

Evolutionary Identification of Chaotic System

Ivan Zelinka*, Roman Senkerik*, Zuzana Oplatkova*, Donald Davendra*

* *Department of Applied Informatics, Faculty of Applied Informatics, Tomas Bata University in Zlin, Nad Stranemi 4511, 762 72 Zlin, Czech Republic (Tel: +420 57 603 5189; e-mails: {senkerik, zelinka, oplatkova, davendra}@fai.utb.cz).*

Abstract: Synthesis, identification and control of complex dynamical systems are usually extremely complicated. When classic methods are used, some simplifications are required which tends to lead to idealized solutions that are far from reality. In contrast, the class of methods based on evolutionary principles is successfully used to solve this kind of problems with a high level of precision. In this paper an alternative method of evolutionary algorithms, which has been successfully proven in many experiments like chaotic systems synthesis, neural network synthesis or electrical circuit synthesis. This paper discusses the possibility of using evolutionary algorithms for the identification of chaotic systems. The main aim of this work is to show that evolutionary algorithms are capable of the identification of chaotic systems without any partial knowledge of internal structure, i.e. based only on measured data. Two different evolutionary algorithms are presented and tested here in a total of 10 versions. Systems selected for numerical experiments here is the well-known logistic equation. For each algorithm and its version, repeated simulations were done, amounting to 50 simulations. According to obtained results it can be stated that evolutionary identification is an alternative and promising way as to how to identify chaotic systems.

Keywords: chaos, identification, regression, genetic algorithms, artificial intelligence, chaotic behaviour

1. INTRODUCTION

Identification of various dynamical systems is vitally important in the case of practical applications as well as in theory. A rich set of various methods for dynamical system identification has been developed. In the case of chaotic dynamics, it is for example the well-known reconstruction of chaotic attractor based on research of Takens (1981) who has shown that, after the transients have died out, one can reconstruct the trajectory on the attractor from the measurement of a single component. Because whole trajectory contains too much information, series of papers by Grassberger, Hentschel and Procaccia (1983), Halsey et al. (1986), Eckmann and Procaccia (1986) is introduced to show a set of averaged coordinate invariant numbers (generalized dimensions, entropies, and scaling indices) by which different strange attractors can be distinguished. Method presented here is based on evolutionary algorithms (EAs), see Back et al. (1997), and allows reconstruction not only of chaotic attractors as a geometrical objects, but also their mathematical description. All those techniques belong to the class of genetic programming techniques; see Koza (1998), Koza et al., 1999]. Generally, when it is used on data fitting, these techniques are called symbolic regression (SR).

The term symbolic regression (SR) represents a process, in which measured data is fitted by a suitable mathematical formula such as $x^2 + C$, $\sin(x)+1/e^x$, etc., Mathematically, this process is quite well known and can be used when data of an unknown process is obtained. Historically SR has been in the preview of manual manipulation, however during the recent

past, a large inroad has been made through the use of computers. Generally, there are two well-known methods, which can be used for SR by means of computers. The first one is called genetic programming or GP, Koza (1998), Koza et al. (1999) and the other is grammatical evolution, O'Neill and Ryan (2002), Ryan et al. (1998).

The idea as to how to solve various problems using SR by means of EA was introduced by John Koza, who used genetic algorithms (GA) for GP. Genetic programming is basically a symbolic regression, which is done by the use of evolutionary algorithms, instead of a human brain. The ability to solve very difficult problems is now well established, and hence, GP today performs so well that it can be applied, e.g. to synthesize highly sophisticated electronic circuits, Koza et al. (2003).

In the last decade of the 20th century, C. Ryan developed a novel method for SR, called grammatical evolution (GE). Grammatical evolution can be regarded as an unfolding of GP due to some common principles, which are the same for both algorithms. One important characteristic of GE is that it can be implemented in any arbitrary computer language compared with GP, which is usually done (in its canonical form) in LISP. In contrast to other evolutionary algorithms, GE was used only with a few search strategies, for example with a binary representation of the populations in O'Sullivan and Ryan, (2002). Another interesting investigation using symbolic regression was carried out by Johnson (2003) working on Artificial Immune Systems or/and systems which are not using tree structures like linear genetic programming (full text is at <https://eldorado.uni-dortmund.de/bitstream>

/2003/20098/2/Brameierunt.pdf) and another similar algorithm to AP, Multi Expression Programming (see <http://www.mep.cs.ubbcluj.ro/>).

Put simply, evolutionary algorithm simulates Darwinian evolution of individuals (solutions of given problem) on a computer and are used to estimate-optimize numerical values of defined cost function. Methods of GP are able to synthesize in an evolutionary way complex structures like electronic circuits, mathematical formulas etc. from basic set of symbolic (nonnumeric) elements.

