Multi-chaotic Differential Evolution: A Preliminary Study

Roman Senkerik[1,⋆], Michal Pluhacek[1], Ivan Zelinka[2], Donald Davendra[2], and Zuzana Kominková Oplatková[1]

[1] Tomas Bata University in Zlin, Faculty of Applied Informatics, Nam T.G. Masaryka 5555, 760 01 Zlin, Czech Republic
{senkerik,pluhacek,oplatkova}@fai.utb.cz
[2] Technical University of Ostrava, Faculty of Electrical Engineering and Computer Science, 17. listopadu 15,708 33 Ostrava-Poruba, Czech Republic
{donald.davendra,ivan.zelinka}@vsb.cz

Abstract. This research deals with the initial investigations on the concept of a multi-chaos-driven evolutionary algorithm Differential Evolution (DE). This paper is aimed at the embedding and alternating of set of two discrete dissipative chaotic systems in the form of chaos pseudo random number generator for DE. Repeated simulations were performed on the selected test function in higher dimensions. Finally, the obtained results are compared with canonical DE.

Keywords: Differential Evolution, Deterministic chaos, Dissipative systems, Optimization.

1 Introduction

These days 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. Differential Evolution (DE) [1] is one of the most potent heuristics available.

This paper is aimed at the investigating the novel concept of multi-chaos driven DE. Although a number of DE variants have been recently developed, the focus of this paper is the embedding of chaotic systems in the form of chaos pseudo random number generator (CPRNG) into the DE.

⋆ This work was supported by Grant Agency of the Czech Republic - GACR P103/13/08195S, partially supported by Grants of SGS No. SP2014/159 and SP2014/170, VSB - Technical University of Ostrava, Czech Republic, by the Development of human resources in research and development of latest soft computing methods and their application in practice project, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic, further was supported by European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089 and by Internal Grant Agency of Tomas Bata University under the project No. IGA/FAI/2014/010.

L. Rutkowski et al. (Eds.): ICAISC 2014, Part I, LNAI 8467, pp. 416–427, 2014.

Firstly, the motivation for this research is proposed. The next sections are focused on the description of evolutionary algorithm DE, the concept of chaos driven DE and the used test function. Results and conclusion follow afterwards.

2 Motivation

This research is an extension and continuation of the previous successful initial experiments with chaos driven DE [2], [3] with test functions in higher dimensions.

In this paper the novel initial concept of DE/rand/1/bin strategy driven alternately by two chaotic maps (systems) is introduced. From the previous research it follows, that very promising results were obtained through the utilization of Delayed Logistic, Lozi, Burgers and Tinkerbelt chaotic maps. The last two mentioned chaotic maps have unique properties with connection to DE: strong progress towards global extreme, but weak overall statistical results, like average CF value and std. dev., and tendency to premature stagnation. While through the utilization of the Lozi and Delayed Logistic map the continuously stable and very satisfactory performance of ChaosDE was achieved. The idea is then to connect these two different influences to the performance of DE into the one multi-chaotic concept.

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 [4]. 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 initial concept of embedding chaotic dynamics into the evolutionary algorithms is given in [5]. Later, the initial study [6] was focused on the simple embedding of chaotic systems in the form of chaos pseudo random number generator (CPRNG) for DE and SOMA [7] in the task of optimal PID tuning

Several papers have been recently focused on the connection of heuristic and chaotic dynamics either in the form of hybridizing of DE with chaotic searching algorithm [8] or in the form of chaotic mutation factor and dynamically changing weighting and crossover factor in self-adaptive chaos differential evolution (SACDE) [9]. Also the PSO (Particle Swarm Optimization) algorithm with elements of chaos was introduced as CPSO [10] or CPSO combined with chaotic local search [11].

The focus of our research is the pure embedding of chaotic systems in the form of chaos pseudo random number generator for evolutionary algorithms.

This idea was later extended with the successful experiments with chaos driven DE (ChaosDE) [2], [3] with both and complex simple test functions and in the task of chemical reactor geometry optimization [12].

