurcation Analysis of Rossler Chaotic System. *Nonlinear Dyn.* **2010**, *61*, 707–715. [CrossRef]
40. Abdmouleh, Z.; Gastli, A.; Brahim, L.; Haouari, M.; Al-Emadi, N. Review of Optimization Techniques Applied for the Integration of Distributed Generation from Renewable Energy Sources. *Renew. Energy* **2017**, *113*, 266–280. [CrossRef]
41. Jiang, H.; Zhao, D.; Zheng, R.; Ma, X. Construction of Pancreatic Cancer Classifier Based on SVM Optimized by Improved FOA. *BioMed Res. Int.* **2015**, *2015*, 781023. [CrossRef]
42. Pan, W. A New Fruit Fly Optimization Algorithm: Taking the Financial Distress Model as an Example. *Knowl. Based Syst.* **2012**, *26*, 69–74. [CrossRef]
43. Yin, L.; Li, X.; Gao, L.; Lu, C. A New Improved Fruit Fly Optimization Algorithm for Traveling Salesman Problem. In Proceedings of the Eighth International Conference on Advanced Computational Intelligence, Chiang Mai, Thailand, 14–16 February 2016; pp. 21–28.
44. 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*, 305. [CrossRef]
45. Lazzus, J.; Rivera, M.; Caraballo, C. Parameter Estimation of Lorenz Chaotic System using a Hybrid Swarm Intelligence Algorithm. *Phys. Lett. A* **2016**, *380*, 1164–1171. [CrossRef]
46. Sun, J.; Zhao, J.; Wu, X.; Fang, W.; Cai, Y.; Xu, W. Parameter Estimation for Chaotic Systems with a Drift Particle Swarm Optimization Method. *Phys. Lett. A* **2010**, *374*, 2816–2822. [CrossRef]
47. Qiang, L.; Wei, P.; Shan, Y.; Bin, L.; Feng, X.; Ning, J. Parameter Estimation for Chaotic Systems with and without Noise using Differential Evolution-based Method. *Chin. Phys. B* **2011**, *20*, 060502. [CrossRef]
48. Gao, W.; Zhang, Z.; Chong, Y. Chaotic System Parameter Identification based on Firefly Optimization. *Appl. Mech. Mater.* **2013**, *347–350*, 3821–3826. [CrossRef]
49. Xiang-Tao, L.; Ming-Hao, Y. Parameter Estimation for Chaotic Systems using the Cuckoo Search Algorithm with an Orthogonal Learning Method. *Chin. Phys. B* **2012**, *21*, 050507.
50. He, Q.; Wang, L.; Liu, B. Parameter Estimation for Chaotic Systems by Particle Swarm Optimization. *Chaos Solitons Fractals* **2007**, *34*, 654–661. [CrossRef]

51. Li, L.; Yang, Y.; Peng, H.; Wang, X. Parameters Identification of Chaotic Systems via Chaotic Ant Swarm. *Chaos Solitons Fractals* **2006**, *28*, 1204–1211. [CrossRef]
52. Gholipour, R.; Khosravi, A.; Mojallali, H. Parameter Estimation of Lorenz Chaotic Dynamic System using Bees Algorithm. *Int. J. Eng.* **2013**, *26*, 257–262. [CrossRef]
53. Wei, J.; Yu, Y. An Effective Hybrid Cuckoo Search Algorithm for Unknown Parameters and Time Delays Estimation of Chaotic Systems. *IEEE Access* **2018**, *6*, 6560–6571. [CrossRef]
54. Sheludko, A. Parameter Estimation for One-Dimensional Chaotic Systems by Guaranteed Algorithm and Particle Swarm Optimization. *IFAC-PapersOnLine* **2018**, *51*, 337–342. [CrossRef]
55. Zhuang, L.; Cao, L.; Wu, Y.; Zhong, Y.; Zhangzhong, L.; Zheng, W.; Wang, L. Parameter Estimation of Lorenz Chaotic System Based on A Hybrid Jaya-Powell Algorithm. *IEEE Access* **2020**, *8*, 20514–20522. [CrossRef]
56. Gupta, S.; Gautam, G.; Vats, D.; Varshney, P.; Srivastava, S. Estimation of Parameters in Fractional Order Financial Chaotic System with Nature Inspired Algorithms. *Procedia Comput. Sci.* **2020**, *173*, 18–27. [CrossRef]
57. Zhang, J.; Huang, S.; Cheng, J. Parameter Estimation for A Chaotic Dynamical System with Partial Observations. *J. Inverse Ill-Posed Probl.* **2021**, *29*, 515–524. [CrossRef]
58. Carlson, E.; Hudson, J.; Larios, A.; Martinez, V.; Ng, E.; Whitehead, J. Dynamically Learning the Parameters of a Chaotic System Using Partial Observations. *arXiv* **2021**, arXiv:2108.08354. [CrossRef]
59. Peng, C.; Li, Y. Parameters Identification of Nonlinear Lorenz Chaotic System for High-Precision Model Reference Synchronization. *Nonlinear Dyn.* **2022**, *108*, 1733–1754. [CrossRef]
60. Rizk-Allah, R.; Farag, M.; Barghout, M.; Hassanien, A. A Memory-Based Particle Swarm Optimization for Parameter Identification of Lorenz Chaotic System. In *Proceedings of International Conference on Computing and Communication Networks*; Springer: Singapore, 2022; pp. 571–587.
