Eur. Phys. J. Spec. Top. (2022) 231:2037–2043
https://doi.org/10.1140/epjs/s11734-022-00452-6

THE EUROPEAN PHYSICAL JOURNAL SPECIAL TOPICS

Regular Article

Optimization of fractional-order chaotic cellular neural networks by metaheuristics

Esteban Tlelo-Cuautle[1,a], Astrid Maritza González-Zapata[1,b], Jonathan Daniel Díaz-Muñoz[1,c], Luis Gerardo de la Fraga[2,d], and Israel Cruz-Vega[1,e]

[1] INAOE, Electronics Department, Puebla 72840, Mexico
[2] CINVESTAV, Computer Department, Zacatenco, Mexico City 07360, Mexico

Received 28 July 2021 / Accepted 13 January 2022 / Published online 21 January 2022
© The Author(s), under exclusive licence to EDP Sciences, Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract Artificial neural networks have demonstrated to be very useful in solving problems in artificial intelligence. However, in most cases, ANNs are considered integer-order models, limiting the possible applications in recent engineering problems. In addition, when dealing with fractional-order neural networks, almost any work shows cases when varying the fractional order. In this manner, we introduce the optimization of a fractional-order neural network by applying metaheuristics, namely: differential evolution (DE) and accelerated particle swarm optimization (APSO) algorithms. The case study is a chaotic cellular neural network (CNN), for which the main goal is generating fractional orders of the neurons whose Kaplan–Yorke dimension is being maximized. We propose a method based on Fourier transform to evaluate if the generated time series is chaotic or not. The solutions that do not have chaotic behavior are not passed to the time series analysis (TISEAN) software, thus saving execution time. We show the best solutions provided by DE and APSO of the attractors of the fractional-order chaotic CNNs.

1 Introduction

Chaos is a discipline that has shown novel applications in different engineering areas [1]. Nowadays, fractional-order chaotic systems are a hot topic for research, but yet the solution of the dynamical system is a challenge, because there is not an analytical method to solve them. Almost all dynamical systems can be converted from its integer-order models to fractional-order ones. In both cases, one can evaluate their performances as dynamical characteristics that are associated to Lyapunov exponents, entropy, and Kaplan–Yorke dimension (D_{KY}). These characteristics can be optimized by metaheuristics [2–4]. For instance, among the available optimization algorithms, differential evolution (DE) and particle swarm optimization (PSO) have shown advantages in maximizing fractional-order systems [5]. In this paper, we apply DE and accelerated PSO (APSO) to optimize fractional-order neural networks that can have commensurate (if the derivatives have the same fractional-order) or incommensurate (if the derivatives have different fractional-order) orders.

Artificial neural networks (ANNs) are associated to model the behavior of a biological neural network, as in the human brain [6]. They are modeled by mathematical equations, some of them involving continuous-time and other discrete time variables. Chaotic neural networks have shown advantages in generating random binary sequences to encrypt color images, as shown in [7]. Other applications can be found in [8], where the authors show recent developments on fuzzy logic, neural networks and optimization algorithms, as well as their hybrid combinations, and their application in areas such as intelligent control and robotics, pattern recognition, medical diagnosis, time series prediction, and optimization of complex problems. More recently, it has been proven that neural networks are a good option for the analysis of the coronavirus pandemic problem [9]. One can list a huge number of applications of neural networks, but in this paper, the efforts are focused on problems involving fractional-order issues [10]. For instance, the following fractional-order neural networks can be suitable for optimization: the fractional-order residual convolutional neural network introduced in [11]; the distributed-order neural networks introduced in [12]; the fractional-order neural network with time-varying delays given in [13]; the fractional-order quaternion-valued neural network given in [14]; the fractional-order complex-valued neural network with impulsive effects given in [15]; the fractional-order memristive neural synaptic weighting given in [16]; the fractional-order complex-valued neural network given in [17], and so on.