In this paper, analytic programming (AP) is applied, see Zelinka et al. (2008), Zelinka (2002a), Zelinka (2002b), Zelinka and Oplatkova (2003), Zelinka and Oplatkova (2004) for the identification of selected chaotic system. Identification is not done on the “level” of strange attractor reconstruction, but it produces a symbolic-mathematical description of the identified system. Investigation reported here is continuation of research done in Zelinka et al. (2008).

2. MOTIVATION

Motivation of this investigation is quite simple. As mentioned in the introduction, evolutionary algorithms are capable of hard problem solving. A lot of examples about evolutionary algorithms can be easily found. Evolutionary algorithms use with chaotic systems is done for example in Richter and Reinschke (2000) where EAs has been used on local optimization of chaos, Richter (2002) for chaos control with use of the multi-objective cost function or in Richter (2005, 2006) where evolutionary algorithms have been studied on chaotic landscapes. Slightly different approach with evolutionary algorithms is presented in Zelinka et al. (2008), selected algorithms were used to synthesize artificial chaotic systems. In Zelinka (2006 & 2008) EAs has been successfully used for real-time chaos control and in Senkerik et al. (2006) and Zelinka et al. (2007) EAs was used for optimization of Chaos Control.

Another examples of evolutionary algorithms use can be found in Dashora et al., (2007) which developed statistically robust evolutionary algorithms, and on the opposite side Hwang et al. (2007) used evolutionary algorithms for fuzzy power system stabilizer which has been applied on single-machine infinite bus system and multi-machine power system. Other research was done by Liu et al. (2007). Parameters of permanent magnet synchronous motors has been optimized by particle swarm algorithm and experimentally validated on the servomotor. In Das and Konar, (2007), swarm intelligence has been used for IIR filter synthesis and He and Wang, (2007) applied co-evolutionary particle swarm optimization (CoPSO) approach for the design of constrained engineering problems, particularly for pressure vessel, compression spring and welded beam.

The main question in the case of this paper is if EAs are able to identify chaos in symbolic i.e. mathematical description. All experiments here were designed to check and either confirm or reject this idea.

3. CHAOS IDENTIFICATION - PROBLEM DESIGN

3.1 Problem selection, used algorithms and computer technology

Based on previous successful experiments of Zelinka et al. (2008), the well-known logistic equation (1) has been selected for experiments.

$$x_{n+1} = Ax_n(1 - x_n) \quad (1)$$

The selection has been made because its structure is simple, well studied and analyzed, however, this does not imply that other systems cannot be used.

For the experiments described here, stochastic optimization algorithms, such as Differential Evolution (DE) (Price, 1999) and Self Organizing Migrating Algorithm (SOMA) (Zelinka, 2004), have been used. Application of alternative algorithms like Genetic algorithm (GA) and Simulated Annealing (SA), are now in process.

All experiments have been done on a special server consisting of 16 Apple XServer (2 x 2 GHz Intel Xeon, 1 GB RAM), each with 4 CPU, so in total 64 CPUs were available for calculations. It is important to note here, that such technology was used to save time due to a large number of calculations (done during 500 simulations), however it must be stated that evolutionary identification described here, is also solvable on a single PC. For all calculations and data processing, *Mathematica*[®] version 6.0.3.0 was used.

3.2 The Cost Function

The cost function (2) has been designed so that its minimization should lead to the identification of a system with the same behaviour as the original system.

$$CV = \sum_{i=300}^{400} \sum_{j=200}^{300} |data_{i,j}^l - data_{i,j}^{ident}| \quad (2)$$

The cost function consists of two sums with $data_{i,j}^l$ (sorted data representing behaviour of logistic equation) and $data_{i,j}^{ident}$ (sorted data representing behaviour of identified system). The first sum ($i \in [300, 400]$) represents the fact that synthesized systems has to be identified for the interval of the control parameter $A \in [3, 4]$ in which chaos is for (1) generated. Parameter A has been changed by step 0.01, so 100 different time series was recorded. For each setting of A , 300 iterations has been done. Last 100 data-points (from 300 in total) were taken into calculation from each time series to calculate final sum (or create bifurcation diagrams) – this is represented by second sum ($j \in [200, 300]$). Based on previous facts, there was generated for each system $100 \times 300 = 30\,000$ values and its cost value calculation was $100 \times 100 = 10\,000$. Minimal value that can be achieved by (2) is 0, i.e. system with this cost value is probably an exact identification of the original system.

For all experiments a threshold has been set, which has been used for decision making, that the identified system belong to similar or exotic class of systems. System with cost value equal 0 which were exact identifications of the original system, with cost value $\in [0, 1500]$ are reported as similar identifications and with cost value > 1500 as exotic identifications.

