Performance Analysis of Chaotic Differential Evolution with the Dissipative Map for the PID Tuning Problem

Roman Senkerik and Michal Pluhacek

Abstract— This research presents results of the utilization of selected discrete chaotic map, which is Dissipative standard map, as the driving pseudo-random number generator for the differential evolution (DE) optimization algorithm in the task of PID controller design for the 4th order dynamical system. Moreover, the detailed performance analysis of ChaosDE and canonical version of DE is present here. Finally the results are compared with previously published results of other heuristics and discussed.

Keywords— Evolutionary algorithms; Differential evolution; Chaos, PRNG, PID controller, Optimization

I. Introduction

CURRENTLY the methods based on soft computing such as neural networks, evolutionary algorithms, fuzzy logic, and genetic programming are known as powerful tool for almost any difficult and complex optimization problem.

In the past decades, PID controllers became a fundamental part of many automatic systems. The successful design of PID controller was mostly based on deterministic methods involving complex mathematics [1, 2].

Recently, different soft-computing methods were used with promising results for solving the complex task of PID controller design [3], [4]. These techniques [6-9] use random operations and typically use various kinds of pseudo-random number generators (PRNGs) that depend on the platform the algorithm is implemented. More recently it was shown that chaotic systems could be used as PRNGs for various stochastic methods with great results. Some of these chaos driven stochastic methods were tested on the task of PID controller design in [5]. In [10] it was shown that Particle Swarm optimization (PSO) algorithm could deal with the task of PID controller design with very good results. Following that in [10 - 13] the performance of chaos driven PSO algorithm was tested on this task with great results.

In this paper, the influence of promising discrete dissipative chaotic system to the performance of chaos driven heuristic algorithm, which is DE, is investigated and results are compared with previously published results of chaos driven evolutionary algorithm PSO [10 - 13] and with other techniques [3,5] as well as with the canonical versions of DE (without chaotic pseudo-random number generator - CPRNG).

II. Motivation

Till now the chaos was observed in many of various systems (including evolutionary one) and in the last few years is also used to replace pseudo-number generators (PRGNs) in evolutionary algorithms (EAs).

This research is a continuation of the previous successful initial application based experiment with chaos driven DE [5], [14], [15]. In this paper the DE/rand/1/bin strategy driven by Dissipative chaotic map (system) was utilized to solve the issue of evolutionary optimization of PID controller settings. Thus the idea was to utilize the hidden chaotic dynamics in pseudo random sequences given by chaotic Dissipative map system to help Differential evolution algorithm in searching for the best controller settings.

Recent research in chaos driven heuristics has been fueled with the predisposition that unlike stochastic approaches, a chaotic approach is able to bypass local optima stagnation. This one clause is of deep importance to evolutionary algorithms. A chaotic approach generally uses the chaotic map 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. The task is then to select a very good chaotic map as the pseudo random number generator.

The primary aim of this work is not to develop a new type of pseudo random number generator, which should pass many statistical tests, but to try to test, analyze and compare the implementation of different natural chaotic dynamics as the CPRNGs, thus to analyze and highlight the different influences to the system, which utilizes the selected CPRNG (including the evolutionary computational techniques).

III. Differential Evolution Algorithm

DE is a population-based optimization method that works

R.Senkerik and M. Pluhacek are with the Department of Informatics and Artificial Intelligence, Tomas Bata University in Zlin, Nad Stranemi 4511, 760 05 Zlin, Czech Republic, (phone: +420576035189; fax: +420576035279; e-mail: senkerik@fai.utb.cz).

This work was supported by Grant Agency of the Czech Republic - GACR P103/15/06700S, further by financial support of research project NPU I No. MSMT-7778/2014 by the Ministry of Education of the Czech Republic 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 Internal Grant Agency of Tomas Bata University under the project No. IGA/FAI/2015/057.

