

Optimizing the maximum Lyapunov exponent and phase space portraits in multi-scroll chaotic oscillators

Luis Gerardo de la Fraga · Esteban Tlelo-Cuautle

Received: 17 July 2013 / Accepted: 23 December 2013 / Published online: 11 January 2014
© Springer Science+Business Media Dordrecht 2014

Abstract This investigation introduces the application of the nondominated sorting genetic algorithm to optimize two characteristics of multiscroll chaotic oscillators: (a) Maximizing the values of the maximum Lyapunov exponent (MLE), and (b) minimizing the dispersions of the phase space portraits (PSP) among all scrolls in an attractor. As shown in this study, these two oscillator's characteristics are in conflict and must be considered at the same time. The cases of study are two multiscroll chaotic oscillators based on piecewise-linear functions, namely: saturated function series and Chua's diode (negative slopes). Basically, a very new procedure to measure the PSP coerture among all generated scrolls is introduced in the optimization loop for each feasible solution maximizing the MLE. The best optimized results are compared with traditional values of the coefficients of the equations describing the oscillators. Finally, we list the values of the optimized MLE and their corresponding PSP when generating from 2- to 6-scroll attractors.

Keywords Chaotic oscillator · Multiobjective optimization · Phase space portrait · Multiscroll · Maximum Lyapunov exponent

1 Introduction

In electronics, a great variety of chaotic oscillators has been implemented with commercially available electronic devices, as well as with integrated circuits technology [1–3]. However, those experimental realizations are not optimized neither to provide a high value of the maximum Lyapunov exponent (MLE), nor to have a good distribution in the phase space portraits (PSP) when generating multiscroll chaotic attractors. The former characteristic can be taken as an indication to quantify the chaoticity related to the unpredictability of the dynamical system [4, 5], and the second one indicates that when the trajectories in the PSP are not well distributed among all the scrolls, some scrolls cannot be formed, thus leading to a bad electronic implementation.

As shown by the recent studies [6–10], computing Lyapunov exponents is in general a good direction to quantify the asymptotic behaviors of nonlinear dynamical systems when the dynamic models of the systems are available. On the other hand, when such models are unknown, as for real world systems, estimating Lyapunov exponents can be performed by time series from experimental data [9]. However, applying that approach one should be aware that time series often assume and

L. G. de la Fraga (✉)
Computer Science Department, Cinvestav,
Av. IPN 2508, 07360 Mexico City, Mexico
e-mail: fraga@cs.cinvestav.mx

E. Tlelo-Cuautle
Department of Electronics, INAOE,
Luis Enrique Erro No. 1, Tonantzintla,
72840 Cholula, PUE, Mexico
e-mail: etlelo@inaoep.mx

do not test for exponential divergence of the orbits, and thus it can have the problem that cannot distinguish chaos from noise. In this article, we compute the MLE using the dynamic model of the multiscroll chaotic oscillators, and also we compute the distribution of the trajectories in the PSP. These tasks are executed herein as optimization problems. That way, this investigation introduces the application of the nondominated sorting genetic algorithm (NSGA-II) [11] to optimize those two characteristics in multiscroll chaotic oscillators [12, 13]: (a) Maximizing the value of MLE, and (b) minimizing the dispersions of the PSP among all scrolls in an attractor. As a contribution, we show that both characteristics are in conflict and must be considered at the same time to design optimized multiscroll chaotic oscillators. Further, once the chaotic oscillators have been optimized to guarantee chaotic regime and to provide a good distribution in their PSP, then those optimized chaotic oscillators can enhance the synchronization [14–17], and realization of secure communication systems [18], for instance.

Henceforth, this article shows by computer simulations, how does the value of MLE is related with the distribution of the trajectories in the PSP for two different multiscroll chaotic oscillators, which are based on piecewise-linear functions (PWL) like saturated function series and Chua’s diode (negative slopes). We show that those two characteristics for multiscroll chaotic oscillators, related to unpredictability and PSP coverage, are in conflict. Therefore, that problem is solved herein by applying a new proposed procedure to measure the PSP coverage for chaotic oscillators based on PWL functions, and by solving the multiobjective optimization problem with NSGA-II.

This article is organized as follows: in Sect. 2 the two multiscroll chaotic oscillators based on PWL functions are described; in Sect. 3 the proposed selection approach is described: the new procedure to take the measurements of the PSP coverage is introduced, and it is included also both, a brief description of how the MLE is calculated, and how the multiobjective optimization algorithm NSGA-II [6] works. Section 4, summarizes the best experimental results that are compared with traditional values of the coefficients of the equations describing the chaotic oscillators, i.e. we list the values of the optimized MLEs and their corresponding PSP when generating from 2- to 6-scroll attractors. Section 5 presents a discussion on the results and finally, in Sect. 6 some conclusions are drawn.

2 Multiscroll chaotic oscillators

2.1 PWL function based on saturated function series

This multiscroll chaotic oscillator is described by the system of ordinary differential equations given by (1) [12], where a, b, c , and d_1 are positive constants that can have values in the interval $[0, 1]$. The dynamical system is controlled by the PWL approximation, e.g. series of a saturated function f ,

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= z \\ \dot{z} &= -ax - by - cz + d_1 f(x; m). \end{aligned} \tag{1}$$

In the following, we describe in detail how the saturated function f in (1) is obtained. Let f_0 be the saturated function:

$$f_0(x; m) = \begin{cases} 1, & \text{if } x > m, \\ \frac{x}{m}, & \text{if } |x| \leq m, \\ -1, & \text{if } x < -m, \end{cases} \tag{2}$$

where $1/m$ is the slope of the middle segment and $m > 0$; the upper radial $\{f_0(x; m) = 1 \mid x > m\}$, and the lower radial $\{f_0(x; m) = -1 \mid x < -m\}$ are called *saturated plateaus*, and the segment $\{f_0(x; m) = x/m \mid |x| \leq m\}$ between the two saturated plateaus is called *saturated slope*.

Lets us consider now the saturated functions f_h and f_{-h} defined as:

$$f_h(x; m, h) = \begin{cases} 2, & \text{if } x > h + m, \\ \frac{x-h}{m}, & \text{if } |x - h| \leq m, \\ 0, & \text{if } x < h - m, \end{cases} \tag{3}$$

and

$$f_{-h}(x; m, -h) = \begin{cases} 0, & \text{if } x > h + m, \\ \frac{x-h}{m}, & \text{if } |x - h| \leq m, \\ -2, & \text{if } x < h - m, \end{cases} \tag{4}$$

where h is called the *saturated delay time* and $h > m$. Therefore, a saturated function series for a chaotic oscillator with s scrolls is defined as the function:

$$f(x; m) = \sum_{i=0}^{s-2} f_{2i-s+2}(x; m, 2i - s + 2), \tag{5}$$

where $s > 2$.

For example, using $f = f_0$ in (1), a 2-scroll chaotic oscillator can be generated. Therefore, the saturated function series to generate 3-scroll is $f(x; m) =$

$f_{-1}(x; m, -1) + f_1(x; m, 1)$. To generate a 4-scroll attractor it will be $f(x; m) = f_{-2}(x; m, -2) + f_0(x; m) + f_2(x; m, 2)$, and so on.

