Chapter 8
Evolutionary Reconstruction of Chaotic Systems

Ivan Zelinka and Ales Raidl

Abstract. This chapter discusses the possibility of using evolutionary algorithms for the reconstruction of chaotic systems. The main aim is to show that evolutionary algorithms are capable of the reconstruction of chaotic systems without any partial knowledge of internal structure, i.e. based only on measured data. Five different evolutionary algorithms are presented and tested in a total of 13 and 12 versions in two different versions of experiments. System selected for numerical experiments here is the well-known logistic equation. For each algorithm and its version, 100 repeated simulations were conducted. According to obtained results it can be stated that evolutionary reconstruction is an alternative and a promising way as to how to identify chaotic systems.

8.1 Introduction

Identification of various dynamical systems is vitally important in theory and in practical applications. A rich set of various methods for dynamical system identification has been developed. In the case of chaotic dynamics, an example is the well-known reconstruction of chaotic attractor based on research of [35] 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, the entire trajectory contains too much information, a series of papers by [12], [8] is introduced

Ivan Zelinka
Tomas Bata University in Zlin, Faculty of Applied Informatics, Nad Stranemi 4511,
Zlin 76001, Czech Republic
and
VSB-TUO, Faculty of Electrical Engineering and Computer Science, 17. listopadu 15,
708 33 Ostrava-Poruba, Czech Republic
e-mail: zelinka@fai.utb.cz

Ales Raidl
Charles University, Faculty of Mathematics and Physics,V Holesovickach 2,
180 00 Prague 8, Czech Republic
e-mail: ales.raidl@mff.cuni.cz

I. Zelinka et al. (Eds.): Evolutionary Algorithms and Chaotic Systems, SCI 267, pp. 265–291.
springerlink.com

to show a set of averaged coordinate invariant numbers (generalized dimensions, entropies, and scaling indices) by which different strange attractors can be distinguished. The method presented here is based on evolutionary algorithms (EAs), see [1], which allows the reconstruction not only of chaotic attractors as a geometrical object, but also their mathematical description. All those techniques belong to the class of genetic programming techniques; see [17],[18]. Generally, when it is used on data fitting, these techniques are called symbolic regression (SR). The term symbolic regression (SR) represents a process, by 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 (GP), [17], [18] and the other is grammatical evolution, [22], [31]. 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, for example to synthesize highly sophisticated electronic circuits, [19]. 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 [23]. Another interesting investigation using symbolic regression was carried out by [14] 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/). Simply put, 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 chapter, analytic programming (AP) is applied, see [43], [36], [37], [41], [42] 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 a continuation of research done in [43] or extended study reported in Chapter 11.

Synthesis, identification and control of complex dynamical systems are usually extremely complicated. When classics methods are used, some simplification is 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 chapter an alternative method of evolutionary algorithms, which has been successfully proven by many experiments like chaotic systems synthesis, neural network synthesis or electrical circuit synthesis. This chapter discusses the possibility of using evolutionary algorithms for the identification (reconstruction) of chaotic systems. The main aim of this work is to show that evolutionary algorithms are capable of reconstruction of chaotic systems without any partial knowledge of internal structure, i.e. based only on measured data. Five different evolutionary algorithms are presented and tested here in a total of 13 and 12 versions. The system selected for numerical experiments here is the well-known logistic equation for discrete systems and Lorenz attractor for continuous systems. For each algorithm and its version, 100 repeated simulations were conducted. According to obtained results it can be stated that evolutionary reconstruction is an alternative and promising way as to how to identify chaotic systems.

8.2 Motivation

Motivation of this investigation is quite simple. As mentioned in the introduction, evolutionary algorithms are capable of hard problem solving. Numerous examples on evolutionary algorithms can be easily found. Evolutionary algorithms use with chaotic systems is done for example in [30] where EAs has been used on local optimization of chaos, [27] for chaos control with use of the multi-objective cost function or in [28] and [29], where evolutionary algorithms have been studied on chaotic landscapes. A slightly different approach of evolutionary algorithms is presented in [43], where selected algorithms were used to synthesize artificial chaotic systems. In [39], [40], EAs has been successfully used for real-time chaos control and in [34] and [44] EAs was used for the optimization of Chaos Control. Other examples of evolutionary algorithms usage can be found in [6] which developed statistically robust evolutionary algorithms, and on the opposite side [11] 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 [20]. Parameters of permanent magnet synchronous motors has been optimized by particle swarm algorithm and experimentally validated on the servomotor. In [5], swarm intelligence has been used for IIR filter synthesis and [26] 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 chapter is if EAs are able to identify chaos in symbolic i.e. mathematical description. All experiments here were designed to check and either affirm or negate this idea.