on real-number-coded individuals [9]. For each individual $\vec{x}_{i,G}$ in the current generation G, DE generates a new trial individual $\vec{x}'_{i,G}$ by adding the weighted difference between two randomly selected individuals $\vec{x}_{r1,G}$ and $\vec{x}_{r2,G}$ to a randomly selected third individual $\vec{x}_{r3,G}$. The resulting individual $\vec{x}'_{i,G}$ is crossed-over with the original individual $\vec{x}_{i,G}$. The fitness of the resulting individual, referred to as a perturbed vector $\vec{u}_{i,G+1}$, is then compared with the fitness of $\vec{x}_{i,G}$. If the fitness of $\vec{u}_{i,G+1}$ is greater than the fitness of $\vec{x}_{i,G}$, then $\vec{x}_{i,G}$ is replaced with $\vec{u}_{i,G+1}$; otherwise, $\vec{x}_{i,G}$ remains in the population as $\vec{x}_{i,G+1}$. DE is quite robust, fast, and effective, with global optimization ability. It does not require the objective function to be differentiable, and it works well even with noisy and time-dependent objective functions. Description of used DERand1Bin strategy is presented in (1). Please refer to [9], [17], [18] and [19] for the description of all other strategies.

$$u_{i,G+1} = x_{r1,G} + F \cdot \left(x_{r2,G} - x_{r3,G}\right) \tag{1}$$

The pseudo-code of DE is depicted in Figure 1, where D is the size of the problem, G_{max} is the maximum number of generations, NP is the total number of solutions, F is the scaling factor, CR is the factor for crossover, $x^{(lo)}$ and $x^{(hi)}$ represents the initial bounds of the solutions; and three random numbers r_1, r_2, r_3 are selected, unique to each other.

1. Input: $D, G_{\max}, NP \geq 4, F \in (0, 1+), CR \in [0,1]$, and initial bounds: $\vec{x}^{(lo)}, \vec{x}^{(hi)}$.

2. Initialize: $\begin{cases} \forall i \leq NP \wedge \forall j \leq D : x_{i,j,G=0} = x_j^{(lo)} + rand_j[0,1] \bullet \left(x_j^{(hi)} - x_j^{(lo)}\right) \\ i = \{1,2,\ldots,NP\},\ j = \{1,2,\ldots,D\},\ G = 0,\ rand_j[0,1] \in [0,1] \end{cases}$

$$\begin{cases} \text{3. While } G < G_{\max} \\ \forall i \leq NP \begin{cases} \text{4. Mutate and recombine:} \\ \quad \text{4.1} \quad r_1, r_2, r_3 \in \{1,2,\ldots,NP\}, \text{ randomly selected, except: } r_1 \neq r_2 \neq r_3 \neq i \\ \quad \text{4.2} \quad j_{rand} \in \{1,2,\ldots,D\}, \text{ randomly selected once each } i \\ \text{4.3} \ \ \forall j \leq D, u_{j,i,G+1} = \begin{cases} x_{j,r_3,G} + F \cdot (x_{j,r_1,G} - x_{j,r_2,G}) \\ \qquad \text{if } (rand_j[0,1] < CR \vee j = j_{rand}) \\ x_{j,i,G} \quad \text{otherwise} \end{cases} \\ \text{5. Select} \\ \vec{x}_{i,G+1} = \begin{cases} \vec{u}_{i,G+1} & \text{if } f(\vec{u}_{i,G+1}) \leq f(\vec{x}_{i,G}) \\ \vec{x}_{i,G} & \text{otherwise} \end{cases} \end{cases} \\ G = G + 1 \end{cases}$$

Fig. 1 DE Pseudocode

IV. The Concept of Chaos DE

The general idea of CPRNG is to replace the default PRNG with the chaotic system. As the chaotic system is a set of equations with a static start position, we created a random start position of the system, in order to have different start position for different experiments. This random position is initialized with the default PRNG, as a one-off randomizer. Once the start position of the chaotic system has been obtained, the system generates the next sequence using its current position.

Generally there exist many other approaches as to how to deal with the negative numbers as well as with the scaling of the wide range of the numbers given by the chaotic systems into the typical range 0 – 1:

- Finding of the maximum value of the pre-generated long discrete sequence and dividing of all the values in the sequence with such a maxval number.
- Shifting of all values to the positive numbers (avoiding of ABS command) and scaling.

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 equations (2) and (3):

$$rndreal = \text{mod}\ (\text{abs}\ (rndChaos)\ ,\ 1.0) \tag{2}$$

$$rndint = \text{mod}\ (\text{abs}\ (rndChaos)\ ,\ 1.0) \times Range + 1 \tag{3}$$

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) till where the number is to be scaled.

