Do Evolutionary Algorithms Indeed Require Random Numbers? Extended Study

Ivan Zelinka, Mohammed Chadli, Donald Davendra, Roman Senkerik, Michal Pluhacek, and Jouni Lampinen

Abstract. An inherent part of evolutionary algorithms, that are based on Darwin theory of evolution and Mendel theory of genetic heritage, are random processes. In this participation, we discuss whether are random processes really needed in evolutionary algorithms. We use n periodic deterministic processes instead of random number generators and compare performance of evolutionary algorithms powered by those processes and by pseudo-random number generators. Deterministic processes used in this participation are based on deterministic chaos and are used to generate periodical series with different length. Results presented here are numerical demonstration rather than mathematical proofs. We propose that a certain class of deterministic processes can be used instead of random number generators without lowering of evolutionary algorithms performance.

Ivan Zelinka · Donald Davendra
VSB-Technical University of Ostrava, 17. listopadu 15 708 33, Ostrava-Poruba, Czech Republic
e-mail: ivan.zelinka@vsb.cz

Roman Senkerik · Michal Pluhacek
Faculty of Applied Informatics, Tomas Bata University in Zlin
e-mail: senkerik@fai.utb.cz

Mohammed Chadli
Laboratory of Modeling, Information and Systems (MIS), University of Picardie Jules Verne (UPJV), 33 rue Saint-Leu, 80039 Amiens Cedex 1, France
e-mail: mchadli@u-picardie.fr

Jouni Lampinen
University of Vaasa, Department of Computer Science, P.O. Box 700, FI-65101 Vaasa, Finland
e-mail: antlam@uwasa.fi

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 61–75.
DOI: 10.1007/978-3-319-00542-3_8

1 Introduction

The term "chaos" covers a rather broad class of phenomena whose behavior may seem erratic, chaotic at first glance. Often, this term is used to denote phenomena which are of a purely stochastic nature, such as the motion of molecules in a vessel with gas and the like. The discovery of the phenomenon of deterministic chaos brought about the need to identify manifestations of this phenomenon also in experimental data. Deterministically chaotic systems are necessarily nonlinear, and conventional statistical procedures, which are mostly linear, are insufficient for their analysis. If the output of a deterministically chaotic system is subjected to linear methods, such signals will appear as the result of a random process. Examples include the Fourier spectral analysis, which will disclose nonzero amplitudes at all frequencies in a chaotic system, and so chaos can be easily mistaken for random noise. Till now, 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). Lets mention for example research papers like [1] (a comprehensive overview of mutual intersection between EAs and chaos is discussed here), [14], [15], [23], [25], discussing use of deterministic chaos inside particle swarm algorithm instead of PRGNs, [2] - [5] investigating relations between chaos and randomness or the latest one [6] and [7] using chaos with EAs in applications. Another papers using deterministic chaos inside EAs are for example [16], [17], [18], [20], [21], discussing combination of differential evolution and chaotic systems, [19] (chaotic optimization and immune co-evolution algorithm) and [22], [24] that also use chaotic dynamics in evolutionary algorithms.

This publication is focused on use of deterministic chaos to generate n periodic deterministic series, that are used inside evolutionary algorithms instead of pseudo-random number generator.

2 Motivation

The motivation of the proposed experiments here is quite simple. For a long time, various PRNGs were used inside evolutionary algorithms. During the last few years, deterministic chaos systems (DCHS) instead of PRNGs have been used. As was demonstrated in [14]-[15], very often the performance of EAs using DCHS better or fully comparable with EAs using PRNGs. See for example [14]. Used EAs (we do not discuss here the special cases, modified for special experiments) of different kinds, such as genetic algorithms [12], differential evolution [9], particle swarm [13], SOMA [8], scatter search [10], evolutionary strategies [11], etc. do not analyze whether used pseudo-random numbers are really random ones. Random numbers are simply used. On the other side, as demonstrated in mentioned references, EAs with DCHS gives the same or often better performance. Difference between series from PRNGs and DCHS is that in the case of DCHS, one can easily reconstruct/calculate whole series generated by DCHS from one point. In the PRNGs it is impossible. Because DCHS generate periodical series (thanks to final numerical precision) it is obvious that EAs performance shall be from a certain numerical precision of DCHS comparable with performance of classical EAs. This is the main aim of this paper - check whether PRNGs can be replaced by deterministic periodical dynamics.