61. Ann, N.; Pebrianti, D.; Abas, M.; Bayuaji, L. Parameter Estimation of Lorenz Attractor: A Combined Deep Neural Network and K-Means Clustering Approach. In *Recent Trends in Mechatronics towards Industry*; Springer: Singapore, 2022; pp. 321–331.
62. 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]
63. Chang, W. Parameter Identification of Rossler's Chaotic System by an Evolutionary Algorithm. *Chaos Solitons Fractals* **2006**, *29*, 1047–1053. [CrossRef]
64. Chang, W. Parameter Identification of Chen and Lü Systems: A Differential Evolution Approach. *Chaos Solitons Fractals* **2007**, *32*, 1469–1476. [CrossRef]
65. Gonzales, O.; Han, G.; De, G.; Sánchez, E. Lorenz-Based Chaotic Cryptosystem: A Monolithic Implementation. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2000**, *47*, 1243–1247. [CrossRef]
66. Roslan, U.; Salleh, Z.; Kilicman, A. Solving Zhou Chaotic System using Fourth-Order Runge-Kutta Method. *World Appl. Sci. J.* **2013**, *21*, 939–944.
67. Garoma, H.; Kabeto, M. Numerical Solution of Fourth Order Ordinary Differential Equations using Fifth Order Runge—Kutta Method. *Asian J. Sci. Technol.* **2017**, *8*, 4332–4339.
68. Meneny, S. *An Introduction to Digital Signal Processing: A Focus on Implementation*, 1st ed.; River Press: Fort Benton, MT, USA, 2008.
69. Mandal, M.; Asif, A. *Continuous and Discrete Time Signals and Systems*, 1st ed.; Cambridge University Press: Cambridge, UK, 2007.
70. Jiang, Z.; Yang, Q. A Discrete Fruit Fly Optimization Algorithm for the Traveling Salesman Problem. *PLoS ONE* **2016**, *11*, e0165804. [CrossRef]
71. Zhang, H.; Li, B.; Zhang, J.; Qin, Y.; Feng, X.; Liu, B. Parameter Estimation of Nonlinear Chaotic System by Improved TLBO Strategy. *Soft Comput.* **2016**, *20*, 4965–4980. [CrossRef]
72. Kundra, H.; Khan, W.; Malik, M.; Rane, K.; Neware, R.; Jain, V. Quantum-Inspired Firefly Algorithm Integrated with Cuckoo Search for Optimal Path Planning. *Int. J. Mod. Phys. C* **2022**, *33*, 2250018. [CrossRef]
73. Li, H.; Bai, P.; Xue, J.; Zhu, J.; Zhang, H. Parameter Estimation of Chaotic Systems using Fireworks Algorithm. *Adv. Swarm Comput. Intell.* **2015**, *9141*, 457–467.
74. Abarbanel, H.; Kennel, M.; Illing, L.; Tang, S.; Chen, H.; Liu, J. Synchronization and Communication using Semiconductor Lasers with Optoelectronic Feedback. *IEEE J. Quantum Electron.* **2001**, *37*, 1301–1311. [CrossRef]