V. Selected Chaotic System – Dissipative Map

This section contains the description of discrete dissipative chaotic map, which was used as the chaotic pseudo random generator. In this research, direct output iterations of the chaotic map were used for the generation of the both integer numbers and real numbers scaled into the typical range for random function: <0 - 1>. Following chaotic system was used: Dissipative Standard Map (4).

The Dissipative Standard map is a two-dimensional chaotic map. The parameters used in this work are b = 0.6 and k = 8.8 as suggested in [20]. The map equations are given in (4).

$$\begin{aligned} X_{n+1} &= X_n + Y_{n+1} (\text{mod}\, 2\pi) \\ Y_{n+1} &= bY_n + k \sin X_n (\text{mod}\, 2\pi) \end{aligned} \tag{4}$$

The x,y plot of Dissipative map is depicted in Fig. 2. The typical chaotic behavior of the utilized chaotic map, represented by the example of simulation of chaotic map for the variable y is depicted in Fig. 3. Whereas Fig. 4 represents the discrete direct outputs of the chaotic system.

The illustrative histograms of the distribution of real numbers transferred into the range <0 - 1> generated by means of studied chaotic system is in Fig. 5.

Finally the Fig. 6 shows the example of dynamical sequencing during the generating of pseudo number numbers by means of studied CPRNG for long data series and Fig. 7 represents the detailed view of hidden chaotic dynamics influencing the generation of pseudo-random numbers.

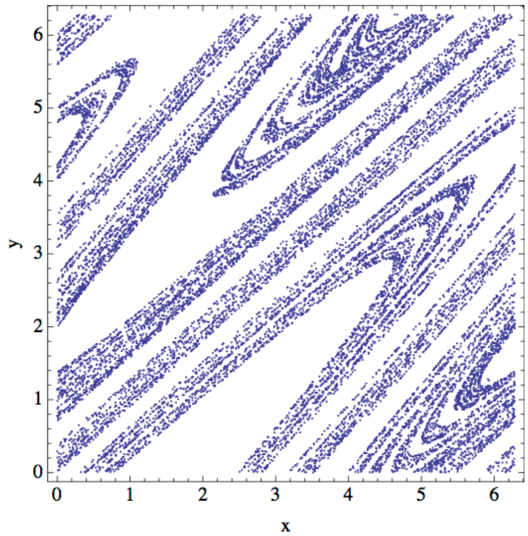

Fig. 2 *x*, *y* plot of the Dissipative standard map

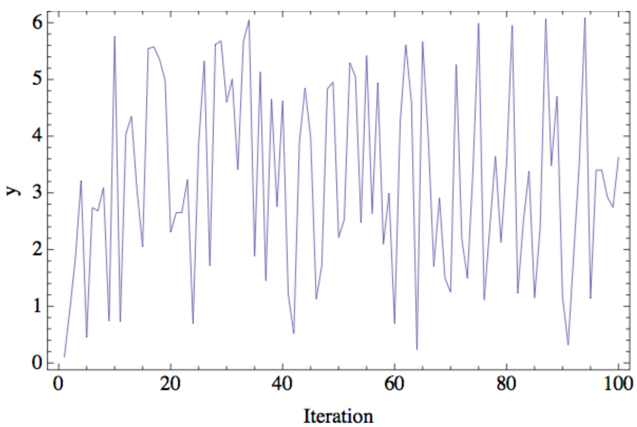

Fig. 3 Simulation of the Dissipative map (variable *y* – line-plot)