3 Experiment Design

Our experiments has been set so that periodical deterministic time series based on deterministic chaos generators (see for example Fig. 1) and Eq. 2, were used instead of PRNGs. Based on the fact that numerical precision has impact on existence of periodicity in deterministic chaos, we have selected logistic equation, Eq. 2, and data series generated by this equation for numerical precisions from interval [1, 13] (numbers behind decimal point) with setting $A = 4$, see Tab. 2 which shows only maximal period for current setting. Algorithms selected for our experiments were SOMA [8] and differential evolution (DERand1Bin and DELocalToBest) [9] . Setting of all three algorithms is in Tab. 1. Based on these setting and algorithm architecture, it is easy to calculate how many times has been used periodic data series (PDS) in EAs generated by DCHS. The total cost function evaluation was for SOMA maximally 172 727 and for both DE versions 130 000 times. Tab. 2 summarize the many precision levels were PDS were repeatedly used. All experiments were done in Mathematica 9, on MacBook Pro, 2.8 GHz Intel Core 2 Duo. Test function used in this experiment was 10th dimensional Schwefel's function (see Eq. 1) and each experiment was 100 times repeated for each precision set. The aim was to find global extreme (10×418.983 at position 420.969, 420.969, ..., 420.969) of this function as precise as possible. So in total 3900 (3 algorithms $\times$ 100 repetitions $\times$ 13 different numerical precisions) evolutionary experiments has been done. In each experiment, the PRNGs used only on the start of Eq. 2 to set initial condition x_{start}. Remaining use of Eq. 2 was PRNGs free, i.e. PRNGs was not further in use. There were in fact three case of studies of our experiments.

$$\sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|}) \tag{1}$$

- **Case 1.** The first one was focused on the use of PDS, instead of PRNG only for mutations, so in DE was used classical PRNG to select individuals from population as well as in special error-correction procedures (when individual leave searched space and has to be returned back).
- **Case 2.** In the second set, all PRNG numbers were fully replaced by PDS numbers (with different precision and thus period length).

- **Case 3.** The last part of our experiments was focused on the use of the same algorithms (SOMA and DE) in its canonical versions with classical PRNGs (standard pseudo-random generator in Wolfram *Mathematica* 9) just to compare performance with both Cases 1 and 2.

$$x_{n+1} = Ax_n\,(1 - x_n) \tag{2}$$

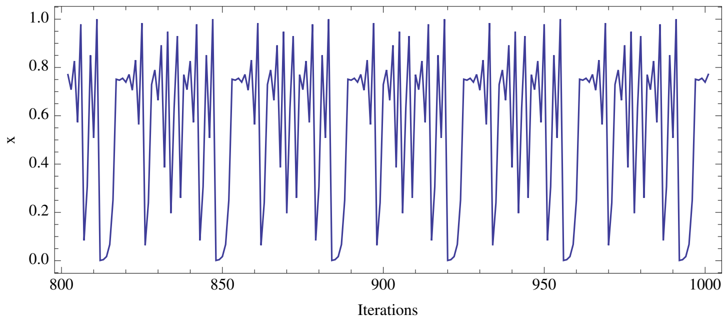

Fig. 1 Time series of period 36 (precision = 4) based on Eq. 2 for $A = 4$, see Tab. 2

Table 1 Algorithms setting. The same settings was used for both versions of DE.

DE		SOMA	
NP	20	PopSize	20
Dimensions	20	Dimensions	20
Generations	500	Migrations	20
F	0.9	PRT	0.1
CR	0.5	PathLength	5
		Step	0.11

4 Results

Results based on all three experiments are reported in Tab. 3 - 10 and Fig. 2 - 8. Tables and figures are organized sequentially according to case examples. Results of the **Case 1** are recorded in Fig. 2 - 4 further in Tab. 3, 5 and 7. Results of **Case 2** are recorded in Fig. 5 - 7 further in Tab. 4, 6 and 8. The last one, **Case 3**, is in Fig. 8 and Tab. 10.

For **Case 1** and **Case 2** has been used precision in DCHS according to Tab. 2 to get PDS. In this table is also recorded how many times was n periodic series repeatedly used in each algorithm. It is visible that in the frame of our experiments it was enough to set precision to 5 (Case 1, Fig. 2, Tab. 3, Tab. 5 and Tab. 7) and 8 (Case 2, Fig. 5, Tab. 4, Tab. 6 and Tab. 8).

Table 2 Periodicity dependance of Eq. 2 on various precision. Table also shows as to how many times was used n periodical series by the discussed algorithm.