2.2 PWL function based on Chua’s diode

The equations describing this multiscroll chaotic oscillator [19] are the following:

$$\begin{aligned} \dot{x} &= \alpha[y - x - g(x)], \\ \dot{y} &= x - y + z, \\ \dot{z} &= \beta y, \end{aligned} \tag{6}$$

where α and β are positive and real constants, and g is the PWL function defined as:

$$g(x) = m_{2n-1}x + \frac{1}{2} \sum_{i=k}^{2n-1} \times (m_{i-1} - m_i)(|x + b_i| - |x - b_i|), \tag{7}$$

where m_i are slopes, and contrary to the PWL function f in (5), here the slopes must have negative values to derive Chua’s diode characteristic. The constants b_i are real numbers and are associated to break point values. For this multiscroll chaotic oscillator, and in order to generate an even number ($2n$) of scrolls, k takes values $k = 1, 2, 3, \dots$. For generating an odd number ($2n + 1$) of scrolls, k takes values $k = 2, 3, 4, \dots$

3 Proposed optimization-selection procedure

The flowchart of our proposed selection approach is shown in Fig. 1. The input information is the number of scrolls to be generated; then, a bi-objective optimization problem is encoded: (i) to maximize the value of

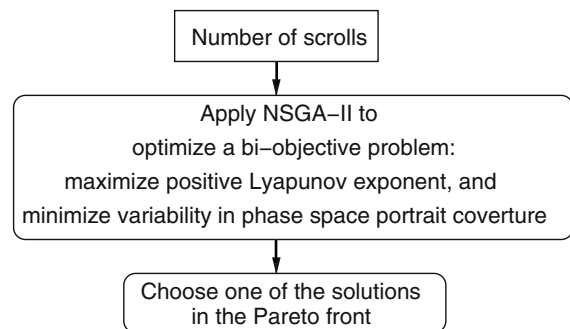


Fig. 1 Flowchart of the proposed optimization-selection procedure

MLE, and (ii) to minimize the variability in the oscillator’s PSP. Those objective functions are solved by using the NSGA-II algorithm [11].

For the case of the multiscroll chaotic oscillator based on saturated function series, the optimization problem is devoted to find the values of the four coefficient variables a, b, c and d_1 in (1) that solve both objectives (i) and (ii), listed above. Those four coefficients can take values within the range [0.0, 1.0]. In our investigation, we keep the value for those variables within four decimal places, i.e. from 0.0001 to 1.0000.

For the case of the multiscroll chaotic oscillator based on Chua’s diode (with s scrolls), we decided to optimize $s + 3$ oscillator’s values: we keep only two slope values (m_0 and m_1 for an even number of scrolls, and m_1 and m_2 for generating an odd number of scrolls) in (7); all the others slope values are copies on those two fixed ones. The number of break points b_i is equal to $s - 1$, and the values for α and β constants must be also optimized.

The MLE was measured like described in [4,5]. In the next subsection, our proposed procedure is described briefly. In addition, a very new procedure to measure the PSP coverage among all generated scrolls is included in the optimization loop. Basically, the procedure consists on counting the number of occurrences of the state trajectory in generating each scroll. Lets us explain this procedure with one example: Fig. 2a shows the PWL function to generate 4 scrolls, the procedure for distributing the trajectories in the PSP basically counts how many times the state variable, e.g. x , crosses the center of saturated levels (horizontal zones) of function f in this Fig. 2a at the set of values $x = \{-3, -1, 1, 3\}$. The taken quantitative measure, that represents the distribution in the PSP, is the standard deviation among all crossings values at the end of the simulation time. The counting procedure for this example is shown in Algorithm 1

For the case of Chua’s oscillator the procedure to count the state variable crossings is slightly different. In this case, the PSP coverage in variable x is divided in s zones, where s is the number of scrolls, and it is counting the number of times that variable y crosses the zero value inside each zone. And again the measure is the variance of the number of crossings inside all the zones, as is highlighted by Algorithm 2. As an example, in Fig. 2b is drawn the PWL function to generate 4-scrolls in Chua’s chaotic oscillator. There exist four zones and three thresholds must be calculated as the midpoint

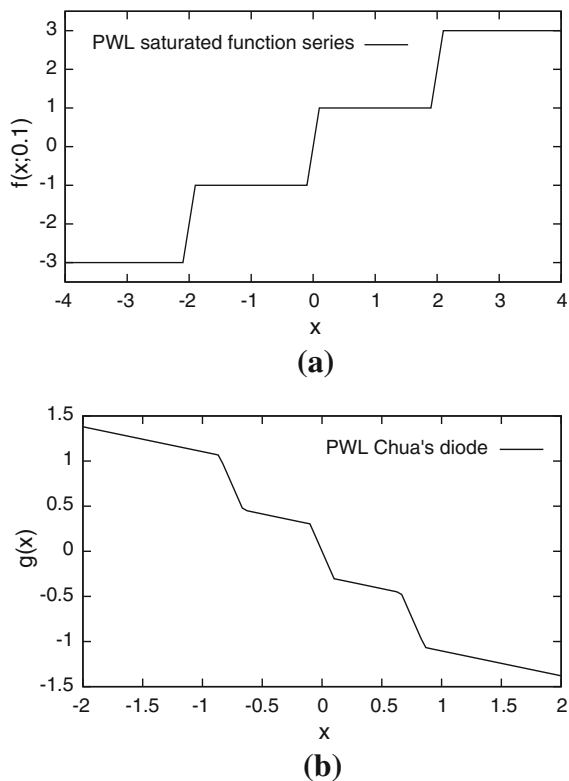


Fig. 2 Examples of two PWL functions for oscillators generating 4-scrolls: **a** saturated function series-based oscillator, and **b** Chua’s circuit oscillator

Algorithm 1 Pseudocode for counting the number of crossings for x state variable for a saturated function series generating 4 scrolls.

```

Ensure: The number of crossings are stored in array ‘phases[]’
1: oldState = 1; ▷ Initialize variable ‘oldState’
2: for Each simulation step do
3:   Simulate the oscillator a time  $t$ 
4:   if  $x < -3$  then
5:     newState = 0;
6:   else if  $x < -1$  then
7:     newState = 1;
8:   else if  $x < 1$  then
9:     newState = 2;
10:  else if  $x < 3$  then
11:    newState = 3;
12:  else ▷  $x > 3$ 
13:    newState = 4;
14:  if newState  $\neq$  oldState then
15:    if newState  $<$  oldState then
16:      phases[ newState ] ++;
17:    else
18:      phases[ oldState ] ++;
19:    oldState = newState;
    
```

inside the three slopes with greater value (as the PWL description is symmetric with the y axis in Fig. 2b, three break points are necessary), here $b_1 = 0.1$, $b_2 = 0.66$, and $b_3 = 0.86$, and the three thresholds are calculated as $q_1 = (-b_3 - b_2)/2 = -0.76$, $q_2 = (-b_1 + b_1)/2 = 0$, and $q_3 = (b_2 + b_3)/2 = 0.76$. The counting procedure for this case is shown in pseudocode 2.