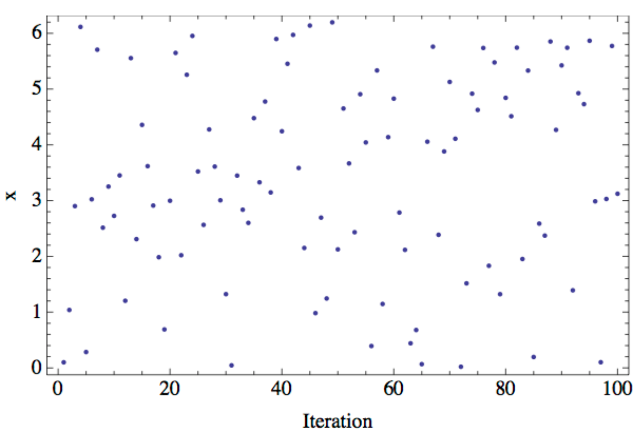

Fig. 4 Direct output iterations of the Dissipative map – variable *x*

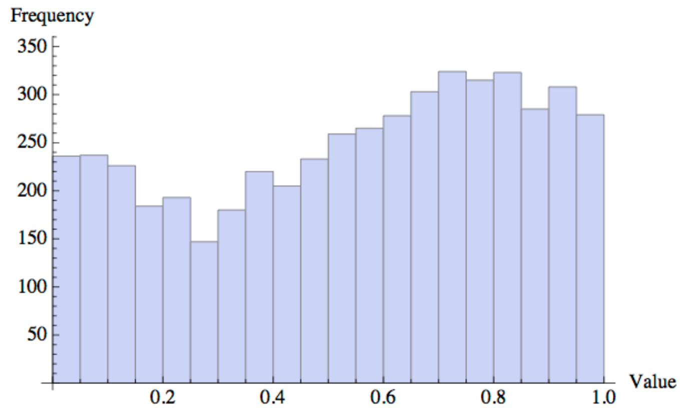

Fig. 5 Histogram of the distribution of real numbers transferred into the range <0 - 1> generated by means of the chaotic Dissipative standard map – 5000 samples

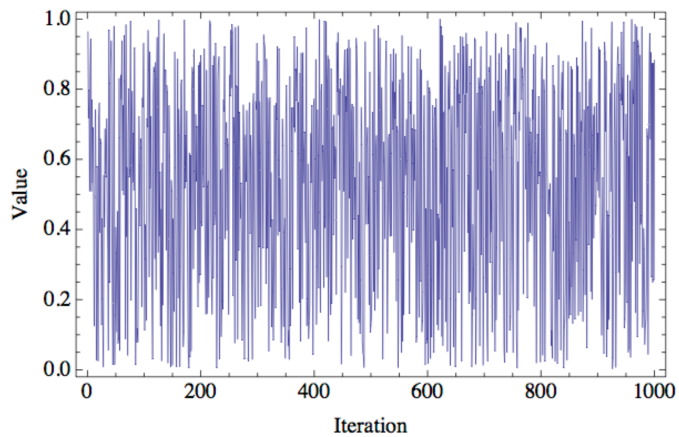

Fig. 6 Example of the chaotic dynamics: range <0 - 1> generated by means of the chaotic Dissipative standard map

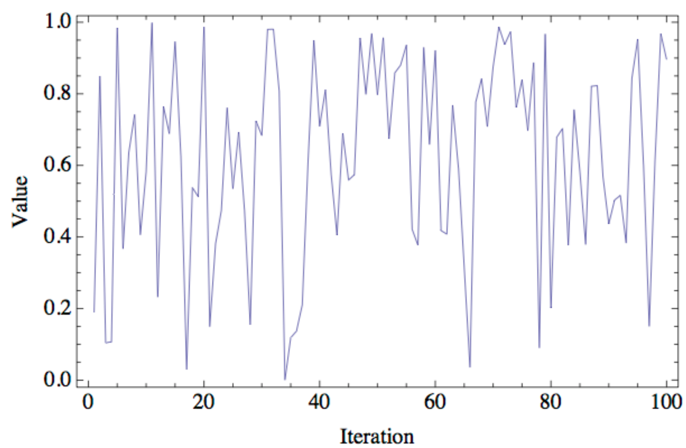

Fig. 7 Detailed example of the chaotic dynamics - sequencing: range <0 - 1> generated by means of the chaotic Dissipative standard map

VI. Problem Design

A. PID Controller

The PID controller contains three unique parts; proportional, integral and derivative controller [1-4]. A simplified form in Laplace domain is given in (5).

$$G(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right) \tag{5}$$

The PID form most suitable for analytical calculations is given in (6).