8.3 Chaos System Reconstruction – Classical Methods

8.3.1 Reconstruction Based on Time Series Analysis

Control of chaos, calculation of quantifiers of chaos etc. requires the trajectories of the dynamic system in the phase space to be examined. This does not pose a serious problem if the control equations of the system are explicitly given. In actual experimental practice, however, one is often faced with the fact that the time sequence of one or, more favorably, more than one variable is measured while the equations of the system are unknown. For instance, only air temperature or air pressure is measured. Such a time series can be interpreted as a projection of the trajectory into one of the axes of the phase space. The task is then to reconstruct the trajectory in the phase space based on the measured time series of the scalar quantity. Under some assumptions, we are really able to reconstruct substantial properties of the system dynamics. The process of estimation of the chaos quantifier values thus separates into two main steps, i.e.:

- **reconstruction** of the trajectory in the phase space on a chaotic attractor, and
- **calculation** of the chaos descriptors itself.

Both steps can be subsequently used for additional modeling of the system, such as nonlinear prediction or reduction of noise in a signal, etc. For those purposes the so called reconstruction of the trajectory in the phase space is used. This will be, although a little bit imprecisely, abbreviated to "phase space reconstruction".

Time delay method is the approach to phase space reconstruction which is most frequently used. Let us start from a situation where a scalar quantity $x(t_i), i = 1, \ldots, N$ is measured at uniformly distributed time moments $t_1, t_2 = t_1 + \Delta t, \ldots, t_i = t_1 + (i-1)\Delta t, \ldots, t_N$. In the early 1980s, Pakard, Crutchfield, Farmer and Shaw [24], Takens [35] and, according to [7], also Ruelle independently proposed constructing an m-dimensional signal as follows:

$$\begin{aligned} &X(t_1) = [x(t_1), x(t_1+\tau), x(t_1+2\tau), \ldots, x(t_1+(m-1)\tau)] \\ &X(t_2) = [x(t_2), x(t_2+\tau), x(t_2+2\tau), \ldots, x(t_2+(m-1)\tau)] \\ &\vdots \\ &X(t_i) = [x(t_i), x(t_i+\tau), x(t_i+2\tau), \ldots, x(t_i+(m-1)\tau)], \qquad i = 1, \ldots, M \end{aligned} \tag{8.1}$$

where m is the dimension of immersion, τ is a suitable time delay and $M = N - (m-1)\tau$. The quantities m and τ together are called immersion parameters. Under rather general assumptions, dynamics reconstructed through the system of eq. (8.1) is equivalent to the system dynamics on the attractor in the initial phase space. This equivalence is understood as the identity of characteristic invariants between the initial and reconstructed attractors. As regards to the full formulation of the immersion theorem, on which the reconstruction eq.(8.1) is based, the interested reader is referred to the original Takens' paper [35] or its extension presented by Sauer, York and Casdagli [33].

Unknown parameters in eq. (8.1) include time delay τ and immersion dimension m. For the latter it has been proven [35] that it is sufficient if $m \geq 2D+1$ and if the attractor is a smooth compact manifold of dimension D; in this case, D takes integer values. However, it is typical where an attractor which has no manifold and has a fractal structure is to be reconstructed. For such situations, Takens' theorem was generalized by the above authors - Sauer, York and Casdagli [33]. According to them, it is sufficient if $m > 2d_c$ where d_c is the capacity of the attractor. In some special cases, the condition for the immersion dimension can be made even less stringent. For the calculation of the correlation dimension d_2 it is even sufficient that $m > d_2$ [32]. As regards to the time delay τ, the immersion theorem does not put any special requirement on its choice, except for the necessity to exclude cases with periodic orbits with periods of τ , 2τ ... etc.