Algorithm 2 Pseudocode for counting the number of crossings for state variable x and y for a Chua’s 4-scroll oscillator

```

Require:  $s$  is the number of scrolls
Ensure: The number of crossings are stored in array ‘phases[]’
1: yOldState = 1; ▷ Initialize variable ‘yOldState’
2: for Each simulation step do
3:   Simulate the oscillator a time  $t$ 
4:   flag = 0;
5:   for  $j = 1 : s$  do
6:     if  $x < q_j$  then
7:       xzone =  $j$ ;
8:       flag ++;
9:     break;
10:  if ! flag then ▷ ‘flag’ is equal to zero
11:    xstate =  $j$ ;
12:  if  $y < 0$  then
13:    yNewState = 0;
14:  else
15:    yNewState = 1;
16:  if yNewState  $\neq$  yOldState then phases[ xzone ] ++;
17:  yOldState = yNewState;
    
```

3.1 Computing Lyapunov exponents

Lyapunov exponents (LE) are asymptotic measures that characterize the average rate of growth (or shrinking) of small perturbations to the solutions of a dynamical system. LEs provide quantitative measures of response sensitivity of a dynamical system to small changes in initial conditions [20].

To measure the three LEs of the original chaotic oscillator system in (1) or (6), this original system is observed by expanding it with other three systems that change according to the derivative of (1) or (6), respectively. If $\mathbf{u} = [\dot{x}, \dot{y}, \dot{z}]^T$, $\mathbf{u} \in \mathbb{R}^3$, represents one state of the original dynamical system at any $t > 0$, the state of the new observed system will be $\mathbf{v} = [\mathbf{u}, \mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]^T$, $\mathbf{v} \in \mathbb{R}^{12}$, where \mathbf{u}_i , for $i = \{1, 2, 3\}$, are the three added systems that will measure precisely the change of those small perturbations on each orthogonal directions, for

each of the three state variables in (1) or (6). The initial state of the expanded system is set to

$$\begin{aligned} \mathbf{v}_0 &\in \mathbb{R}^{12} \\ \mathbf{v}_0 &= [\mathbf{u}_0^T, \mathbf{e}_1^T, \mathbf{e}_2^T, \mathbf{e}_3^T]^T \end{aligned} \tag{8}$$

where \mathbf{u}_0 is the vector of initial conditions $[x_0, y_0, z_0]^T$; $[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3] = I$, and I is the identity matrix of size 3×3 . Thus, \mathbf{e}_i , for $i = \{1, 2, 3\}$, are each unitary column vectors of the identity matrix I .

The observational system is integrated by several steps until a period T_O is reached. After this step, the state of the variational system is orthonormalized by using the standard Gram-Schmidt method [21]. The next integration is carried out by using the new orthonormalized vectors as initial conditions.

The LEs measure the long time sensitivity of the flow in \mathbf{u} with respect to the initial data \mathbf{u}_0 at the directions of every orthonormalized vector. This measure is taken when the variational system is orthonormalized. If $\mathbf{v} = [\mathbf{u}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3]^T$ is the state after the matrix $[\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]$ is orthonormalized, the LE λ_i , for $i = \{1, 2, 3\}$ is calculated by

$$\lambda_i \approx \frac{1}{T} \sum_{j=1}^k \ln \|\mathbf{p}_i\|, \tag{9}$$

where the number of summations k is calculated as $\lfloor \frac{T}{T_O} \rfloor$, and T is the simulation time.

As in [4], the period of time T_O is selected by using the minimum absolute value of all the eigenvalues of the system as

$$T_O = \frac{1}{l_{\min}}$$

where l_{\min} represents the value of the minimum eigenvalue of system in (1) or (6).

3.2 NSGA-II algorithm

In multiobjective problems, there exist several optimal solutions: one solution that improves one objective could not improve the other objective. Then, to compare two solutions, the Pareto dominance is used: one vector solution dominates other one if all its values are less or equal to the other, and at least in one element it is strictly lesser than the other. All the nondominated solutions form the set called ‘‘Pareto front’’.

In our problem, one could define a new single function that sum both objectives, perhaps weighting each objective, and optimizing this new single problem. This

solution leads to a single point in the Pareto front and in general it is not clear in which part of the Pareto front this new solution is.

The NSGA-II was proposed by Deb et al. [11] and is now a state of the art algorithm to solve multiobjective problems. The main goal in which NSGA-II was designed is to give and to spread all solutions on the Pareto front. As any evolutionary algorithm, it uses a set of solutions (or a set of vectors, called *population*) that is evolved by applying genetic operators (as mutation and crossover) to generate new solutions (a vector is called *individual*); and only the best solutions survive to the iteration of the algorithm. NSGA-II presents two ideas that try to improve the common performance of traditional multiobjective algorithms.

The *nondominated sort* is the first mechanism used in NSGA-II. It sorts the solutions according to Pareto nondomination. A rank value is created according to the number of individuals that dominated each solution. The nondominated solutions receive a rank equal to one, and the others receive a rank value according to how many subsets they dominate.

Also, to preserve a good spread of the final solutions, NSGA-II uses the second mechanism called *crowding distance*. This distance measures the average size of the cuboid formed with the points that enclose a solution in the population.

The main idea behind NSGA-II is to use a selection that preserves the individuals with the lowest rank value. When almost all the solutions exist in the first rank, the algorithm selects the solutions with the highest value of the crowding distance. Algorithm 3 describes how NSGA-II evolves the solutions.

Algorithm 3 Pseudocode of NSGA-II

- Require:** number of generations G , number of individuals
- 1: Randomly generate the population P_0
 - 2: Evaluate the population
 - 3: Apply genetic operators in P_0 to generate Q_0
 - 4: **for** $i := 0 : G - 1$ **do**
 - 5: Set $R_i = P_i \cup Q_i$
 - 6: Calculate the rank value of R_i
 - 7: Calculate the crowding distance of R_i
 - 8: $P_i \leftarrow$ select the N individuals with the lowest rank and highest crowding distance
 - 9: Apply genetic operators in P_i to generate Q_i
-

4 Results

For all the simulations, NSGA-II was executed by using a population of 100 individuals, crossover probability

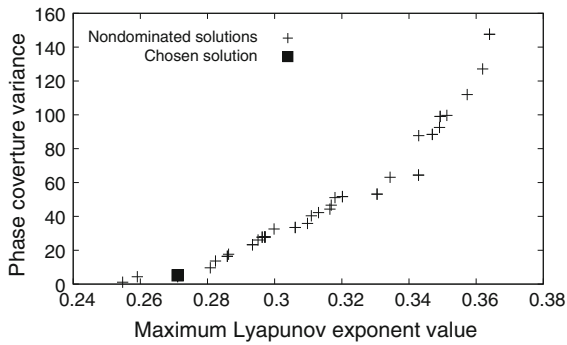


Fig. 3 Solutions obtained for a 4-scroll chaotic oscillator. The final chosen solution is marked with ■

of 0.9 and mutation probability of 0.5. NSGA-II was run three times with different seeds for the random number generator. From the three runs, the nondominated solutions were calculated again and only one solution was chosen from the Pareto front. We use the fourth order Runge–Kutta integrator with a fixed inte-

gration time step of 0.01 sec. T_O was rounded to the values nearest multiple of 0.01. Also, the initial condition was set for all the simulations to $\mathbf{u}_0 = [0.1, 0.0, 0.0]^T$.