The concept of Chaos DE has proved itself to be a powerful heuristic also in combinatorial problems domain [13].

At the same time the chaos embedded PSO with inertia weigh strategy was closely investigated [14], followed by the introduction of a PSO strategy driven alternately by two chaotic systems [15] and novel chaotic Multiple Choice PSO strategy (Chaos MC-PSO) [16].

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 use and test the implementation of natural chaotic dynamics into evolutionary algorithm as a multi-chaotic pseudo random number generator.

3 Differential Evolution

DE is a population-based optimization method that works on real-number-coded individuals [1]. For each individual $\boldsymbol{x}_{i,G}$ in the current generation G, DE generates a new trial individual $\boldsymbol{x}'_{i,G}$ by adding the weighted difference between two randomly selected individuals $\boldsymbol{x}_{r1,G}$ and $\boldsymbol{x}_{r2,G}$ to a randomly selected third individual $\boldsymbol{x}_{r3,G}$. The resulting individual $\boldsymbol{x}'_{i,G}$ is crossed-over with the original individual$\boldsymbol{x}_{i,G}$. The fitness of the resulting individual, referred to as a perturbed vector $\boldsymbol{u}_{i,G+1}$, is then compared with the fitness of $\boldsymbol{x}_{i,G}$. If the fitness of $\boldsymbol{u}_{i,G+1}$ is greater than the fitness of$\boldsymbol{x}_{i,G}$, then $\boldsymbol{x}_{i,G}$ is replaced with$\boldsymbol{u}_{i,G+1}$; otherwise, $\boldsymbol{x}_{i,G}$ remains in the population as$\boldsymbol{x}_{i,G+1}$. 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. Please refer to [1], [17] for the detailed description of the used DE-Rand1Bin strategy (1) (both for Chaos DE and Canonical DE) as well as for the complete description of all other strategies.

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

4 The Concept of ChaosDE

The general idea of ChaosDE and CPRNG is to replace the default PRNG with the discrete chaotic map. As the discrete chaotic map is a set of equations with a static start position, we created a random start position of the map, 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. Once the start position of the chaotic map has been obtained, the map generates the next sequence using its current position.

The first possible way is to generate and store a long data sequence (approx. 50-500 thousand numbers) during the evolutionary process initialization and keep the pointer to the actual used value in the memory. In case of the using up of the whole sequence, the new one will be generated with the last known value as the new initial one.

The second approach is that the chaotic map is not re-initialized during the experiment and no long data series is stored, thus it is imperative to keep the current state of the map in memory to obtain the new output values.

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

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

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

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.

5 Chaotic Maps

This section contains the description of discrete dissipative chaotic maps used as the chaotic pseudo random generators for DE. In this research, direct output iterations of the chaotic maps were used for the generation of real numbers in the process of crossover based on the user defined CR value and for the generation of the integer values used for the selection of individuals. Following chaotic maps were used: Burgers (4), and Lozi map (5).

The Burgers mapping is a discretization of a pair of coupled differential equations which were used by Burgers [18] to illustrate the relevance of the concept of bifurcation to the study of hydrodynamics flows. The map equations are given in (4) with control parameters $a = 0.75$ and $b = 1.75$ as suggested in [19].

$$\begin{aligned} X_{n+1} &= aX_n - Y_n^2 \\ Y_{n+1} &= bY_n + X_nY_n \end{aligned} \tag{4}$$

The Lozi map is a discrete two-dimensional chaotic map. The map equations are given in (5). The parameters used in this work are: $a = 1.7$ and $b = 0.5$ as suggested in [19]. 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.

$$\begin{aligned} X_{n+1} &= 1 - a\left|X_n\right| + bY_n \\ Y_{n+1} &= X_n \end{aligned} \tag{5}$$

5.1 Graphical Example – Lozi Map and Burgers Map

The x, y plots of the chaotic maps are depicted in Fig. 1 - left (Lozi map) and Fig. 3 - left (Burgers map). The typical chaotic behavior of the utilized maps, represented by the examples of direct output iterations is depicted in Fig. 1 - right (Lozi map) and Fig. 3 - right (Burgers map).

