Chaos Enhanced Differential Evolution in the Task of Evolutionary Control of Discrete Chaotic Lozi Map

Roman SENKERIK[1,2], *Ivan ZELINKA*[1,3], *Michal PLUHACEK*[2]

[1]Modeling Evolutionary Algorithms Simulation and Artificial Intelligence, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, 19 Nguyen Huu Tho Street, 700000 Ho Chi Minh City, Vietnam
[2]Department of Informatics and Artificial Intelligence, Faculty of Applied Informatics, Tomas Bata University in Zlin, T. G. Masaryka 5555, 760 01 Zlin, Czech Republic
[3]Department of Computer Science, Faculty of Electrical Engineering and Computer Science, VSB–Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava, Czech Republic

senkerik@fai.utb.cz, ivan.zelinka@vsb.cz, pluhacekk@fai.utb.cz

DOI: 10.15598/aeee.v14i3.1166

Abstract. *In this paper, evolutionary technique Differential Evolution (DE) is used for the evolutionary tuning of controller parameters for the stabilization of selected discrete chaotic system, which is the two-dimensional Lozi map. The novelty of the approach is that the selected controlled discrete dissipative chaotic system is used within Chaos enhanced heuristic concept as the chaotic pseudo-random number generator to drive the mutation and crossover process in the DE. The idea was to utilize the hidden chaotic dynamics in pseudo-random sequences given by chaotic map to help Differential evolution algorithm in searching for the best controller settings for the same chaotic system. The optimizations were performed for three different required final behavior of the chaotic system, and two types of developed cost function. To confirm the robustness of presented approach, comparisons with canonical DE strategy and PSO algorithm have been performed.*

Keywords

Differential evolution, deterministic chaos, chaos control, optimization.

1. Introduction

In many applications, one of the most challenging tasks is the controlling of highly nonlinear dynamical systems in order to either eliminate or synchronize the chaos. The first successful approach to control chaotic dynamics by means of a simple linearization technique was introduced in the 1990s by Ott, Grebogy and Yorke (i.e. OGY method) [1]. Later, rapid development of methods for stabilizing chaotic dynamics has arisen, and more advanced modern techniques have been applied for chaos control and synchronization including unconventional methods from the soft computing field.

The most current intelligent methods are mostly based on soft computing, which is a discipline tightly bound to computers, representing a set of methods including special algorithms, belonging to the artificial intelligence paradigm. The most popular of these methods are neural networks, Evolutionary Algorithms (EA's) and fuzzy logic. Currently, EA's are known as a powerful set of tools for almost any difficult and complex optimization problem.

The interest about the connection between evolutionary techniques and (not only) control of chaotic systems is rapidly spreading. The initial research was conducted in [2], whereas [3] and [4] was more concerned with the tuning of parameters inside the existing chaos control technique based on the Pyragas Extended Delay Feedback Control (ETDAS), [5]. Later works [6], [7], and [8] introduce a novel approach of generating the entire control law (control method) for the purpose of stabilization of any chaotic system.

Other approaches utilizing the EA's for the stabilizing of chaotic dynamics have mostly applied the Particle Swarm Optimization algorithm (PSO), [9], and multi-interval gradient-method [10] or minimum entropy control technique [11]. EA's have been also frequently used in the task of synchronization of chaos [12], [13] and [14]. In [15] an EA for optimizing local control of chaos based on a Lyapunov approach is presented. Another example of the connection between deterministic chaos and EA's represents the embedding of chaotic dynamics into the EA's. Recent research has proven that chaotic approach is able to bypass local optima stagnation. A chaotic approach uses any chaotic system in the place of a pseudo-random number generator [16]. This causes the heuristic to map unique regions since the chaotic map iterates to new regions due to the basic property of deterministic chaos, which is the density of periodic orbit.

The initial concept of embedding chaotic dynamics into EA's is given in [17]. Later, the initial study [18] was focused on the simple embedding of chaotic systems in the form of Chaos Pseudo-Random Number Generator (CPRNG) for Differential Evolution (DE), [19], and Self Organizing Migrating Algorithm (SOMA), [20], in the task of optimal PID tuning. Also, the PSO algorithm with elements of chaos was introduced as the CPSO [21]. This field of research was later extended with the successful experiments with chaos driven DE [22] in real domain as well as in combinatorial problems domain [23] and [24].

At the same time, the chaos embedded PSO with inertia weight strategy was closely investigated [25], followed by the introduction chaotic firefly algorithm [26]. The organization of this paper is as follows: firstly, used evolutionary technique, which is DE, is described, and followed by the description of the ChaosDE concept. Thereafter, the problem design and appropriate corresponding cost functions are investigated and proposed. Results and conclusion follow afterward.

2. Motivation

This paper extends the research of evolutionary chaos control optimization by means of ChaosDE algorithm [27]. In this paper the DE/rand/1/bin strategy driven by different chaotic map (system) was utilized to solve the issue of evolutionary optimization of chaos control for the same chaotic system used as a CPRNG in the particular case study. Thus, the idea was to utilize the hidden chaotic dynamics in pseudo-random sequences given by chaotic map to help Differential evolution algorithm in searching the best controller settings for the very same chaotic system. Since the positive contribution of the chaotic dynamics to the performance of DE in the task of evolutionary chaos control optimization was proven in comparison with original canonical DE within the initial study [28], this paper is not primarily focused on the performance comparisons with the different heuristic.

This research extends the initial work with the aforementioned idea and with the several case studies combining different required states of the system (i.e. different Unstable Periodic Orbits - UPOs) and different utilized cost functions.

3. Used Heuristic - Differential Evolution

DE is a simple and powerful population-based optimization method that works either on real-number-coded individuals or with small modifications on discrete type individuals [19], [29] and [30]. DE is quite robust, fast, and effective, with global optimization ability. This global optimization ability has been proven in many interdisciplinary types of research. It works well even with noisy and time-dependent objective functions. Recently hybridized DE strategies have been developed [31], [32] and also self-adaptive DE variants [33], [34] and [35] have proven to be powerful heuristics. Basic canonical principle is following.

For each individual $x_{i,\vec{G}}$ in the current generation G, DE generates a new trial individual $x'_{i,\vec{G}}$ by adding the weighted difference between two randomly selected individuals $x_{r1,\vec{G}}$ and $x_{r2,\vec{G}}$ to a randomly selected third individual $x_{r3,\vec{G}}$. The resulting individual $x'_{i,\vec{G}}$ is crossed-over with the original individual $x_{i,\vec{G}}$. The fitness of the resulting individual, referred to as a perturbed vector $u_{i,\vec{G}+1}$, is then compared with the fitness of $u_{i,\vec{G}}$. If the fitness of $u_{i,\vec{G}+1}$ is greater than the fitness of $x_{i,\vec{G}}$, then $x_{i,\vec{G}}$ is replaced with $u_{i,\vec{G}+1}$; otherwise, $x_{i,\vec{G}}$ remains in the population as $x_{i,\vec{G}+1}$.

Please refer to Eq. (1) for notation of crossover, and to [19] for the detailed description of used DERand1Bin strategy and all other DE strategies:

$$u_{i,\vec{G}} = x_{r1,\vec{G}} + \mathbf{F}\left(x_{r2,\vec{G}} - x_{r3,\vec{G}}\right). \qquad (1)$$

4. Concept of ChaosDE

This section contains the description of discrete dissipative chaotic map, which can be used as the chaotic pseudo-random generators for DE as well as the main principle of the ChaosDE concept. In this research, direct output iterations of the chaotic map were used for the generation of pseudo random numbers. Two types of numbers are required: real numbers in the process of crossover based on the user defined CR value and integer values used for selection of individuals.

The general idea of ChaosDE and CPRNG is to replace the default PRNG with the discrete chaotic map. Since the discrete chaotic map is a set of equations with a static start position, a random start position of the map is created in each run of EA, in order to have different start position for different experiments (runs of EA's). This random position is initialized with the default PRNG, as a one-off randomizer. Thus, the concept of ChaosDE is utilizing the famous and well-known "butterfly effect" i.e. extreme sensitivity of chaotic system to the initial conditions.

As two different types of numbers are required in ChaosDE; real and integers, the use of modulo operators is used to obtain values between the specified ranges, as given in the following Eq. (2) and Eq. (3):

$$rndreal = \mathrm{mod}\left(\mathrm{abs}\left(rndChaos\right), 1.0\right), \qquad (2)$$

$$rndint = \mathrm{mod}\left(\mathrm{abs}\left(rndChaos\right), 1.0\right) \times Range + 1, \quad (3)$$

where abs refers to the absolute portion of the chaotic map generated number *rndChaos*, and mod is the modulo operator. *Range* specifies the value (inclusive), where the number is to be scaled.