4.1 Saturated function series-based oscillators

For this type of oscillator, the simulations were executed first for 400 sec and then for another 3,500-s duration, where the Lyapunov exponents were measured.

As an example of our procedure, in Fig. 3, we show the solutions found for a 4-scroll saturated function series-based chaotic oscillator. The solutions are in the Pareto front trading both objectives: the value of the MLE and the PSP coverage. The final chosen solution is marked with symbol ■, it was the one with nearest and below the value of 10 in its PSP coverage. Phase portraits of the two outermost solutions in Fig. 3 are shown in Fig. 4: the oscillator with the lesser MLE value and minimum variance in PSP is shown in Fig. 4a, and

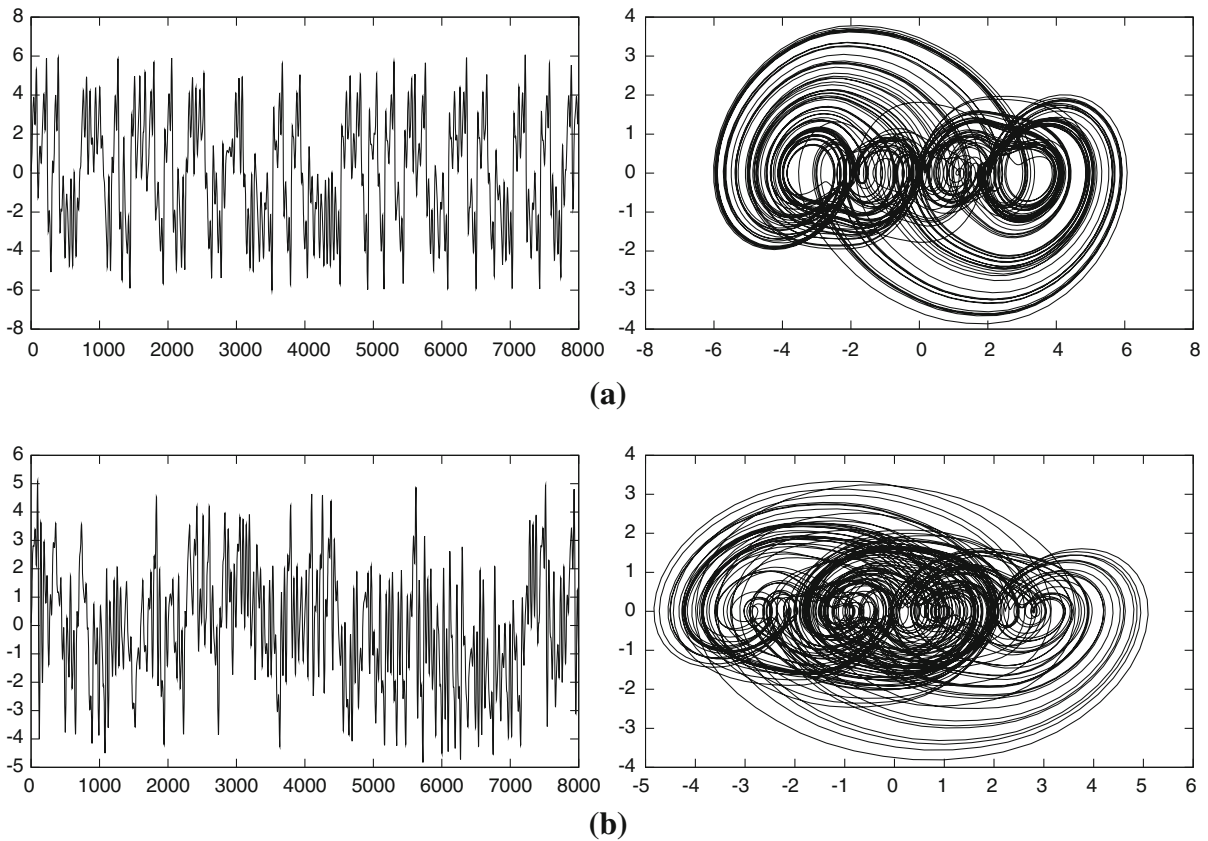


Fig. 4 PSPs of the two outermost points in Fig. 3. Details in text

Table 1 Coefficient values, MLE values, and standard deviation, of PSP coerture [σ (PSPC)] for nonoptimized and optimized saturated function series chaotic oscillators

Osc.	Scr.	Coefficients [a, b, c, d_1]	MLE	σ
1	2	0.7000 0.7000 0.7000 0.7000	0.11177	8
2	2	0.9997 0.8158 0.5369 0.7387	0.23998	0
3	3	0.7000 0.7000 0.7000 0.7000	0.14263	214
4	3	0.8116 0.7550 0.3641 0.9900	0.25217	6
5	4	0.7000 0.7000 0.7000 0.7000	0.16535	122
6	4	0.9225 0.8420 0.3577 0.9909	0.27110	4
7	5	0.7000 0.7000 0.7000 0.7000	0.15667	89
8	5	0.9059 0.8151 0.2153 0.9950	0.27487	8
9	6	0.7000 0.7000 0.7000 0.7000	0.17212	69
10	6	0.9043 0.6229 0.2600 0.9919	0.28405	7

Table 2 Counting values, means, and standard deviations, σ , of the phase transitions for the nonoptimized and optimized saturated function series chaotic oscillators showed in Table 1

Oscillator	Counting values	Mean	σ
1	472, 488	480	8
2	502, 502	502	0
3	188, 611, 130	310	214
4	464, 477, 466	469	6
5	80, 325, 368, 134	227	122
6	346, 342, 334, 338	340	4
7	96, 272, 255, 233, 56	182	89
8	348, 344, 342, 338, 324	339	8
9	64, 229, 167, 143, 235, 66	151	69
10	307, 321, 305, 323, 318, 320	316	7

the oscillator with greater MLE and maximum variance in PSP coerture is shown in Fig. 4b. The chaotic oscillator corresponding to diagrams in Fig. 4a was obtained with optimized coefficients $a = 0.8658$, $b = 0.6964$, $c = 0.4116$, and $d_1 = 0.9995$, and the one in Fig. 4b was obtained with coefficient values $a = 0.9944$, $b = 0.7339$, $c = 0.2532$, and $d_1 = 0.9992$. As one sees, both objectives are in conflict, thus one can select the appropriated one from the Pareto front in Fig. 3.

Table 1 shows the chosen solutions and their comparison with the traditional design of the multiscroll chaotic oscillator using the coefficients values [0.7, 0.7, 0.7, 0.7]. In Table 2 are shown the counting values that help us estimate the distribution of the trajectories in the PSP for each oscillator in Table 1. In Fig. 5, one can see the chaotic oscillator behavior when generating from 2 to 6 scrolls. It is pretty clear from this figure that the optimized oscillators not only present a better chaotic behavior, but also the distributions of the trajectories among the scrolls is well done.

To have more clarity in the description of the distribution of trajectories vs the MLE, in Fig. 6, we show how the surface of the search space is for the 4-scroll oscillator optimized by NSGA-II.