The illustrative histograms of the distribution of real numbers transferred into the range <0 - 1> generated by means of studied chaotic maps are in Figures 2 and 4.

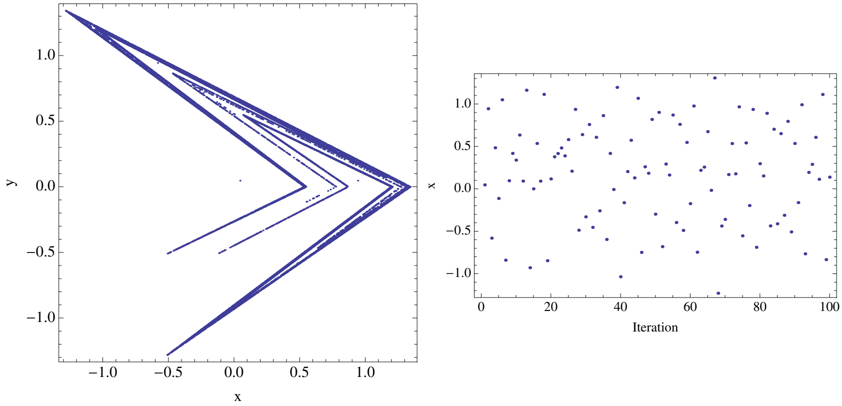

Fig. 1. x, y plot of the Lozi map (left); Iterations of the Lozi map (variable x) (right)

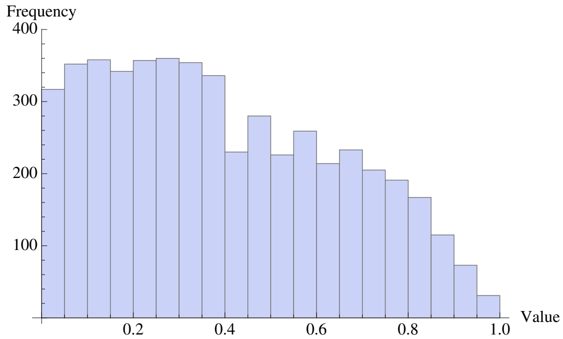

Fig. 2. Histogram of the distribution of real numbers generated by means of the chaotic Lozi map transferred into the range <0 - 1> – 5000 samples

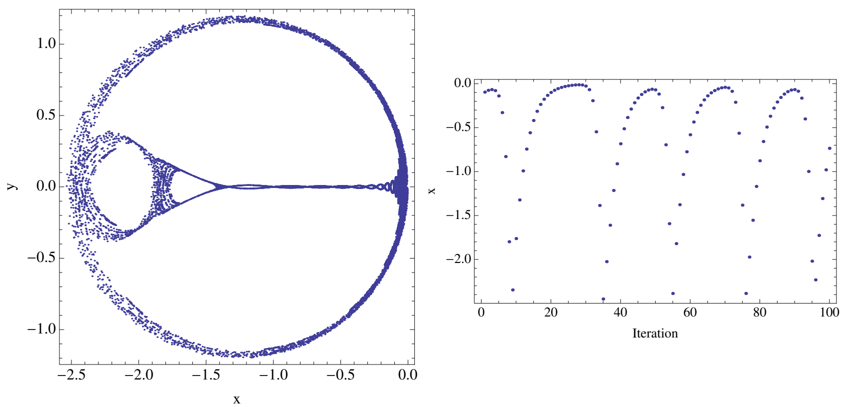

Fig. 3. x, y plot of the Burgers map (left); Iterations of the Burgers map (variable x) (right)