It should be noted that the immersion theorem works absolutely precisely with infinitely long time series. The above requirements for the time delay and for the immersion dimension apply to such cases. In reality, however, a researcher works with a finite volume of data involving some error - error of measurement and/or rounding error introduced by the computer. This should be borne in mind when determining immersion parameters based on experimental data. In fact, it appears that an inappropriate choice of the immersion parameters can affect the result of the reconstruction of the system dynamics substantially and can result in a wrong interpretation of the results of estimation of the attractor's characteristic invariants.

The time delay method represented by Scheme (8.1) is not the only way to construct a multidimensional signal from a one-dimensional time series. Immersion theorems [35], [32] enable us, within trajectory reconstruction in a phase space, to select from a wide choice of operations including a number of smooth transformations, both with the initial time series and with the reconstructed states. This allows a number of techniques to be used, such as principal component analysis, signal differentiation and integration, linear combination of time delayed coordinates or their filtration or utilization of variables simultaneously measured at different sites.

Let us describe the application of differentiated coordinates, which have a simple physical interpretation. For the Lorenz system [21] the system of 3 differential equations (8.2) for the 3 variables x, y, z could be replaced by a single differential equation for a single variable, x, which, however, is a 3rd order quantity. In this way it is possible to pass from the phase space formed by the coordinates (x, y, z) to new coordinates, viz. $(x, \frac{dx}{dt}, \frac{\partial^2 x}{\partial t^2})$. When working with a scalar time series $x(t_i)$, $i = 1, \ldots, N$ recorded in equidistant time intervals Δt, the 1st, 2nd, ... derivatives have to be estimated numerically, e.g.

$$\begin{aligned}
&\frac{dx}{dt}(t)_i \approx \frac{1}{2\Delta t}[x(t_i+\Delta t) - x(t_i-\Delta t)] \\
&\frac{d^2x}{dt^2}(t)_i \approx \frac{1}{\Delta t^2}[x(t_i+\Delta t) + x(t_i-\Delta t) - 2x(t_i)] \\
&\frac{d^3x}{dt^3}(t)_i \approx \frac{1}{2\Delta t^3}[x(t_i+2\Delta t) - 2x(t_i+\Delta t) + 2x(t_i-\Delta t) - x(t_i-2\Delta t)] \\
&\vdots
\end{aligned} \tag{8.2}$$

or, alternatively, higher order formulas have to be used. The coordinates of point $\mathbf{X}(t_i)$ in the phase space will then be

$$\mathbf{X}(t_i) = [x(t)_i, \frac{dx}{dt}(t)_i, \frac{d^2x}{dt^2}(t)_i, \frac{d^3x}{dt^3}(t)_i], \tag{8.3}$$

where m is, as usual, the dimension of the reconstructed phase space satisfying the condition $m > d_c$.

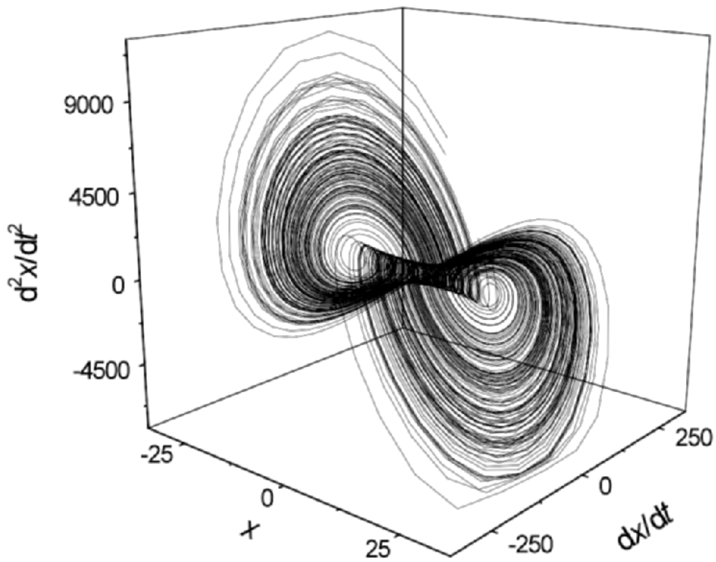

Fig. 8.1 Reconstruction of the Lorenz attractor by using differentiated coordinates.

Comparing the position of the point in the phase space so constructed with the reconstruction which was based on the time delay method (eq. (8.1)) and recalling that time delay τ is an integer multiple of the sampling step Δt, one can see that the differentiated coordinates are nothing more than a linear combination of coordinates obtained from the time delay method.