4. EXPERIMENT SETUP

Four versions of SOMA and six versions of DE have been applied in order with AP and were used for all simulations in this paper. See **Table 1** and Table 2 for relation between each version and index corresponding to another Tables. Parameters for the optimizing algorithm were set up in such a way as to reach the same value of maximal cost function evaluations for all used versions. Each version of EAs has been applied 50 times in order to synthesize an appropriate structure which can serve as models of the observed chaotic system.

The primary aim here is not to show which version is better or worse, but to show that the EA can in reality be used for identification of chaotic systems without knowledge of internal structure or/and auxiliary information. The ranges of all estimated parameters are reported in Table 3 and Table 4. Basic set of symbolic element (GFS) used for synthesis consist of : A, x, +, -, *, /.

Table 1 Used Versions of SOMA

Algorithm	Index
AllToOne	S1
AllToRandom	S2
AllToAll	S3
AllToAllAdaptive	S4

Table 2 Used Versions of DE

Algorithm	Index
DERand1Bin	D1
DERand2Bin	D2
DEBest2Bin	D3
DELocalToBest	D4
DEBest1JIter	D5
DERand1DIter	D6

Table 3 SOMA setting

	S1	S2	S3	S4
PathLength	3	3	3	3
Step	0.11	0.11	0.11	0.11
PRT	0.1	0.1	0.1	0.1
PopSize	200	200	40	40
Migrations	8	8	4	4
MinDiv	-1	-1	-1	-1
Individual Length	50	50	50	50
Max. CF Evaluations	42984	42984	42120	42120

Table 4 DE setting

	D1 - D6
NP	200
F	0.9
CR	0.3
Generations	200
Individual Length	50
Max. CF Evaluations	40000

Results from all experiments are reported in detail in the following sections. In totality, it can be stated that during all 500 simulations, original logistic equation has been identified on 32 occasions (6.4% from all simulations), similar systems that less or more fit the behaviour of the logistic equation on 79 occasions (15.8%). Therefore in total 111 identified cases (22.2%), as given in Table 5.

Table 5 Experiment summarization

Note	Total value	%
Total number of simulations	500	100
Exact identifications	32×	6.4
Similar identifications	79×	15.8
Total number of acceptable identifications	111×	22.2

5. EXPERIMENTAL RESULTS

5.1 Exact identification

During all simulations, the canonical version of the logistic equation has been synthesized 32× in total, (see Table 6). Logistic equation has been identified in 5 various versions which are clearly algebraic variation of its canonical version, i.e. after simple algebraic manipulations we get (1), see (3) - (7).

Table 6 Summarization of canonical version synthesis

Equation No.	Synthesized
(3)	6×
(4)	14×
(5)	4×
(6)	3×
(7)	5×
Total	32×

$$A(x - x^2) \quad (3)$$

$$x(A - Ax) \quad (4)$$

$$Ax - Ax^2 \quad (5)$$

$$A(1 - x)x \quad (6)$$

$$-x(-A + Ax) \quad (7)$$

5.2 Similar systems

Beside the canonical version of the logistic equation there also been synthesized systems, which less or more fit the behaviour of the original system. Selected examples of very good approximation of (1) are for example systems (8) and (9), see Fig. 1. Significantly “worst” approximations are for example (10) and (11), see Fig. 2 and Fig. 3. Corresponding cost values are given in Table 8. Minimal, maximal and average cost values of accepted similar systems (according to threshold) in this “category” are reported in Table 7.

$$x \left(A - Ax + \frac{(1-A)x}{A^2(2A-x+Ax)} \right) \tag{8}$$

$$x \left(A - Ax + \frac{x^2}{A \left(\frac{1}{A} + A + \frac{A}{x} + Ax(A+x) \right)} \right) \tag{9}$$

$$\frac{A(1-x)x(x+(-A+x)(A+x))}{-A^2-x} \tag{10}$$

$$x \left(A - \frac{A^3x}{A+A^2+x} - \frac{A^2-2A(A-x)}{-\frac{A^2}{x}+2x} \right) \tag{11}$$

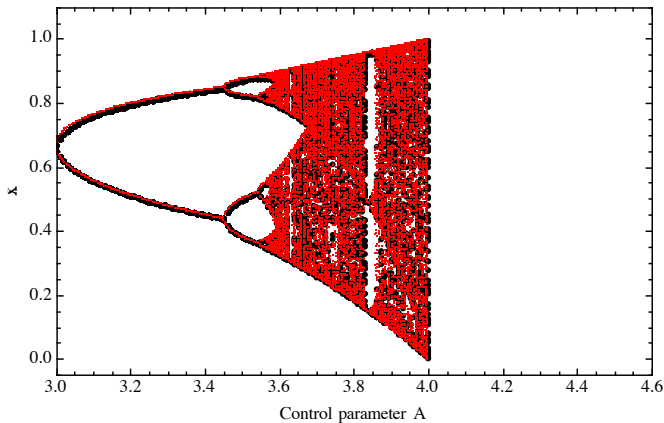


Fig. 1 The best-synthesized solution, see (8) and (9). Red (thin) points represent the canonical logistic equation; black (thick) points represent the synthesized system.