4.2 Chua’s circuit chaotic oscillator

All the simulations of this type of oscillator were executed during 5 s, and then additional 3,200 s time was needed for measuring the Lyapunov exponents.

As an example of the suggested procedure for this class of oscillator, Fig. 7 shows all the nondominated solutions found for a Chua’s 4-scroll oscillator. As one sees from this figure, both objectives are in conflict, and thus one needs to select the appropriate solution from the Pareto front. In this case, the solution with value PSP coerture nearer and below to the value of 10 was chosen, and it is marked with symbol ▲ in Fig. 7. Phase portraits of the two outermost solutions in Fig. 7 are shown in Fig. 8: the oscillator with the lesser MLE

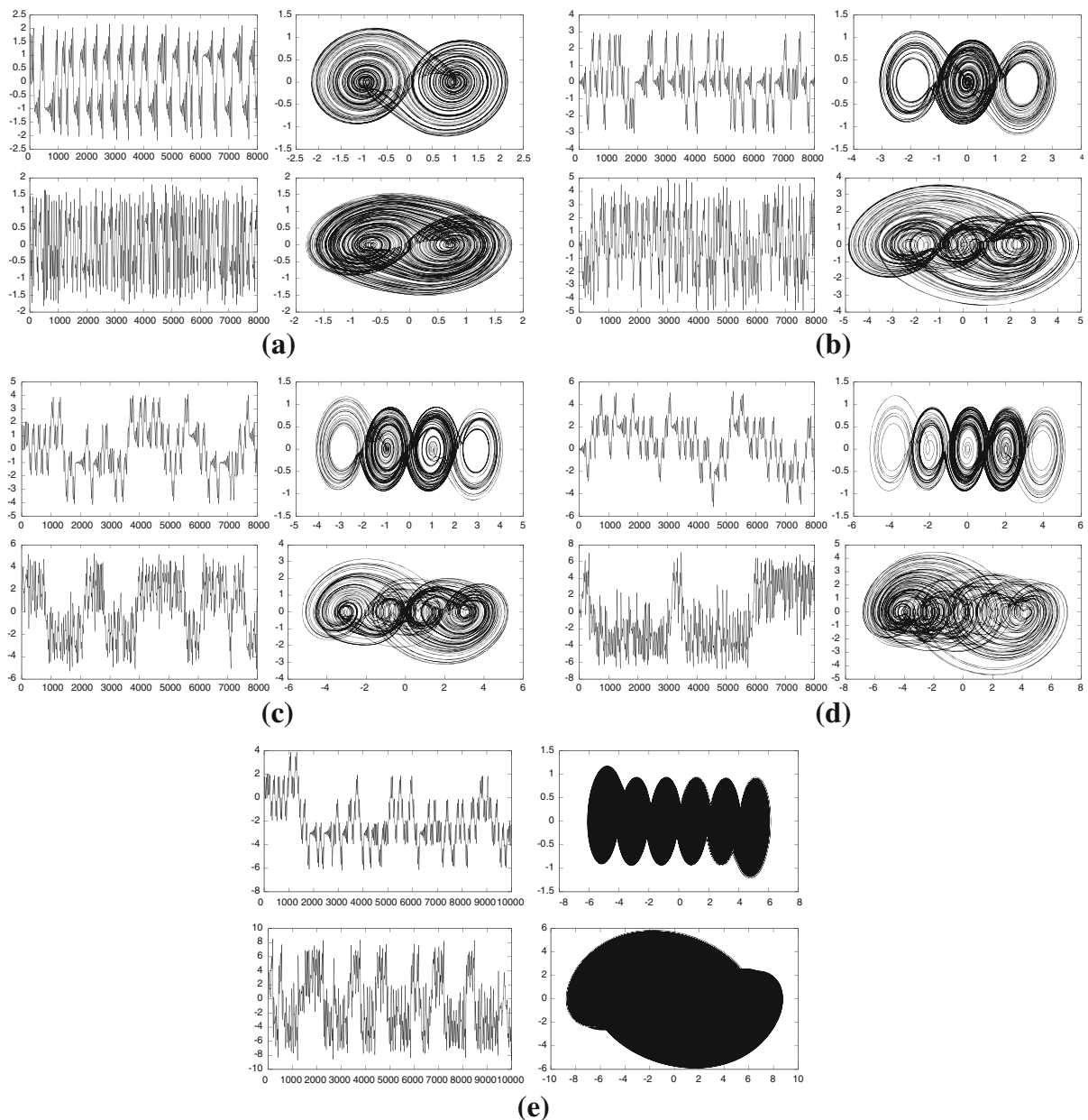


Fig. 5 Diagrams of the cases in Table 1. For each case, the *upper row* and the *bottom row* correspond to the nonoptimized and the optimized ones, respectively. Two diagrams are shown: the x val-

value, and the minimum variance in PSP is shown in Fig. 8a; and the oscillator with greater MLE and maximum variance in PSP coverage is shown in Fig. 8b. The chaotic oscillators corresponding to diagrams in Fig. 8a were obtained with optimized variables $m_0 = -2.234$, $m_1 = -0.329$, $b_1 = 0.392$, $b_2 = 1.386$, $b_3 = 1.492$, $\alpha = 9.012$, and $\beta = 13.128$; the ones in Fig. 8b

were obtained with variable values $m_0 = -1.148$, $m_1 = -0.370$, $b_1 = 0.400$, $b_2 = 0.618$, $b_3 = 1.450$, $\alpha = 11.874$, and $\beta = 14.192$.

In Fig. 9 are shown the PSPs for the chaotic oscillators before and after the optimization. For the nonoptimized oscillators in Table 3 and shown in Fig. 9, for generating an even number of scrolls, the values $m_0 =$

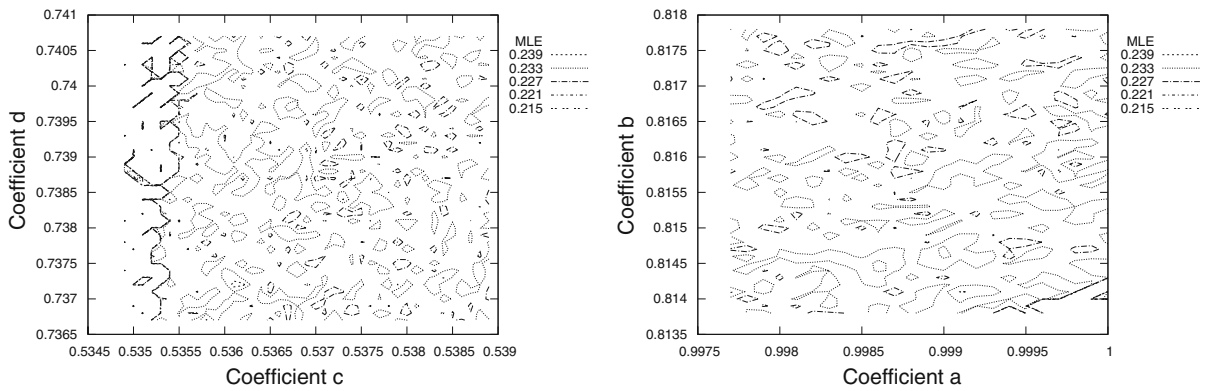


Fig. 6 The search space around the solution (a, b, c, d_1) 0.9997, 0.8158, 0.5369, and 0.7387 for the optimized 4-scroll oscillator. The figure to the *left* was made for 0.9997, 0.8158, 0.5369 ± 0.0020 , and 0.7387 ± 0.0020 , and the figure to the *right* was made with values $[1.0 - 0.9977]$, 0.8158 ± 20 , 0.5369, and 0.7387. The

solution is a local optimum in a zone where the MLE value varies in the range $[0.215-0.245]$. In two zones, one to the *left* on first figure, and the other in *bottom-right* in figure to the *right*, the MLEs do not exist