A different method of constructing the phase space, which obviates some problems encountered with differentiated coordinates and creates an orthogonal base of this space, is based on the principal component analysis approach. In the context of dynamic system analysis, this method was first used in the mid-1980s [3], [9], although its mathematical basis dates back to the early 20th century [10]. Currently, this method can be encountered under various names. Apart from the principal component analysis they include, for instance: decomposition into singular values, empirical orthogonal function, singular spectrum analysis and the Karhunen Loeve transformation [10], [15]. The different names given to the method actually reflect the different methods of covariant matrix estimation from relatively short

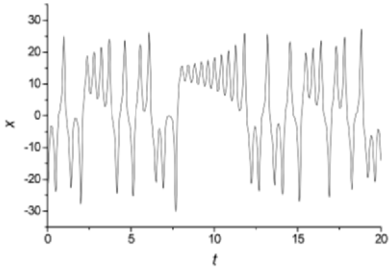

Fig. 8.2 Time development of the first four coordinates during reconstruction of the Lorenz system by using differentiated coordinates.

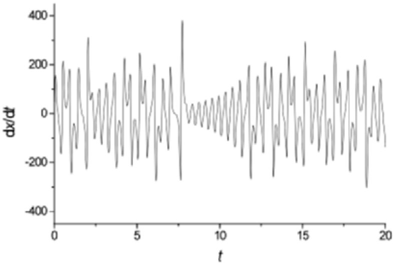

Fig. 8.3 Time development of the first four coordinates during reconstruction of the Lorenz system by using differentiated coordinates.

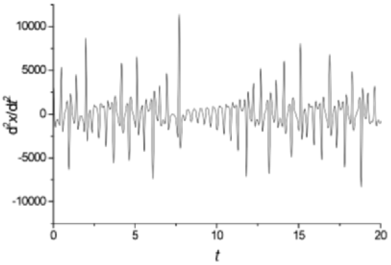

Fig. 8.4 Time development of the first four coordinates during reconstruction of the Lorenz system by using differentiated coordinates.

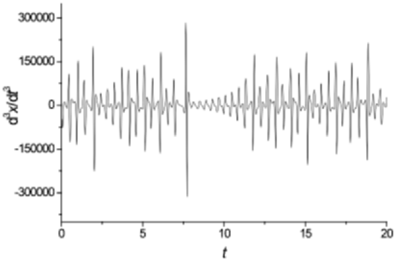

Fig. 8.5 Time development of the first four coordinates during reconstruction of the Lorenz system by using differentiated coordinates.

time series. The differences, however, are not appreciable provided that the immersion dimension m is substantially shorter than the length of the time series [15].

8.4 Evolutionary Reconstruction of Chaotic Systems

Another approach entirely different from classical methods (see previous section or Chapter 7), which is demonstrated in this chapter, is the use of evolutionary algorithms. They are applied on selected examples to demonstrate how evolutionary algorithms can be applied to the reconstruction of chaotic systems. The first example uses data from bifurcation diagram (discrete systems) to synthesize a suitable solution and the second one is using measured time series to partially reconstruct the mathematical description of the Lorenz attractor.

8.4.1 Problem Selection, Used Algorithms and Computer Technology

Based on previous successful experiments of [43], the well-known logistic equation (8.4) has been selected for experiments.

$$x_{n+1} = Ax_n(1 - x_n) \tag{8.4}$$

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. Main idea was to reconstruct mathematical description as described in detail in the Chapter 11, so that two bifurcation diagrams (original and synthesized solution) has been compared. The difference between them (see next section) is calculated like fitness and "says" of what quality the synthesized system is. For the experiments described here, stochastic optimization algorithms (see also Chapter 6), such as Differential Evolution (DE) [25], Self Organizing Migrating Algorithm (SOMA) [38], Genetic Algorithms (GA) [13], Simulated Annealing (SA) [16], [4] and Evolutionary Strategies (ES) [2] were selected. 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 (1300 simulations), however it must be stated that evolutionary reconstruction described here, is also solvable on a single PC. For all calculations and data processing, *Mathematica* version 7 was used.