Precision	Maximal Period	Repeated in SOMA	Repeated in DE
1	4	43181	32500
2	10	17272	13000
3	29	5956	4482
4	36	4797	3611
5	170	1016	764
6	481	359	270
7	758	227	171
8	4514	38	28
9	11227	15	11
10	35200	4	3
11	57639	2	2
12	489154	0	0
13	518694	0	0

All results from both cases can be compared with Fig. 8 and Tab. 10 from **Case 3** that was using only PRNG (standard pseudo-random generator in Wolfram *Mathematica* 9). When compared with Tab. 10 according to the Median value, then it is visible that in Case 1 it is comparable with precision equal to 4 (SOMA), 3 (DERand1Bin and DELocalToBest). For Case 2, it is enough to have precision to 5 (SOMA), 7 (DERand1Bin) and 8 (DELocalToBest). From Tab. 2 it is visible as to how many times was n periodic PDS used with given precisions. Results are also summarized in Tab. 9.

Table 3 Case 1. SOMA AllToOne.

Precision	Max	75%	Median	25%	Min
1	-585.311	-933.89	-1076.61	-1294.98	-1984.45
2	-771.596	-1225.85	-1392.08	-1572.34	-2324.94
3	-4144.75	-4175.6	-4179.39	-4182.09	-4186.95
4	-4051.62	-4182.91	-4186.16	-4187.5	-4189.39
5	-4070.28	-4186.44	-4187.86	-4188.75	-4189.68
6	-4172.77	-4185.4	-4187.13	-4188.04	-4189.42
7	-4068.95	-4187.24	-4188.75	-4189.03	-4189.65
8	-4058.85	-4184.46	-4187.2	-4188.05	-4189.29
9	-4069.58	-4185.51	-4187.4	-4187.98	-4189.48
10	-4177.38	-4186.37	-4187.5	-4188.46	-4189.55
11	-4174.2	-4185.52	-4187.49	-4188.56	-4189.35
12	-4066.18	-4185.32	-4187.4	-4188.61	-4189.71
13	-4144.51	-4186.16	-4187.58	-4188.7	-4189.51

Table 4 Case 2. SOMA AllToOne.

Precision	Max	75%	Median	25%	Min
1	823.851	687.668	687.668	-12.9572	-3042.29
2	-50.7378	-58.9429	-58.9429	-58.9429	-2146.33
3	220.071	220.071	-4057.14	-4132.14	-4169.95
4	-3761.36	-3987.13	-4044.72	-4068.93	-4182.16
5	-3914.18	-4136.2	-4185.91	-4189.17	-4189.62
6	-4157.77	-4187.88	-4188.76	-4189.12	-4189.60
7	-4069.75	-4187.	-4188.64	-4189.33	-4189.61
8	-4104.29	-4188.22	-4188.99	-4189.39	-4189.75
9	-4181.43	-4187.81	-4188.59	-4189.13	-4189.70
10	-4182.85	-4188.05	-4188.76	-4189.15	-4189.53
11	-4179.54	-4188.1	-4188.89	-4189.23	-4189.67
12	-4185.73	-4188.35	-4188.93	-4189.24	-4189.77
13	-4185.14	-4187.93	-4188.6	-4189.12	-4189.60

Table 5 Case 1. DERand1Bin.

Precision	Max	75%	Median	25%	Min
1	-1518.23	-1844.57	-2030.3	-2267.56	-2850.71
2	-2113.45	-2676.34	-2851.81	-3071.15	-3401.61
3	-4189.83	-4189.83	-4189.83	-4189.83	-4189.83
4	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83
5	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83
6	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83
7	-4071.39	-4189.82	-4189.83	-4189.83	-4189.83
8	-4189.81	-4189.83	-4189.83	-4189.83	-4189.83
9	-4189.75	-4189.83	-4189.83	-4189.83	-4189.83
10	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83
11	-4189.73	-4189.83	-4189.83	-4189.83	-4189.83
12	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83
13	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83

Table 6 Case 2. DERand1Bin.