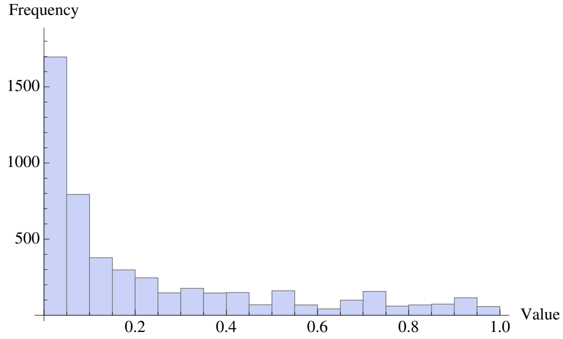

Fig. 4. Histogram of the distribution of real numbers generated by means of the chaotic Burgers map transferred into the range <0 - 1> – 5000 samples

6 Benchmark Function

For the purpose of evolutionary algorithm performance comparison within this initial research, the multimodal Schwefel's test function (6) was selected.

$$f(x) = \sum_{i=1}^{D} -x_i \sin\left(\sqrt{|x_i|}\right) \tag{6}$$

Function minimum:
Position for E_n: $(x_1, x_2 \ldots x_n) = (420.969, 420.969, \ldots, 420.969)$
Value for E_n: $y = -418.983 \cdot Dimension$

7 Results

The novelty of this approach represents the utilization of discrete chaotic maps as the multi-chaotic pseudo random number generator for the DE. In this paper, the canonical DE strategy DERand1Bin and the Multi-Chaos DERand1Bin strategy driven alternately by two different chaotic maps (ChaosDE) were used.

The previous research [2], [3] showed that through the utilization of Burgers and Tinkerbelt maps the unique properties with connection to DE were achieved: strong progress towards global extreme, but weak overall statistical results, like average CF value and std. dev. Whereas through the utilization of the Lozi and Delayed Logistic maps, the continuously stable and very satisfactory performance of ChaosDE was achieved. The idea is then to connect these two different influences to the performance of DE into the one novel multi-chaotic concept. The moment of manual switching over between two chaotic maps as well as the parameter settings for both canonical DE and ChaosDE were obtained analytically based on numerous experiments and simulations (see Table 1)

Experiments were performed in the combined environment of *Wolfram Mathematica* and *C language*, canonical DE therefore used the built-in *C language*

Table 1. Parameter set up for Chaos DE and Canonical DE

Parameter	Value
PopSize	75
F	0.8
CR	0.8
Dimensions	30
Generations	$100 \cdot D = 3000$
Max Cost Function Evaluations (CFE)	225000

pseudo random number generator *Mersenne Twister C* representing traditional pseudorandom number generators in comparisons. All experiments used different initialization, i.e. different initial population was generated within the each run of Canonical or Chaos driven DE.

Within this initial research, one type of experiment was performed. It utilizes the maximum number of generations fixed at 3000 generations. This allowed the possibility to analyze the progress of DE within a limited number of generations and cost function evaluations.

The statistical results of the experiments are shown in Table 2, which represent the simple statistics for cost function 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 several versions of ChaosDE and Multi-ChaosDE.

Table 3 compares the progress of several versions of ChaosDE, Multi-ChaosDE and Canonical DE. This table contains the average CF values for the generation No. 750, 1500, 2250 and 3000 from all 50 runs. The bold values within the both Tables 2 and 3 depict the best obtained results. Following versions of Multi-ChaosDE were studied:

- *Burgers-Lozi-Switch-500*: Start with Burgers map CPRNG, switch to the Lozi map CPRNG after 500 generations.
- *Lozi-Burgers-Switch-1500*: Start with Lozi map CPRNG, switch to the Burgers map CPRNG after 1500 generations.