[a] e-mail: etlelo@inaoep.mx (corresponding author)
[b] e-mail: amgonzalez@inaoep.mx
[c] e-mail: jdiazm@inaoep.mx
[d] e-mail: fraga@cs.cinvestav.mx
[e] e-mail: icruzv@inaoep.mx

As it is well known, the challenge on applying metaheuristics is the reduction of the execution time in the optimization loop. In this manner, as the evaluation of the dynamical characteristics of a fractional-order chaotic system consumes large time, we introduce a procedure to eliminate the evaluation of D_{KY} of a fractional-order CNN when the behavior is not chaotic. Henceforth, the proposed method consists on applying Fourier transform to the chaotic time series, to evaluate if the CNN generates chaotic behavior or not. If the CNN is chaotic, its time series are introduced to the time series analysis (TISEAN) software to evaluate D_{KY}. If the CNN is not chaotic, the individual in DE or particle in APSO is eliminated and this saves execution time in the optimization loop.

The organization is done as follows: Sect. 2 describes the mathematical model of the cellular neural network (CNN) in its fractional-order version. Section 3 describes the metaheuristics, namely: DE and APSO algorithms that are used to optimize the CNN. The formulation of the optimization problem is shown in Sect. 4, where the goal is maximizing D_{KY}, and we show the application of Fourier transform to reduce execution time of DE and APSO. Section 5 summarizes the optimization results and lists the best individuals provided by DE and solutions by APSO along their corresponding chaotic attractors. Finally, the conclusions are given in Sect. 6.

2 Fractional-order chaotic cellular neural network

Some recent fractional-order chaotic systems have been introduced in [18,19], and the commensurate and incommensurate characteristics have been described in [20,21]. This paper focuses on the Cellular Neural Network (CNN) that was proposed in [22]. Basically, it consists of a series of modeled cells by an analog nonlinear circuit. In [23], the authors analyzed the chaotic behavior of the autonomous CNN composed by two or three cells. For instance, the CNN modeled by three cells is given in (1). In this mathematical model: x, y, and z are neural states; $f(x)$, $f(y)$, and $f(z)$ are neural activation functions shown in (2); and p_1, p_2, p_3, s, and r are the weights or coupling forces of the network, which values generating chaotic behavior are set to $p_1 = 1.25$, $p_2 = 1.1$, $p_3 = 1$, $s = 3.2$, and $r = 4.4$

$$\begin{aligned}\dot{x} &= -x + p_1 f(x) - s f(y) - s f(z)\\ \dot{y} &= -y - s f(x) + p_2 f(y) - r f(z)\\ \dot{z} &= -z - s f(x) + r f(y) + p_3 f(z)\end{aligned} \tag{1}$$

$$\begin{aligned}f(x) &= \tfrac{|x+1|-|x-1|}{2}\\ f(y) &= \tfrac{|y+1|-|y-1|}{2}\\ f(z) &= \tfrac{|z+1|-|z-1|}{2}.\end{aligned} \tag{2}$$

Table 1 Proposed search space ranges of the design variables for the optimization process

Design Variable	Range
p_1	[1.00000000, 1.50000000]
p_2	[0.50000000, 1.50000000]
p_3	[0.50000000, 1.50000000]
s	[2.50000000, 4.00000000]
r	[4.00000000, 5.00000000]
q_i	[0.60000000, 0.99999999]

As one sees, (1) is a dynamical system of integer order, and it can be transformed to its fractional-order version, as shown in (3). It can be observed that the differential equations changed to have fractional orders denoted by q_1, q_2, q_3. One can have two cases for the optimization process of the fractional orders: if $q_1 = q_2 = q_3$ the CNN has commensurate orders, otherwise, the CNN has incommensurate orders, i.e., $q_1 \neq q_2 \neq q_3$

$$\begin{aligned}{}_0D_t^{q_1} x(t) &= -x(t) + p_1 f(x) - s f(y) - s f(z)\\ {}_0D_t^{q_2} y(t) &= -y(t) - s f(x) + p_2 f(y) - r f(z)\\ {}_0D_t^{q_3} z(t) &= -z(t) - s f(x) + r f(y) + p_3 f(z).\end{aligned} \tag{3}$$

In general, the optimization of a continuous-time system can have infinite possibilities for the design variables, in this case the coefficients and fractional orders of the derivatives. However, not all the possibilities will lead to generate chaotic behavior, so that one can estimate reduced search spaces for p_1, p_2, p_3, s, and r, and for the fractional orders, one can also establish fractional ranges for q_1, q_2, and q_3, as listed in Table 1, which have high possibilities of generating chaotic behavior than other random values.