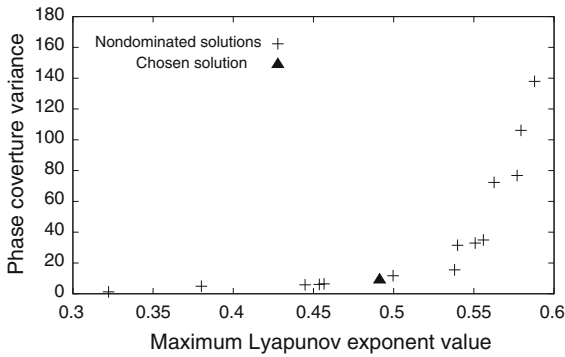


Fig. 7 Solutions obtained for a Chua’s 4-scroll chaotic oscillator. The final chosen solution is marked with \blacktriangle

-3.036 , $m_1 = -0.276$, $b_1 = 0.1$, $\alpha = 10$, and $\beta = 15$ were used. For generating an odd number of scrolls, the same values were used except $m_1 = -0.276$ and $m_2 = -3.036$, and the initial break point is $b_2 = 0.8$. The subsequent break points are calculated as in [22], evaluating

$$b_{i+1} = \frac{2 \sum_j^i (m_j - m_{j-1}) b_j}{1 + m_i} - b_i, \tag{10}$$

where the slopes with an even index are equal to m_0 (or m_2), and those with the odd ones are equal to m_1 . In (10), j is equal to 1 for an even scroll number, and j is equal to 2 for an odd scroll number.

Finally, in Table 3 are shown the variable values for the nonoptimized and optimized oscillators, and in Table 4 are shown the counting values that help us esti-

mate the distribution of trajectories in the PSP for each oscillator in Table 3.

5 Discussion

The optimization problem for the saturated function series oscillators is encoded to search for the feasible values of variables a, b, c , and d_1 to design a more robust chaotic oscillator. The search space for a single variable is $2 \times 10 \times 10 \times 10 \times 10$ (first digit can take two values 0 or 1, and each one of the other four decimal places can take ten values, from 0 to 9), equal to 2×10^4 . The total search space is (for four variables) $(2 \times 10^4)^4$, or equal to 1.6×10^{17} . Because of this huge search space, the use of heuristics, as the NSGA-II algorithm, is well justified.

For the Chua’s oscillator, the optimization problem is more difficult because it is search for $s + 3$ variables, where s is the number of scrolls; therefore, five variables are optimized for 2-scroll oscillator, and up to 9 variables for a 6-scroll oscillator. The methodology presented in this article opens the possibility to design oscillators with different values for each slope, and also for different values of the break points, so that the PWL function for the Chua’s diode will increase in a nonsystematic way, but the description remains the same as for (7).

After performing the optimization process with NSGA-II, several feasible solutions of the bi-objective optimization problem are in the Pareto front; however,

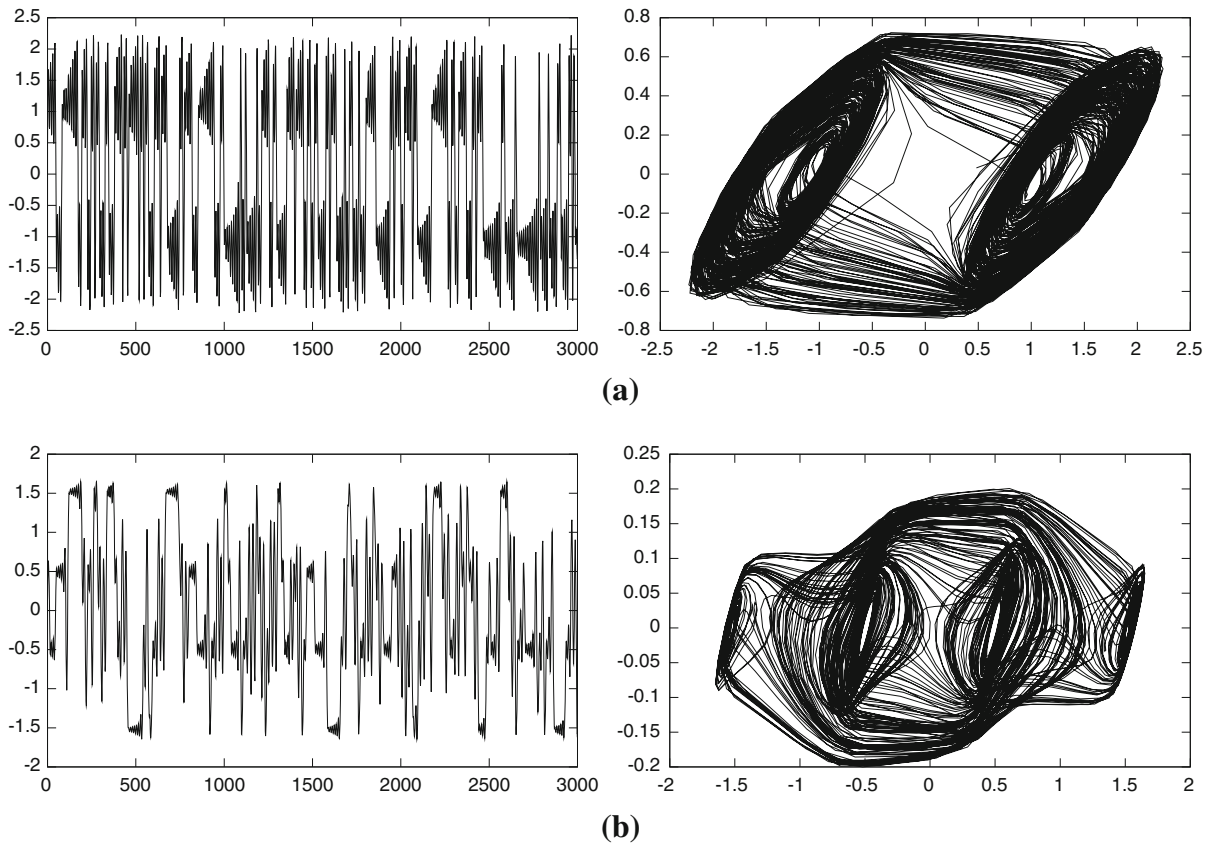


Fig. 8 PSPs of the two outermost points in Fig. 7. Details in text

we decided to take a solution with a low variance in the PSP distribution, and it was with a value around 10.