5. Lozi Map

This section contains the mathematical and graphical description of the selected discrete dissipative system, which serves both as for CPRNG and also as the example of the system to be evolutionary controlled.

The Lozi map is a simple discrete two-dimensional chaotic map. The map equations are given in Eq. (4). The parameters used in this work are $a = 1.7$ and $b = 0.5$ as suggested in [36]. For these values, the system exhibits typical chaotic behavior and with this parameter setting it is used in the most research papers and other literature sources [37]:

$$\begin{array}{c} X_{n+1} = 1 - a\left|X_n\right| + b \cdot Y_n \\ Y_{n+1} = X_n \end{array}. \qquad (4)$$

The x, y plot of the selected map is depicted in Fig. 1. The chaotic behavior of the chaotic map, represented by the example of direct output iterations is depicted in Fig. 2, whereas the Fig. 3 shows the example of chaotic dynamics transferred into the range $\langle 0, -1 \rangle$. Finally, the illustrative histogram of the distribution of real numbers transferred into the range $\langle 0, -1 \rangle$ generated by means of chaotic Lozi map is shown in Fig. 4.

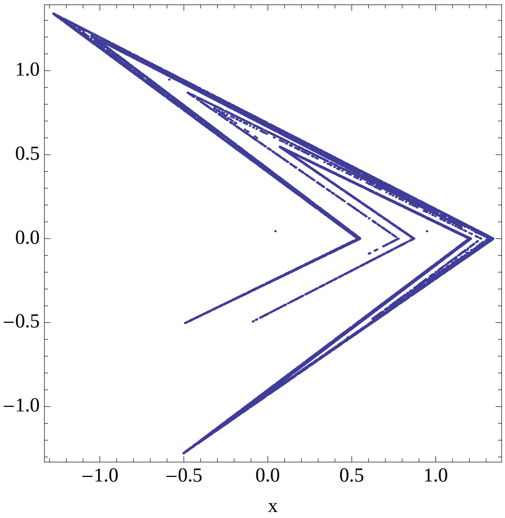

Fig. 1: x, y plot of the Lozi map.

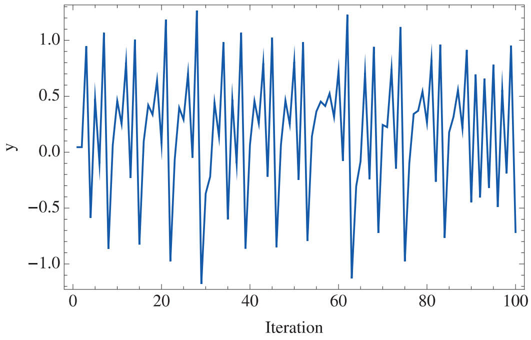

Fig. 2: Iterations of the uncontrolled Lozi map (variable x).

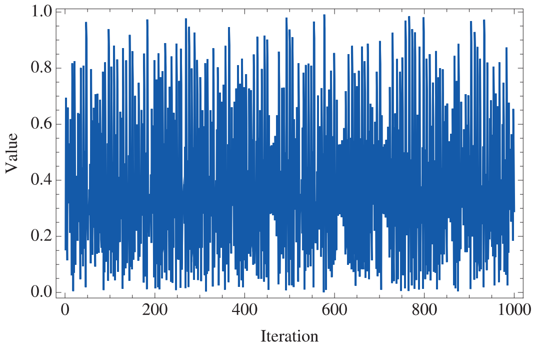

Fig. 3: Example of the chaotic dynamics: range $\langle 0, -1 \rangle$ generated by means of the Lozi map.

6. Cost Function Design

The idea of the basic cost function (CF_{Simple}), which could be used problem-free only for the stabilization of $p-1$ orbit, was to minimize the area created by the difference between the required state and the real system output in the whole simulation interval $-\tau_i$ Eq. (5). This CF design is very convenient for the evolutionary searching process due to the relatively favorable CF surface. The disadvantage of the approach is that CF value is influenced by chaotic transient behavior of the non-stabilized system. As a result of this, the small change in control method setting for extremely sensitive chaotic system (given by the very small change of CF value), can be suppressed by the above-mentioned

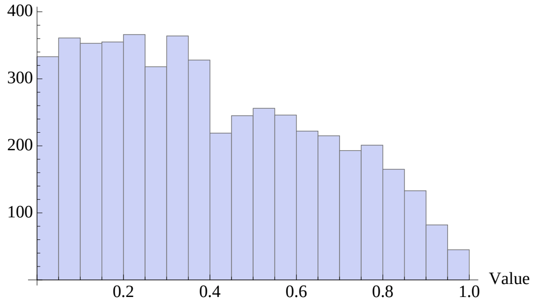

Fig. 4: Histogram of the distribution of real numbers transferred into the range $\langle 0, -1\rangle$ generated by the Lozi map (5000 samples).

including of initial chaotic transient:

$$CF_{Simple} = \sum_{t=0}^{\tau_i} |TS_t - AS_t|, \tag{5}$$

where TS is target state and AS is actual state.

Different type of universal cost function is purely based on searching for the desired stabilized periodic orbit and thereafter calculation of the difference between desired and actual periodic orbit in the short time interval $-\tau_s$ (20 iterations) from the point, where the first minimal value of difference between desired and actual system output is found (i.e. floating window for minimization, Fig. 5).

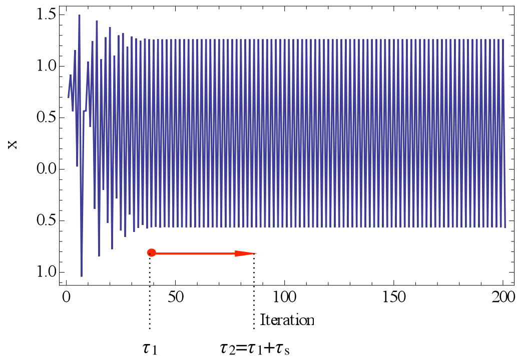

Fig. 5: Floating window for the optimization.

Such a design of universal CF should secure the successful stabilization of either $p-1$ orbit (stable state) or any higher periodic orbit anywise phase shifted. Furthermore, due to CF values converging towards zero, this CF also allows using decision rules and avoiding very time demanding simulations. This rule stops EA immediately, when the first individual with good parameter structure is reached, since the value of CF is lower than the acceptable one (CF_{acc}). Based on the numerous experiments, typically $CF_{acc} = 0.001$ at time interval $\tau_s = 20$ iterations, thus the difference between desired and actual output has the value of 0.0005 per iteration – i.e. successful stabilization for the used control technique. The CFUNI has the form described in Eq. (6):

$$CF_{UNI} = pen_1 + \sum_{t=\tau_1}^{\tau_2} |TS_t - AS_t|, \tag{6}$$

where τ_1 is the first min value of difference between TS and AS, τ_2 is the end of optimization interval ($\tau_1 + \tau_s$), $pen_1 = 0$ if $\tau_i - \tau_2 \geq \tau_s$ and $pen_1 = 10 \cdot (\tau_i - \tau_2)$ if $\tau_i - \tau_2 < \tau_s$ (i.e. late stabilization).

7. Experiment Design

This research encompasses six case studies. Three different required behavior of the chaotic system and two different cost functions were combined in the following form:

- Case study 1: $p-1$ UPO, Lozi map as CPRNG/Controlled system with CF_{Simple}.
- Case study 2: $p-1$ UPO, Lozi map as CPRNG/Controlled system with CF_{UNI}.
- Case study 3: $p-2$ UPO, Lozi map as CPRNG/Controlled system with CF_{Simple}.
- Case study 4: $p-2$ UPO, Lozi map as CPRNG/Controlled system with CF_{UNI}.
- Case study 5: higher order $p-4$ UPO, Lozi map as CPRNG/Controlled system with CF_{Simple}.
- Case study 6: higher order $p-4$ UPO, Lozi map as CPRNG/Controlled system with CF_{UNI}.

