

# EVOLUTIONARY ALGORITHMS AND DETERMINISTIC CHAOS

*Ivan Zelinka*

Department of Applied Informatics, Faculty of Applied Informatics,  
Tomas Bata University in Zlin, Nad Stranemi 4511, 762 72 Zlín, Czech Republic  
[zelinka@fai.utb.cz](mailto:zelinka@fai.utb.cz)

## Abstract

This keynote speech discusses mutual intersection of evolutionary techniques and deterministic chaos. Three possible application of evolutionary computation on deterministic chaos are discussed here. The first one discuss chaos synthesis by means of evolutionary computation and the second one chaos control of simple chaotic system. The last, third, part discuss use of evolutionary computation on spatiotemporal chaos control.

**Key words:** evolutionary techniques, chaos, genetic programming, control, synthesis, CML

## 1. Introduction

Deterministic chaos, discovered by E. Lorenz is a fairly active area of research in the last few decades. The Lorenz system produces a well-known chaotic attractor in a simple three-dimensional autonomous system of ordinary differential equations, whereas for discrete chaos, there is another famous chaotic system, called logistic equation, which was found based on a predator-prey model showing complex dynamical behaviors. Chaotic behavior of various systems is actual topic in engineering and science today. It can be used for example in secret communication or can cause serious problems in various engineering devices in the case of wrong design. Chaotic behavior can be observed also in power systems and in its extreme consequences can cause power systems colaps, so called blackout. Blackout can be also modeled by Thom's catastrophe theory which is tightly joined with chaos via bifurcations. In almost the same period has been discovered so called evolutionary algorithms, which are very powerful in modern engineering design of various devices. This paper introduces a few fruitful intersections of deterministic chaos and evolutionary techniques.

In the first part will be discussed the notion of chaos synthesis by means of evolutionary algorithms and demonstrate a new method for chaotic systems synthesis. The aim of this part is to show how to synthesize new and "simple" chaotic systems based on some elements contained in a pre-chosen existing chaotic system and a properly defined cost function by means of evolutionary techniques.

In the second part is discussed investigation on optimization of the feedback control of chaos based on the use of evolutionary algorithms. The main

objective is to show that evolutionary algorithms are capable of optimization of chaos control. As models of deterministic chaotic systems, one dimensional Logistic equation and two dimensional Henon map were used. The optimizations were realized in several ways, each one for another set of parameters of evolution algorithms or separate cost functions.

The last part discuss continuation of an investigation on deterministic spatiotemporal chaos real-time control by means of selected evolutionary techniques. Real-time like behavior is specially defined and simulated with spatiotemporal chaos model based on mutually nonlinearly joined  $n$  equations, so called Coupled Map Lattices - CML. CML system has been considered like black-box system for controlling by evolutionary algorithms.

In total four evolutionary algorithms are used throughout all simulations and results: differential evolution, self-organizing migrating algorithm, genetic algorithm and simulated annealing in 12 versions.

This keynote paper is based on author research and represents only main and important informations from all three above mentioned applications of evolutionary algorithms on deterministic chaos. For more information are in each part representative references, containing full experiment description.

## 2. Example case studies

All presented results here are available in full description in authors previous articles, referenced in each case study.

### 2.1. Chaos synthesis

This part introduces the outline of chaos synthesis by means of evolutionary algorithms and develops a new method for chaotic systems synthesis. This method is similar to genetic programming and grammatical evolution and is being applied along with three evolutionary algorithms: differential evolution, self-organizing migrating, and genetic algorithm. The aim of this investigation is to synthesize new and "simple" chaotic systems based on some elements contained in a pre-chosen existing chaotic system and a properly defined cost function. The investigation consists of 11 case studies: the aforementioned three evolutionary algorithms in 11 versions. For all algorithms, 100 simulations of chaos synthesis were repeated and then averaged to guarantee the reliability and robustness of the

proposed method. The most significant results were carefully selected, visualized and commented, see [1].