8.4.2 The Cost Function

The cost function 8.5 has been designed so that its minimization should lead to the reconstruction of a system with the same behavior as the original system.

$$CV = \sum_{i=300}^{400} \sum_{j=200}^{300} \left| data^{L}_{i,j} - data^{ident}_{i,j} \right| \tag{8.5}$$

The cost function consists of two sums calculating the difference between two datasets. The first one, $data^{L}_{i,j}$, represents sorted data of the behavior of the logistic equation and the second one, $data^{ident}_{i,j}$, represent sorted data of the behavior of the identified system. The first sum ($i \in [300, 400]$) represents the fact that the synthesized systems has to be identified for the interval of the control parameter $A \in [3, 4]$ in which chaos is by eq. (8.4) 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 the final sum (or create bifurcation diagrams) - this is represented by the second sum ($j \in [200, 300]$). Based on previous facts, there were generated for each system, $100 \times 300 = 30\,000$ values and for cost value calculation, $100 \times 100 = 10\,000$ values were used. The minimal value that can be achieved by eq. (8.5) is 0, i.e. system with this cost value is probably an exact reconstruction of the original system. For all experiments a threshold has been set, which has been used for decision making, whether the identified system belongs to similar or exotic class of systems. System with cost value equal to 0 which were the exact reconstruction of the original system, with cost value $\in (0, 1500]$ are reported as similar reconstruction and with cost value > 1500 as exotic reconstruction.

8.4.3 Experiment Setup

Four versions of SOMA, six versions of DE, one version of GA, SA and ES have been applied in order with AP and were used for all simulations in this chapter. In Table 8.1 - Table 8.6 abbreviations of used algorithms and their setting is described. Parameters for the optimizing algorithm were set up in such a way as to reach approximately the same value of maximal cost function evaluations for all used versions. Each version of EAs has been applied 100× in order to synthesize

Table 8.1 Algorithms abbreviation

Algorithm	Version	Abbreviation
SOMA	AllToOne	S1
	AllToOneRandomly	S2
	AllToAll	S3
	AllToAllAdaptive	S4
Differential Evolution	DERand1Bin	D1
	DERand2Bin	D2
	DEBest2Bin	D3
	DELocalToBest	D4
	DEBest1JIter	D5
	DERand1DIter	D6
Genetic Algorithm		G
Evolutionary strategies (μ,λ)		ES
Simulated annealing		SA

Table 8.2 SOMA setting for 4 basic search strategies: S1, S2, S3 and S4

Algorithm	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	-0.1	-0.1	-0.1	-0.1
Individual Length	50	50	50	50
Max. CF Evaluations	42984	42984	42120	42120

Table 8.3 DE setting for 6 basic search strategies: D1, D2, D3, D4, D5 and D6

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

Table 8.4 GA setting for canonical version of GA: G

Algorithm	G
PopSize	200
Mutation	0.4
Generations	100
Individual Length	50
Max. CF Evaluations	40000

Table 8.5 ES setting for search strategy: ES

Algorithm	ES
μ ,λ	200
σ	0.8
Iterations	200
Individual Length	50
Max. CF Evaluations	40000

Table 8.6 SA setting for search strategy: SA

Algorithm	SA
No. of particles	200
σ	0.5
k_{max}	66
T_{min}	0.0001
T_{max}	1000
α	0.93
Individual Length	50
Max. CF Evaluations	44600

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 the reconstruction of chaotic systems without knowledge of internal structure or/and auxiliary information. The basic set of symbolic element (GFS) used for synthesis consist of : $A, x, +, -, *, /$.

Results from all experiments are reported in detail in the following sections. In totality, it can be stated that during all 1300 simulations (100%), original logistic equation has been identified on 73 occasions (5.6% from all simulations) and similar systems that less or more fit the behavior of the logistic equation on 186 occasions (14.3%). Therefore, in total 259 identified cases (19.92%), as given in Table 8.7.

Table 8.7 Experiment summarization

Note	Total value	%
Total number of simulations	1300	100
Exact reconstruction	73	5.6
Similar reconstruction	186	14.3
Total number of acceptable reconstruction	259	19.92

8.4.4 Experimental Results

8.4.4.1 Exact Reconstruction

During all simulations, the canonical version of the logistic equation has been synthesized 73× in total, (see Table 8.8). Logistic equation has been identified in 7 various versions which are clearly algebraic variation of its canonical version, i.e. after simple algebraic manipulations we get eq. 8.4, see eq. (8.6) - (8.10).