This work is focused on the utilization of the chaos driven DE for tuning of parameters for ETDAS control method to stabilize desired Unstable Periodic Orbits (UPO). In the described research, desired UPO was $p-1$ (stable state). The original control method, ETDAS, in the discrete form suitable for discrete chaotic maps has the form Eq. (7) and Eq. (8):

$$F_n = K\left[(1-R)\, S_{n-m} - x_n\right], \tag{7}$$

$$s_n = x_n \cdot R \cdot S_{n-m}, \tag{8}$$

where K and R are adjustable constants, F is the perturbation; S is given by a delay equation utilizing previous states of the system, m is the period of m-periodic orbit to be stabilized. The perturbation F_n in Eq. (8) may have arbitrarily large value, which can cause diverging of the system. Therefore, F_n should fall between $-F_{max}$, F_{max}. The ranges of all evolutionary estimated parameters are given in Tab. 1.

Tab. 1: Estimated parameters.

Parameter	Min	Max
K	-2	2
R	0	0.99
$F_{\max}$	0	0.9

Tab. 2: DE versions settings.

Parameter	Value
PopSize	25
F CanonicalDE	0.5
C_r Canonical DE	0.9
F ChaosDE	0.5
C_r ChaosDE	0.6
Generations	300
Max. CF Evaluations (CFE)	7500

Within the research, a total number of 50 simulations for each case study and each DE version were carried out. The parameter settings for both ChaosDE and Canonical DE were given following way (Tab. 2). Besides the parameters defining the size of population and number of generations, other two internal tuning parameters of DE, which are mutation constant F and crossover parameter C_r, are very important in the DE performance issue. Canonical DE utilizes the well-proven and recommended settings [38] Recent research in the chaos and complex dynamics driven heuristics shows that DE requires lower values for aforementioned internal parameters [30]. Therefore, simple tuning with the incremental step of 0.1 has been conducted for ChaosDE (Tab. 3).

Experiments were performed in an environment of *Wolfram Mathematica*, PRNG operations. Therefore, used the built-in Mathematica software pseudo-random number generator. All experiments used different initialization, i.e. different initial population was generated in each run of Canonical/ChaosDE.

Tab. 4: The values for desiredUPOs.

UPO	Values of UPO of unperturbed system
$p-1$	$x_F = 0.454545$
$p-2$	$x_1 = -0.382166$; $x_2 = 0.700637$
$p-4$	$x_1 = -0.691899$; $x_2 = 0.334059$; $x_3 = 0.086151$; $x_4 = 1.020573$

8. Results

All simulations were successful and gave new optimal settings for ETDAS control method securing the fast stabilization of the chaotic system at required behaviors, which were $p-1$ UPO (stable state), $p-2$ UPO (oscillation between two values) and finally $p-4$ UPO.

Performances of both studied DE strategies are compared with the representative of swarm algorithm, which is Particle Swarm Optimizer (PSO). The canonical version with inertia weight strategy [9] has been utilized. The maximal cost function evaluation value, population size and the number of iterations were set identically as for the both DE strategies.

The organization of the results is following. Table 5, Tab. 6, Tab. 8, Tab. 9, Tab. 11, and Tab. 12 are focused on the performance comparisons between canonical DE, ChaosDE driven by Lozi map; and swarm based PSO algorithm. These tables contain simple statistical overview of evolutionary optimization/simulation results i.e. average, median maximum, minimum (the best solution), std. dev. values for the particular cost function and for all 50 runs of both compared heuristics. Italic numbers represent the best result.

Tab. 3: Results for tuning of internal parameters F and C_r for ChaosDE from the interval $\langle 0.1, 0.9 \rangle$ and incremental step of 0.1 – Average cost function results for 30 runs of ChaosDE optimizing the higher nonlinear task: Case study 2.

	C_r=0.1	C_r=0.2	C_r=0.3	C_r=0.4	C_r=0.5	C_r=0.6	C_r=0.7	C_r=0.8	C_r=0.9
F=0.1	$4.39444 \cdot 10^{-15}$	$4.14344 \cdot 10^{-15}$	$4.16289 \cdot 10^{-15}$	$4.37956 \cdot 10^{-15}$	$4.14627 \cdot 10^{-15}$	$5.83432 \cdot 10^{-7}$	$3.52617 \cdot 10^{-10}$	0.0000271935	0.172951
F=0.2	$4.37891 \cdot 10^{-15}$	$4.09678 \cdot 10^{-15}$	$3.88405 \cdot 10^{-15}$	$3.95291 \cdot 10^{-15}$	$3.92013 \cdot 10^{-15}$	$3.99240 \cdot 10^{-15}$	$4.15958 \cdot 10^{-15}$	$3.16606 \cdot 10^{-12}$	0.0000531224
F=0.3	$4.35117 \cdot 10^{-15}$	$3.96566 \cdot 10^{-15}$	$3.84068 \cdot 10^{-15}$	$3.82903 \cdot 10^{-15}$	$3.90125 \cdot 10^{-15}$	$3.86901 \cdot 10^{-15}$	0.0000501611	$3.96350 \cdot 10^{-15}$	$4.16068 \cdot 10^{-15}$
F=0.4	$4.47881 \cdot 10^{-15}$	$3.98678 \cdot 10^{-15}$	$3.85901 \cdot 10^{-15}$	$3.81791 \cdot 10^{-15}$	$3.82793 \cdot 10^{-15}$	$3.79960 \cdot 10^{-15}$	$3.82458 \cdot 10^{-15}$	$3.85238 \cdot 10^{-15}$	$3.90238 \cdot 10^{-15}$
F=0.5	$4.43944 \cdot 10^{-15}$	$4.00844 \cdot 10^{-15}$	$3.85903 \cdot 10^{-15}$	$3.81515 \cdot 10^{-15}$	$3.78464 \cdot 10^{-15}$	$3.75962 \cdot 10^{-15}$	$3.77407 \cdot 10^{-15}$	$3.79903 \cdot 10^{-15}$	$3.87791 \cdot 10^{-15}$
F=0.6	$4.44328 \cdot 10^{-15}$	$4.01397 \cdot 10^{-15}$	$3.92736 \cdot 10^{-15}$	$3.82460 \cdot 10^{-15}$	$3.81850 \cdot 10^{-15}$	$3.79684 \cdot 10^{-15}$	$3.79572 \cdot 10^{-15}$	$3.79625 \cdot 10^{-15}$	$3.85182 \cdot 10^{-15}$
F=0.7	$4.43664 \cdot 10^{-15}$	$4.00505 \cdot 10^{-15}$	$3.87791 \cdot 10^{-15}$	$3.80962 \cdot 10^{-15}$	$3.82570 \cdot 10^{-15}$	$3.79680 \cdot 10^{-15}$	$3.76240 \cdot 10^{-15}$	$3.81350 \cdot 10^{-15}$	$3.88070 \cdot 10^{-15}$
F=0.8	$4.45275 \cdot 10^{-15}$	$4.04009 \cdot 10^{-15}$	$3.90736 \cdot 10^{-15}$	$3.83625 \cdot 10^{-15}$	$3.83625 \cdot 10^{-15}$	$3.79125 \cdot 10^{-15}$	$3.76964 \cdot 10^{-15}$	$3.80127 \cdot 10^{-15}$	$3.86570 \cdot 10^{-15}$
F=0.9	$4.44226 \cdot 10^{-15}$	$4.13174 \cdot 10^{-15}$	$3.91846 \cdot 10^{-15}$	$3.82625 \cdot 10^{-15}$	$3.83793 \cdot 10^{-15}$	$3.78905 \cdot 10^{-15}$	$3.79184 \cdot 10^{-15}$	$3.78738 \cdot 10^{-15}$	$3.85848 \cdot 10^{-15}$

Tab. 5: Comparison for ChaosDE, canonical DE and PSO case study 1.

Statistical	**ChaosDE**	**Canonical DE**	**PSO**
data	CF **Value**	CF **Value**	CF **Value**
Min	0.520639	0.520639	0.530679
Max	0.522148	0.527132	0.573742
Average	0.520696	0.520769	0.548688
Median	0.520639	0.520639	0.549588
Std. Dev.	$2.78{\cdot}10^{-4}$	$9.18{\cdot}10^{-4}$	$1.07{\cdot}10^{-2}$

Tab. 6: Comparison for ChaosDE, canonical DE and PSO case study 2.

Statistical	**ChaosDE**	**Canonical DE**	**PSO**
data	CF **Value**	CF **Value**	CF **Value**
Min	$3.53307 \cdot 10^{-15}$	$3.55511 \cdot 10^{-15}$	$4.41022 \cdot 10^{-15}$
Max	$4.05511 \cdot 10^{-15}$	$3.91062 \cdot 10^{-15}$	$5.21022 \cdot 10^{-15}$
Average	$3.75362 \cdot 10^{-15}$	$3.7514 \cdot 10^{-15}$	$4.79336 \cdot 10^{-15}$
Median	$3.75511 \cdot 10^{-15}$	$3.75511 \cdot 10^{-15}$	$4.78267 \cdot 10^{-15}$
Std. Dev.	$9.96 \cdot 10^{-17}$	$7.42 \cdot 10^{-17}$	$2.02977 \cdot 10^{-16}$

Tab. 7: Best solutions – Joined case studies 1 and 2, $p-1$ UPO.