$$G(s) = k_p + \frac{k_i}{s} + k_d s \tag{6}$$

The parameters are related to the standard form through: $k_p = K$, $k_i = K/T_i$ and $k_d = KT_d$. Acquisition of the combination of these three parameters that gives the lowest value of the test criterions was the objective of this research.
Selected controlled system was the 4[th] order system that is given by Eq. 7.

$$G(s) = \frac{1}{s^4 + 6s^3 + 11s^2 + 6s} \tag{7}$$

B. Cost function

Test criterion measures properties of output transfer function and can indicate quality of regulation [1-4]. Following four different integral criterions were used for the test and comparison purposes: IAE (Integral Absolute Error), ITAE (Integral Time Absolute Error), ISE (Integral Square Error) and MSE (Mean Square Error). These test criterions (given by Eq. 8 – 11) were minimized within the cost functions for the both chaos enhanced and canonical heuristic.

a) Integral of Time multiplied by Absolute Error (ITAE)

$$I_{ITAE} = \int_0^T t\left|e(t)\right|dt \tag{8}$$

b) Integral of Absolute Magnitude of the Error (IAE)

$$I_{IAE} = \int_0^T \left|e(t)\right|dt \tag{9}$$

c) Integral of the Square of the Error (ISE)

$$I_{ISE} = \int_0^T e^2(t)dt \tag{10}$$

d) Mean of the Square of the Error (MSE)

$$I_{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(e(t)\right)^2 \tag{11}$$

VII. Results

In this section, the results obtained within experiments with ChaosDE algorithm driven by Dissipative standard map are compared with previously published works [3, 5, 10 - 13]. Table 1 shows the typical used settings for the both ChaosDe and Canonical DE.

TABLE I. DE SETTINGS

DE Parameter	Value
PopSize	25
F	0.8
CR	0.8
Generations	50
Max. CF Evaluations (CFE)	750

Best results obtained for each method are given in Table 2. The bold values within the all Tables depict the best obtained results.

Furthermore an example of comparison of the step responses of the system with PID controllers designed by means of Chaos DE and Canonical DE is depicted in Fig 8.

TABLE II. COMPARISONS OF RESULTS FOR OTHER HEURISTICS – 4[TH] ORDER SYSTEM PID CONTROLLER DESIGN

Criterion	ZN Step Response	Canonical DE	Chaos DE	Chaos SOMA	PSO	Chaos PSO
IAE	34.9413	12.3262	**12.3260**	12.3305	12.3738	12.3479
ITAE	137.5650	15.1935	**15.1919**	15.3846	16.4079	15.5334
ISE	17.8426	**6.40515**	**6.40515**	6.41026	6.40538	6.40516
MSE	0.089213	**0.032026**	**0.032026**	0.032027	0.032030	**0.032026**

