Chaos Powered Selected Evolutionary Algorithms

Lenka Skanderova, Ivan Zelinka, and Petr Šaloun

Abstract. It is well known that the evolution algorithms use pseudo-random numbers generators for example to generate random individuals in the space of possible solutions, crossing etc. In this paper we are dealing with the effect of different pseudo-random numbers generators on the course of evolution and the speed of their convergence to the global minimum. From evolution algorithms the differential evolution and self organizing migrating algorithm have been chosen because they have different strategies. As the random generators Mersenne Twister and chaotic system - logistic map have been used.

1 Introduction

Evolution algorithms are based on the Darwin's and Mendel's principles. There is the population of individuals, which is improving in the time. The individuals are crossing and mutated and the best survive, while worse die. The population is developing during the generation cycles, we call them *Generations* or *Migrations* according to the used algorithm [1]. From the view of pseudo-random numbers generators, we need to generate individuals in the space of possible solutions. The individual consists of parameters (each parameter has its low and high bound) and these parameters are usually real numbers. In the first generation cycle the individuals are chosen randomly - their parameters are chosen randomly in their low and high borders. Next the pseudo-random numbers generators plays the essential role in the process of crossing.

In this paper we connect together three basic research areas – evolution algorithms, chaos and pseudo-random numbers generators. With connection

Lenka Skanderova · Ivan Zelinka · Petr Šaloun
Department of Computer Science, VSB-Technical university Ostrava,
17. listopadu 15, 708 33 Ostrava-Poruba, Czech Republic
e-mail: {lenka.skanderova,ivan.zelinka,petr.saloun}@vsb.cz

I. Zelinka et al. (Eds.): *Nostradamus 2013: Prediction, Model. & Analysis*, AISC 210, pp. 111–124.
DOI: 10.1007/978-3-319-00542-3_12

of evolutionary algorithms (especially differential evolution and SOMA) and chaos [2] - [4] have been written. Artificial bee colony algorithm is connected with chaos in [5]. New adaptive differential evolution technique based on logistic map for optimal distribution placement and sizing is presented in [7] and in [6]the differential evolution with chaos theory for self adaptation of differential evolution's parameters is combined. In [8] chaotic Logistic equation is mentioned in connection with SOMA and differential evolution. In recent years, some new pseudo-random number generators were described, e.g. [11] and [12]. The pseudo-random numbers generators based on chaos are presented in [9], [10]. In [13] Sandpile model - the complex system operating at a critical state between chaos and order - is proposed and author state that cellular automata can be used as a generator of pseudo-random numbers.

This paper is divided into the sections, where in the first one we describe the used evolution algorithms - DE and SOMA, next we are dealing with used pseudo-random numbers generators, motivation and design experiments. In section Results, the experiment's results are shown and in Conclusions we sumarize achieved results.

2 Evolution Algorithms

2.1 Differential Evolution (DE)

Differential evolution works at the principle of improving population during the generation's cycles, as it is mentioned above. The population consists of individuals, each individual has its own parameters and fitness value – the value of the cost function. The fitness says how good this individual is in the population. In generation cycle for each individual three different individuals from the population are chosen randomly and the noise vector is generated, see Eq.(1.

$$v_i^{G+1} = x_{r1}^G + F(x_{r2}^G - x_{r3}^G), r_1 \neq r_2 \neq r_3 \tag{1}$$

where v_i^{G+1} is the i-th noise vector in the next generation, $x_{\mathbf{r1}}$ is the first randomly chosen individual, $x_{\mathbf{r2}}^G$ is the second randomly chosen individual and $x_{\mathbf{r3}}^G$ is the third randomly chosen individual. F is the mutation constant [14].

When the noise vector is made, the trial individual creation can start. To the trial individual the parameters from the noise vector or from the actual individual are chosen according to Eq.(2).

$$u_{i,G+1}^j = \begin{cases} v_{i,G+1}^j & if\ r(j) \leq CR \\ x_{i,G}^j & otherwise \end{cases} \tag{2}$$

where $u_{i,G+1}$ is the j-th parameter of the trial vector, $r(j)$ is the random number from the interval $[0, 1]$ [14] and CR is the crossover probability.

The selection of a new individual is described by Eq.(3).