Parameter	**Case study 1,** CN_{Simple}, **ChaosDE**	**Case study 2,** CN_{UNI}, **ChaosDE**
K	-1.11259	-0.859989
F_{max}	0.9	0.65695
R	0.289232	0.065673
CF Value	0.520639	$3.53307 \cdot 10^{-15}$
Istab. Value	21	9
Avg. error per iteration	$7.21 \cdot 10^{-15}$	$2.07 \cdot 10^{-15}$

Tab. 8: Comparison for ChaosDE, canonical DE and PSO case study 3.

Statistical	**ChaosDE**	**Canonical DE**	**PSO**
data	CF **Value**	CF **Value**	CF **Value**
Min	7.04967	6.99829	7.29818
Max	7.54409	7.33379	8.05143
Average	7.30428	7.2827	7.67487
Median	7.33379	7.33379	7.6944
Std. Dev.	$7.77 \cdot 10^{-2}$	$9.29 \cdot 10^{-2}$	0.216771

Tab. 9: Comparison for ChaosDE, canonical DE and PSO case study 4.

Statistical	**ChaosDE**	**Canonical DE**	**PSO**
data	CF **Value**	CF **Value**	CF **Value**
Min	$1.62665 \cdot 10^{-9}$	$1.68548 \cdot 10^{-9}$	$1.81061 \cdot 10^{-6}$
Max	$1.28207 \cdot 10^{-2}$	$1.36095 \cdot 10^{-2}$	0.136879
Average	$7.57338 \cdot 10^{-4}$	$6.40434 \cdot 10^{-4}$	0.008665
Median	$1.42814 \cdot 10^{-4}$	$1.35227 \cdot 10^{-4}$	0.001515
Std. Dev.	$1.86 \cdot 10^{-3}$	$1.96 \cdot 10^{-3}$	0.023862

Tab. 10: Best solutions – Joined case studies 3 and 4, $p-2$ UPO.

Parameter	**Case study 3,** CN_{Simple}, **ChaosDE**	**Case study 4,** CN_{UNI}, **ChaosDE**
K	0.574025	-0.614527
F_{max}	0.430788	0.508694
R	0.445453	0.528986
CF Value	6.99829	$1.62665 \cdot 10^{-9}$
Istab. Value	22	18
Avg. error per iteration	$2.98 \cdot 10^{-8}$	$1.99 \cdot 10^{-11}$

Tab. 11: Comparison for ChaosDE, canonical DE and PSO case study 5.

Statistical data	**ChaosDE**	**Canonical DE**	**PSO**
	CF **Value**	CF **Value**	CF **Value**
Min	13.8025	13.7305	14.9151
Max	19.2155	51.9391	29.6333
Average	15.4354	21.7940	14.5647
Median	14.2014	14.2548	22.4428
Std. Dev.	1.1312	5.4219	3.5719

Tab. 12: Comparison for ChaosDE, canonical DE and PSO case study 6.

Statistical data	**ChaosDE**	**Canonical DE**	**PSO**
	CF **Value**	CF **Value**	CF **Value**
Min	$1.39076 \cdot 10^{-3}$	$1.45946 \cdot 10^{-3}$	$2.04937 \cdot 10^{-3}$
Max	$1.4131 \cdot 10^{-2}$	$1.28539 \cdot 10^{-2}$	1.14558
Average	$2.79004 \cdot 10^{-3}$	$2.09308 \cdot 10^{-3}$	0.15435
Median	$1.60824 \cdot 10^{-3}$	$1.62911 \cdot 10^{-3}$	0.11778
Std. Dev.	$3.49 \cdot 10^{-3}$	$2.19 \cdot 10^{-3}$	0.25369

Tab. 13: Best solutions – Joined case studies 5 and 6, $p-4$ UPO.

Parameter	**Case study 5,** CN_{Simple}, **ChaosDE**	**Case study 6,** CN_{UNI}, **ChaosDE**
K	-0.869336	-0.935913
F_{max}	0.255559	0.623172
CF Value	13.7305	$1.39076 \cdot 10^{-3}$
Istab. Value	46	39
Avg. error per iteration	$2.66 \cdot 10^{-4}$	$1.39 \cdot 10^{-6}$

Results given in Tab. 7, Tab. 10 and Tab. 13 represent the direct comparison of chaos stabilization properties for the joined case studies related to the identical UPO to be controlled. Tables show the best founded individual solutions of parameters set up for ETDAS control method, corresponding final CF value. Also these tables show the Istab. value representing the number of iterations required for stabilization on the desired UPO and further the average error between desired output value and real system output from the last 20 iterations.

Graphical simulation outputs of the best individual solutions for particular case studies are depicted in Fig. 6, Fig. 7, Fig. 10, Fig. 11, Fig. 14 and Fig. 15 whereas the Fig. 8, Fig. 9, Fig. 12, Fig. 13, Fig. 16 and Fig. 17 show the simulation output of all 50 runs of ChaosDE confirming the robustness of this approach. For the illustrative purposes, all graphical simulations outputs are depicted only for the variable x of the chaotic systems.

The values for desired UPOs of unperturbed chaotic Lozi map based on the mathematical analysis of the systems are given in Tab. 4.

From the results presented in the Tab. 5, Tab. 6, Tab. 7, Tab. 8, Tab. 9, Tab. 10, Tab. 11, Tab. 12 and Tab. 13 it is clear that the CF_{Simple} is very convenient for the evolutionary process, which means that repeated runs of EA are providing identical optimal

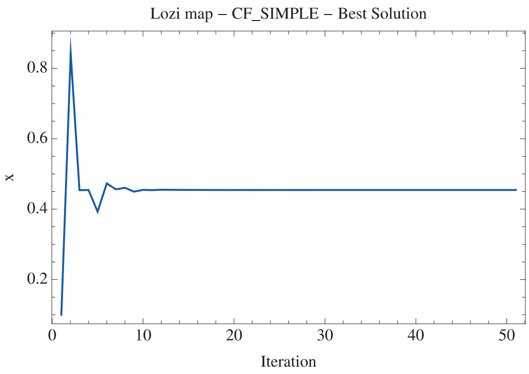

Fig. 6: Simulation of the best individual solution – ChaosDE and Lozi map: Case study 1, CN_{Simple}, $p-1$ UPO.

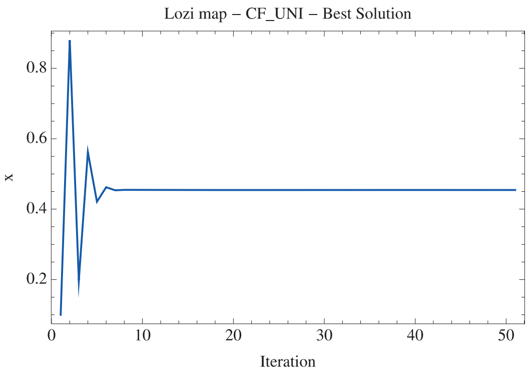

Fig. 7: Simulation of the best individual solution – ChaosDE and Lozi map: Case study 2, CN_{UNI}, $p-1$ UPO.

results (i.e. very close to the possible global extreme). This is graphically confirmed in the Fig. 8, Fig. 12 and Fig. 16, which show all 50 simulations. All the runs are merged into the one line.

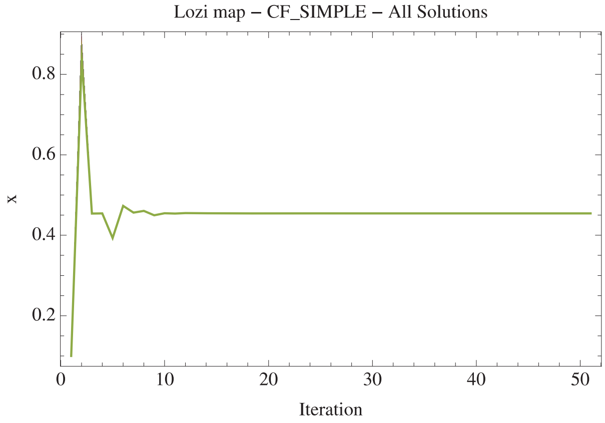

Fig. 8: All 50 runs of EA – ChaosDE and Lozi map: Case study 1, CN_{Simple}, $p-1$ UPO.