Precision	Max	75%	Median	25%	Min
1	823.851	687.668	687.668	123.225	-12.9572
2	687.668	-58.9429	-58.9429	-58.9429	-58.9429
3	220.071	220.071	-1615.84	-2050.05	-2635.02
4	-1138.12	-1936.64	-2183.85	-2477.27	-3026.53
5	-1656.53	-2347.12	-2625.79	-3036.31	-4159.5
6	-2602.	-3712.17	-3975.73	-4189.82	-4189.83
7	-2417.19	-4189.32	-4189.81	-4189.83	-4189.83
8	-2591.58	-4189.82	-4189.83	-4189.83	-4189.83
9	-4189.65	-4189.83	-4189.83	-4189.83	-4189.83
10	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83
11	-4189.76	-4189.83	-4189.83	-4189.83	-4189.83
12	-4188.91	-4189.83	-4189.83	-4189.83	-4189.83
13	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83

Table 7 Case 1. DELocalToBest.

Precision	Max	75%	Median	25%	Min
1	-1339.95	-1872.95	-2060.63	-2292.03	-3371.71
2	-834.557	-2592.75	-2767.6	-2918.88	-3428.83
3	-971.065	-3429.78	-4189.83	-4189.83	-4189.83
4	-2004.78	-4172.27	-4188.8	-4189.68	-4189.83
5	-4052.35	-4189.76	-4189.82	-4189.83	-4189.83
6	-3886.55	-4186.71	-4189.71	-4189.82	-4189.83
7	-4071.28	-4189.77	-4189.82	-4189.83	-4189.83
8	-3980.9	-4189.76	-4189.83	-4189.83	-4189.83
9	-4071.39	-4189.82	-4189.83	-4189.83	-4189.83
10	-4071.39	-4189.82	-4189.83	-4189.83	-4189.83
11	-4071.38	-4189.81	-4189.83	-4189.83	-4189.83
12	-3895.89	-4189.8	-4189.83	-4189.83	-4189.83
13	-4015.18	-4189.8	-4189.83	-4189.83	-4189.83

Table 8 Case 2. DELocalToBest.

Precision	Max	75%	Median	25%	Min
1	823.851	687.668	687.668	123.225	-12.9572
2	-58.9429	-58.9429	-58.9429	-848.729	-2146.33
3	220.071	220.071	-2390.42	-2647.82	-3341.34
4	-1700.31	-2360.66	-2587.63	-2836.82	-3402.46
5	-1457.35	-2878.78	-3613.31	-4056.6	-4189.83
6	-2451.4	-3426.59	-3601.66	-3850.79	-4189.83
7	-2567.36	-3694.01	-4079.63	-4189.81	-4189.83
8	-3335.31	-4189.57	-4189.83	-4189.83	-4189.83
9	-4162.43	-4189.81	-4189.83	-4189.83	-4189.83
10	-3109.22	-4189.83	-4189.83	-4189.83	-4189.83
11	-4132.83	-4189.79	-4189.83	-4189.83	-4189.83
12	-4071.39	-4189.81	-4189.83	-4189.83	-4189.83
13	-3985.53	-4189.8	-4189.83	-4189.83	-4189.83

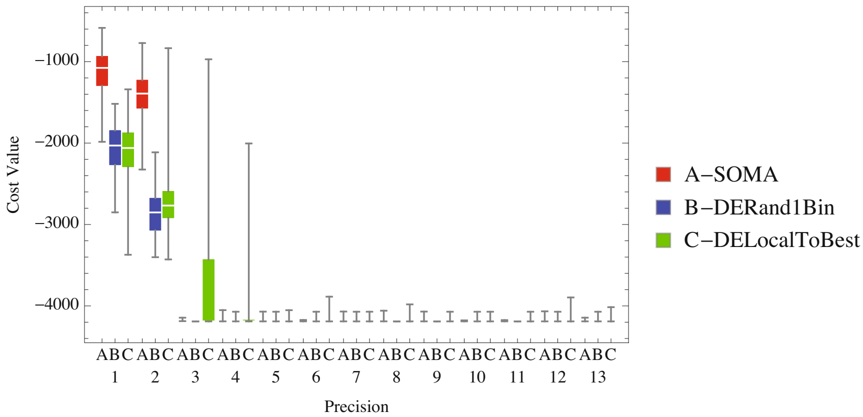

Fig. 2 Case 1. Dependance of algorithm performance on precision.