Table 2. Simple results statistics for the Schwefel's function – 30D

DE Version	Avg CF	Median CF	Max CF	Min CF	StdDev
Canonical DE	-5822.8	-5754.4	-5443.23	-6500.44	**226.4365**
Lozi-No-Switch	-11296.9	-11581	-7842.25	-12235.5	879.1985
Burger-No-Switch	-11052.1	-11192.9	-8473.79	-12105	667.7065
Burger-Lozi-Switch-500	-11332.9	-11459.1	-7871.2	**-12486.9**	799.7749
Lozi-Burger-Switch-1500	**-11475.5**	**-11489.6**	**-10354.5**	-12279.7	373.059

Table 3. Comparison of progress towards the minimum for the Schwefel's function

DE Version	Generation No. 750	Generation No. 1500	Generation No. 2250	Generation No. 3000
Canonical DE	-5231.94	-5537.79	-5738.96	-5822.8
Lozi-No-Switch	-5839.69	-7998.35	-9965.25	-11296.9
Burger-No-Switch	-6075.91	**-8854.6**	-10564.1	-11052.1
Burger-Lozi-Switch-500	**-6538.11**	-8658.15	-10356.3	-11332.9
Lozi-Burger-Switch-1500	-5701.57	-7719.37	**-10663.1**	**-11475.5**

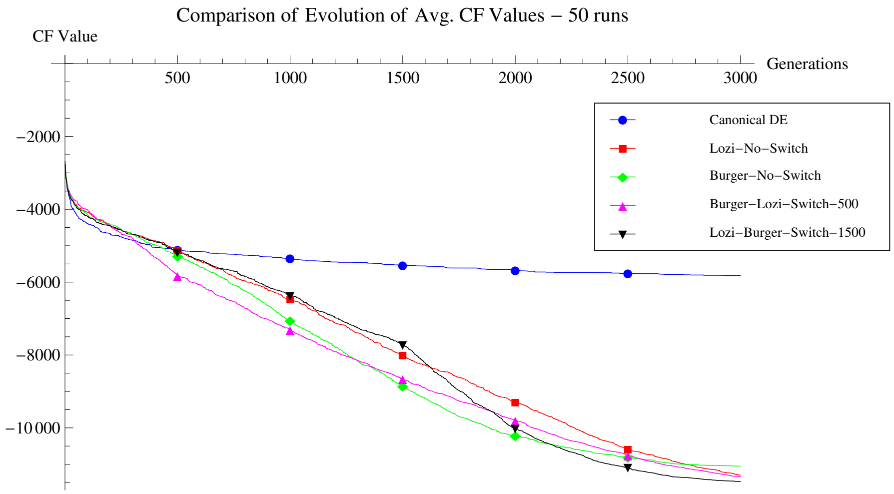

Fig. 5. Comparison of the time evolution of avg. CF values for the all 50 runs of Canonical DE, ChaosDE and Multi-ChaosDE. Schwefel's function, $D = 30$.

The graphical comparison of the time evolution of average CF values for all 50 runs of ChaosDE/Multi-ChaosDE and canonical DERand1Bin strategy is depicted in Fig. 5. Finally the Figures 6 - 8 confirm the robustness of Multi-ChaosDE in finding the best solutions for all 50 runs.

Obtained numerical results given in Tables 2 and 3 and graphical comparisons in Figures 5 - 8 support the claim that all Multi-Chaos/ChaosDE versions have given better overall results in comparison with the canonical DE version. From the presented data it follows, that Multi-Chaos DE versions driven by Lozi/Burgers Map have given the best overall results.

For the *Burgers-Lozi-Switch-500* version the progressive Burgers map CPRNG secured the faster approaching towards the global extreme from the very beginning of evolutionary process. The very fast switch over to the Lozi map based CPRNG helped to avoid the Burgers map based CPRNG weak spots, which are the weak overall statistical results, like average CF value and std. dev.; and