It is clear by observing the graph in Fig. 3 that the two objectives used in our proposed selection procedure—the value of the oscillator’s MLE, and the variability in the oscillator’s PSP—are in conflict. This means that an oscillator with high variability in the oscillator’s trajectories has a higher value in its MLE, and vice versa; the PSPs of these two extreme oscillators are shown in Fig. 4. If both objectives have not in conflict at all, then a single solution should be obtained, meaning that the optimization problem could be solved with any single objective solver and not with a multiobjective solver.

All our results show new optimized coefficient values of the multiscroll chaotic oscillators with higher MLE and low variance in the PSP, as can be verified from Tables 1 and 3. Although there is no rule for selecting a solution from the Pareto front, the elec-

tronic implementation can be performed considering the trade-off between the values of the MLE and the variance for the PSP coverage. In this manner, one can select the appropriate one according to the application at hand. For instance, as inferred from the Pareto fronts, a design showing a good distribution in the PSP, i.e., a low standard deviation will have a low value for the MLE. The unpredictability may not be robust, but chaos is guaranteed, and the oscillator will show all scrolls uniformly. On the other hand, when selecting a solution with the maximum MLE, the scrolls will not appear uniformly and in the worst case, one scroll may appear just for a few times. This leads us to suppose how to implement a random number generator: for instance, where one can trade unpredictability versus uniformity in the PSP. In this case, our contribution on identifying this trade-off between the values of the MLE and its PSP coverage provides a solution by applying evolutionary algorithms, e.g. NSGA-II.

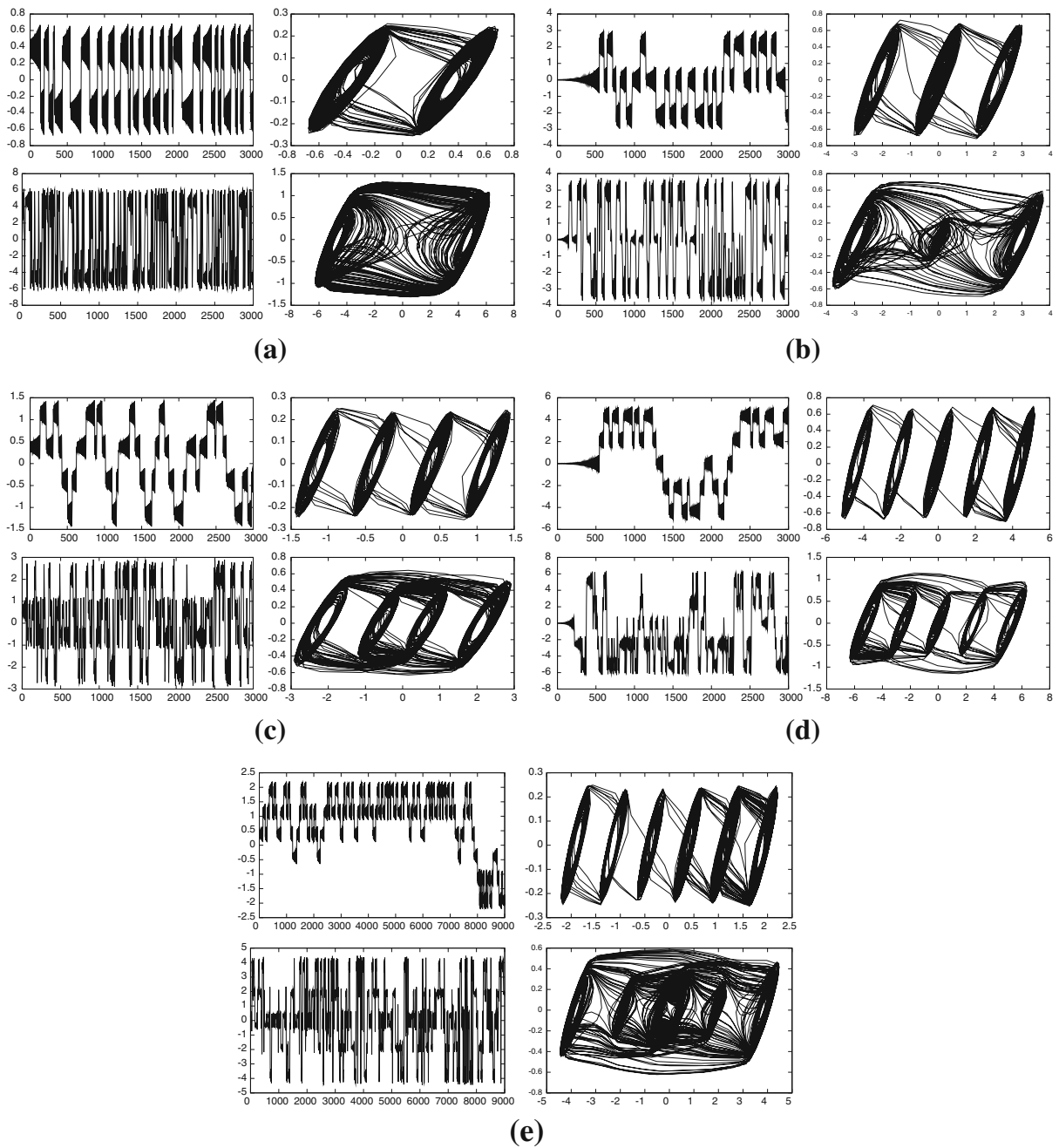


Fig. 9 Diagrams of the cases in Table 3 for Chua’s circuit chaotic oscillators. For each case, the *upper row* and the *bottom row* correspond to the nonoptimized and the optimized one, respectively.

Two diagrams are shown: the *x* values against time, and the PSP between *x*-*y* state variables. **a** Two scrolls, **b** three scrolls, **c** four scrolls, **d** five scrolls, and **e** six scrolls

6 Conclusion

A methodology to select the best parameter values of chaotic oscillators, based on saturated function series

and in Chua’s circuit, to guarantee chaotic regime by maximizing the MLE and higher distribution of scrolls in the PSP coverage, has been proposed. We demonstrated that both objectives are in conflict. We proposed

Table 3 Comparison between nonoptimized and optimized coefficients for the Chua’s circuit chaotic oscillator

Osc.	Scrolls	Variable values			MLE	σ (PSPC)
		Slopes m_0 (or m_2), m_1	Break points	Coef. (α , β)		
11	2	-3.036, -0.276	0.1	10, 15	0.21256	9.0
12	2	-1.131, -0.395	2.618	11.991, 14.997	0.52559	7.5
13	3	-3.036, -0.276	0.8, 1.37	10, 15	0.21121	272
14	3	-1.220, -0.385	0.234, 2.394	10.297, 12.565	0.43266	4
15	4	-3.036 - 0.276	0.1, 0.66, 0.86	10, 15	0.16851	241
16	4	-3.981 - 0.397	0.101, 1.206, 1.467	11.489, 14.674	0.49122	9
17	5	-3.036, -0.276,	0.8, 1.37, 2.97, 3.54	10, 15	0.22363	62
18	5	-2.813, -0.356,	0.893, 1.559, 3.432, 4.114	10.227, 11.786	0.58500	11
19	6	-3.036, -0.276	0.10, 0.66, 0.86, 1.42, 1.62	10, 15	0.12587	227
20	6	-1.464, -0.381	0.19, 0.70, 1.58, 2.21, 3.31	10.556, 11.415	0.43326	17