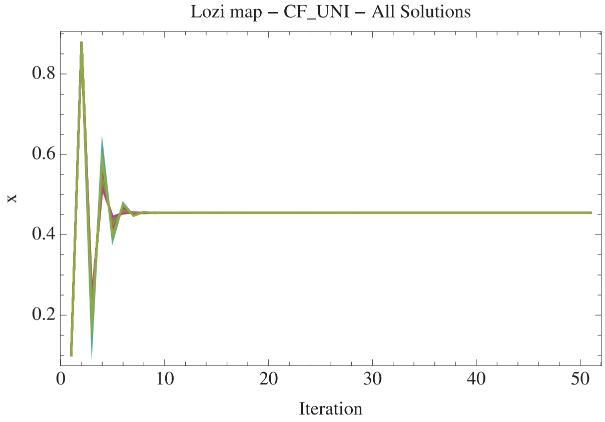

Fig. 9: All 50 runs of EA – ChaosDE and Lozi map: Case study 2, CN_{UNI}, $p-1$ UPO.

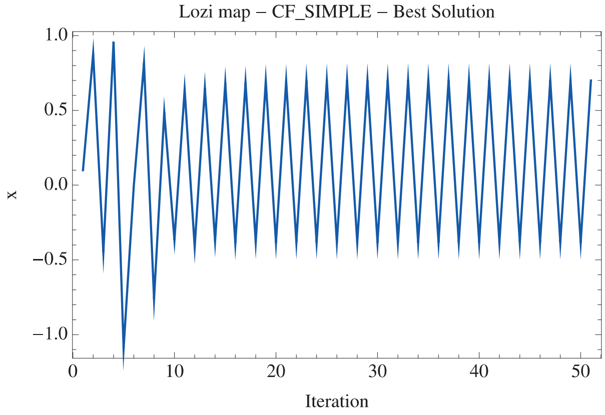

Fig. 10: Simulation of the best individual solution – Canonical DE and Lozi map: Case study 3, CN_{Simple}, $p-2$ UPO.

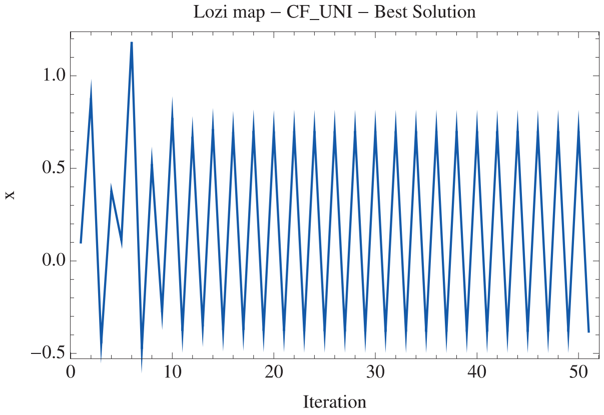

Fig. 11: Simulation of the best individual solution – ChaosDE and Lozi map: Case study 4, CN_{UNI}, $p-2$ UPO.

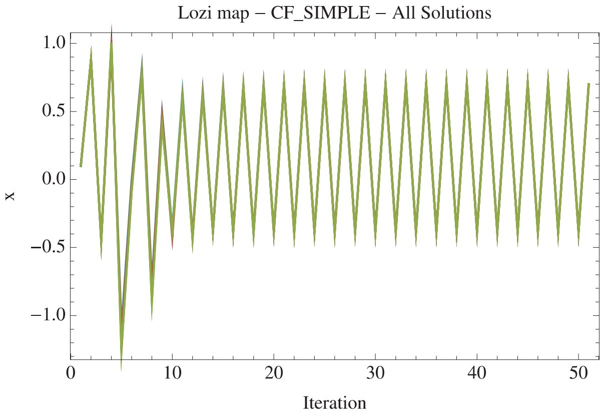

Fig. 12: All 50 runs of EA– Canonical DE and Lozi map: Case study 3, CN_{Simple}, $p-2$ UPO.

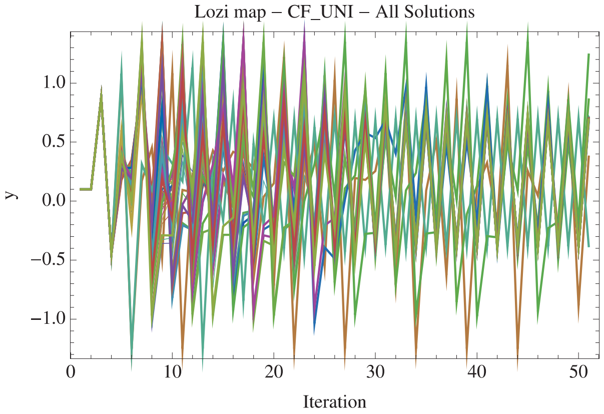

Fig. 13: All 50 runs of EA – ChaosDE and Lozi map, Case study 4, CN_{UNI}, $p-2$ UPO.

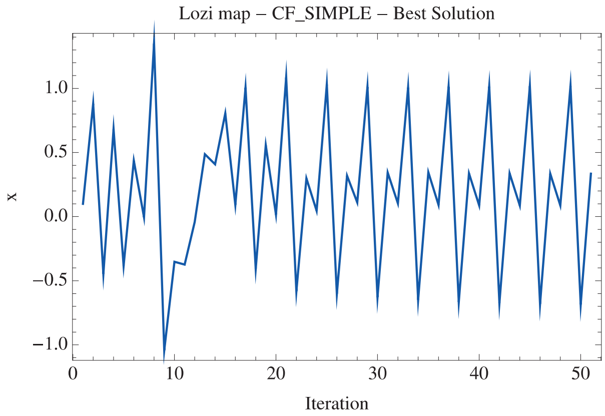

Fig. 14: Simulation of the best individual solution – Canonical DE and Lozi map: Case study 5, CN_{Simple}, $p-4$ UPO.

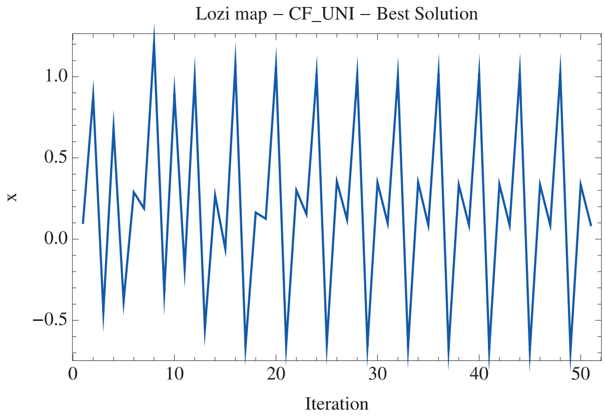

Fig. 15: Simulation of the best individual solution – Canonical DE and Lozi map: Case study 5, CN_{Simple}, $p-4$ UPO.

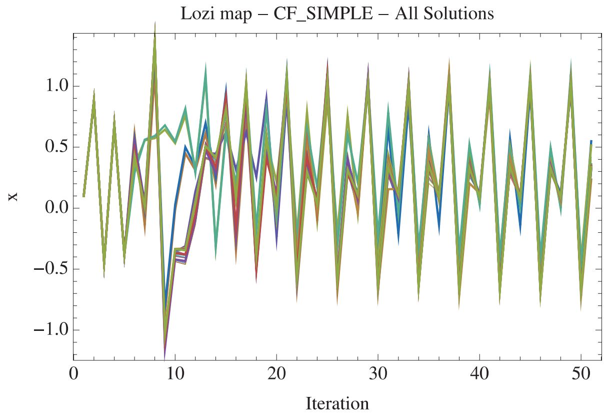

Fig. 16: All 50 runs of EA – Canonical DE and Lozi map: Case study 5 - CN_{Simple}, $p-4$ UPO.