### 2.1.1. Motivation

In recent years, interests in softcomputing methods are increasing, including in particular evolutionary algorithms. These algorithms are based on similar principles of biological evolution in the real world. The aim of EAs is to solve computationally hard problems which are too complex to be solved by conventional methods. In its canonical form, EAs can be used only for numerical estimation of parameters (usually, arguments of a given cost function). Together with EAs in the canonical form, another modification allows to use EAs as a symbolic “constructors”, i.e., a processor, for synthesizing complex structures in a symbolic way, based on some predefined simple elements (mathematical operators or electronic elements like diode, transistor, etc.). The term “symbolic way” specifies that mathematical structures and equations, electronic systems, etc., are generated from those simple elements just mentioned.

Given the above background, the main motivation of this investigation was the question “Is it possible to synthesize the mathematical description of a new chaotic system, based on simple and elementary mathematical objects, by means of evolutionary computation?” This question was also based partially on the fact that in engineering applications, it is very often vitally important to know not only when chaos can be generated but also how to generate it [2], [3]. This is extremely important in cryptography, for example, where chaotic systems are often used in the design. From a mathematical point of view, it is quite clear that there are some classes of chaotic systems which can be represented by one canonical form (one class – one canonical form) [4]. However, generally speaking, it is not so easy to exactly synthesize a chaotic system with specified features by means of classical mathematical methods. A positive answer to the question mentioned above would open possibilities to synthesize not only a set of not-yet-described chaotic systems, but also some chaotic systems with predefined features. It is believed that such possibilities would have an important impact on engineering design of various complex nonlinear systems, especially chaotic systems.

### 2.1.1. Experiment results

Three evolutionary algorithms in 11 versions, has been used in this investigation [1]. Symbolic objects (e.g., variables, constants,...) for manipulation and complex structure synthesis were selected from the well-known logistic equation:

$$x_{n+1} = Ax(1 - x) \quad (1)$$

This selection was based on the fact that the logistic equation is a well-known simplest system that can produce chaotic behavior. This equation is also well analysed. It was expected that evolutionary search would be possible to synthesize the logistic equation. Evolutionary synthesis of logistic equation was actually observed, as discussed in [1]. Another reason behind the selection of the logistic equation is that results from designed experiments can be easily compared, verified and analyzed. In this investigation, a total of 1100 independent simulations were completed, 100 trials by each of the 11 algorithms. Each simulation was started at randomly selected initial conditions (i.e., each initial population was randomly generated).

### 2.1.1. Selected results and discussion

The aim of this investigation is to show how various chaotic systems can be synthesized by means of evolutionary algorithms. Evolutionary synthesis of chaotic systems has been applied to 11 basic comparative simulations in this paper. Each comparative simulation was repeated 100 times and all 1100 results (100 simulations for each algorithm) were used to create figure (see samples Fig. 1 - Fig. 3 for overall performance evaluation of evolutionary chaos synthesis. The results look quite promising and convincing.

For comparative studies, three algorithms were used – differential evolution (DE) [5], SOMA [6] and genetic algorithm (GA) [7], [8]. They were chosen to show that evolutionary synthesis of chaos can be implemented via any evolutionary algorithm and that they all give reasonable results.

The method of symbolic regression described in this paper is relatively simple, but feasible to implement and easy to use. Based on its principles and its possible universality (as just mentioned, it was tested with 3 evolutionary algorithms – SOMA, DE and GA in 11 versions), symbolic regression seems quite capable of synthesizing new dynamical systems for generating chaos.

As a summary, the following statements are presented:

**Result verification.** To be sure that the results as presented in the paper are correct, all written synthesized functions were used for automatic generation of bifurcation diagrams and Lyapunov exponents.

**Simulation results.** Based on the results in [1] (see selected equations (2) – (4)) and the selected bifurcation diagrams (Fig. 1 - Fig. 3), it can be stated that all simulations give satisfactory results and that

evolutionary synthesis of chaos is capable of solving this class of problems.