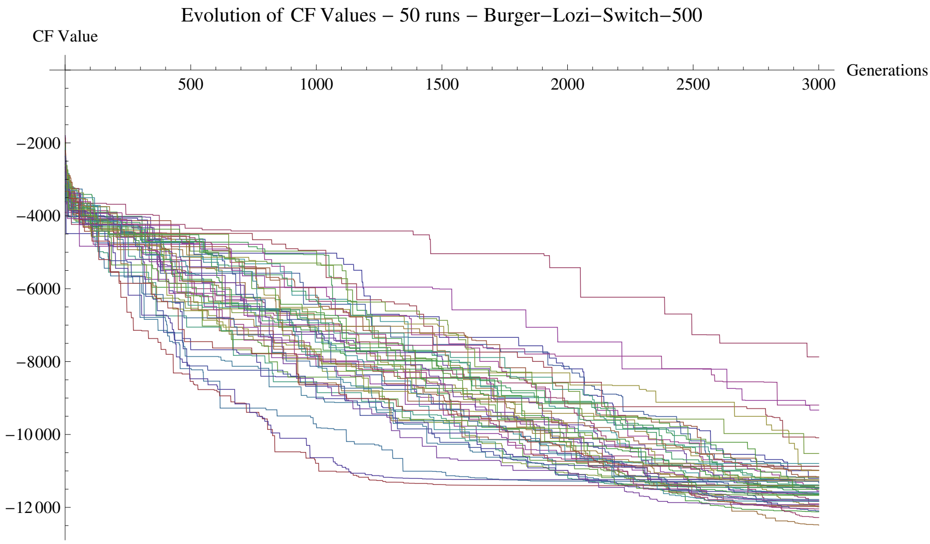

Fig. 6. Comparison of the time evolution of CF values for all 50 runs of Multi-ChaosDE version: Burgers-Lozi-Switch-500

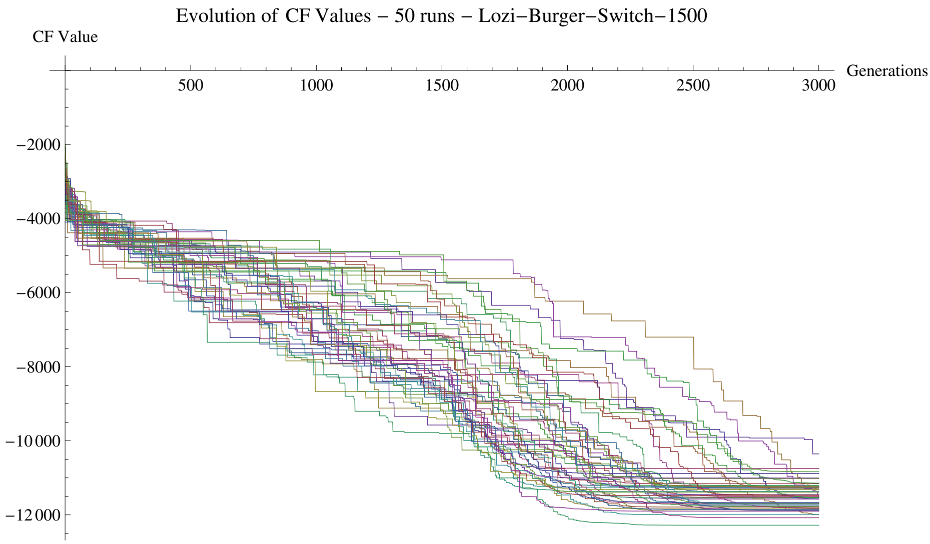

Fig. 7. Comparison of the time evolution of CF values for all 50 runs of Multi-ChaosDE version: Lozi- Burgers -Switch-1500