Table 7 Similar systems – an overview

	Cost value
Minimum (the best, see Fig. 1)	129.549
Average	1001.02
Maximum	1456.19

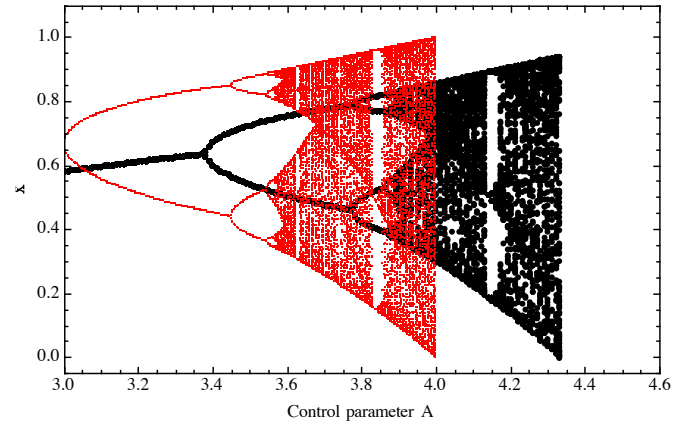


Fig. 2 Another solution, basically the same behaviour of (1), only shifted along axis x, see (10). Red (thin) points represent the canonical logistic equation; black (thick) points represent the synthesized system.

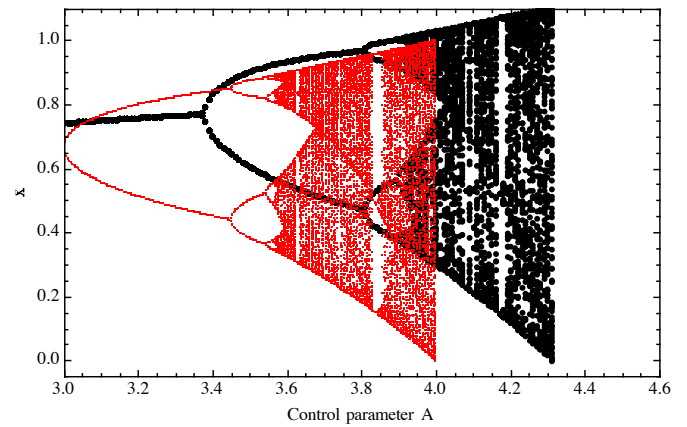


Fig. 3 Basically the same case as in Fig. 1, see (11). Red (thin) points represent the canonical logistic equation; black (thick) points represent the synthesized system.

Table 8 Cost values of similar systems

System (Eq.) No.	Cost value
(8)	129.549
(9)	136.706
(10)	1048.72
(11)	993.346

Behaviour of other similar systems is reported in Fig. 4. From the given figures, it is visible that evolution has found really similar systems and their precise “evolutionary adjustment” to the logistic equation is probably only question of better setting of evolution parameters.

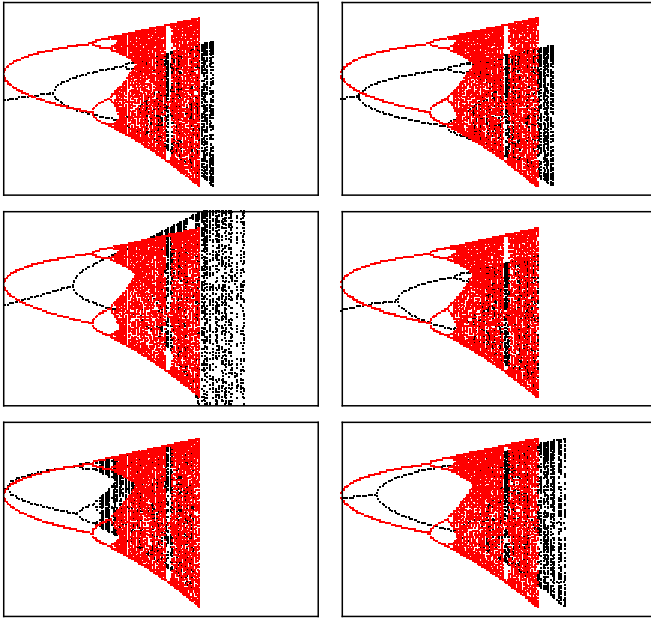


Fig. 4 A few another similar systems. Red (thin) points represent the canonical logistic equation; black (thick) points represent the synthesized system. Figures are drawn again in intervals $x \in [3, 4.6]$ and $y \in [0, 1]$