**Range of chaos and interval of observation.** During evolutions, chaos was searched by focusing on interval  $\langle 0, 4 \rangle$ , based on the priori known behavior of the logistic equation, whose elements were used in the evolution. Despite the a priori known information, a few chaotic systems were located outside of this interval, see [1]. That was due to the fact that a part of chaotic behavior was inside the interval  $\langle 0, 4 \rangle$  and thus EA was able to identify it. From these facts, it is clear that EA are able to locate chaos in a wider range than those expected from some textbook exemplary systems.

$$\frac{(1-x)(-2A+x)}{-2A + \frac{A}{x} + x + x(A+x)} \quad (2)$$

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

$$\frac{A(-A+x)}{2+A+x-x^2+2x(A+x^2)+\frac{x}{x+x^2}} \quad (4)$$

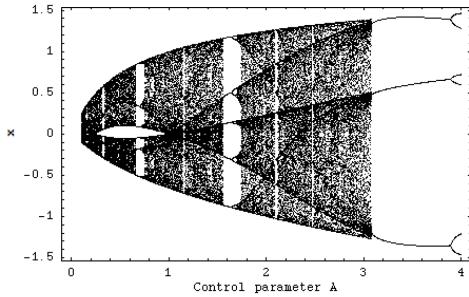


Figure 1. Bifurcation diagram of synthesized system (2)

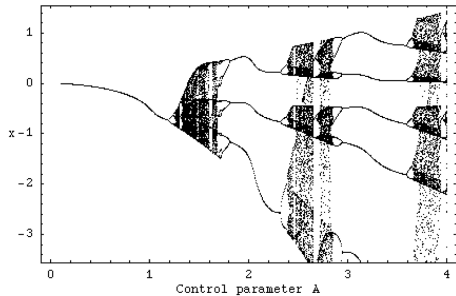


Figure 2. Bifurcation diagram of synthesized system (3)

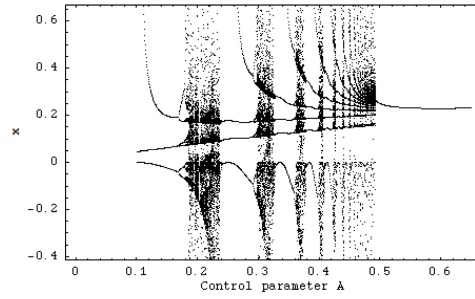


Figure 3. Bifurcation diagram of synthesized system (4)

## 2.2. Evolutionary chaos control

This part discuss an investigation on optimization of the feedback control of chaos [9] based on the use of evolutionary algorithms. The main objective is to show that evolutionary algorithms are capable of optimization of chaos control. As models of deterministic chaotic systems, one dimensional Logistic equation and two dimensional Henon map were used. The optimizations were realized in several ways, each one for another set of parameters of evolution algorithms or separate cost functions. The evolutionary algorithm SOMA was used in four versions. For each version simulations were repeated several times to show and check for robustness of the applied method. Reason why EAs were used for finding of optimal control parameter setting comes from Fig.4. There is typical graph of cost function comprising chaotic dynamics. It is clear that standard optimization methods would not work there.

In this work [9], the following design of the cost function (CF) was developed and tested for stabilization of  $p-1$  orbit (fixed point) and higher periodic orbits. The CF has been calculated in general from the distance between desired state and actual system output. The minimal value of this cost function revealing the best solution is zero. The aim of all the simulations was to find the best solution that returns the cost function value as close as possible to zero. This proposal of the simplest cost function ( $CF_1$ ) could be used only for the stabilization of  $p-1$  orbit. The idea was to minimize the area created by the difference between the required state (stabilized fixed point) and the real system output on the whole simulation interval  $-\tau$ , thus this proposal of CF should secure fast targeting into the close neighborhood of  $p-1$  orbit and its stabilization. The  $CF_1$  is given by (5).