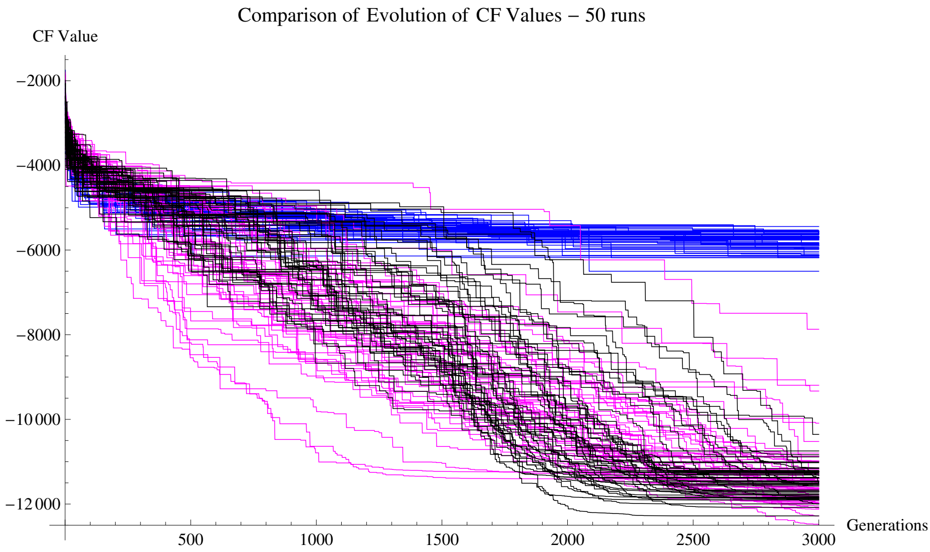

Fig. 8. Comparison of the time evolution of CF values for all 50 runs of canonical DE (blue) and Multi-ChaosDE versions: Burgers-Lozi-Switch-500 (magenta), Lozi-Burgers-Switch-1500 (black)

tendency to stagnation. This version was able to reach the best individual minimum CF value. The initial faster convergence (starting of evolutionary process) and subsequent continuously stable searching process without premature stagnation issues are visible from Fig. 5 (magenta line), Fig. 6 and Fig. 8 (magenta lines).

Through the utilization of *Lozi-Burgers-Switch-1500* version, the strong progress towards global extreme given by Burgers map CPRNG helped to the evolutionary process driven from the start by mans of Lozi map CPRNG to achieve the best avg. CF and median CF values. The moment of switch (at 1500 generations) is clearly visible from Fig. 5 (black line) and Fig. 7 and Fig. 8 (black lines).

8 Conclusions

In this paper, the novel concept of multi-chaos driven DERand1Bin strategy was tested and compared with the canonical DERand1Bin strategy on the selected benchmark function in higher dimension. Based on obtained results, it may be claimed, that the developed Multi-ChaosDE gives considerably better results than other compared heuristics.

Since this was a preliminary study of the novel presented concept, only one single benchmark function in higher dimensions was utilized to test and more deeply analyze the influence of alternating several CPRNGs to the performance of original previous ChaosDE concept. Nevertheless the original concept of ChaosDE itself was tested on huge set of both simple and complex benchmark functions based mostly on the IEEE CEC 2005 benchmark set and with nine different discrete dissipative chaotic systems. Thus based on the deeper analysis of results from the previous research the composition of the presented experiment was prepared.

Future plans are including the testing of combination of different chaotic systems as well as the adaptive switching and obtaining a large number of results to perform statistical tests.

Furthermore chaotic systems have additional parameters, which can by tuned. This issue opens up the possibility of examining the impact of these parameters to generation of random numbers, and thus influence on the results obtained by means of ChaosDE.