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & if\ f(u_{i,G+1}) \prec f(x_{i,G}) \\ x_{i,G} & otherwise \end{cases} \tag{3}$$

where $f(u_{i,G+1})$ is the fitness of the trial vector and $f(x_{i,G})$ denote fitness of the actual individual. The sign $\prec$ is used because we search global minimum [14].

2.2 Self Organized Migrating Algorithm (SOMA)

The strategy of SOMA differs from DE. In DE new offspring is creating during the evolution. In SOMA there is no offspring. The individuals migrate in the space of possible solutions, they just change their positions. In this paper version *AllToOne* of SOMA is used, where all individuals migrate to the one individual, we call it *Leader*. Except this strategy, *AllToAll*, *AllToOne Random*, *AllToAll Adaptive* and *AllToOne Adaptive* exist.

The begin of the algorithm is the same like in DE – the random individuals are generated in population. Following principle is different: Each individual migrates to the *Leader* in steps (the length of all step is united), while the sum of steps do not reach or even cross the parameter denoted as $PathLength$, see Eq.(4). In SOMA the step is denoted by the parameter *Step* and its value should be odd, usually it is the value 0.11. For each step, where individual reaches a new position, new fitness is computed. In the end of migration cycle the best reached position is chosen.

Mutation is replaced by perturbation in SOMA, see Eq.4.

$$\mathbf{r} = \mathbf{r_0} + \mathbf{m}t\mathbf{PRTvector} \tag{4}$$

where $\mathbf{r}$ is a new candidate solution, $\mathbf{r_0}$ denotes actual individual, m is a difference between *Leader* and start position of individual and t is the parameter *Step*, $t \in [0, PathLength]$ [15].

The direction of the individual's migration is given by the parameter PRT, ussually with the value 0.1. According to the PRT the $PRTvector$ is generated by this way: for each parameter of $PRTvector$ a random number from interval [0,1] is generated. If this number is smaller than PRT, the parameter of $PRTvector$ will have value 1 else it will have value 0, see Eq.(5).

$$if\ rnd_j \prec PRT\ then\ PRTvector_j = 1\ else\ 0 \tag{5}$$

where rnd_j means random number from the interval [0,1], $PRTvector_j$ denoted the j-th parameter of the perturbation vector. If the fitness of the best found position od the individual is better than its actual fitness, the individual will migrate to the best position. Otherwise individual stays at its old position.

3 Pseudo-random Numbers Generator

3.1 Mersenne Twister (MT)

Mersenne Twister has been proposed by M. Matsumoto and T. Nishimura in the year 1997. Its period is $2^{19937} - 1$ and it has 632-dimensional equidistribution property. It is the variant of previously proposed generators, TGFSR. For more information see [16].

MT has been used for example in the Monte Carlo Localization Algorithm in [17]. It has been also used as a comparative tool in the developement of new pseudo-random numbers generators, see [19]. In [18] authors state that MT cannot be used efficiently without substantial changes as a random number generator for massively parallel simulations on GPU. In connection with evolution algorithm MT has been used in [20], where autors are specialized in genetic algorithms and simulated annealing, [21] in parallel evolutionary algorithm for RNA holding and [22] in Particle Swarm Optimization.

3.2 Logistic Map

Each definition of chaos describes some kind of unpredictability in the evolution of the system. At this idea theory of Li and Yorke is based, this theory says that in the logistic map in interval [0,1] it is possible to find two close trajectories, which are moving away from each other with increasing time. In other words small change in starting conditions may cause very different results. The system can behave equally if and only if the starting conditions are absolutely same [33]. Good example is the butterfly effect.

In connection with chaos control [24] and [27] have been written. In 2013 [25] and [26] have been written about chaos and evolution algorithms connection.

Logistic map is a one – dimensional quadratic map defined by Eq.6.

$$x_{n+1} = ax_n(1 - x_n) \tag{6}$$

where a is an external parameter and x_n value moves in interval [0,1] [28].