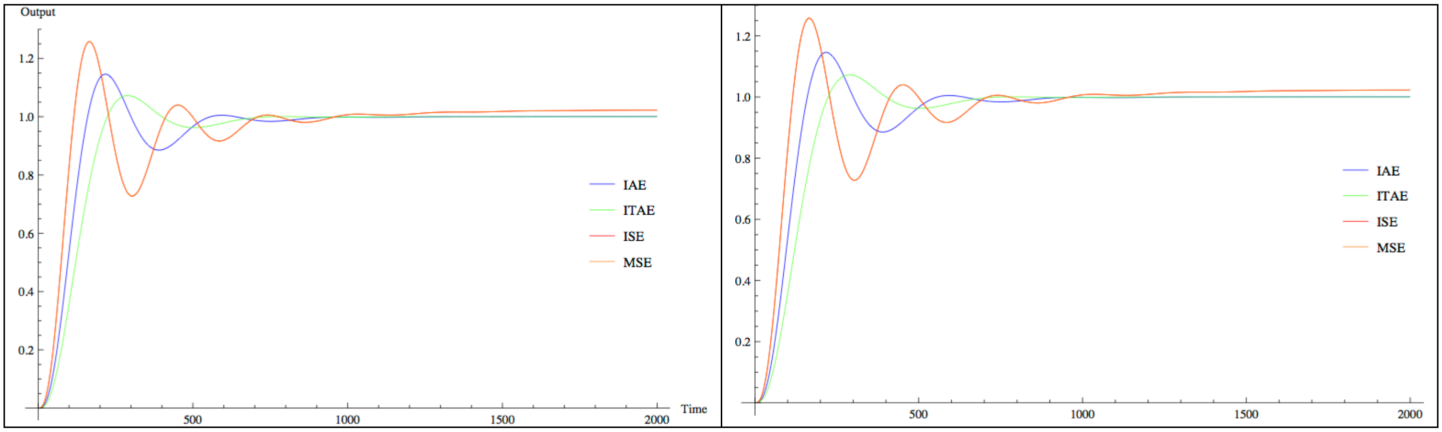

Fig. 8 Comparison of system responses – 4th order system – best results given by ChaosDE (left) and Canonical DE (right)

VIII. Performance Analysis of ChaosDE

The statistical results of the experiments for all four evaluated criterions are shown in Tables 3 – 6. These Tables represent the simple statistics for cost function (CF) values, e.g. average, median, maximum values, standard deviations and minimum values representing the best individual solution for all 50 repeated runs of canonical DE and ChaosDE. Illustrative graphical comparisons of time evolution of CF values (average values from 50 repeated runs) for both ChaosDE and Canonical DE and all four criterions are depicted in Fig 9. Based on the numerical and graphical results, it is possible to claim, that the chaos driven (embedded) heuristics has given better overall performance in finding the best settings of PID controller for IAE and ITAE criterions. For the remaining two criterions (ISE and MSE), the performance of both studied heuristics is comparable, a slightly better results are obtained through the utilization of ChaosDE (See StdDev values – although the average and median values are identical).

TABLE III. Statistical results of all 50 runs of Chaos DE and Canonical DE -IAE Criterion

DE Version	Avg CF	Median CF	Max CF	Min CF	StdDev
Canonical DERand1Bin	**12.3274**	12.327	**12.3314**	12.3262	**0.00121591**
Chaos DE with Dissipative Map	**12.3274**	**12.3268**	12.3338	**12.326**	0.00141141

TABLE IV. Statistical results of all 50 runs of Chaos DE and Canonical DE -ITAE Criterion

DE Version	Avg CF	Median CF	Max CF	Min CF	StdDev
Canonical DERand1Bin	15.2292	15.2171	15.3834	15.1935	0.0413201
Chaos DE with Dissipative Map	**15.2251**	**15.2127**	**15.3212**	**15.1919**	**0.0337992**

TABLE V. Statistical results of all 50 runs of Chaos DE and Canonical DE -ISE Criterion

DE Version	Avg CF	Median CF	Max CF	Min CF	StdDev
Canonical DERand1Bin	**6.40516**	**6.40516**	6.40517	**6.40515**	2.51955*10^-6
Chaos DE with Dissipative Map	**6.40516**	**6.40516**	**6.40516**	**6.40515**	**1.9712*10^-6**

TABLE VI. Statistical results of all 50 runs of Chaos DE and Canonical DE -MSE Criterion

DE Version	Avg CF	Median CF	Max CF	Min CF	StdDev
Canonical DERand1Bin	**0.0320258**	**0.0320258**	**0.0320258**	**0.0320258**	9.91478*10^-9
Chaos DE with Dissipative Map	**0.0320258**	**0.0320258**	**0.0320258**	**0.0320258**	**8.83727*10^-9**