Table 8.8 Summarization of canonical version synthesis

Equation No.	Synthesized
(8.6)	14×
(8.7)	25×
(8.8)	9×
(8.9)	8×
(8.10)	11×
(8.11)	5×
(8.12)	2×
Total	73×

$$A(x-x^2) \tag{8.6}$$

$$x(A-Ax) \tag{8.7}$$

$$Ax-Ax^2 \tag{8.8}$$

$$A(1-x)x \tag{8.9}$$

$$-x(-A+Ax) \tag{8.10}$$

$$x(1-x)A \tag{8.11}$$

$$x^2(1/x-1)A \tag{8.12}$$

8.4.5 Reconstruction of Similar Systems

Beside the canonical version of the logistic equation, there has also been synthesized systems, which less or more fit the behavior of the original system. Selected examples of very good approximation of eq. (8.4) are for example systems eq. (8.13) and eq. (8.14), see for example Fig. 8.6. Significantly "worst" approximations are for example eq. 8.15 and 8.16, see Fig. 8.7. and Fig. 8.8. Corresponding cost values are given in Table 8.9 and 8.10. Minimal, maximal and average cost values of accepted similar systems (according to threshold) in this "category" are reported there. Behavior of other similar systems is reported in Fig. 8.9 - Fig. 8.16. 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 a question of better setting of evolutionary algorithm parameters.

Table 8.9 Similar systems – an overview

	Cost Value
Minimum	117.538
Average	1053.92
Maximum	1487.85

$$x\left(A-Ax+\frac{(1-A)x}{A^2(2A-x+Ax)}\right) \tag{8.13}$$

$$x\left(A-Ax+\frac{x^2}{A\left(\frac{1}{A}+A+\frac{A}{x}+Ax(A+x)\right)}\right) \tag{8.14}$$

$$\frac{A(1-x)x(x+(-A+x)(A+x))}{-A^2-x} \tag{8.15}$$

$$x\left(A-\frac{A^3x}{A+A^2+x}-\frac{A^2-2A(A-x)}{-\frac{A^2}{x}+2x}\right) \tag{8.16}$$

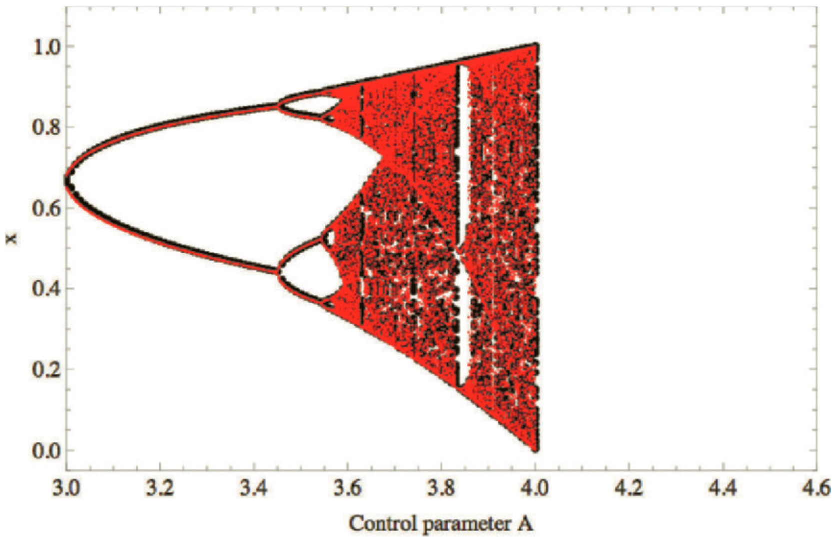

Fig. 8.6 The best synthesized solution, see eq. (8.13) and eq. (8.14). Red (thin) points represent the canonical logistic equation; black (thick) points represent the synthesized system.