Table 4 Counting values, their means, and standard deviations (σ) of the phase transitions for the nonoptimized and optimized Chua’s circuit chaotic oscillators shown in Table 3

Oscillator	Counting values	Mean	σ
11	1833, 1815	1824.0	9.0
12	1411, 1426	1418.5	7.5
13	996, 1603, 1061	1220	272
14	900, 908, 908	905	4
15	597, 897, 1276, 887	914	241
16	784, 780, 771, 761	774	9
17	620, 745, 774, 801, 716	731	62
18	585, 579, 583, 558, 587	578	11
19	942, 862, 379, 357, 641, 475	609	227
20	400, 434, 403, 404, 434, 438	419	17

a new procedure to take the measurement of the PSP coverage. In this manner, we applied NSGA-II to solve a biobjective optimization problem encoded to maximize the value of MLE exponent and minimize the variability in the oscillator’s phase transitions. The proposed approach has been demonstrated for the optimization of two kinds of chaotic oscillators: one based on saturated functions series, and other one based on Chua’s circuit.

Both oscillators are based on PWL functions that, in all the reported studies, there is an increase in the number of their segments in a systematic way according to the number of scrolls to be generated. However, as mentioned in Sect. 2, both PWL functions described by (5) and (7) can be increased in a nonsystematic way. It means that for the saturated function series, the saturated levels, the slopes, and the shifted values can be all different, as well as for Chua’s diode where the negative slopes and break points can be different. Further,

by applying the proposed approach of this study, one may find optimal solutions that maximize the MLE and minimize the dispersions in the PSPs in a better way. Finally, in Figures 5 and 9, we list the optimized results for both kinds of oscillators showing a high MLE and a very good distribution of the trajectories in the PSP.

Acknowledgments This study is partially supported by the CONACyT-Mexico under project numbers CB2011/168357 and 131839-Y

References

1. Trejo-Guerra, R., Tlelo-Cuautle, E., Carbajal-Gomez, V.H., Rodriguez-Gomez, G.: A survey on the integrated design of chaotic oscillators. *Appl. Math. Comput.* **219**(10), 5113–5122 (2013)
2. Sanchez Lopez, C.: A 1.7 MHz chua’s circuit using VMs and CF+s. *Rev. Mex. Fis.* **58**, 86–93 (2012)

3. Trejo-Guerra, R., Tlelo-Cuautle, E., et al.: Integrated circuit generating 3- and 5-scroll attractors. *Commun. Nonlinear Sci. Numer. Simul.* **17**(11), 4328–4335 (2012)
4. Carbajal-Gomez, V.H., Tlelo-Cuautle, E., Fernandez, F.V.: Optimizing the positive Lyapunov exponent in multi-scroll chaotic oscillators with differential evolution algorithm. *Appl. Math. Comput.* **219**(15), 8163–8168 (2013)
5. Yang, C.J., Zhu, W.D., Ren, G.X.: Approximate and efficient calculation of dominant Lyapunov exponents of high-dimensional nonlinear dynamic systems. *Commun. Nonlinear Sci. Numer. Simul.* **18**(12), 3271–3277 (2013)
6. Caballero, M.V., Balibrea, F.: Stability of orbits via Lyapunov exponents in autonomous and nonautonomous systems. *Int. J. Bifurcation Chaos* **23**(07), 1350127 (2013)
7. Fazanaro, F.I., Soriano, D.C., Suyama, R., Attux, R., Madrid, M.K., de Oliveira, J.R.: Characterization of multiscroll attractors using Lyapunov exponents and lagrangian coherent structures. *Chaos: An Interdisciplinary. J. Nonlinear Sci.* **23**(2), 023105 (2013)
8. Li, Chunbiao, Sprott, J.C.: Amplitude control approach for chaotic signals. *Nonlinear Dyn.* **73**(3), 1335–1341 (2013)
9. Sun, Y., Wu, C.: A radial-basis-function network-based method of estimating Lyapunov exponents from a scalar time series for analyzing nonlinear systems stability. *Nonlinear Dyn.* **70**(2), 1689–1708 (2012)
10. Li, S.-Y., Huang, S.-Ch., Yang, C.-H., Ge, Z.-M.: Generating tri-chaos attractors with three positive Lyapunov exponents in new four order system via linear coupling. *Nonlinear Dyn.* **69**(3), 805–816 (2012)
11. Deb, K., Pratap, A., Agarwal, S.: A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
12. Lu, J., Chen, G.: Generating multiscroll chaotic attractors: Theories, methods and applications. *Int. J. Bifurcation Chaos* **16**, 775–858 (2006)
13. Lu, J., Yu, S., Leung, H., Chen, G.: Experimental verification of multidirectional multiscroll chaotic attractors. *IEEE Trans. Circuits Syst. I* **56**, 149–165 (2006)
14. Munoz-Pacheco, J.M., Zambrano-Serrano, E., Felix-Beltran, O., Gomez-Pavon, L.C., Luis-Ramos, A.: Synchronization of PWL function-based 2D and 3D multi-scroll chaotic systems. *Nonlinear Dyn.* **70**(2), 1633–1643 (2012)
15. Munoz-Pacheco, J.M., Zambrano-Serrano, E., Felix-Beltran, O.G., Tlelo-Cuautle, E., Gomez-Pavon, L.C., Trejo-Guerra, R., Luis-Ramos, A., Sanchez-Lopez, C.: In: *Selected Topics in Nonlinear Dynamics and Theoretical Electrical Engineering*, volume 459 of *Studies in Computational Intelligence*, Chapter On the Synchronization of 1D and 2D Multi-scroll Chaotic Oscillators, pp. 19–40. Springer (2013)
16. Mata-Machuca, J.L., Martinez-Guerra, R., Aguilar-Lopez, R.: Chaotic systems synchronization via high order observer design. *J. Appl. Res. Technol.* **9**(1), 57–68 (2011)
17. Pehlivan, İ., Çiçek, S., Uyaroglu, Y.: Simulation and circuit implementation of sprott case H chaotic system and its synchronization application for secure communication systems. *J. Circuits Syst. Comput.* **22**(04), 1350022 (2013)
18. Gamez-Guzman, L., Cruz-Hernández, C., Lopez-Gutierrez, R.M., Garcia-Guerrero, E.E.: Synchronization of multi-scroll chaos generators: application to private communication. *Rev. Mex. Física* **54**(4), 299–305 (2008)
19. Zhong, G.-Q., Man, K.-F., Chen, G.: A systematic approach to generating n -scroll attractors. *Int. J. Bifurcation Chaos* **12**(12), 2907–2915 (2002)
20. Dieci, L.: Jacobian free computation of Lyapunov exponents. *J. Dyn. Differ. Equ.* **14**(3), 697–717 (2002)
21. Golub, G.H., Loan, C.V.: *Matrix Computations*, 3rd edn. The Johns Hopkins University Press, Baltimore (1996)
22. Yu, S.M., Qiu, S.S., Lin, Q.H.: New results of study on generating multiple-scroll chaotic attractors. *Sci. China Ser. F* **46**, 104–115 (2003)