The authors in [24] demonstrated that for a chaotic oscillator consisting of three fractional-order derivatives, and considering different types of model nonlinearities, and using the proper control parameters, chaotic attractors are obtained with system orders as low as 2.1. Consequently, those authors introduced a conjecture that third-order systems can still produce chaotic behavior with a total system order higher than 2. In this manner, the fractional-order chaotic CNN modeled by (3) has three differential equations, and its minimum fractional order to generate chaotic behavior must be higher than two. This can be accomplished if all the commensurate fractional orders of the derivatives have values equal or higher than 0.7. However, for incommensurate fractional orders, one can have combinations of values, so that the sum of the three fractional orders be higher than two. To know the dimension of the problem: By setting the search spaces from 0.60000000 to 0.99999999, it leads us to have $(10^8)^3 = 10^{24}$ possibilities of having $q_1 \neq q_2 \neq q_3$. This number increases exponentially when varying p_1, p_2, p_3, s, and r. For this reason, metaheuristics are a good option to optimize a fractional-order chaotic CNN.

3 Metaheuristics

The pseudocodes of DE and PSO algorithms have already given in [5] for the optimization of fractional-order chaotic oscillators. In this paper, DE and APSO are applied to optimize a fractional-order chaotic CNN. The pseudocodes are summarized in the following subsections.

3.1 Differential evolution (DE)

DE algorithm is a metaheuristic that uses real numbers in its representation and then can be applied to optimize continuous-time problems. DE requires as input, the size of the population, number of generations, the value for the recombination constant (CR), the value for the differential constant (F), and the threshold value s. The pseudocode for the optimization of the chaotic CNN is given in Algorithm 1. The evaluation of D_{KY} is performed, as shown in Sect. 4.2, including the verification of chaotic behavior of the times series using the spectrum.

The generations are updated by creating new vectors by performing mutation (4) and crossover (5) operations. Where, a, b, and c are different numbers randomly chosen; g denotes the current generation; $F \in [0, 2]$ is the differential constant; $j = \{1, 2, \ldots, D\}$; $randb(j) \in [0, 1]$ is the jth evaluation of a generated random number; $CR \in [0, 1]$ is a crossover coefficient chosen by the user; and $rnbr(i) \in [0, D-1]$ is an index randomly generated [5]

$$v_i^{g+1} \leftarrow x_c^g + F(x_a^g - x_b^g) \tag{4}$$

$$u_{ij}^{g+1} \leftarrow \begin{cases} v_{ij}^{g+1} & \text{if } randb(j) \le CR \text{ or } rnbr(i) = j \\ x_{ij}^{g+1} & \text{if } randb(j) > CR \text{ and } rnbr(i) \ne j. \end{cases} \tag{5}$$

Algorithm 1 Differential evolution algorithm.

```
Initialize the population randomly (x)
Evaluate the position of the individuals of the CNN described by func(x)
Evaluate D_KY
Save the results of evaluating D_KY in score
for (counter = 1; counter ≤ G; counter++) do
    for (i = 1; i ≤ N_p; i++) do
        Select three different indexes randomly (a, b and c in (4))
        for (j = 1; j ≤ D; j++) do
            if U(0,1) < CR||j = D then
                trial_j ← x_aj + F(x_bj − x_cj)
            else
                trial_j ← x_ij
            end if
        end for
        f_x ← func(trial)
        if f_x is better than score_i then
            score_i ← f_x
            x_i ← trial
        end if
    end for
end for
return x and score
```

3.2 Accelerated particle swarm optimization (APSO)

Similar to DE, APSO algorithm is a metaheuristic that performs a direct random search in an intelligent sense. Generally, APSO requires the following parameters as input: size of the population, number of generations, β, and α. The pseudocode of APSO for the optimization of the chaotic CNN is given in Algorithm 2. The evaluation of D_{KY} is performed, as shown in Sect. 4.2, including the verification of chaotic behavior of the times series using the spectrum.

The particle behavior is defined by two equations: velocity (6) and position (7). Where, i is the index of the particle; j its dimension; p_i the best position found in i; p_g the best position found during the optimization; $\alpha \in \mathbb{R}$ is the inertial weight; $\beta \in \mathbb{R}$ the acceleration constant; and $U(\bullet)$ a random number generator with uniform distribution [5]

$$v_{ij}^{t+1} \leftarrow \alpha v_{ij}^t + U(0, \beta)(p_{ij} - x_{ij}^t) + U(0, \beta)(g_j - x_{ij}^t) \tag{6}$$

$$x_{ij}^{t+1} \leftarrow x_{ij}^t + v_{ij}^{t+1}. \tag{7}$$

4 Optimizing D_{KY} of a chaotic CNN by DE and APSO

This section shows the fitness function of the optimization process, and the handling of constraints highlighting the use of Fourier transform to verify if the time series is chaotic or not and, therefore, save execution time when evaluating D_{KY} by TISEAN, who is time-consuming.