5.3 Exotic solutions

Together with acceptable systems, other systems were also synthesized, which did not fit the threshold mentioned in the section *Cost function*. This category is termed “exotic”, i.e. systems that are very different from the logistic equation, however there is still visible as a similar structure of logistic equation. An example can be the systems (12) and (13), which had been synthesized during all 500 experiments. For behaviour of systems (12) see Fig. 5, and for (13) see Fig. 6. Another selected example is depicted in Fig. 7.

$$A - x + x^2 - \frac{(A - x)x^3(-x + x^2)(x + A(2A + A^2 + x))}{A\left(1 - \frac{A}{x}\right)} \quad (12)$$

$$x \left(\frac{2A - A\left(-x - \frac{x^2}{A}\right)\left(-2x + \frac{x}{A - \frac{1+2x}{x^2}}\right)}{A} \right) \quad (13)$$

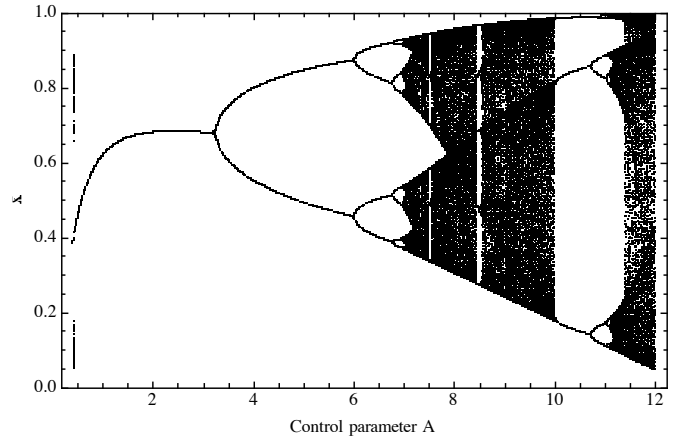


Fig. 5 Example of “exotic” solution, see (12)

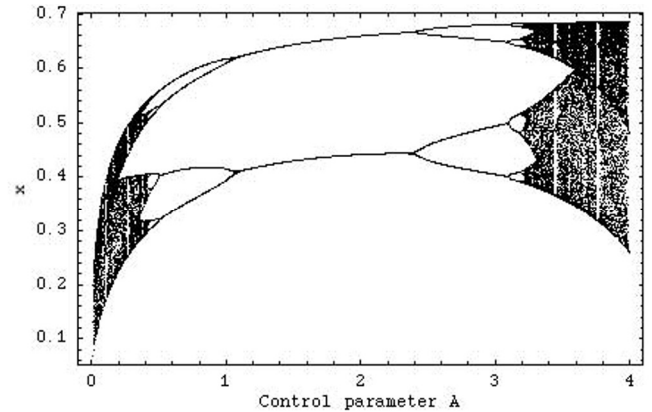


Fig. 6 Another “exotic” solution, see (13)

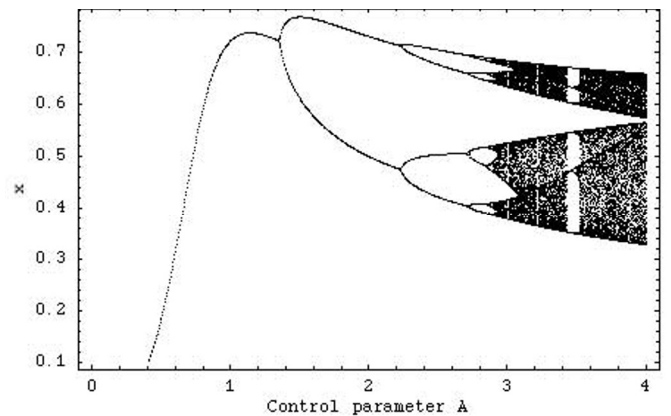


Fig. 7 Another “exotic” solution

6. CONCLUSIONS

Based on obtained results, it may be stated that simulations provided promising results, which shows that EAs are capable of model identification of chaotic systems. In this participation two evolutionary algorithms in 10 versions were

used and tested. Exact descriptions of logistic equation as well as its variations have been identified from the results (see for example (8)). The question is why such complex equation like (8) have similar behaviour to (1). The answer is simple. After expansion of (8), equation (14) is obtained. The first part is basically the logistic equation (see Fig. 8). The remaining part participates on the final behaviour without significant impact (see Fig. 9).