$$CF_1 = \sum_{t=0}^{\tau} |TS_t - AS_t| \quad (5)$$

Where: TS - target state, AS - actual state

But another cost function ( $CF_2$ ) had to be used for stabilizing of higher periodic orbit. It was synthesized from the simple  $CF_1$  (5) and other terms

were added. In this case it is not possible to use the simple rule of minimizing the area created by the difference between the required and actual state on the whole simulation interval –  $\tau$ , due to the many serious reasons, for example: degrading of the possible best solution by phase shift of periodic orbit. This CF is in general based on searching for desired stabilized periodic orbit and thereafter calculation of the difference between desired and found actual periodic orbit on the short time interval –  $\tau_s$  (approx. 20 - 50 iterations) from the point, where the first min. value of difference between desired and actual system output is found. Such a design of CF should secure the successful stabilization of higher periodic orbit anyway phase shifted. Furthermore because of CF values very close to zero this CF also allows using of decision rule avoiding very time demanding simulations. This rule stops EA immediately, when the first individual with good parameter structure is reached, thus the value of CF is lower then acceptable ( $CF_{acc}$ ) one. Typically  $CF_{acc} = 0.001$  at time interval  $\tau_s = 20$  iterations, thus difference between desired and actual output has value 0.0005 per iteration – i.e. successful stabilization for used control technique. This CF can also be used for p-1 orbit. The  $CF_2$  has form (6).

$$CF_2 = \text{penalization}_1 + \sum_{i=\tau_1}^{\tau_2} |TS_i - AS_i| \quad (6)$$

Where: TS - target state, AS - actual state  
 $\tau_1$  - the first min. value of difference between TS and AS  
 $\tau_2$  – the end of optimization interval ( $\tau_1 + \tau_s$ )  
 $\text{penalization}_1 = 0$  if  $\tau - \tau_2 \geq \tau_s$ ;  $\text{penalization}_1 = 10 * (\tau - \tau_2)$  if  $\tau - \tau_2 < \tau_s$  (i.e. late stabilization)

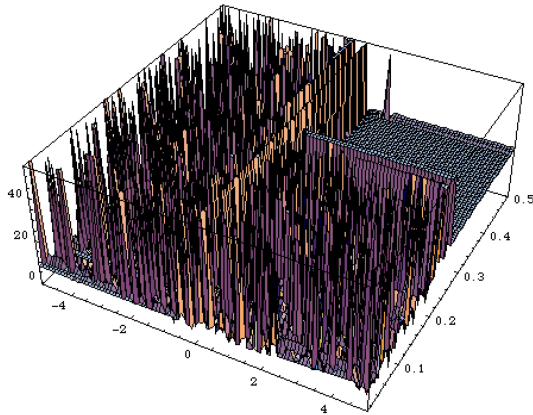


Figure 4. Typical surface of cost function comprising chaotic dynamic

### 2.2.1. Experiment results

The optimization of chaos control described here is relatively simple and easy to implement. Based on obtained results, see [9], it may be claimed that all simulations give satisfactory results and thus EA are capable of solving this class of difficult problems and the quality of results does not depend only on the problem being solved but also on the proper definition of the CF.

From the comparison with classical control technique – OGY follows that TDAS or ETAS based control method can be simply considered as targeting and stabilizing algorithm and their performance is much better than OGY.

As can be seen from the optimization results, see [9], they are extremely sensitive to the construction of used CF, for example the problem with fast stabilization not only for initial conditions used in optimization process, but for the whole range of the initial conditions or the problem with the best individual solution which was not suitable for simulation with distributed conditions in the case of Logistic equation. Any small change in the design of CF can cause radical improvement of system behavior.

There is no problem for the future research in defining much more complex CF comprising as subcriteria control of stability, costs, time-optimality, controllability, or any of their arbitrary combinations. Furthermore parameter settings for EA were based on heuristic approach; therefore there is also possibility for the future research. According to all results (see selected figures Fig. 5-7) it is planned that the main activities will be focused on testing more complex cost functions together with searching for better settings of EA and certainly on testing of evolutionary deterministic chaos control in continuous-time and high-order systems and finally testing of evolutionary real-time chaos control.