Kuznetsov N.V. and Leonov G.A. in [34] say about Lyapunov exponent: "*In 1930 O. Perron found the effects of Lyapunov exponent sign inversion. It has been shown that the negativeness of the largest Lyapunov exponent of the first approximation system does not always result in the stability of zero solution of the original system. Small neighborhood of zero solution, the solutions of the original system with positive Lyapunov exponent (Lyapunov characteristic exponent) can exist. A. M. Lyapunov has introduced the notion of regular linear system and showed that for regular linearizations the stability is defined by the negativeness of Lyapunov exponents of linearized system - that was the first sufficient condition of asymptotic stability by the first approximation for nonstationary linearizations.*"

We can say that Lyapunov exponent is the basic tool for dynamic system description. If the Lyapunov exponent is negative, the dynamic system is not sensitive to beginning conditions. On the other hand if the Lyapunov exponent is positive, the system will be sensitive to basic condition. The chaotic system must have at least one positive exponent [36].

The behavior of the logistic map is described in [28] where authors investigate behavior of logistic map for $x \geq x_\infty$, they say that Lyapunov exponent λ (it characterizes the rate of separation of infinitesimally close trajectories) of the logistic map at x_∞ is zero. Lyapunov exponent λ becomes mostly possitive for $x > x_\infty$ and therefore authors say that chaos starts at the end of the bifurcation region, see Fig.1. Next authors state that: "*the detailed behavior of the iterates of the logistic map appears rather complicated in this region, it shows regularities which are again dictated by doubling operator and therefor universal. For $x_\infty \prec r$, periodic and chaotic regions are densely interwoven, and one finds a sensitive dependence on the parameter value*".

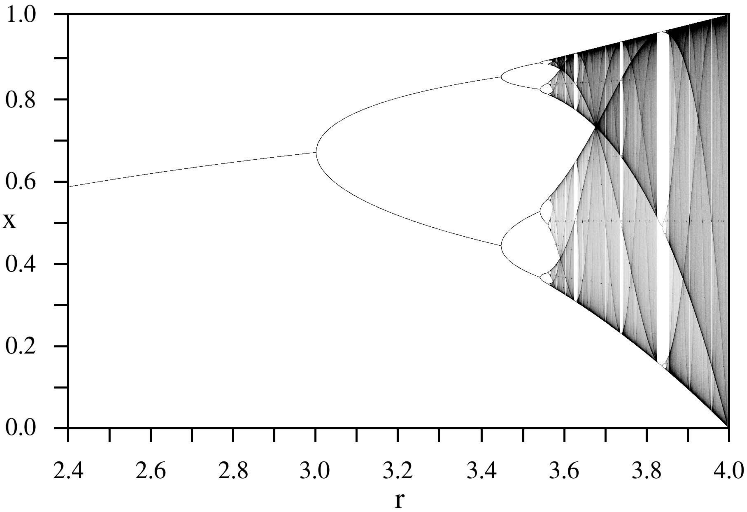

Fig. 1 Logistic map

In [29] the logistic map has been used for example to generate cycle time in series of signals, in cryptography - Baptista's cryptosystem, [30], and image encryption [31]. From the view of connection between evolution algorithms logistic map is mentioned for example in [32].

4 Motivation

The main motivation was to compare pseudo-random numbers generators from the view of influence to developing of the population in evolutionary algorithms. Mersenne Twister has been chosen because it has a big period

$(2^{19937} - 1)$ and as it was mentioned above it is very often connected with evolution algorithms as a confirmed random number generator.

5 Experiment Design

For experiments, SOMA and DE have been chosen because these algorithms have different strategies. As it was mentioned above, DE's population is still developing during the generation cycles, while in SOMA the individuals are migrating in the space of possible solutions and no offspring is generated. Each experiment in the experiment group has been repeated one hundred times and exact settings of both algorithms are mentioned in Tabs. 1 and 2 where NP means the number of individuals in one population, D dimension (how many parameters will be contained in one individual), $Migrations$ and $Generations$ denote the number of evolution cycles.

For experiments HP Pavilion dv7-6050 with processor Intel Core i7 with frequency 2 GHz, 4 GB RAM and graphic card AMD Radeon HD 6770M and Microsoft Visual Studio 2010 have been used. The experiments have beed processed by Mathematica 8.

Table 1 DE setting

NP	1000
D	20
$Generations$	500
F	0.8
CR	0.5

Table 2 SOMA setting

NP	1000
D	20
$Migrations$	500
PRT	0.11
$PathLength$	3
$Step$	0.11

As the trial functions 1^{st} de Jong's function, Schwefel's function and Ranna's function have been chosen. Schwefel's function has many local minimums, it is very jagged and we know the global minimum - $f(x) = -418.9829D$. 1^{st} de Jong's function has just one minimum - global minimum - $f(x) = 0$ and this function is not jagged. And in Ranna's function there the global minimum has not been found. And it is not mentioned in any literature.