4.1 Fitness function

In chaotic systems, D_{KY} is evaluated considering the number of ordinary differential equations. In this case, the chaotic CNN has three equations and its dimension can be evaluated by applying (8). This dimension is related to the Lyapunov exponents λ_1, which are equal to the number of state variables. In this case, the chaotic CNN has three state variables x, y, z, so that it has three Lyapunov exponents, and to show a chaotic behavior, one must be positive, one zero, and another negative, and therefore, $D_{KY} > 2$ [25]. Maximizing D_{KY} leads

Algorithm 2 Accelerated particle swarm optimization algorithm.

1: Initialize the particle's position randomly ($\mathbf{x}$)
2: Initialize the velocity of the particles v
3: Evaluate the position of the particles associated to the CNN described by $func(x)$
4: Evaluate D_{KY}
5: Save the results of evaluating D_{KY} in $score$ and $p \leftarrow x$
6: Find the best value from p and save it in g
7: **for** $(counter = 1; counter \leq G; counter++)$ **do**
8: **for** $(i = 1; i \leq N_p; i++)$ **do**
9: **for** $(j = 1; j \leq D; j++)$ **do**
10: $v_{ij} \leftarrow \alpha v_{ij} + U(0,\beta)(p_{ij} - x_{ij}) + U(0,\beta)(g_j - x_{ij})$ ▷ This evaluates the new velocity using (6)
11: $x_{ij} \leftarrow x_{ij} + v_{ij}$ ▷ This evaluates the new position using (7)
12: **end for**
13: $f_x \leftarrow func(x_i)$
14: **if** f_x is better than $score_i$ **then**
15: $score_i \leftarrow f_x$
16: $p_i \leftarrow x_i$
17: **if** p_i is better than g **then**
18: $g \leftarrow p_i$
19: **end if**
20: **end if**
21: **end for**
22: **end for**
23: **return** x, p, g and $score$

us to deal with the fitness function given in (9)

$$D_{KY} = j + \frac{\lambda_1 + \cdots + \lambda_j}{|\lambda_{j+1}|} \tag{8}$$

$$f(\lambda) = 2 + \frac{\lambda_1 + \lambda_2}{|\lambda_3|}. \tag{9}$$

Among the available methods to evaluate Lyapunov spectrum, one can use the online available software called TISEAN, who requires the chaotic time series of the associated state variable x, y, z, as input. This software is used herein to evaluate D_{KY}, but is time-consuming, so that we propose to verify if the chaotic time series is chaotic or not by computing the Fourier transform, and it is taken as a constraint.

4.2 Verification of chaotic behavior using the spectrum

The optimization algorithms based on populations as DE and APSO are time-consuming, so that to reduce the execution time, we propose the application of Fourier transform to select those individuals in DE or particles in APSO that can be passed to TISEAN to compute D_{KY}. Besides, the first constraint is imposed by (10) for both commensurate or incommensurate fractional orders