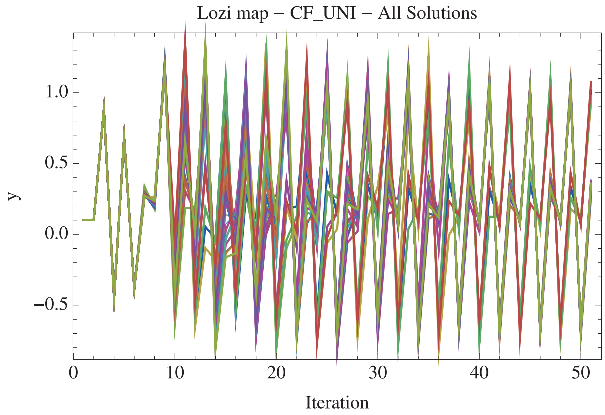

Fig. 17: All 50 runs of EA – ChaosDE and Lozi map: Case study 6, CN_{UNI}, $p-4$ UPO.

On the other hand the disadvantage of including of initial chaotic transient behavior of not stabilized system into the cost function value and resulting very tiny change of control method setting for the extremely sensitive chaotic system is causing suppression of stabilization speed and numerical precision.

Results obtained in the cases utilizing the CN_{UNI} lend weight to the argument that the technique of pure searching for periodic orbits is advantageous for faster and more precise stabilization of the chaotic system.

The graphical comparisons of the performance analysis of ChaosDE, Canonical DE and PSO within all 6 case studies are given in complex Fig. 18 and Fig. 19. The first one represents the time evolution of the cost function value for the best individual solution, which are given in Tab. 10, Tab. 11 and Tab. 13. Moreover, these solutions were also used for the graphical simulations in Fig. 6, Fig. 7, Fig. 10, Fig. 11, Fig. 14 and Fig. 15. Figure 19 shows the comparisons of time evolution of average CF values for all 50 runs of ChaosDE/Canonical DE/PSO, confirming the robustness of both used DE strategies within many repeated runs. More about the findings is given in the conclusion section of this paper.

9. Conclusion

In this paper, evolutionary algorithm Differential Evolution was used for the evolutionary tuning of controller parameters for the stabilization of selected discrete chaotic system, which was the two-dimensional Lozi map. The originality of the presented approach is that the selected controlled discrete dissipative chaotic system is used also within chaos enhanced heuristic concept as the chaotic pseudo-random number generator to drive the mutation and crossover process in the Differential Evolution. The idea was to utilize the hidden chaotic dynamics in the pseudo-random sequences given by chaotic map to help Differential Evolution algorithm in searching for the best controller settings for the identical chaotic system.

The findings of the two different cost function designs can be summarized as follows:

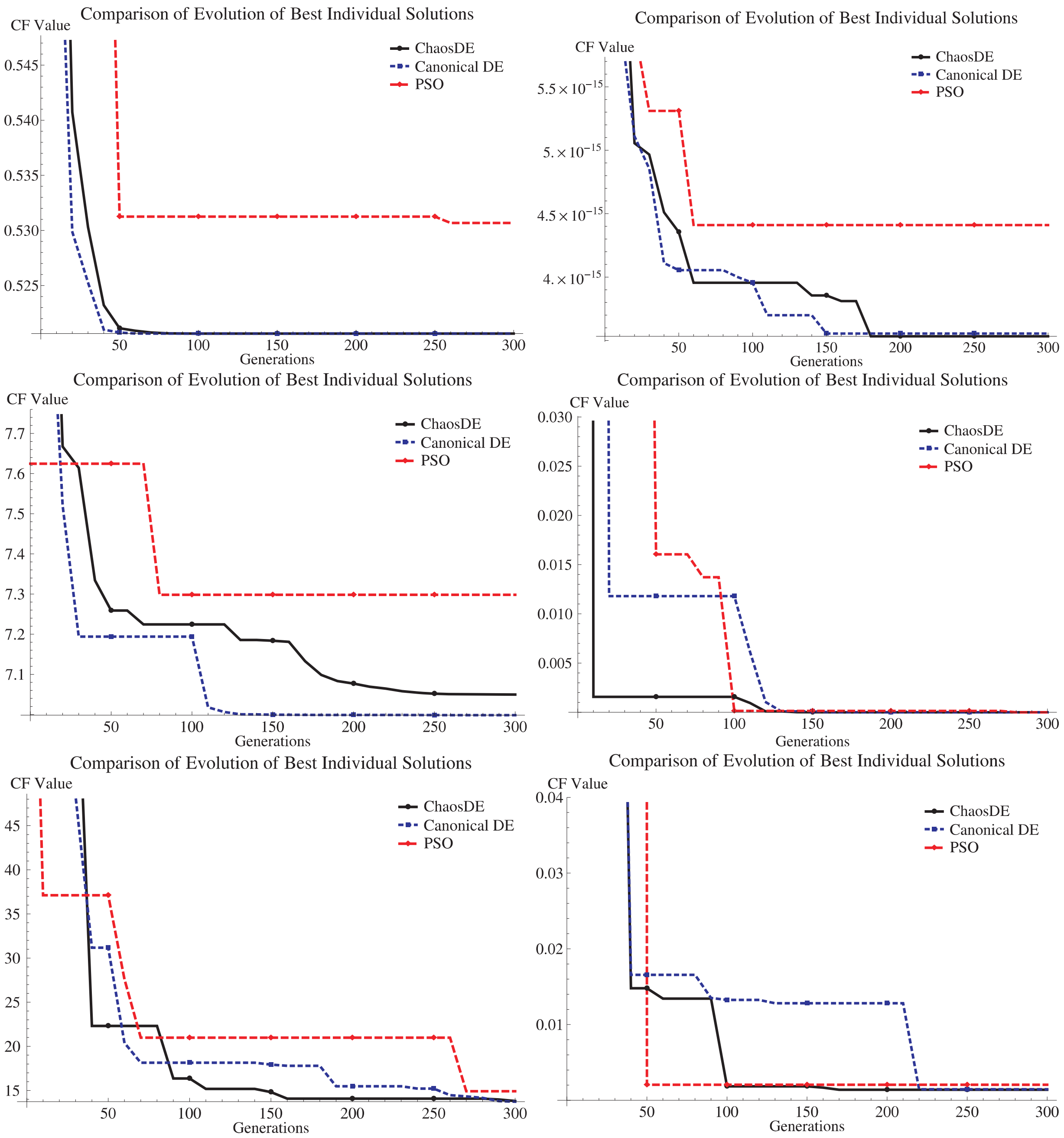

Fig. 18: Comparisons of time evolution of the best individual solutions for ChaosDE, Canonical DE and PSO; from left to right and from above to bottom: Case study 1 – Case study 6.

- CF_{Simple} is very convenient for the evolutionary process, which means that most of the repeated runs of EA are giving identical optimal results. On the other hand the disadvantage of including of initial chaotic transient behavior of not stabilized system into the cost function value and resulting small change of control method setting for extremely sensitive chaotic system is causing suppression of stabilization speed and numerical precision . Moreover, this design has many limitations for stabilizing of higher order oscillations.
- The second universal CF design brings the possibility of using it problem-free for any desired behavior of arbitrary chaotic systems, but at the cost of the highly chaotic CF surface. Nevertheless, the embedding of the chaotic dynamics into the evolutionary algorithms helped to deal with such an issue. Direct searching for stabilization points brings faster and more precise stabilization of chaotic oscillations.