$$Ax - Ax^2 + \frac{x^2}{A^2(2A - x - Ax)} - \frac{x^2}{A(2A - x - Ax)} \quad (14)$$

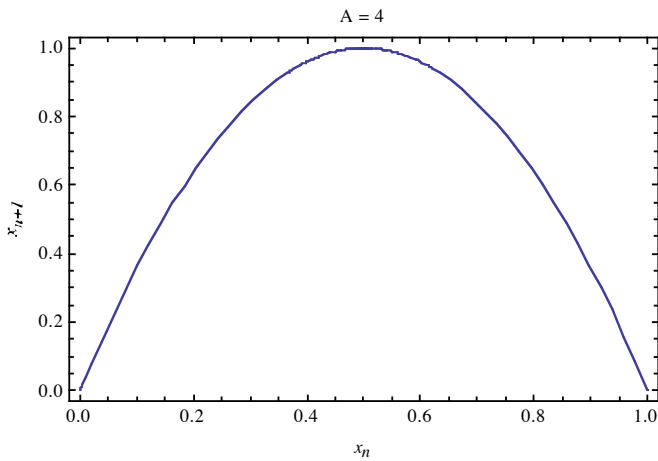


Fig. 8 Graph of the first part ($Ax - Ax^2$) of (14).

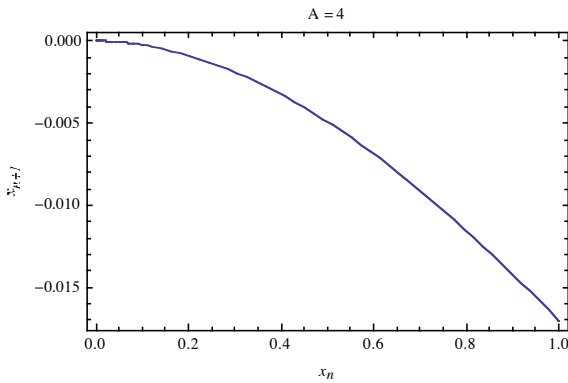


Fig. 9 Graph of the second part (14).

Based on previously mentioned facts and all experimental results, conclusions and statements can be made as follows:

Experiment overview. The cost function (2) consist of two sums where the total number of synthesized data-points was 10 000 from 30 000 (see section “Cost Function”). Based on the fact that 500 experiments were conducted, 15 000 000 data-points and 5 000 000 of these points were used for evaluation of all synthesized systems.

Number of successful identification. In this participation, results were divided into three categories: *exact*, *similar* and *exotic* identifications. The representation is as follows: *exact* implies that logistic equation has been recovered in its canonical version (or its algorithmic variations), *similar*

means that behaviour of the synthesized systems was visually the same (or same and shifted along x axis) like that of the logistic equation, however with different mathematical description (see (8), (9), (10) and (11)). *Exotic* identification is partially similar to the original one. Based on all data analysis, it can be stated that a) exact form of logistic equation has been synthesized 32 times (see Table 6). Number of synthesized similar systems was 79. For general overview see Table 7.

Used algorithms and experiment settings. All algorithms has been initialized so that a) population size remained the same, b) cost function evaluations was similar amongst algorithms as much as possible. The first “condition” has not been followed for algorithms S3 and S4 compared to S1 and S2 (see Table 3). It is caused by different internal algorithm structure for new individuals calculations. Due to this fact condition b) has been kept with highest priority.

Behaviour preciseness. It should be noted here that identification has not been focused on exact behaviour identification for each time development of logistic equation, but on similarity of behaviour via data used later for bifurcation diagrams. Despite the fact that some of them were precisely estimated, it is our duty to say that sometimes, **very rarely** and only for special setting of parameter A , were trajectories of synthesized systems running to infinity. To avoid this “side effect” the above-mentioned cost function should contain in future a penalization for such kinds of effects. From Fig. 4, it is also visible that a little bit longer time is needed to better estimation of system description. For some identified systems it has been observed that while for $A \in [3, 4]$ behavior identical or very similar with of logistic equation was produced, whereas other values of A (for example negative) other chaotic behavior were generated, see for example $-A + (-1/A) + A/x - 2x + (A - x)x$ with $A \in [0, 1]$