$$\sum_{i=1}^{3} q_i \geq 2.1 \tag{10}$$

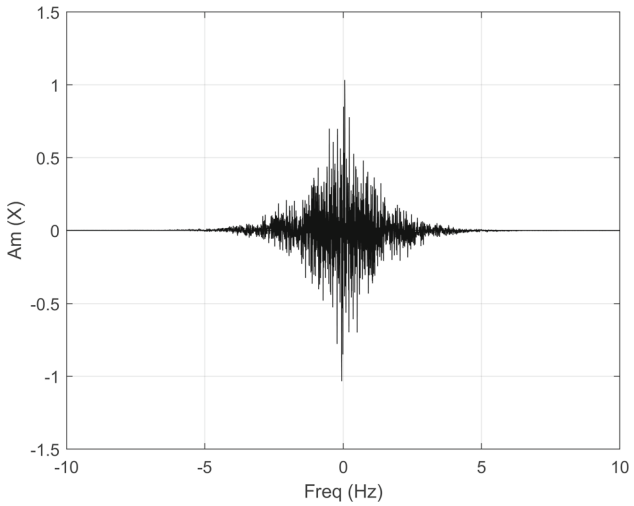

Fig. 1 Fourier transform of a chaotic time series

A second constraint consists on verifying the eigenvalues that must be complex to generate oscillating behavior. If the mathematical model does not have equilibrium points, as the case of the fractional-order chaotic CNN given in (3), then one can apply a numerical method as Newton–Raphson [26], which is given in (11). This method is very fast, so that the execution time of the metaheuristic is not increased as for the one taken by TISEAN

$$X_{i+1} = X_i - J^{-1}(X_i) f(X_i). \tag{11}$$

Guaranteeing the constraint in (10) and that the eigenvalues are complex conjugated may not lead to chaotic behavior. In fact, in many cases, the time series may have periodic behavior or they can be damped. In this manner, as DE and APSO work with populations, if the individuals are verified to have chaotic behavior, the execution time can be reduced, because D_{KY} is evaluated by TISEAN and it is the bottleneck in the optimization loop. See for example the spectrum of a chaotic time series shown in Fig. 1, and the spectrum of a non-chaotic time series shown in Fig. 2. It is very notable that the spectrum of a chaotic time series has higher amplitude than a non-chaotic one. For this reason, we propose to evaluate the Fourier transform of each time series before the individual or particle pass to TISEAN and this is decided by setting a threshold for the spectrums. In this work, the threshold was set to 0.2.

The pseudocode including the Fourier transform as constraint is given in the procedure named *Evaluate()*, shown below. This procedure is executed in Algorithm 1 at step 3, and in Algorithm 2 at step 4 to evaluate D_{KY}. As one can see, the first step is verifying the fractional order to accomplish (10). If this is accomplished, the eigenvalues are computed applying Newton–Raphson. If the eigenvalues are complex conjugated, the chaotic CNN is simulated by applying a numerical method, as the Grüwald–Letnikov [1]. The generated time series is

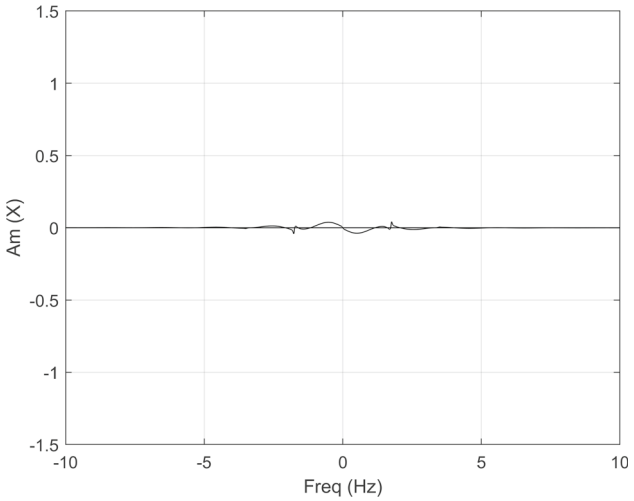

Fig. 2 Fourier transform of a non-chaotic time series

Table 2 *DE* input parameters

Parameter	Value
Population size	40
Number of generations	20
Number of variables	8
Difference constant	0.4
Recombination constant	0.1
min_p_1[1], max_p_1[1]	1.00000000, 1.50000000
min_p_2[2], max_p_2[1]	0.50000000, 1.50000000
min_p_3[3], max_p_3[1]	0.50000000, 1.50000000
min_s[4], max_s[1]	2.50000000, 4.00000000
min_r[5], max_r[1]	4.00000000, 5.00000000
min_q_i[6 − 8], max_q_i[6 − 8]	0.60000000, 0.99999999

evaluated with the Fourier transform to double check that is chaotic behavior. Finally, the chaotic time series is processed by TISEAN computing D_{KY}.

```
1. def Evaluate(_n, vpar):
2.   D_{KY} = 0.0
2.   # Constraint 1: fractional order
3.   const1 = Cal.FractOrder(vpar)

4.   if const1 == 1:
5.     # Contraint 2: eigenvalues
6.     const2 = Cal.EvalEigenvalues(vpar)

7.     if const2 == 1:
8.       # Calculate time series
9.       t = Cal.CalTimeSeries(vpar)
10.      # Contraint 3: Fourier transform
11.      const3 = Cal.fftDiscrete(t)

12.      if const3 == 1:
13.        # Calculate Kaplan-Yorke
             dimension (TISEAN)
14.        D_{KY} = Cal.CalDimKY()

15. return D_{KY}
```