Table 3 Setting of logistic map

Experiment's group	Value of parameter a
1^{st} group	3.58
2^{nd} group	[3.8280, 3.8285]
3^{rd} group	3.855
4^{th} group	[3.8567, 3.8570]
5^{th} group	4

In logistic map of chaos as the beginning value of x_n 0.02 has been set. This value has been chosen randomly. Each algorithm's experiments have been divided into 5 groups according setting of parameter a, see Tab. 3. 2^{nd} and 4^{th} groups are special, because the parameter a has been changing for each x_n by the step 0.0001. Otherwise the parameter a has been constant.

6 Results

The Figs. 2 and 3 show comparing of results of MT and Chaos random number generators, where in chaos $a = 4$ and Ranna's function has been used as a testing function. It is known that if $a = 4$ logistic equation will generate numbers from all interval [0,1], see 1. This fact is important for evolution developing. As it is obvious, evolutions where chaos pseudo-random number generator has been used, convergate faster than evolutions, where MT has been used. On the other hand when $a = 3.58$ and 1^{st} de Jong's and Schwefel's functions have been testing function, some experiments from the collection, where chaos has been used convergate much slower than MT and the total results have been much worse than MT.

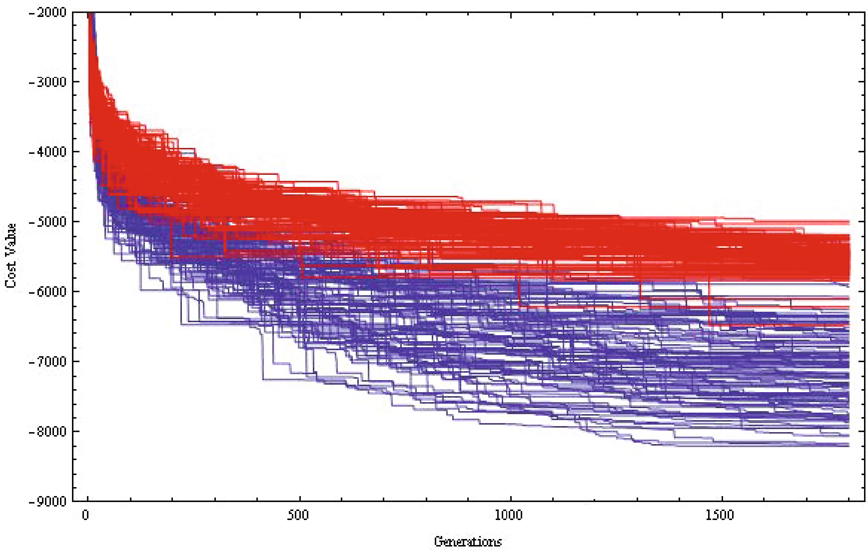

Fig. 2 Comparing MT and Chaos. Differential evolution, Ranna's function, $a = 4$. Blue represents Chaos, red represents MT.