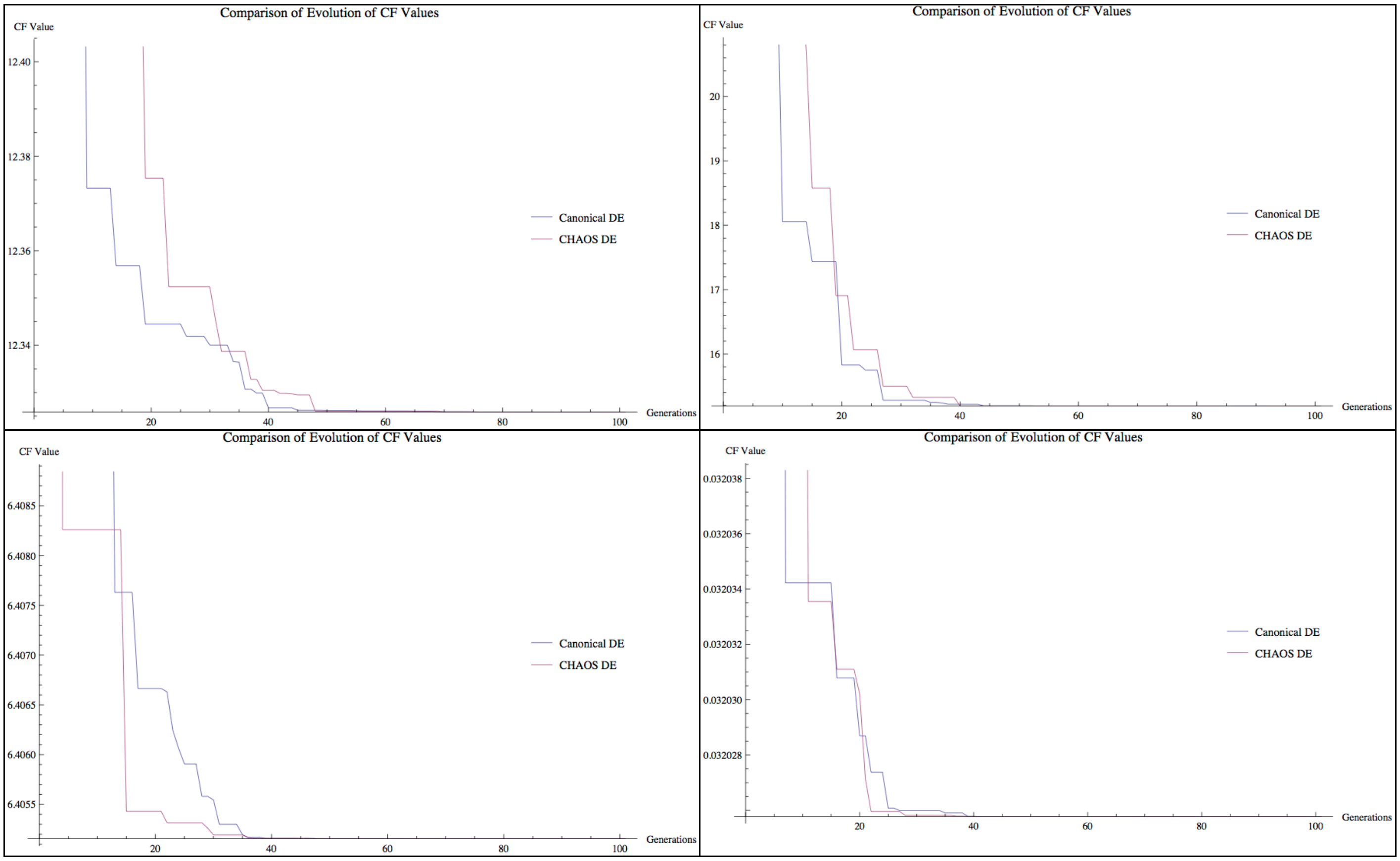

Fig. 9 Comparison of time evolution of CF values (average from 50 repeated runs) for both ChaosDE and Canonical DE – IAE Criterion (upper left); ITAE criterion (upper right); ISE Criterion (below left) and MSE criterion (below right)

IX. Conclusion

In this paper the chaotic dissipative standard map was presented and investigated over their capability of enhancing the performance of DE algorithm (ChaosDE) in the task of PID controller design.

From the comparisons, it follows that through the utilization of chaotic systems; the best overall results were obtained and entirely different statistical characteristics of CPRNGs-based heuristic can be achieved. Thus the different influence to the system, which utilizes the selected CPRNG, can be chosen through the implementation of particular inner hidden complex chaotic dynamics given by the particular chaotic system.

Promising results were presented, discussed and compared with other methods of PID controller design. More detail experiments are needed to prove or disprove these claims and explain the effect of the chaotic systems on the optimization and controller design.

References

[1] K. J. Åström. (2002). *Control System Design - Lecture Notes for ME 155A*.Available: http://www.cds.caltech.edu/~murray/courses/cds101/fa02/caltech/astrom.html

[2] I. D. Landau and Z. Gianluca, *Digital Control Systems*: Springer London, 2006.