References

1. Price, K.V.: An Introduction to Differential Evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 79–108. McGraw-Hill Ltd. (1999)
2. Senkerik, R., Pluhacek, M., Zelinka, I., Oplatkova, Z.K., Vala, R., Jasek, R.: Performance of Chaos Driven Differential Evolution on Shifted Benchmark Functions Set. In: Herrero, A., et al. (eds.) International Joint Conference SOCO'13-CISIS'13-ICEUTE'13. AISC, vol. 239, pp. 41–50. Springer, Heidelberg (2014)
3. Senkerik, R., Davendra, D., Zelinka, I., Pluhacek, M., Kominkova Oplatkova, Z.: On the Differential Evolution Driven by Selected Discrete Chaotic Systems: Extended Study. In: 19th International Conference on Soft Computing, MENDEL 2013, pp. 137–144 (2013)
4. Aydin, I., Karakose, M., Akin, E.: Chaotic-based hybrid negative selection algorithm and its applications in fault and anomaly detection. Expert Systems with Applications 37(7), 5285–5294 (2010)
5. Caponetto, R., Fortuna, L., Fazzino, S., Xibilia, M.G.: Chaotic sequences to improve the performance of evolutionary algorithms. IEEE Transactions on Evolutionary Computation 7(3), 289–304 (2003)
6. Davendra, D., Zelinka, I., Senkerik, R.: Chaos driven evolutionary algorithms for the task of PID control. Computers & Mathematics with Applications 60(4), 1088–1104 (2010)
7. Zelinka, I.: SOMA — Self-Organizing Migrating Algorithm. In: Onwubolu, G.C., Babu, B.V. (eds.) New Optimization Techniques in Engineering. STUDFUZZ, vol. 141, pp. 167–217. Springer, Heidelberg (2004)
8. Liang, W., Zhang, L., Wang, M.: The chaos differential evolution optimization algorithm and its application to support vector regression machine. Journal of Software 6(7), 1297–1304 (2011)
9. Zhenyu, G., Bo, C., Min, Y., Binggang, C.: Self-Adaptive Chaos Differential Evolution. In: Jiao, L., Wang, L., Gao, X.-B., Liu, J., Wu, F. (eds.) ICNC 2006. LNCS, vol. 4221, pp. 972–975. Springer, Heidelberg (2006)
10. dos Santos Coelho, L., Mariani, V.C.: A novel chaotic particle swarm optimization approach using Henon map and implicit filtering local search for economic load dispatch. Chaos, Solitons & Fractals 39(2), 510–518 (2009)

11. Hong, W.-C.: Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model. Energy Conversion and Management 50(1), 105–117 (2009)
12. Senkerik, R., Pluhacek, M., Oplatkova, Z.K., Davendra, D., Zelinka, I.: Investigation on the Differential Evolution driven by selected six chaotic systems in the task of reactor geometry optimization. In: 2013 IEEE Congress on Evolutionary Computation (CEC), June 20-23, pp. 3087–3094 (2013)
13. Davendra, D., Bialic-Davendra, M., Senkerik, R.: Scheduling the Lot-Streaming Flowshop scheduling problem with setup time with the chaos-induced Enhanced Differential Evolution. In: 2013 IEEE Symposium on Differential Evolution (SDE), April 16-19, pp. 119–126 (2013)
14. Pluhacek, M., Senkerik, R., Davendra, D., Kominkova Oplatkova, Z., Zelinka, I.: On the behavior and performance of chaos driven PSO algorithm with inertia weight. Computers & Mathematics with Applications 66(2), 122–134 (2013)
15. Pluhacek, M., Senkerik, R., Zelinka, I., Davendra, D.: Chaos PSO algorithm driven alternately by two different chaotic maps - An initial study. In: 2013 IEEE Congress on Evolutionary Computation (CEC), June 20-23, pp. 2444–2449 (2013)
16. Pluhacek, M., Senkerik, R., Zelinka, I.: Multiple Choice Strategy Based PSO Algorithm with Chaotic Decision Making – A Preliminary Study. In: Herrero, A., et al. (eds.) International Joint Conference SOCO'13-CISIS'13-ICEUTE'13. AISC, vol. 239, pp. 21–30. Springer, Heidelberg (2014)
17. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution - A Practical Approach to Global Optimization. Natural Computing Series. Springer, Heidelberg (2005)
18. ELabbasy, E., Agiza, H., EL-Metwally, H., Elsadany, A.: Bifurcation Analysis, Chaos and Control in the Burgers Mapping. International Journal of Nonlinear Science 4(3), 171–185 (2007)
19. Sprott, J.C.: Chaos and Time-Series Analysis. Oxford University Press (2003)