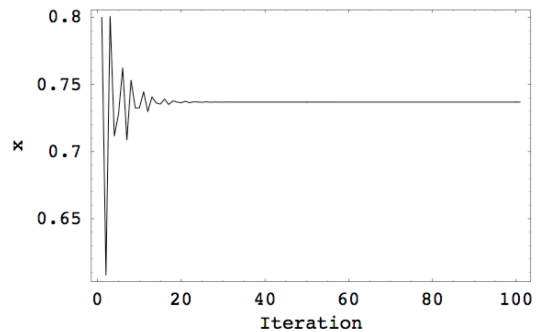


Figure 5. The best solution – algorithm SOMA ATAA.



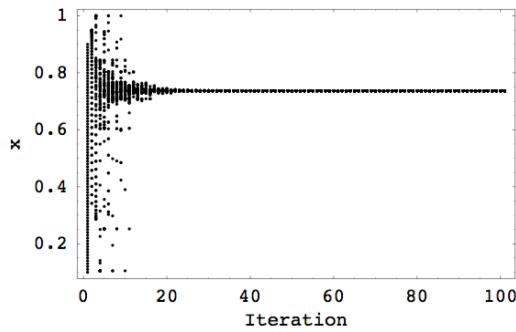


Figure 6. 100× repeated stabilization for 1 UPO

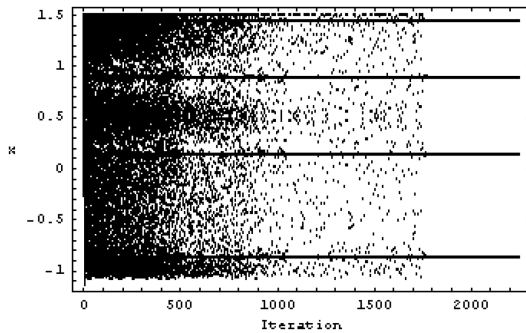


Figure 7. 100× repeated stabilization for 4 UPO

### 2.3. Spatiotemporal chaos control

This part introduces continuation of an investigation [14] and [15] on deterministic spatiotemporal chaos real-time control by means of selected evolutionary techniques [19]. Real-time like behavior is specially defined and simulated with spatiotemporal chaos model based on mutually nonlinear joined  $n$  equations, so called Coupled Map Lattices - CML. Four evolutionary algorithms are used for chaos control here: differential evolution, self-organizing migrating algorithm, genetic algorithm and simulated annealing in total of 12 versions. For modelling of real-time spatiotemporal chaos behavior, so called coupled map lattices were used based on logistic equation to generate chaos. The main aim of this investigation was to show that evolutionary algorithms, are under certain conditions, are capable of real-time control of deterministic chaos, when the cost function is properly defined as well as parameters of selected evolutionary algorithm. Investigation consists of four different case studies with increasing simulation complexity. For all used algorithms each simulation was 100 times repeated to show and check robustness of used methods. All data were processed and used in order to get summarizing results and graphs.

The class of CML problems chosen for this comparative study was based mostly on case studies reported in and partially partially on [10], [11]. In

general, controlling of the CML means setting of such pinning sites (control CML sites) and their pinning values (control inputs) so that system stabilizes itself on expected spatiotemporal pattern (in the case of this article on T1S2 – time period 1, space period 2). CML was chosen, as an object of study, because it shows chaotic behavior and its level of complexity can be quite rich.