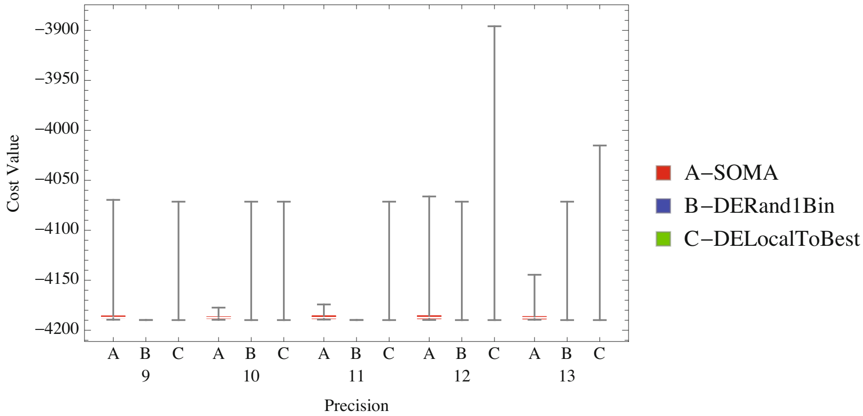

Fig. 3 Case 1. Dependance of algorithm performance on precision - detail.

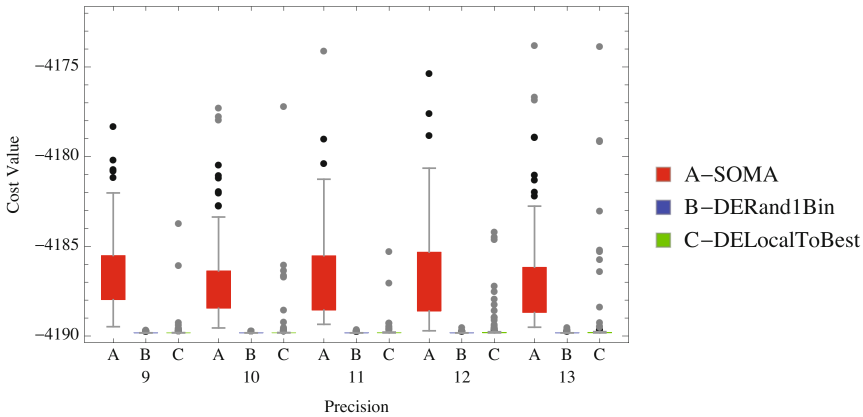

Fig. 4 Case 1. Dependance of algorithm performance on precision - detail with outliers.

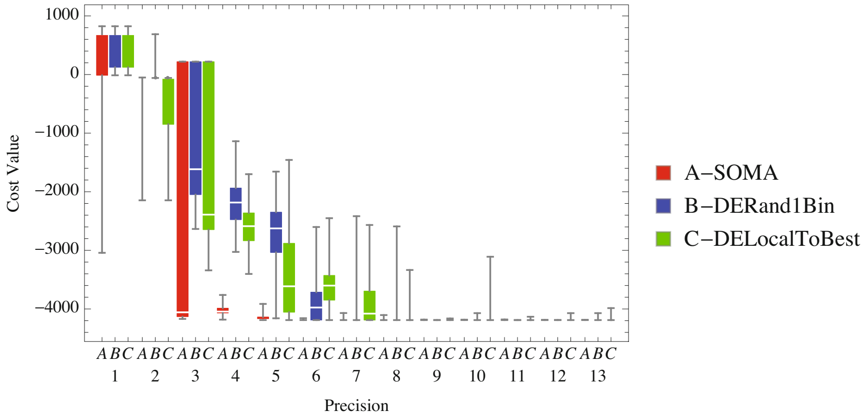

Fig. 5 Case 2. Dependance of algorithm performance on precision.

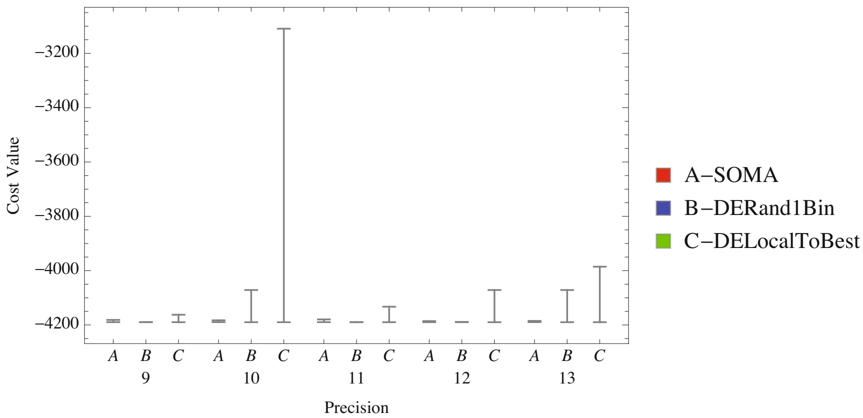

Fig. 6 Case 2. Dependance of algorithm performance on precision - detail.