5 Results

Both DE and APSO were programmed in *python* language. The simulation of the fractional-order CNN modeled in (3) is done applying Grünwald–Letnikov with a step-size $h = 0.01$ and a short memory length of 1% of the total length, e.g., $Lm = 2$ for 20,000 samples. The initial conditions for all the cases were set to $x_0 = -0.1$, $y_0 = 0.2$, and $z_0 = -0.1$.

5.1 Best feasible solutions provided by DE

The input parameters that were used for DE algorithm are listed in Table 2. It can be noted that it includes the parameters given in Subsect. 3.1

The four best feasible solutions provided by DE are given in Table 3. It can be noted that solution 2 has incommensurate, while solutions 1, 3, and 4 have commensurate fractional order, but the other parameters are totally different, and all these best solutions have good values for the fitness function, i.e., D_{KY}.

Figure 3 shows the chaotic attractors of the solutions provided by DE for the fractional-order CNN. It can be appreciated that they have very similar ranges in the state variables x and y.

Table 3 Four best solutions provided by DE

Variable	Solution 1	Solution 2	Solution 3	Solution 4
p_1	1.2249	1.22556604	1.19199827	1.23340808
p_2	1.08	1.13855862	1.1	1.17102638
p_3	1.0	0.96259797	1.0	1.0
s	3.2	3.19792497	3.02918670	2.95860449
r	4.4	4.55205362	4.4	4.4
q_1	0.99	0.98373450	0.99	0.99
q_2	0.99	0.99857154	0.99	0.99
q_3	0.99	0.98559591	0.99	0.99
D_{KY}	**2.41112811**	**2.52850903**	**2.59403606**	**2.73015194**

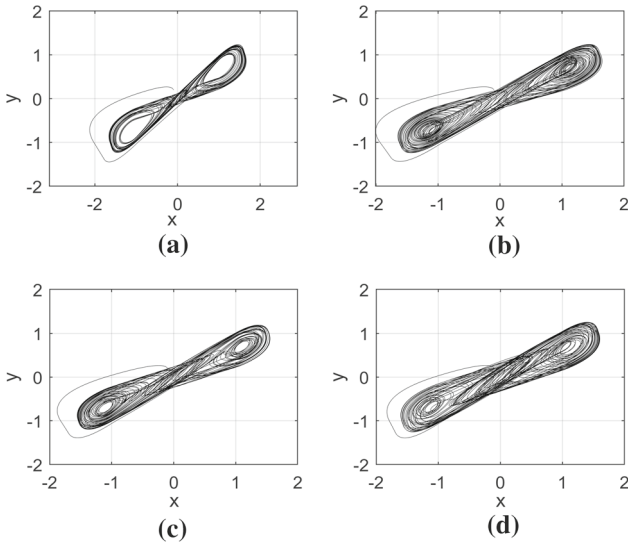

Fig. 3 Chaotic attractors for the solutions given in Table 3: **a** Solution 1, **b** Solution 2, **c** Solution 3, and **d** Solution 4

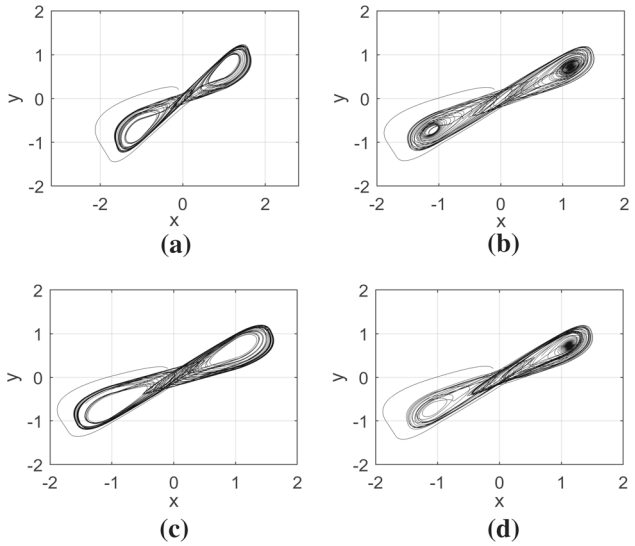

Fig. 4 Chaotic attractors for the solutions given in Table 5. **a** Solution 1, **b** Solution 2, **c** Solution 3, and **d** Solution 4