Computer simulations were designed so that they would simulate real-time CML process, based on following philosophy: real-time means that there is no way to return back CML state and initial conditions, which means that CML has to run, for new calculations of pinning inputs, further in time. To combine use of EAs, a little change was done in contrary with “non real-time control” published in for example in [12], [13], [14], [15]. Simply, EA process, i.e. individuals and population development has followed CML behavior in time in a way that a certain number of CML iterations were reserved for evaluation of each individual in the population. This number has been set just heuristically and should reflect time gap of real system needed for new control input processing, before new individual - input to CML is used. It means that if for example population has size say 20 individuals and number of CML iterations between two individual evaluation was 10, then in case of 30 generations (presumed to needed to reach suitable control input) gives total number of CML iterations as 6000 ( $20 \times 10 \times 30$ ). From individual point of view it means that the 1st individual in the 1st generation has started at 1st CML iteration, 1st individual in the 2nd generation has started at  $(20 \times 10) + 1 \dots 201$ st iteration, etc. Despite the fact that all CML calculation were done in PC environment (no real model was presented) are these calculations / simulations are called experiments, because the combination of EA and CML, less or more or less simulate possible real time control of the CML system. The author is convicted that this is competent assumption, because as was demonstrated in [16] and [17], EAs are capable of real-time process control.

Investigation here consists of four parts in increasing order from calculation complexity point of view and was based on paper [18] and [14], [15]. The first one is focused on pinning values estimation for a priori given number of the pinning sites. In the second one, the pinning sites with a priori given pinning values were estimated by EA. The third simulation was focused on mutual estimation of pinning sites and values, i.e. EA was searching for the number of pinning sites and optimal (as much as possible) pinning values. The fourth and last case was similar to the previous one. One modification was done so that together with pinning values also minimal number of pinning sites has been estimated. All simulations were based on the same CML model of

size 20 possible pinning sites, and were repeated 100 times for each EA with new initial conditions for each simulation. In total, there were done 4800 (12 algorithms x 4 case studies x 100 repeated experiments) independent simulations were conducted of real-time control of spatiotemporal chaos on Power Mac G5 Quad 2.5GHz with software Grid Mathematica®.

### 2.2.1. Cost function

The fitness (cost function) has been calculated according to the distance between desired CML state and actual CML output (7) and (8) as in [14], [15]. The minimal value of the cost function (7), guarantee of the best solution, is 0, and for the cost function (8) value  $p1$ . The aim of all simulations based on (7) and (8) was to find the best solution, i.e. a solution that returns as lowest cost value as possible. The cost function (7), was used for the first two experiments (pinning values setting, pinning sites setting). In the last two experiments, cost function (8) was used. It is synthesized from cost function (7) so that two terms are heuristically added. The first one  $p_1$ , represents number of pinning sites in CML. The second one  $p_2$ , is added here to “attract attention” of evolutionary process on main part of cost function. It is in fact, the number of used pinning sites, i.e. in the optimal case cost function (8) should return  $p_1$ . If the second part of (8) would not be there, then mainly  $p_1$  would be optimized and results would not be acceptable (proved by simulations). Indexes  $i$  and  $j$  are coordinates of lattice element, i.e.  $CML_{i,j}$  is  $i^{\text{th}}$  site (equation) in  $j^{\text{th}}$  iteration. In all the simulations for the target of control,  $TS_{i,j}$  was set to pattern T1S2 (see [19]) so that CML behavior was controlled to periodical state.

Based on real-time control ideas described here, one can in contrary with simulations in [14], [15] expect, that (8) will take no effect because evolution is stopped before all generations are done. Cost function (8) was used here to prove that penalization, at least of this kind, has no effect on quality of real-time evolutionary control here, as described later.

$$f_{\text{cost}} = \sum_{i=1}^{20} \sum_{j=m}^{m+n} |TS_{i,j} - CML_{i,j}|^2 \quad (7)$$

$TS_{i,j}$  - target state of CML

$CML_{i,j}$  - actual state of controlled CMI

$m$  - actual iteration of the CML

$n$  - iterations, see Tab. 1.

$$f_{\text{cost}} = p1 + \left( p2 \sum_{i=1}^{20} \sum_{j=m}^{m+n} |TS_{i,j} - CML_{i,j}| \right)^2 \quad (8)$$