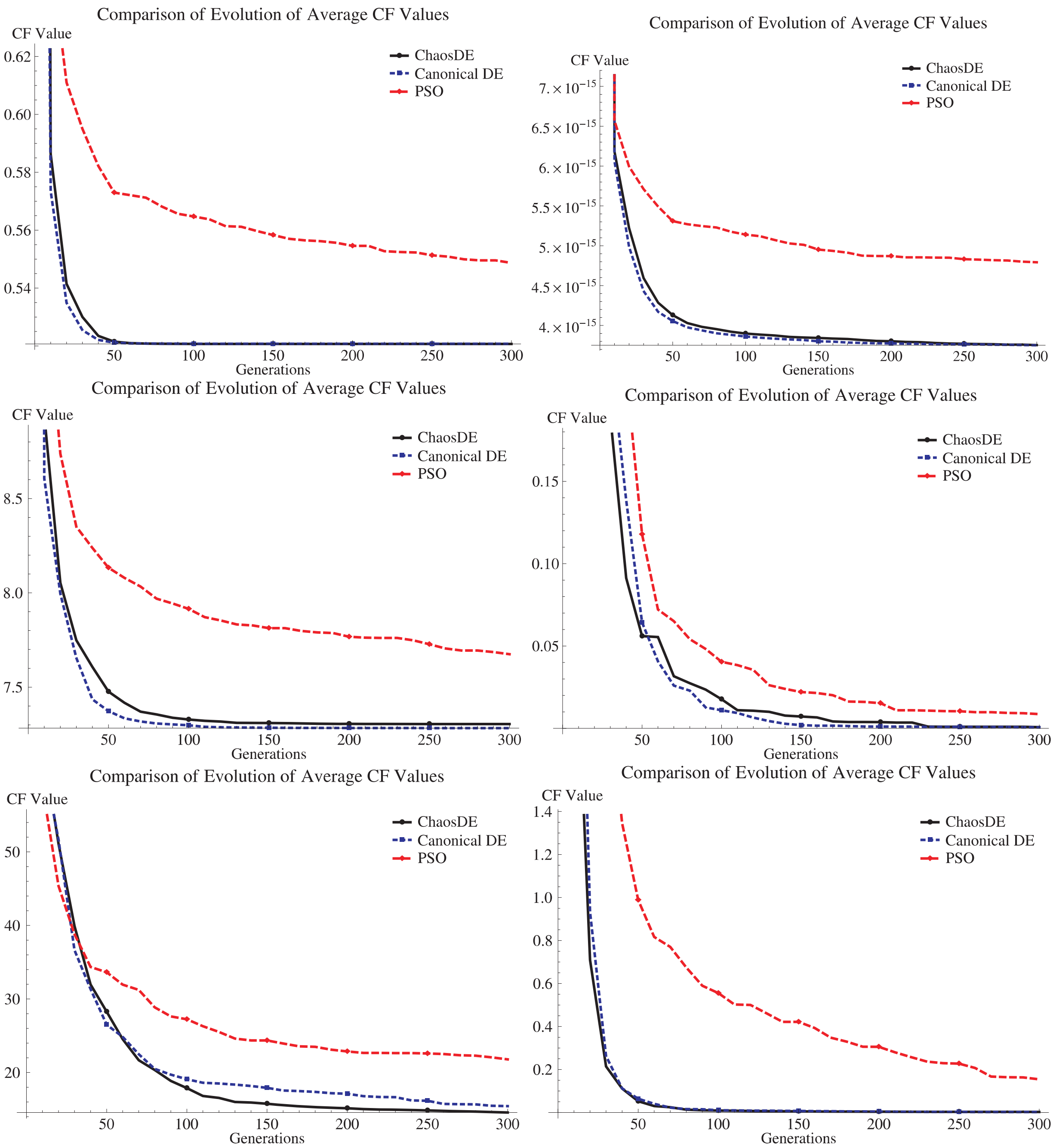

Fig. 19: Comparisons of time evolution of average cost function values of the best solutions from all 50 runs of ChaosDE, Canonical DE and PSO; from left to right and from above to bottom: Case study 1 – Case study 6.

Performance comparisons between canonical version, chaos driven heuristic and swarm based algorithm PSO revealed following findings:

- The overall performance of canonical version of DE is comparable with ChaosDE within simpler case studies with CF_{Simple}. There is the only moderate difference in selected observed statistical properties of results. ChaosDE shows significantly better performance with complex universal CF and in case studies dealing with higher order of oscillations. Furthermore, ChaosDE gives solutions with very fast convergence towards cost function extremes (see Fig. 18 – more complex case studies 4, 5 and 6). Thus, less iterations of heuristics is required for acceptable level of chaos system stabilization. Comparisons between both DE versions and swarm based PSO algorithm show, that PSO is not good choice in the task of nonlinear/chaos control optimization.

- The results show that embedding of the chaotic dynamics in the form of chaotic pseudo-random number generator into the differential evolution algorithm may help to improve the performance and robustness of the DE. Thus, obtaining optimal solutions secure the very fast and precise stabilization, especially for very chaotic and nonlinear CF surface in case of the CF_{UNI}.

This work has experimentally confirmed that embedding the hidden chaotic dynamics into the evolutionary process in the form of chaotic pseudo-random number generators may help to obtain better results and avoid problems connected with evolutionary computation such as premature convergence and stagnation in local extremes.

Acknowledgment

This work was supported by Grant Agency of the Czech Republic GACR P103/15/06700S, further this work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Pro- gramme project No. LO1303 (MSMT-7778/2014) and also by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089., partially supported by Grant of SGS No. SP2016/175 of VSB–Technical University of Ostrava, Czech Republic and by Internal Grant Agency of Tomas Bata University under the project No. IGA/CebiaTech/2016/007.

References

[1] OTT, E, C. GREBOGI and J. A. YORKE. Controlling chaos. *Physical Review Letters*. 1990, vol. 64, iss. 11, pp. 1196–1199. ISSN 1079-7114. DOI: 10.1103/PhysRevLett.64.1196.

[2] ZELINKA, I. Real-time deterministic chaos control by means of selected evolutionary techniques. *Engineering Applications of Artificial Intelligence*. 2009, vol. 22, iss. 2, pp. 283–297. ISSN 0952-1976. DOI: 10.1016/j.engappai.2008.07.008.

[3] ZELINKA, I., R. SENKERIK and E. NAVRATIL. Investigation on evolutionary optimization of chaos control. *Chaos, Solitons & Fractals*. 2009, vol. 40, iss. 1, pp. 111–129. ISSN 0960-0779. DOI: 10.1016/j.chaos.2007.07.045.

[4] SENKERIK, R., I. ZELINKA, D. DAVENDRA and Z. OPLATKOVA. Utilization of SOMA and differential evolution for robust stabilization of chaotic Logistic equation. *Computers & Mathematics with Applications*. 2010, vol. 60, iss. 4, pp. 1026–1037. ISSN 0898-1221. DOI: 10.1016/j.camwa.2010.03.059.

[5] PYRAGAS, K. Control of chaos via extended delay feedback. *Physics Letters A*. 1995, vol. 206, iss. 5–6, pp. 323–330. ISSN 0375-9601. DOI: 10.1016/0375-9601(95)00654-L.

[6] SENKERIK, R., Z. OPLATKOVA, I. ZELINKA and D. DAVENDRA. Synthesis of feedback controller for three selected chaotic systems by means of evolutionary techniques: Analytic programming. *Mathematical and Computer Modelling*. 2013, vol. 57, iss. 1–2, pp. 57–67. ISSN 0895-7177. DOI: 10.1016/j.mcm.2011.05.030.

[7] KOMINKOVA OPLATKOVA, Z., R. SENKERIK, I. ZELINKA and M. PLUHACEK. Analytic programming in the task of evolutionary synthesis of a controller for high order oscillations stabilization of discrete chaotic systems. *Computers & Mathematics with Applications*. 2013, vol. 66, iss. 2, pp. 177–189. ISSN 0898-1221. DOI: 10.1016/j.camwa.2013.02.008.

[8] SENKERIK, R., Z. KOMINKOVA OPLATKOVA, I. ZELINKA, B. CHRAMCOV, D. D. DAVENDRA and M. PLUHACEK. Utilization of analytic programming for the evolutionary synthesis of the robust multi-chaotic controller for selected sets of discrete chaotic systems. *Soft Computing*. 2014, vol. 18, iss. 4, pp. 651–668. ISSN 1432-7643. DOI: 10.1007/s00500-014-1220-1.

[9] KENNEDY, J. and R. EBERHART. Particle swarm optimization. In: *IEEE International Conference on Neural Networks*. Perth: IEEE, 1995, pp. 1942–1948. ISBN 0-7803-2768-3.

[10] ABEDINI, M., R. VATANKHAH and N. ASSADIAN. Stabilizing chaotic system on periodic orbits using multi-interval and modern optimal control strategies. *Communications in Nonlinear Science and Numerical Simulation*. 2012, vol. 17, iss. 10, pp. 1007–5704. ISSN 1432-7643. DOI: 10.1016/j.cnsns.2012.02.004.

[11] SADEGHPOUR, M., H. SALARIEH, G. VOSSOUGHI and A. ALASTY. Multi-variable control of chaos using PSO-based minimum entropy control. *Communications in Nonlinear Science and Numerical Simulation*. 2011, vol. 16, iss. 6, pp. 2397–2404. ISSN 1432-7643. DOI: 10.1016/j.cnsns.2010.09.019.

[12] COELHO, L. D. S. and R. B. GREBOGI. Chaotic synchronization using PID control combined with population based incremental learning algorithm. *Expert Systems with Applications*. 2010, vol. 37, iss. 7, pp. 5347–5352. ISSN 0957-4174. DOI: 10.1016/j.eswa.2010.01.022.

[13] SHIRAZI, M. J., R. VATANKHAH, M. BOROUSHAKI, H. SALARIEH and A. ALASTY. Application of particle swarm optimization in chaos synchronization in noisy environment in presence of unknown parameter uncertainty. *Communications in Nonlinear Science and Numerical Simulation*. 2012, vol. 17, iss. 2, pp. 742–753. ISSN 1432-7643. DOI: 10.1016/j.cnsns.2011.05.032.