[3] B. Nagaraj, S. Subha, and B. Rampriya, "Tuning Algorithms for PID Controller Using Soft Computing Techniques " *International Journal of Computer Science and Network Security,* vol. 8, pp. 278-281, 2008.

[4] Saad, Mohd S., Hishamuddin Jamaluddin, and Intan ZM Darus. "PID controller tuning using evolutionary algorithms." WSEAS TRANSACTIONS on SYSTEMS and control, vol. 7, No 4, pp. 139-149, 2012.

[5] D. Davendra, I. Zelinka, and R. Senkerik, "Chaos driven evolutionary algorithms for the task of PID control," *Computers & Mathematics with Applications,* vol. 60, pp. 1088-1104, 2010.

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.

[7] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," presented at the CEC 99. Proceedings of the 1999 Congress on Evolutionary Computation, 1999.

[8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*: Addison-Wesley, 1989.

[9] R. Storn and K. Price, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," *Journal of Global Optimization,* vol. 11, pp. 341-359, 1997.

[10] M. Pluhacek, R. Senkerik, D. Davendra, I. Zelinka, "Designing PID Controller For DC Motor System By Means Of Enhanced PSO Algorithm With Discrete Chaotic Lozi Map", in *Proc. 26th European Conference on Modelling and Simulation, ECMS 2012*, pp. 405 - 409, 2012, ISBN 978-0-9564944-4-3.

[11] M. Pluhacek, R. Senkerik, D. Davendra, I. Zelinka, "PID Controller Design For 4th Order System By Means Of Enhanced PSO algorithm With Lozi Chaotic Map", in *Proc. 18th International Conference on Soft*

Computing, MENDEL 2012, pp. 35 - 39, 2012, ISBN 978-80-214-4540-6.

[12] M. Pluhacek, R. Senkerik, D. Davendra, I. Zelinka, “Designing PID controller for DC motor by means of enhanced PSO algorithm with dissipative chaotic map”. In: Snasel, V., Abraham, A., Corchado, E.S. (eds.) *SOCO Models in Industrial & Environmental Appl*. AISC, vol. 188, pp. 475–483. Springer, Heidelberg (2013)

[13] M. Pluhacek, R. Senkerik, D. Davendra, I. Zelinka: “Designing PID Controller for 4th Order System By Means of Enhanced PSO Algorithm with Discrete Chaotic Dissipative Standard Map”. in *Proc. 24th European Modeling & Simulation Symposium, EMSS 2012*, pp. 396–401 (2012) ISBN 978-88-97999-09-6

[14] R. Senkerik, M. Pluhacek, D. Davendra, I. Zelinka and Z. Oplatkova, " Chaos Driven Evolutionary Algorithm: A New Approach for Evolutionary Optimization," *International Journal of Mathematics and Computers in Simulation*, vol. 7, pp. 363-368, 2013.

[15] R. Senkerik, I. Zelinka, M. Pluhacek, and Z. Oplatkova, "Evolutionary Control of Chaotic Burgers Map by means of Chaos Enhanced Differential Evolution," *International Journal of Mathematics and Computers in Simulation*, vol. 8, pp. 39-45, 2014.

[16] R. Caponetto, L. Fortuna, S. Fazzino, and M. G. Xibilia, "Chaotic sequences to improve the performance of evolutionary algorithms," *IEEE Transactions on Evolutionary Computation,* vol. 7, pp. 289-304, 2003.

[17] K. Price, “An Introduction to Differential Evolution”, *In: New Ideas in Optimization*, (D. Corne, M. Dorigo and F. Glover, Eds.), pp. 79–108, McGraw-Hill, London, UK, ISBN 007-709506-5, 1999.

[18] K. Price and R. Storn, “*Differential evolution homepage*” (2001) [Online]. Available: http://www.icsi.berkeley.edu/~storn/code.html.

[19] K. Price, R. Storn and J. Lampinen, "*Differential Evolution - A Practical Approach to Global Optimization*", 2005, Springer, ISBN: 3-540-20950-6.

[20] J. C. Sprott, *Chaos and Time-Series Analysis*: Oxford University Press, 2003.