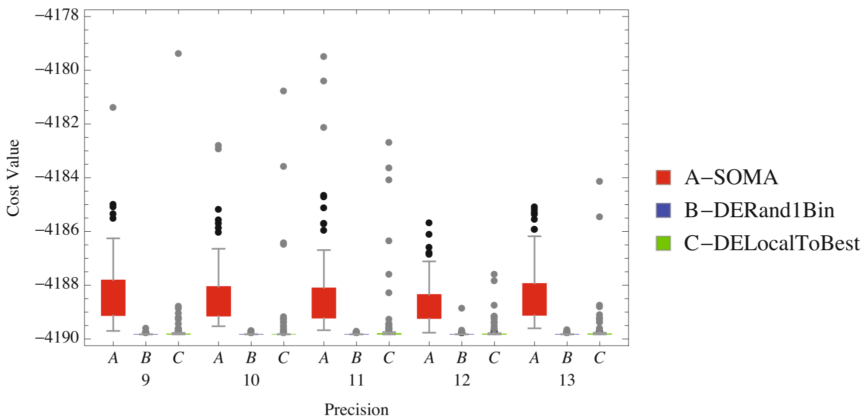

Fig. 7 Case 2. Dependance of algorithm performance on precision - detail with outliers.

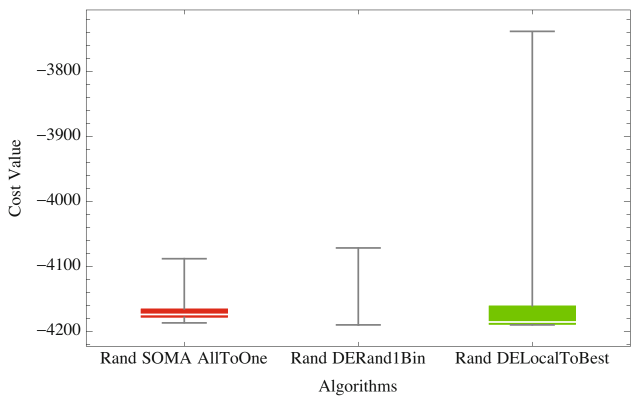

Fig. 8 Case 3. Results of EAs with PRNG, see Tab. 10.

Table 9 Summarization of all the best results from Tab. 10 from experiments based on EAs and PDS

Precision	SOMA		DERand1Bin		DELocalToBest	
	Case 1	Case 2	Case 1	Case 2	Case 1	Case 2
1	-1984.45	-3042.29	-2850.71	-12.9572	-3371.71	-12.9572
2	-2324.94	-2146.33	-3401.61	-58.9429	-3428.83	-2146.33
3	-4186.95	-4169.95	-4189.83	-2635.02	-4189.83	-3341.34
4	-4189.39	-4182.16	-4189.83	-3026.53	-4189.83	-3402.46
5	-4189.68	-4189.62	-4189.83	-4159.5	-4189.83	-4189.83
6	-4189.42	-4189.60	-4189.83	-4189.83	-4189.83	-4189.83
7	-4189.65	-4189.61	-4189.83	-4189.83	-4189.83	-4189.83
8	-4189.29	-4189.75	-4189.83	-4189.83	-4189.83	-4189.83
9	-4189.48	-4189.70	-4189.83	-4189.83	-4189.83	-4189.83
10	-4189.55	-4189.53	-4189.83	-4189.83	-4189.83	-4189.83
11	-4189.35	-4189.67	-4189.83	-4189.83	-4189.83	-4189.83
12	-4189.71	-4189.77	-4189.83	-4189.83	-4189.83	-4189.83
13	-4189.51	-4189.60	-4189.83	-4189.83	-4189.83	-4189.83

Table 10 Case 3. Results of EAs with PRNG.

Algorithm	Max	75%	Median	25%	Min
SOMA AllToOne	-4087.96	-4165.07	-4173.78	-4178.38	-4186.74
DERand1Bin	-4071.39	-4189.83	-4189.83	-4189.83	-4189.83
DELocalToBest	-3738.01	-4160.41	-4185.35	-4189.32	-4189.83

5 Conclusion

The main motivation of the research reported in this paper is whether it is possible to replace random number generators by deterministic processes originated in systems of deterministic chaos. In this participation were done three kind of experiments the first two used deterministic generators inside evolutionary algorithms (SOMA and differential algorithms) instead of pseudo-random number generators and the last one used standard pseudo-random generator in Wolfram *Mathematica* 9 in selected evolutionary algorithms to compare efficiency of proposed and tested methods.