[14] ZELINKA, I. and A. RAIDL. *Evolutionary Synchronization of Chaotic Systems*. erlin: Springer, 2010. ISBN 978-3-642-10706-1.

[15] RICHTER, H. and K. J. REINSCHKE. Optimization of local control of chaos by an evolutionary algorithm. *Physica D: Nonlinear Phenomena*. 2000, vol. 144, iss. 3–4, pp. 309–334. ISSN 0167-2789. DOI: 10.1016/S0167-2789(00)00080-4.

[16] AYDIN, I., M. KARAKOSE and E. AKIN. Chaotic-based hybrid negative selection algorithm and its applications in fault and anomaly detection. *Expert Systems with Applications*. 2010, vol. 37, iss. 7, pp. 5285–5294. ISSN 0957-4174. DOI: 10.1016/j.eswa.2010.01.011.

[17] CAPONETTO, R., L. FORTUNA, S. FAZZINO and M. G. XIBILIA. Chaotic sequences to improve the performance of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*. 2003, vol. 7, iss. 3, pp. 289–304. ISSN 1089-778X. DOI: 10.1109/TEVC.2003.810069.

[18] DAVENDRA, D., I. ZELINKA and R. SENKERIK. Chaos driven evolutionary algorithms for the task of PID control. *Computers & Mathematics with Applications*. 2010, vol. 60, iss. 4, pp. 1088–1104. ISSN 0898-1221. DOI: 10.1016/j.camwa.2010.03.066.

[19] PRICE, K. V., R. M. STORN and J. A. LAMPINEN. *Differential Evolution - A Practical Approach to Global Optimization*. Berlin: Springer, 2005. ISBN 978-3-540-31306-9.

[20] ONWUBOLU, G. C. and B. V. BABU. *New Optimization Techniques in Engineering*. Berlin: Springer, 2004. ISBN 978-3-642-05767-0.

[21] COELHO, L. D. S. and V. C. MARIANI. A novel chaotic particle swarm optimization approach using Henon map and implicit filtering local search for economic load dispatch. *Chaos, Solitons & Fractals*. 2009, vol. 39, iss. 3, pp. 510–518. ISSN 0960-0779. DOI: 10.1016/j.chaos.2007.01.093.

[22] SENKERIK, R., D. DAVENDRA, I. ZELINKA, M. PLUHACEK,. An Investigation on the Chaos Driven Differential Evolution: An Initial Study. In: *5th International Conference on Bioinspired Optimization Methods and Their Applications*. Bohinj: Bioma, 2012, pp. 185–194. ISBN 978-961-264-043-9.

[23] DAVENDRA, D. *Traveling Salesman Problem, Theory and Applications*. Rijeka: InTech, 2010. ISBN 978-953-307-426-9.

[24] DAVENDRA, D., M. BIALIC-DAVENDRA and R. SENKERIK. Scheduling the Lot-Streaming Flowshop scheduling problem with setup time with the chaos-induced Enhanced Differential Evolution. In: *IEEE Symposium on Differential Evolution*. Singapore: IEEE, 2013, pp. 119–126. ISBN 978-146-735-873-6. DOI: 10.1109/SDE.2013.6601451.

[25] PLUHACEK, M., R. SENKERIK, D. DAVENDRA, Z. KOMINKOVA OPLATKOVA and I. ZELINKA. On the behavior and performance of chaos driven PSO algorithm with inertia weight. *Computers & Mathematics with Applications*. 2013, vol. 66, iss. 2, pp. 122–134. ISSN 0898-1221. DOI: 10.1016/j.camwa.2013.01.016.

[26] COELHO, L. D. S. and V. C. MARIANI. Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning. *Computers & Mathematics with Applications*. 2012, vol. 64, iss. 8, pp. 2371–2382. ISSN 0898-1221. DOI: 10.1016/j.camwa.2012.05.007.

[27] ZELINKA, I., V. H. DUY and J. CHA. *Recent Advances in Electrical Engineering and Related Sciences*. Berlin: Springer, 2014. ISBN 978-3-319-27247-4.

[28] SENKERIK, R., D. DAVENDRA, I. ZELINKA and Z. OPLATKOVA. Chaos driven Differential Evolution in the task of chaos control optimization. In: *Proceedings of the IEEE Congress on Evolutionary Computation*. Barcelona: IEEE, 2010, pp. 1–8. ISBN 978-1-4244-8126-2. DOI: 10.1109/CEC.2010.5585989.

[29] STORN, R. and K. PRICE. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*. 1997, vol. 11, iss. 4, pp. 341–359. ISSN 0925-5001. DOI: 10.1023/A:1008202821328.

[30] CORNE, D., M. DORIGO and F. GLOVER. *New Ideas in Optimization*. London: McGraw-Hill, 1999. ISBN 978-0077095062.

[31] DAS, S., A. ABRAHAM, U. K. CHAKRABORTY and A. KONAR. Differential evolution using

a neighborhood-based mutation operator. *IEEE Transactions on Evolutionary Computation*. 2009, vol. 13, iss. 3, pp. 526–553. ISSN 1089-778X. DOI: 10.1109/TEVC.2008.2009457.

[32] CHOI, T. J., C. W. AHN and J. AN. An adaptive cauchy differential evolution algorithm for global numerical optimization. *The Scientific World Journal*. 2013, vol. 2013, id. 969734, pp. 1–7. ISSN 2356-6140. DOI: 10.1155/2013/969734.

[33] QIN, A. K., V. L. HUANG and P. N. SUGANTHAN. Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization. *IEEE Transactions on Evolutionary Computation*. 2009, vol. 13, iss. 2, pp. 398–417. ISSN 1089-778X. DOI: 10.1109/TEVC.2008.927706.

[34] HANG, J. and A. C. SANDERSON. JADE: Self-adaptive differential evolution with fast and reliable convergence performance. *IEEE Transactions on Evolutionary Computation*. 2009, vol. 13, iss. 5, pp. 945–958. ISSN 1089-778X. DOI: 10.1109/TEVC.2009.2014613.

[35] ZHANG, J. and A. C. SANDERSON. Self-adaptive multi-objective differential evolution with direction information provided by archived inferior solutions. In: *IEEE World Congress on Computational Intelligence*. Hong Kong: IEEE, 2008, pp. 2801–2810. ISBN 978-1-4244-1823-7. DOI: 10.1109/CEC.2008.4631174.

[36] SPROTT, J. C. *Chaos and Time-Series Analysis*. Oxford: Oxford University Press, 2003. ISBN 978-0198508403.

[37] AZIZ-ALAOUI, M. A., C. ROBERT and C. GREBOGI. Dynamics of a Henon–Lozi-type map. *Chaos, Solitons & Fractals*. 2001, vol. 12, iss. 12, pp. 2323–2341. ISSN 0960-0779. DOI: 10.1016/S0960-0779(00)00192-2.

[38] SENKERIK, R., M. PLUHACEK, Z. K. OPLATKOVA and D. DAVENDRA. On the parameter settings for the chaotic dynamics embedded differential evolution. In: *Proceedings of the IEEE Congress on Evolutionary Computation*. Sendai: IEEE, 2015, pp. 1410–1417. ISBN 978-147-997-492-4. DOI: 10.1109/CEC.2015.7257053.

About Authors

Roman SENKERIK was born in the Czech Republic, and went to the Tomas Bata University in Zlin, where he studied Technical Cybernetics and obtained his M.Sc. degree in 2004, Ph.D. degree in Technical Cybernetics in 2008 and Assoc. prof. in 2013 (Informatics). He is now an Assoc. prof. at the same university (research and courses in: Evolutionary Computation, Applied Informatics, Cryptology, Artificial Intelligence, Mathematical Informatics).

Ivan ZELINKA was born in the Czech Republic, and went to the Technical University of Brno, where he studied Technical Cybernetics and obtained his degree in 1995. He obtained Ph.D. degree in Technical Cybernetics in 2001 at Tomas Bata University (TBU) in Zlin. Now he is a professor at TBU in Zlin and VSB–Technical University of Ostrava (research in: Artificial Intelligence, Theory of Information).

Michal PLUHACEK was born in the Czech Republic, and went to the Tomas Bata University in Zlin, where he studied Information Technologies and obtained his M.Sc. degree in 2011. He is now a Ph.D. student at the same university.