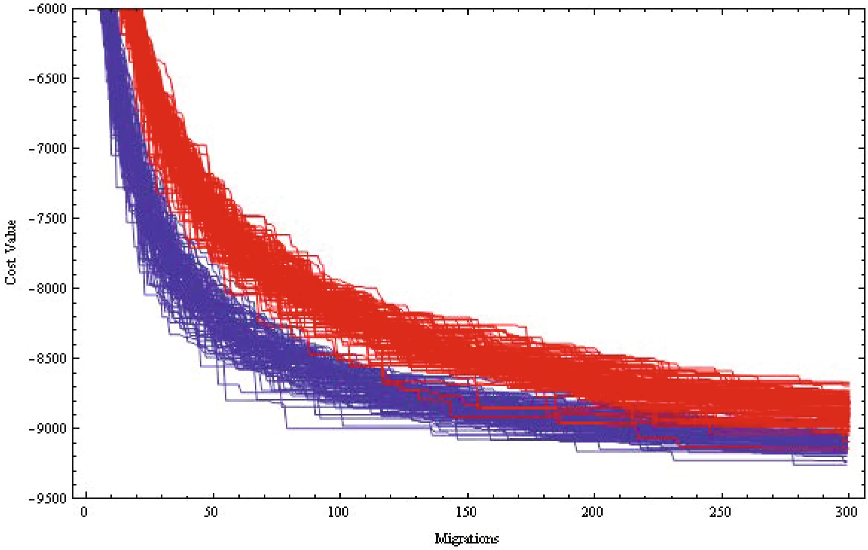

Fig. 3 Comparing MT and Chaos. SOMA, Ranna's function, $a = 4$. Blue represents Chaos, red represents MT.

Table 4 Minimum, maximu and average fitness value of DE experiments for 1^{st} de Jong's, Ranna's and Schwefel's function with settings mentioned in Tab. 1.

	Function	Min. fitness value	Max. fitness value	Average fitness value
	1^{st} de Jong's	0.000	0.000	0.000
Chaos $a = 3.58$	Ranna's	-7669.411	-5426.377	-6656.126
	Schwefel's	-8379.658	-7539.422	-8160.376
	1^{st} de Jong's	11.674	15220.172	4206.105
Chaos $a = 3.828$	Ranna's	-8081.707	-5125.884	-6894.176
	Schwefel's	-7763.678	-6167.574	-7107.649
	1^{st} de Jong's	0.003	0.074	0.023
Chaos $a = 3.855$	Ranna's	-8585.980	-6383.648	-7730.503
	Schwefel's	-8379.657	-7712.970	-8231.763
	1^{st} de Jong's	0.004	0.121	0.026
Chaos $a = 3.8567$	Ranna's	-8494.416	-6136.087	-7687.413
	Schwefel's	-8379.657	-7692.009	-8221.569
	1^{st} de Jong's	0.008	0.144	0.039
Chaos $a = 4$	Ranna's	-8732.198	-6175.018	-7860.793
	Schwefel's	-8379.651	-7882.383	-8234.940
	1^{st} de Jong's	0.000	0.003	0.000
MT	Ranna's	-8013.129	-6213.274	-7052.287
	Schwefel's	-8379.656	-7144.871	-7976.479

Table 5 Minimum, maximu and average fitness value of SOMA experiments for 1^{st} de Jong's, Ranna's and Schwefel's function with settings mentioned in Tab. 2.

	Function	Min. fitness value	Max. fitness value	Average fitness value
	1^{st} de Jong's	0.000	0.000	0.000
Chaos $a = 3.58$	Ranna's	-8588.349	-6946.222	-7716.781
	Schwefel's	-8379.658	-6543.776	-7277.547
	1^{st} de Jong's	0.000	0.000	0.000
Chaos $a = 3.828$	Ranna's	-9180.587	-8415.158	-8857.963
	Schwefel's	-8379.658	-8379.658	-8379.658
	1^{st} de Jong's	0.000	0.000	0.000
Chaos $a = 3.855$	Ranna's	-9110.511	-8435.843	-8830.062
	Schwefel's	-8379.658	-8379.658	-8379.658
	1^{st} de Jong's	0.000	0.000	0.000
Chaos $a = 3.8567$	Ranna's	-9264.553	-8535.640	-8887.881
	Schwefel's	-8379.658	-8379.657	-8379.658
	1^{st} de Jong's	0.000	0.000	0.000
Chaos $a = 4$	Ranna's	-9258.219	-8947.286	-9090.472
	Schwefel's	-8379.658	-8379.658	-8379.658
	1^{st} de Jong's	0.000	0.000	0.000
MT	Ranna's	-9139.584	-8667.654	-8866.831
	Schwefel's	-8379.658	8379.658	8379.658

7 Conclusion

From the experiments we can make some conclusions:

- It depends on the parameter a in logistic equation. When $a = 4$ the evolution convergence has been faster with using chaos pseudo-random numbers generator than MT. When a has been from interval [3.828, 3.8285], $a = 3.855$ and a has been from interval [3.8567, 3.857] chaos random number generator and MT results have been comparable except 1^{st} de Jong's function, which converged faster with MT using. If $a = 3.58$ the results have been much worse in chaos than in MT. It is logical, because in this area in the bifurcation diagram we can see deterministic windows (white areas), see 1. That means many numbers from interval [0,1] have not been contained in the computation. This area is weighted by a big periodicity, that influence the results negatively. Much better results with using chaos have been reached when $a = 4$, see Figs. 2 and 3. It is the area of chaos in bifurcation diagram and random numbers are generated from all interval [0,1].
- In chaos pseudo-random numbers generator some experiments differ from total average by their convergence trajectories, while in MT there was no experiment, which would be much different than others from the same group.
- In Tabs. 4 and 5 minimum, maximum and average cost function values of the best individuals in evolution are shown for DE and SOMA, where MT and Chaos - logistic map as the pseudo-random numbers generators

have been used. As it was mentioned above 1^{st} de Jong's, Ranna's and Schwefel's function have ben used as test functions. Minimums in Tabs. 4 and 5 are the best values of the cost function, maximums are the worst values of the cost function, which have been reached during the evolution process.
- There is a big difference between SOMA and DE in 1^{st} de Jong's function's searched minimum, SOMA has got a smaller value than DE, while in the other functions the minimum values are comparable, and it does not matter if MT or Chaos - logistic map has been used.
- In Figs. 2 and 3 it is shown that if $a = 4$ in logistic equation and the Ranna's function has been choosen, evolution converged faster than when MT has been used as a pseudo-random numbers generator.

In the future the experiments will be extended by next kinds of SOMA - *AllToAll*, *AllToOne Random*, *AllToAll Adaptive* and *AllToOne Adaptive* as well as some other kinds of DE. As testing functions next testing functions (e.g. Rastrigin's, Ackley's, Michalewicz's) will be tried. Further research will be focused on more extensive and intensive testing of our ideas proposed here. Our aim is to try algorithms like scatter search [46], evolutionary strategies [47], genetic algorithms [48], [52] or particle swarm [49]. Also novel algorithms will be tested for its performance under our proposed approach in [50], [51] and alternative methods of symbolic regression [53].

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 and by grant SV 4603351.

References

1. Zelinka, I., et al.: Evolutionary Algorithms and Chaotic Systems. SCI. Springer (2010) ISBN-10: 3642107060, ISBN-13: 978-364210706
2. Zelinka, I.: On evolutionary synthesis of chaotic systems. In: Zelinka, I., Snasel, V., Rössler, O.E., Abraham, A., Corchado, E.S. (eds.) Nostradamus: Mod. Meth. of Prediction, Modeling. AISC, vol. 192, pp. 29–34. Springer, Heidelberg (2013)
3. Brandejsky, T., Zelinka, I.: Specific behaviour of GPA-ES evolutionary system observed in deterministic chaos regression. In: Zelinka, I., Snasel, V., Rössler, O.E., Abraham, A., Corchado, E.S. (eds.) Nostradamus: Mod. Meth. of Prediction, Modeling. AISC, vol. 192, pp. 73–81. Springer, Heidelberg (2013)