Problem complexity and algorithm performance. Based on the fact that individual can consist of 50 symbolic elements, there are 3.04×10^{64} possible combinations of synthesized structures – systems, including senseless combinations. This is of course only the theoretical number, because some combinations will be avoided due to process of synthesis (only mathematically acceptable {functions with appropriate number of arguments, ...} structures are synthesized). However, in layman’s terms, it can be stated that all 111 synthesized solutions (from 500 in total) represents 3.64×10^{-61} % of such defined searched space. If we will follow maximal allowed number of cost function evaluations (see Table 3 + Table 4 = 410 208 cost function evaluations, i.e. tested solutions) then evolutions searched maximally 1.34×10^{-57} % of the search space. Lets take a simplified time point of view. When for example MacBook, 2.33 MHz Intel Core Duo with 3 GB RAM is used, then one cost function evaluation needs (if we omit time needed for formula synthesis) approx. 0.3659 s. Then to evaluate all possible combinations by simple enumeration would take approx. 3.52×10^{56} years. This is 2.35×10^{46} longer than our universe exist. All those numbers clearly shows that EAs are

powerful enough to handle such tasks and obtained results are not simply a matter of randomness.

Other evolutionary techniques. In this paper, the so-called analytic programming has been used, however we have to say that another and more well known techniques like genetic programming, see Koza (1998), Koza et al. (1999) or grammatical evolution, see O'Neill and Ryan (2002), should give similar results as reported here.

In conclusion, it has to be stated that, a) EAs use on chaos identification is a promising direction of research; b) to increase the number of successful identifications (see Table 6 and Table 7) the cost function or/and algorithm settings should be improved. Also, more extensive set of simulations based on more rich set of evolutionary algorithms and chaotic systems is needed – which is now in process at Tomas Bata University in Zlin.

ACKNOWLEDGEMENT

This work was supported by grant No. MSM 7088352101 of the Ministry of Education of the Czech Republic and by grants of the Grant Agency of the Czech Republic GACR 102/06/1132.

REFERENCES

- Back T., Fogel D. B. and Michalewicz Z., (1997), Handbook of Evolutionary Computation, *Institute of Physics*, London,
- Dashora, Y., et al., (2007). Improved and generalized learning strategies for dynamically fast and statistically robust evolutionary algorithms. *Engineering Applications of Artificial Intelligence*, doi:10.1016/j.engappai.2007.06.005
- Eckmann, J. P., and Procaccia, I. (1986): “Fluctuation of Dynamical Scaling Indices in Non-Linear Systems“, *Phys. Rev.* 34 A, 659.
- Gi-Hyun H., Dong-Wan K., Jae-Hyun L., and Young-Joo A., (2008). Design of fuzzy power system stabilizer using adaptive evolutionary algorithm, *Engineering Applications of Artificial Intelligence*, Volume 21, Issue 1, February 2008, 86-96
- Grassberger, P., and Procaccia, I. (1983 a), On the Characterization of Strange Attractors, *Phys. Rev. Lett.* 50, 346.
- Grassberger, P., and Procaccia, I. (1983 b). Estimation of the Kolmogorov Entropy From a Chaotic Signal, *Phys. Rev.* 29 A, 2591.
- Grassberger, P., and Procaccia, I. (1983 c), Measuring the Strangeness of Strange Attractors, *Physica 9 D*, 189.
- Halsey, T. C., Jensen, M. H., Kadanoff, L. P., Procaccia, I., and Schraiman, B. I. (1986): “Fractal Measures and Their Singularities: the Characterization of Strange Sets”, *Phys. Rev.* 33 A, 1141.
- Johnson C. G., (2003), “Artificial immune systems programming for symbolic regression”, In C. Ryan, T. Soule, M. Keijzer, E. Tsang, R. Poli, and E. Costa, editors, *Genetic Programming: 6th European Conference*, LNCS 2610, p. 345-353
- Koza J. R., Keane M. A., and Streeter M. J., (2003), “Evolving Inventions”, *Scientific American*, February 2003, p. 40-47, ISSN 0036-8733
- Koza J.R. (1998), *Genetic Programming II*, MIT Press, ISBN 0-262-11189-6, 1998
- Koza J.R., Bennet F.H., Andre D., and Keane M., (1999), “*Genetic Programming III*”, Morgan Kaufmann pub., ISBN 1-55860-543-6
- Li L., Wenxin L., and David A. C., (2007), Particle swarm optimization-based parameter identification applied to permanent magnet synchronous motors, *Engineering Applications of Artificial Intelligence*, doi:10.1016/j.engappai.2007.10.002
- O'Neill M. and Ryan C. (2002), *Grammatical Evolution. Evolutionary Automatic Programming in an Arbitrary Language*. Kluwer Academic Publishers, ISBN 1402074441
- Oplatkova Z., and Zelinka I. (2006) : Investigation on Artificial Ant using Analytic Programming, *GECCO 2006*, Seattle, WA, USA, 8-12.7.2006, 11th conference on Genetic Programming
- O'Sullivan J., and Conor R., (2002), An Investigation into the Use of Different Search Strategies with Grammatical Evolution”, *Proceedings of the 5th European Conference on Genetic Programming*, p.268 - 277, 2002, Springer-Verlag London, UK, ISBN:3-540-43378-3
- Price K. (1999), An Introduction to Differential Evolution, In D. Corne, M. Dorigo and F. Glover, Eds., *New Ideas in Optimization*, p. 79–108, McGraw-Hill, London, UK, ISBN 007-709506-5
- Qie H., and Ling W. , (2007), An effective co-evolutionary particle swarm optimization for constrained engineering design problems, *Engineering Applications of Artificial Intelligence*, Volume 20, Issue 1, February 2007, 89-99
- Richter H. and Reinschke K. J., (2000), Optimization of local control of chaos by an evolutionary algorithm, *Physica D*, 144, pp. 309-334
- Richter H., (2002). An evolutionary algorithm for controlling chaos: The use of multi-objective fitness functions, in Parallel Problem Solving from Nature-PPSN VII (Eds.: Guervós M., Panagiotis J. J., Beyer A., Villacanas F. H. G., J. L. & Schwefel H. P.), *Lecture Notes in Computer Science*, Vol. 2439 (Springer-Verlag, Berlin), pp. 308-317
- Richter H., (2006) Evolutionary Optimization in Spatio-temporal Fitness Landscapes, *Lecture Notes In Computer Science*, Springer-Verlag, NUMB 4193, pp. 1-10, ISSN 0302-9743
- Richter H., (2005). A study of dynamic severity in chaotic fitness landscapes, *Evolutionary Computation*, 2005. The *IEEE Congress*, Volume 3, Issue 2-5 Sept. 2005, pp. 2824 – 2831, Vol. 3
- Ryan C., Collins J.J., and O'Neill (1998), Grammatical Evolution: Evolving Programs for an Arbitrary Language, *Lecture Notes in Computer Science 1391*. First European Workshop on Genetic Programming