$TS_{i,j}$  - target state of CML

$CML_{i,j}$  - actual state of controlled CML

$p1$  - number of actually selected pinning sites

$p2$  - 1000, heuristically set weight constant

$m$  - actual iteration of the CML

$n$  - iterations

### 2.2.1. Experiment results

The main aim of this case study was to show how various CML real-time control problems were solved by means of evolutionary algorithms. Evolutionary deterministic chaos control was used here in four basic comparative simulations. Each comparative simulation was 100 times repeated and all 4800 results (100 simulations for each of 12 algorithm and for all four experiments) were used to create Fig. 8 - Fig. 10 for performance evaluation of real-time evolutionary deterministic chaos control (for more see [19]).

Algorithms chosen here are - DE [5] and SOMA [6], GA [7] and SA [8]. They were chosen to show that evolutionary deterministic chaos control can be regarded as a “black box” method (no a priori information about CML structure is needed) and that it can be implemented using arbitrary evolutionary algorithms as shown also in [14], [15].

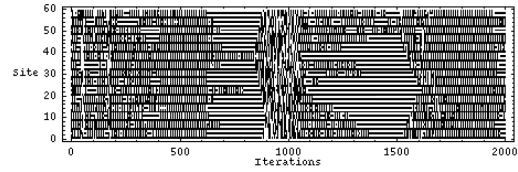


Figure 8. Typical example of evolutionary controlled CML, “regions” of stabilized behavior can be observed

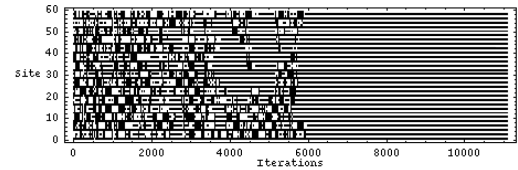


Figure 9. Typical example of evolutionary controlled CML – extended phase

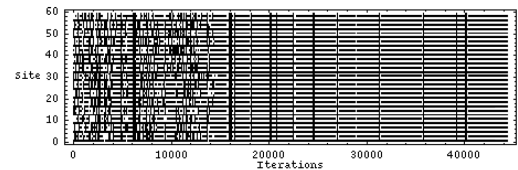


Figure 10. Another example of evolutionary controlled CML

As a conclusion the following statements are presented:

1. **Reached results.** Based on results reported in [19] it can be stated that all simulations give satisfactory results and thus evolutionary real-time deterministic chaos

control is capable of solving this class of problems.

2. **No. of successful experiments.** In all 4800 experiments, only a few failure had been observed, which means that almost all 12 algorithms had found a suitable solution before evolution has finished.
3. **Mutual comparison.** When comparing all algorithms, it is visible that algorithms give good results. Parameter setting for all algorithms was based on heuristic approach and thus there is a possibility that better settings can be found there.
4. **Penalization effect.** The penalization in the cost function (8) basically had no effect on reached results (case D), which seems to be in contradiction with results in [14], [15]. Explanation is quite simple – evolution had found a suitable solution before effect of the penalization could be graphically visible. The cost function (8) should be redefined if more intensive penalization effect would be desired.
5. **No. of pinning sites.** The same principle applies as in the previous paragraph again minimal values should be taken into consideration because this number of pinning sites was large enough for T1S2 pattern stabilizing.
6. **Random control.** To check if the described results are not a matter of randomness, another test, based on random control sequence, was designed. The random generator was used to generate control inputs in a random way for experiment A. In total, there was repeated 10 simulations, each for random control input within 100 000 CML iterations, i.e. 1 000 000 CML iterations was used in total. When positive control was recorded during experiment, then CML has been “restarted” and remaining iterations were used for repeating of experiment, till 100000 iterations have been reached.

### 3. Conclusions

Based on results, it can be stated that methods of evolutionary deterministic chaos control described here are relatively simple, easy to implement and easy to use. Their capability of evolutionary deterministic chaos control is obvious.