For different numerical precessions were generated periodic series; see Tab. 2, that were used instead of random ones. Based on the obtained results it can be stated that at least in our case studies, all experiments exhibit fact that random number generators can be replaced by deterministic processes with small period (29 - 4514), with the repeated use in evolutionary algorithms with quite big frequency (5956 - 28 times). Results of the best reached minimum of each algorithm are also summarized in Tab. 9. An advantage of the proposed use of deterministic processes is the fact that in such case it is possible to fully repeat runs of given algorithm, analyze its behavior deterministically, including its full path on searched fitness landscape. We also believe that mathematical proofs can be in such case more easily constructed for such class of evolutionary algorithms.

Despite the widely presumed fact that pseudo-random number generators (also for evolutionary algorithms use) has to have as big period as possible (for example Mersenne twister with $2^{19937}-1$) and such as the 2^{32} common in many software packages, we show here that deterministic periodical processes with period 29 - 4514 is enough for our experiments.

Our further research is focused on more extensive and intensive testing of our ideas proposed here. Our aim is to try algorithms like scatter search [10], evolutionary strategies [11], genetic algorithms [12], [28] or particle swarm [13]. Also novel algorithms will be tested for its performance under our proposed approach in [26], [27] and alternative methods of symbolic regression [29].

Wider class of different algorithms, test functions and deterministic processes will be selected for future experiments to prove and specify the domain of validity of our ideas proposed here.

Acknowledgements. The following two grants are acknowledged for the financial support provided for this research: Grant Agency of the Czech Republic - GACR P103/13/08195S, 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, partially supported by Grant of SGS No. SP2013/114, VŠB - Technical University of Ostrava, Czech Republic, and by European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089.

References

1. Zelinka, I., Celikovsky, S., Richter, H., Chen, G.: Evolutionary Algorithms and Chaotic Systems, p. 550. Springer, Germany (2010)
2. Lozi, R.: Emergence Of Randomness From Chaos. International Journal of Bifurcation and Chaos 22(2), 1250021 (2012), doi:10.1142/S0218127412500216
3. Wang, X.-Y., Qin, X.: A new pseudo-random number generator based on CML and chaotic iteration. Nonlinear Dynamics An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems 70(2), 1589–1592 (2012) ISSN 0924-090X, doi:10.1007/s11071-012-0558-0
4. Pareek, N.K., Patidar, V., Sud, K.K.: A Random Bit Generator Using Chaotic Maps. International Journal of Network Security 10(1), 32–38 (2010)
5. Wang, X.-Y., Lei, Y.: Design Of Pseudo-Random Bit Generator Based On Chaotic Maps. International Journal of Modern Physics B 26(32), 1250208 (9 pages) (2012), doi:10.1142/S0217979212502086