- Das S., and Konar A., (2007), A swarm intelligence approach to the synthesis of two-dimensional IIR filters, *Engineering Applications of Artificial Intelligence*, Volume 20, Issue 8, December 2007, 1086-1096
- Zelinka I., (2002) a, Analytic programming by Means of Soma Algorithm, In: *Proc. 8th International Conference on Soft Computing Mendel'02*, Brno, Czech Republic, 2002, 93-101., ISBN 80-214-2135-5
- Zelinka I., (2002) b, Analytic programming by Means of Soma Algorithm, ICICIS'02, *First International Conference on Intelligent Computing and Information Systems*, Egypt, Cairo, ISBN 977-237-172-3
- Zelinka I., and Oplatkova Z., (2003), Analytic programming – Comparative Study”, CIRAS'03, *The second International Conference on Computational Intelligence, Robotics, and Autonomous Systems*, Singapore, 2003, ISSN 0219-6131
- Zelinka I., and Oplatkova Z., (2004), Boolean Parity Function Synthesis by Means of Arbitrary Evolutionary Algorithms - Comparative Study, In: *8th World Multiconference on Systemics, Cybernetics and Informatics* (SCI 2004), Orlando, USA, in July 18-21, 2004
- Zelinka I., (2004), SOMA – Self Organizing Migrating Algorithm“, Chapter 7, 33 p. In: B.V. Babu, G. Onwubolu (eds), *New Optimization Techniques in Engineering*, Springer-Verlag, ISBN 3-540-20167X
- Zelinka I., (2006), Investigation on Realtime Deterministic Chaos Control by Means of Evolutionary Algorithms, In: *1st IFAC Conference on Analysis and Control of Chaotic Systems*, Reims, France
- Senkerik R., Zelinka I., and Navratil E., (2006), Optimization of feedback control of chaos by evolutionary algorithms, In: *1st IFAC Conference on Analysis and Control of Chaotic Systems*, Reims, France,
- Zelinka I., Senkerik R., and Navratil E., (2007), Investigation on Evolutionary Optimization of Chaos Control, *CHAOS, SOLITONS & FRACTALS*, doi:10.1016/j.chaos.2007.07.045
- Zelinka I., Chen G., Celikovsky S., (2008), Chaos Synthesis by Means of Evolutionary Algorithms, *International Journal of Bifurcation and Chaos*, University of California, Berkeley USA, Vol 18, No 4, p. 911 – 942
- Zelinka, I., (2008) Real-time deterministic chaos control by means of selected evolutionary algorithms, *Engineering Applications of Artificial Intelligence*, doi:10.1016/j.engappai.2008.07.008