Three case studies were reported here: chaos synthesis, chaos control and spatiotemporal chaos control. In all three cases EAs demonstrated that their use is important and alternative way in chaos control and synthesis.

### 4. References

- [1] Zelinka, I., Guanrong Ch., Celikovskiy S., Chaos Synthesis by Means of Evolutionary algorithms, International Journal of Bifurcation and Chaos, Vol. 18, No. 4 (2008) 911–942
- [2] Chen G. & Dong X. [1998] From Chaos to Order: Methodologies, Perspectives and Applications (World Scientific, Singapore)
- [3] Perruquetti W. & Barbot J. P. [2005] Chaos in Automatic Control, (CRC, Boca Raton)
- [4] Gilmore R. & Lefranc M. [2002] The Topology of Chaos: Alice in Stretch and Squeezeland (Wiley-Interscience, New York)
- [5] Price K. [1999] An Introduction to Differential Evolution, New Ideas in Optimization, Ed.: Corne D., Dorigo M. Glover F. (McGraw-Hill, London, UK), 79–108
- [6] Zelinka I. [2004] SOMA – Self Organizing Migrating Algorithm, in New Optimization Techniques in Engineering, Eds.: Babu B. V. & Onwubolu G. (Springer-Verlag, New York), 167-218
- [7] Holland J. H. [1975] Adaptation in Natural and Artificial Systems, (Univ. Michigan Press, Ann Arbor)
- [8] Davis L. [1996] Handbook of Genetic Algorithms, (Van Nostrand Reinhold, Berlin)
- [9] Zelinka I., Senkerik R., Navratil E., Investigation on Evolutionary Optimizatiom of Chaos Control, CHAOS, SOLITONS & FRACTALS (2007), doi:10.1016/j.chaos.2007.07.045
- [10] Xu Zhang and Ke Shen, [2001], Controlling spatiotemporal chaos via phase space compression, Phys. Rev. E 63, 046212
- [11] Ogorzalek, M.J.; Adachi, M., [1996], Control strategies for spatio-temporal chaotic systems and neural networks, ISCAS '96., 'Connecting the World', IEEE International Symposium on Circuits and Systems, 3, 9 - 12
- [12] Just W., Benner H. & Reibold E. [2003] Theoretical and experimental aspects of chaos control by time-delayed feedback, Chaos, 13, 259-266
- [13] Mizutani, S.; Sano, T.; Uchiyama, T.; Sonehara, N., [1995], Controlling Chaos in Chaotic Neural Networks, IEEE International Conference on Neural Networks, 6, 3038
- [14] Zelinka I. [2005a]. Investigation on Evolutionary Deterministic Chaos Control, IFAC, Prague 2005
- [15] Zelinka Ivan [2005b]. Investigation on Evolutionary Deterministic Chaos Control – Extended Study, In: 19th International Conference on Simulation and Modeling, (ECMS 2005), Riga, Latvia, in June 1-4, 2005
- [16] Nolle L., Zelinka I., Hopgood A. A. & Goodyear A. [2005] Comparison of an self organizing migration algorithm with simulated

annealing and differential evolution for automated waveform tuning, *Advances in Engineering Software*, 36 (10), 645-653

- [17] Zelinka I. & Nolle L. [2006] Plasma reactor optimizing using differential evolution, in *Differential Evolution: A Practical Approach to Global Optimization*, Eds.: Price K.V., Lampinen J. & Storn R. (Springer-Verlag, New York), 499-512
- [18] Hu G., Xie F., Xiao J., Yang J. & Qu Z. [1999] Control of patterns and spatiotemporal chaos

and its application, in *Handbook of Chaos Control*, Ed.: Schuster H. G. (Wiley VCH, New York)

- [19] Zelinka, I., Real-time deterministic chaos control by means of selected evolutionary algorithms *Engineering Applications of Artificial Intelligence* (2008), doi:10.1016/j.engappai.2008.07.008



Copyright of AIP Conference Proceedings is the property of American Institute of Physics and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.