4. 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)
5. Tien, J.P., Li, T.H.S.: Hybrid Taguchi-chaos of multilevel immune and the artificial bee colony algorithm for parameter identification of chaotic systems. Computer & Mathematics with Applications 64, 1108–1119 (2012)
6. Manal, K.K., et al.: Emission Constrained Economic Dispatch Using Logistic Map Adaptive Differential Evolution. In: Proceedings of the International Conference on Information Systems Design and Intelligent Applications 2012, vol. 132, pp. 387–394 (2012)
7. Mandal, K.K., Bhattacharya, B., Tudu, B., Chakraborty, N.: Logistic Map Adaptive Differential Evolution for Optimal Capacitor Placement and Sizing. In: Panigrahi, B.K., Suganthan, P.N., Das, S., Satapathy, S.C. (eds.) SEMCCO 2011, Part I. LNCS, vol. 7076, pp. 68–76. Springer, Heidelberg (2011)
8. Senkerik, R., et al.: Utilization of SOMA and differential evolution for robust stabilization of chaotic Logistic equation. 3rd Global Conference on Power Control Optimization. Computers & Mathematics with Applications 60, 1026–1037 (2010)
9. Hu, H.P., et al.: Pseudorandom sequence generator based on the Chen chaotic system. Computer Physicics Communications 184, 765–768 (2013)
10. Wang, X.Y., Qin, X.: A new pseudo-random number generator based on CML and chaotic iteration. Nonlinear Dynamics 70, 1589–1592 (2012)
11. Song, H.L.: New Pseudorandom Number Generator Artin-Schreier Tower for p=5. China Communications 9, 60–67 (2012)
12. Marquardt, P., et al.: Pseudorandom number generators based on random covers for finite groups. Designs Codes and Cryptography 64, 209–220 (2012)
13. Karimi, H., et al.: On the combination of self-organized systems to generate pseudo-random numbers. Information Science 221, 371–388 (2013)
14. Zhou, Y., Li, X., Gao, L.: A differential evolution algorithm with intersect mutation operator. Applied Soft Computing 13, 390–401 (2013)
15. Nolle, L., Zelinka, I., Hopgood, A.A., Goodyear, A.: Comparison of an self-organizing migration algorithm with simulated annealing and differential evolution for automated waveform tuning
16. Matsumoto, M., Nishimura, T.: Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator. ACM Transactions on Modeling and Computer Simulation 8, 3–30 (1998)
17. Bonato, V., et al.: A Mersenne Twister Hardware Implementation for the Monte Carlo Localization Algorithm. Journal of Signal Processing Systemsfor Signal, Image, and Video Technology (formerly the Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology) (2012)
18. Manssen, M., et al.: Random number generators for massively parallel simulations on GPU. The European Physical Journal Special Topics, EDP Sciences, 53–71 (2012)
19. Leiserson, et al.: Deterministic Parallel Random-Number Generation for Dynamic-Multithreading Platforms. In: 17th ACM SIGPLAN symposium on Principles and Practice of Parallel Programming, pp. 193–204. ACM, New York (2012)

20. Maucher, M., Schning, U., Kestler, H.A.: Search heuristics and the influence of non-perfect randomness: examining Genetic Algorithms and Simulated Annealing. Springer (2011)
21. Wiese, K.C., et al.: P-RnaPredict - A parallel evolutionary algorithm for RNA folding: Effects of pseudorandom number quality. IEEE Transactions on Nanobioscience 4, 219–227 (2005)
22. Igarashi, J., Sonoh, S., Koga, T.: Particle Swarm Optimization with SIMD-Oriented Fast Mersenne Twister on the Cell Broadband Engine. In: Köppen, M., Kasabov, N., Coghill, G. (eds.) ICONIP 2008, Part II. LNCS, vol. 5507, pp. 1065–1071. Springer, Heidelberg (2009)
23. http://msdn.microsoft.com/en-us/library/system.random.aspx
24. Hegazi, A.S., et al.: On chaos control and synchronization of the commensurate fractional order Liu system. Communications in Nonlinear Science and Numerical Simulation 18, 1193–1202 (2013)
25. Senkerik, R.: On the Evolutionary Optimization of Chaos Control - A Brief Survey. Nostradamus: Modern Methods of Prediction, Modeling and Analysis of Nonlinear Systems 192, 35–48 (2013)
26. Senkerik, R., Davendra, D., Zelinka, I., Oplatkova, Z., Pluhacek, M.: Optimization of the batch reactor by means of chaos driven differential evolution. In: Snasel, V., Abraham, A., Corchado, E.S. (eds.) SOCO Models in Industrial & Environmental Appl. AISC, vol. 188, pp. 93–102. Springer, Heidelberg (2013)
27. Chen, D.Y., et al.: Control and synchronization of chaos in an induction motor system. International Journal of Innovative Computing Information and Control 8, 7237–7248 (2012)
28. Schuster, H.G., Just, W.: Deterministic Chaos An Introduction. Wiley-VCH Verlag GmbH & Co., KGaA, Weinheim (2005)
29. Nagatani, T., Sugiyama, N.: Vehicular traffic flow through a series of signals with cycle time generated by a logistic map. Physica A: Statistical Mechanics and its Applications 392, 851–856 (2013)
30. Hussain, I., et al.: An efficient approach for the construction of LFT S-boxes using chaotic logistic map. Nonlinear Dynamics 71, 133–140 (2013)
31. Akhshani, A., et al.: An image encryption scheme based on quantum logistic map. Communications in Nonlinear Science and Numerical Simulation 17, 4653–4661 (2012)
32. He, Y.Y., et al.: A fuzzy clustering iterative model using chaotic differential evolution algorithm for evaluating flood disaster. Expert Systems with Applications 38, 10060–10065 (2011)
33. Wu, X., Zhu, P.: Chaos in a class of non-autonomous discrete systems. Applied Mathematics Letters 26, 431–436 (2013)
34. Kuznetsov, N.V., Leonov, G.A.: On stability by the first approximation for discrete systems. In: Proceedings of International Conference on Physics and Control, PhysCon 2005, vol. 2005, pp. 596–599 (2005)
35. Leonov, G.A., Kuznetsov, N.V.: Time-Varying Linearization and the Perron effects. International Journal of Bifurcation and Chaos 17, 1079–1107 (2007)
36. Kaclek, J., Mca, I.: Nelinern analza a predikce stovho provozu Elektrorevue (2009) ISSN 1213 – 1539