Table 4 *APSO* input parameters

Parameter	Value
Population size	40
Number of generations	20
Number of variables	8
β	0.3
α	0.7
min_p_1[1], max_p_1[1]	1.00000000, 1.50000000
min_p_2[2], max_p_2[1]	0.50000000, 1.50000000
min_p_3[3], max_p_3[1]	0.50000000, 1.50000000
min_s[4], max_s[1]	2.50000000, 4.00000000
min_r[5], max_r[1]	4.00000000, 5.00000000
min_q_i[6 − 8], max_q_i[6 − 8]	0.60000000, 0.99999999

5.2 Best feasible solutions provided by APSO

APSO algorithm was executed by setting the input parameters, as shown in Table 4. It includes the values of the search spaces ranges of the design variables given in Sect. 3.2.

The four best solutions provided by APSO are given in Table 5. In this case, just one commensurate case associated with solution 1 is listed. the other three solutions 2, 3, and 4 have incommensurate fractional order, but in all cases, again the fitness function is high. In fact, APSO provided higher D_{KY} than DE.

Figure 4 shows the chaotic attractors associated to the four best solutions provided by APSO.

6 Conclusions

This paper showed the optimization of a fractional-order cellular neural network by applying metaheuristics, such as DE and APSO. The fitness function was associated to maximize the Kaplan–Yorke dimension, which is evaluated by TISEAN program. To reduce execution time of the optimization loops in both metaheuristics, we proposed the evaluation of the Fourier transform to the time series of the CNN and apply a threshold to the spectrum to verify chaotic behavior. However, to verify that the individual or particle in the population will generate chaotic behavior, some constraints were evaluated, namely: verification that the fractional order is higher than two, verifica-

Table 5 Four best solutions provided by APSO

Variable	Solution 1	Solution 2	Solution 3	Solution 4
p_1	1.2249	1.21524764	1.28832790	1.21529417
p_2	1.08	1.02487968	1.14787860	1.02466857
p_3	1.0	1.16548903	1.05392120	1.16572212
s	3.2	3.20087397	3.10834142	3.20088438
r	4.4	4.60504481	4.44069088	4.46546546
q_1	0.99	0.99057132	0.94994949	0.99040112
q_2	0.99	0.98569995	0.97372856	0.98570651
q_3	0.99	0.97725932	0.94780070	0.97719746
D_{KY}	**2.41112811**	**2.63592020**	**2.64600627**	**2.83383367**

tion that the eigenvalues are complex, and verification that the Fourier transform has high amplitude values in the spectrum, e.g., 0.2, in this work. These constraints driven DE and APSO algorithms to maximize D_{KY}, providing feasible solutions whose attractors have similar ranges in their phase portraits. The best D_{KY} results provided by the metaheuristics were associated to incommensurate fractional-order CNNs. They can be implemented on embedded systems for the development of real applications as encryption systems under a protocol for IoT.