6. Zhang, S.Y., Xingsheng, L.G.: A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems. In: International Conference on Life System Modeling and Simulation / International Conference on Intelligent Computing for Sustainable Energy and Environment (LSMS-ICSEE), Wuxi, Peoples R China, September 17-20 (2010)
7. Hong, W.-C., Dong, Y., Zhang, Wen, Y., Chen, L.-Y., B.K., P.: Cyclic electric load forecasting by seasonal SVR with chaotic genetic algorithm. International Journal of Electrical Power and Energy Systems 44(1), 604–614, doi:10.1016/j.ijepes.2012.08.010
8. Zelinka, I.: SOMA – Self Organizing Migrating Algorithm. In: Babu, B.V., Onwubolu, G. (eds.) New Optimization Techniques in Engineering, pp. 167–218. Springer, New York
9. Price, K.: An Introduction to Differential Evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 79–108. McGraw-Hill, London
10. Glover, F., Laguna, M.: Scatter Search. In: Ghosh, A., Tsutsui, S. (eds.) Advances in Evolutionary Computation: Theory and Applications, pp. 519–537. Springer, New York (2003)
11. Beyer, H.-G.: Theory of Evolution Strategies. Springer, New York (2001)
12. Holland, J.H.: Genetic Algorithms. Scientific American, 44–50 (July 1992)
13. Clerc, M.: Particle Swarm Optimization. ISTE Publishing Company (2006) ISBN 1905209045
14. Pluhacek, M., Senkerik, R., Zelinka, I.: Impact of various chaotic maps on the performance of chaos enhanced PSO algorithm with inertia weight – an initial study. In: Zelinka, I., Snasel, V., Rössler, O.E., Abraham, A., Corchado, E.S. (eds.) Nostradamus: Mod. Meth. of Prediction, Modeling. AISC, vol. 192, pp. 153–166. Springer, Heidelberg (2013)
15. Pluhacek, M., Senkerik, R., Davendra, D., Zelinka, I.: 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)
16. Yang, M., Guan, J., Cai, Z., Wang, L.: Self-adapting differential evolution algorithm with chaos random for global numerical optimization. In: Cai, Z., Hu, C., Kang, Z., Liu, Y. (eds.) ISICA 2010. LNCS, vol. 6382, pp. 112–122. Springer, Heidelberg (2010)
17. Coelho, L., Mariani, V.: Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. IEEE Transactions On Power Systems 21(2), 989–996 (2006), doi:10.1109/TPWRS.2006.873410
18. Hu, G.-W.: Chaos-differential evolution for multiple sequence alignment. In: 3rd International Symposium on Intelligent Information Technology Application, Nanchang, Peoples R China, vol. 2, pp. 556–558., doi:10.1109/IITA.2009.511
19. Zhao, Q., Ren, J., Zhang, Z., Duan, F.: Immune co-evolution algorithm based on chaotic optimization. In: Workshop on Intelligent Information Technology Application (IITA 2007), Zhang Jiajie, Peoples R China, pp. 149–152 (2007)
20. Peng, C., Sun, H., Guo, J., Liu, G.: Dynamic economic dispatch for wind-thermal power system using a novel bi-population chaotic differential evolution algorithm. International Journal of Electrical Power & Energy Systems 42(1), 119–126 (2012), doi:10.1016/j.ijepes.2012.03.012

21. Zhang, H., Zhou, J., Zhang, Y., Fang, N., Zhang, R.: Short term hydrothermal scheduling using multi-objective differential evolution with three chaotic sequences. International Journal of Electrical Power & Energy Systems 47, 85–99 (2013), doi:10.1016/j.ijepes.2012.10.014
22. Zou, X., Wang, M., Zhou, A., McKay, B.: Evolutionary optimization based on chaotic sequence in dynamic environments. In: 2004 IEEE International Conference on Networking, Sensing and Contro, vol. 2, pp. 1364–1369 (2004)
23. Liua, B., Wanga, L., Jina, Y.-H., Tangb, F., Huanga, D.-X.: Improved particle swarm optimization combined with chaos. Chaos, Solitons & Fractals 25(5), 1261–1271 (2005)
24. Song, Y.: A bi-directional chaos optimization algorithm. In: 2010 Sixth International Conference on Natural Computation (ICNC), August 10-12, vol. 5, pp. 2202–2206 (2010)
25. Gandomi, A., Yun, G., Yang, X., Talatahari, S.: Chaos-enhanced accelerated particle swarm optimization. Communications In Nonlinear Science and Numerical Simulation 18(2), 327–340 (2013), doi:10.1016/j.cnsns.2012.07.017
26. Matousek, R.: HC12: The Principle of CUDA Implementation. In: Matousek (ed.) 16th International Conference on Soft Computing, MENDEL 2010, Brno, pp. 303–308 (2010)
27. Matousek, R., Zampachova, E.: Promising GAHC and HC12 algorithms in global optimization tasks. Journal Optimization Methods & Software 26(3), 405–419 (2011)
28. Matousek, R.: GAHC: Improved Genetic Algorithm. In: Krasnogor, N., Nicosia, G., Pavone, M., Pelta, D. (eds.) Nature Inspired Cooperative Strategies for Optimization (NICSO 2007). SCI, vol. 129, pp. 507–520. Springer, Heidelberg (2008)
29. Zelinka, I., Davendra, D., Senkerik, R., Jasek, R., Oplatkova, Z.: Analytical Programming - a Novel Approach for Evolutionary Synthesis of Symbolic Structures. In: Kita, E. (ed.) Evolutionary Algorithms. InTech (2011) ISBN: 978-953-307-171-8, doi:10.5772/16166, http://www.intechopen.com/books/evolutionary-algorithms/analytical-programming-a-novel-approach-for-evolutionary-synthesis-of-symbolic-structures