37. Pluhacek, M., et al.: On the Behaviour and Performance of Chaos Driven PSO Algorithm with Inertia Weight. Computers & Mathematics with Applications (2012) (accepted for publication) ISSN 0898-1221
38. Pluhacek, M., Budikova, V., Senkerik, R., Oplatkova, Z., Zelinka, I.: Extended initial study on the performance of enhanced PSO algorithm with lozi chaotic map. In: Zelinka, I., Snasel, V., Rössler, O.E., Abraham, A., Corchado, E.S. (eds.) Nostradamus: Mod. Meth. of Prediction, Modeling. AISC, vol. 192, pp. 167–177. Springer, Heidelberg (2013)
39. 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)
40. 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)
41. Pluhacek, M., et al.: PID Controller Design For 4th Order system By Means Of Enhanced PSO algorithm With Lozi Chaotic Map. In: Proceedings of 18th International Conference on Soft Computing, MENDEL 2012, pp. 35–39 (2012) ISBN 978-80-214-4540-6
42. Pluhacek, M., et al.: On The Performance Of Enhanced PSO algorithm With Lozi Chaotic Map –An Initial Study. In: Proceedings of 18th International Conference on Soft Computing, MENDEL 2012, pp. 40–45 (2012) ISBN 978-80-214-4540-6
43. Pluhacek, M., et al.: Designing PID Controller For DC Motor System By Means Of Enhanced PSO Algorithm With Discrete Chaotic Lozi Map. In: Proceedings of 26th European Conference on Modelling and Simulation, ECMS 2012, pp. 405–409 (2012) ISBN 978-0-9564944-4-3
44. Pluhacek, M., et al.: Designing PID Controller for 4th Order System By Means of Enhanced PSO Algorithm with Discrete Chaotic Dissipative Standard Map. In: Proceedings of 24th European Modeling & Simulation Symposium, EMSS 2012, pp. 396–401 (2012) ISBN 978-88-97999-09-6
45. Pluhacek, M., et al.: On the Performance of Enhanced PSO Algorithm with Lozi Chaotic Map. In: Application of Modern Methods of Prediction, Modeling and Analysis of Nonlinear Systems. SCI, vol. 1, p. 18. Springer (November 2012) (accepted for publication) ISSN: 1860-949X
46. Glover, F., Laguna, M., Mart, R.: Scatter Search. In: Ghosh, A., Tsutsui, S. (eds.) Advances in Evolutionary Computation: Theory and Applications, pp. 519–537. Springer, New York (2003)
47. Beyer, H.-G.: Theory of Evolution Strategies. Springer, New York (2001)
48. Holland, J.H.: Genetic Algorithms. Scientific American, 44–50 (July 1992)
49. Clerc, M.: Particle Swarm Optimization. ISTE Publishing Company (2006) ISBN 1905209045
50. Matousek, R.: HC12: The Principle of CUDA Implementation. In: Matousek (ed.) 16th International Conference on Soft Computing, MENDEL 2010, Brno, pp. 303–308 (2010)

51. Matousek, R., Zampachova, E.: Promising GAHC and HC12 algorithms in global optimization tasks. Journal Optimization Methods & Software 26(3), 405–419 (2011)
52. 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)
53. 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, http://www.intechopen.com/books/evolutionary-algorithms/analytical-programming-a-novel-approach-for-evolutionary-synthesis-of-symbolic-structures , doi:10.5772/16166