References

1. E. Tlelo-Cuautle, A.D. Pano-Azucena, O. Guillén-Fernández, A. Silva-Juárez, *Analog/Digital Implementation of Fractional Order Chaotic Circuits and Applications* (Springer, New York, 2020)
2. B. Abdollahzadeh, F. Soleimanian Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. Comput. Ind Eng. **158**, 107408 (2021)
3. Human Shayanfar and Farhad Soleimanian Gharehchopogh, Farmland fertility: a new metaheuristic algorithm for solving continuous optimization problems. Appl. Soft Comput. **71**, 728–746 (2018)
4. Farhad Soleimanian Gharehchopogh and Hojjat Gholizadeh, A comprehensive survey: Whale optimization algorithm and its applications. Swarm Evol. Comput. **48**, 1–24 (2019)
5. A. Silva-Juarez, E. Tlelo-Cuautle, L. G. de la Fraga, R. Li. Optimization of the Kaplan–Yorke dimension in fractional-order chaotic oscillators by metaheuristics. Appl. Math. Comput. **394**, 125831 (2021)
6. W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. **5**(4), 115–133 (1943)
7. E. Tlelo-Cuautle, J. D. Díaz-Muñoz, A. M. González-Zapata, R. Li, W.D. León-Salas, F.V. Fernandez, O. Guillén-Fernández, I. Cruz-Vega, Chaotic Image Encryption Using Hopfield and Hindmarsh-Rose Neurons Implemented on FPGA. Sensors **20**(5), 1326 (2020). Number: 5 Publisher: Multidisciplinary Digital Publishing Institute
8. O. Castillo, Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms: Theory and Applications, volume 940. Springer Nature (2021)
9. P. Melin, O. Castillo, Spatial and temporal spread of the coronavirus pandemic using self organizing neural networks and a fuzzy fractal approach. (2021)
10. A. Mohammadzadeh, O. Castillo, S.S. Band, A. Mosavi, A novel fractional-order multiple-model type-3 fuzzy control for nonlinear systems with unmodeled dynamics. Int. J. Fuzzy Syst. pp. 1–19 (2021)
11. M. Chen, P. Yi-Fei, Y.-C. Bai, Low-dose ct image denoising using residual convolutional network with fractional tv loss. Neurocomputing **452**, 510–520 (2021)
12. G.M. Mahmoud, T. Aboelenen, T. M. Abed-Elhameed, A. A. Farghaly, On boundedness and projective synchronization of distributed order neural networks. Appl. Math. Comput. **404**, 126198 (2021)
13. F. Zhang, T. Huang, W. Qiujie, Z. Zeng, Multistability of delayed fractional-order competitive neural networks. Neural Netw. **140**, 325–335 (2021)
14. K. Udhayakumar, R. Rakkiyappan, X. Li, J. Cao, Mutiple ψ-type stability of fractional-order quaternion-valued neural networks. Appl. Math. Comput. **401**, 126092 (2021)
15. H. Li, Y. Kao, H.-L. Li, Globally β-mittag-leffler stability and β-mittag-leffler convergence in lagrange sense for impulsive fractional-order complex-valued neural networks. Chaos Solit. Fract. **148**, 111061 (2021)
16. P. Yifei, Yu. Bo, Q. He, X. Yuan, Fractional-order memristive neural synaptic weighting achieved by pulse-based fracmemristor bridge circuit. Front. Inf. Technol. Electron. Eng. **22**(6), 862–876 (2021)
17. S. Wang, H. Zhang, W. Zhang, H. Zhang, Finite-time projective synchronization of caputo type fractional complex-valued delayed neural networks. Mathematics **9**(12), 1406 (2021)
18. X. Li, J. Mou, L. Xiong, Z. Wang, X. Ji, Fractional-order double-ring erbium-doped fiber laser chaotic system and its application on image encryption. Opt. Laser Technol. **140**, 107074 (2021)
19. T. Liu, H. Yan, S. Banerjee, J. Mou, A fractional-order chaotic system with hidden attractor and self-excited attractor and its DSP implementation. Chaos Solit. Fract. **145**, 110791 (2021)
20. F. Yang, J. Mou, C. Ma, Y. Cao, Dynamic analysis of an improper fractional-order laser chaotic system and its image encryption application. Opt. Lasers Eng. **129**, 106031 (2020)
21. T. Liu, S. Banerjee, H. Yan, J. Mou, Dynamical analysis of the improper fractional-order 2D-SCLMM and its DSP implementation. Eur. Phys. J. Plus **136**(5), 506 (2021)
22. L.O. Chua, L. Yang, Cellular neural networks: theory. IEEE Trans. Circ. Syst. **35**(10), 1257–1272 (1988)
23. F. Zou, J.A. Nossek, Bifurcation and chaos in cellular neural networks. IEEE Trans. Circ. Syst. I: Fundamental Theory Appl. **40**(3), 166–173 (1993)
24. Wajdi M. Ahmad, J.C. Sprott. Chaos in fractional-order autonomous nonlinear systems. Chaos Solit. Fract. **16**(2), 339–351 (2003)
25. S. Thomas Parker, L. Chua, *Practical Numerical Algorithms for Chaotic Systems* (Springer, New York, 1989)
26. J. Verbeke, R. Cools, The Newton–Raphson method. Int. J. Math. Educ. Sci. Technol. **26**(2), 177–193 (1995)