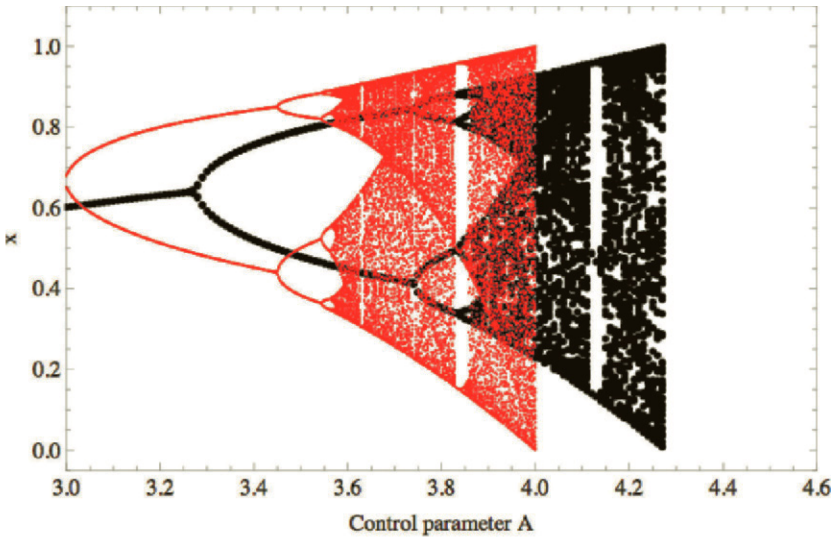

Fig. 8.7 Another solution, basically the same behavior of eq. (8.4), only shifted along axis x, see eq. (8.15). Red (thin) points represent the canonical logistic equation; black (thick) points represent the synthesized system.

Table 8.10 Cost values of similar systems

Equation No.	Cost Value
(8.13)	129.549
(8.14)	136.706
(8.15)	1048.72
(8.16)	993.346

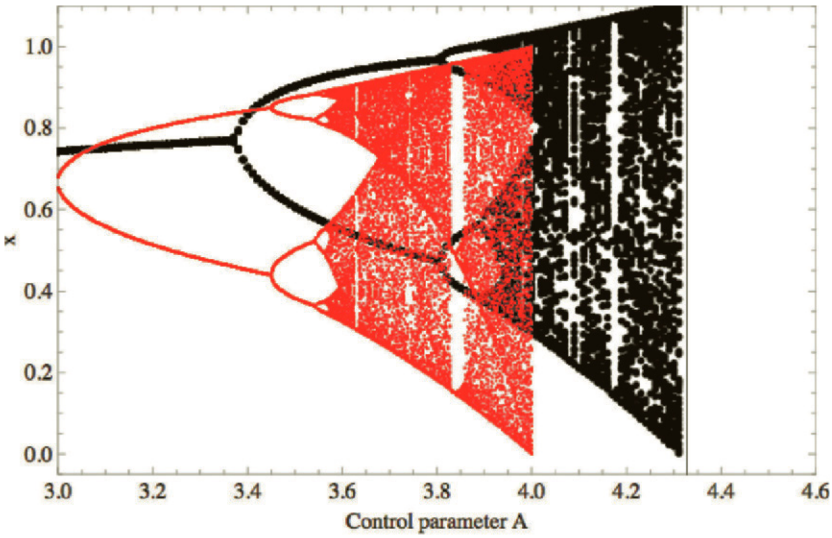

Fig. 8.8 Basically the same case as in Fig. 8.6, see eq. (8.16). Red (thin) points represent the canonical logistic equation; black (thick) points represent the synthesized system.

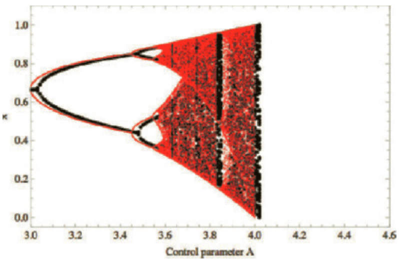

Fig. 8.9 Another similar solution.

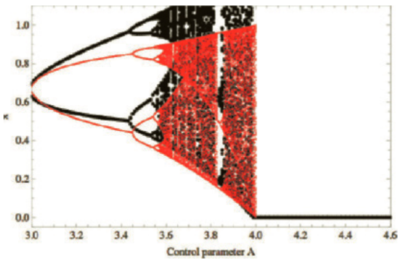

Fig. 8.10 Another similar solution.

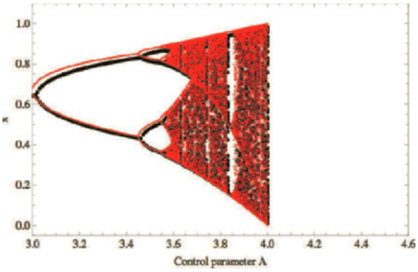

Fig. 8.11 Another similar solution.

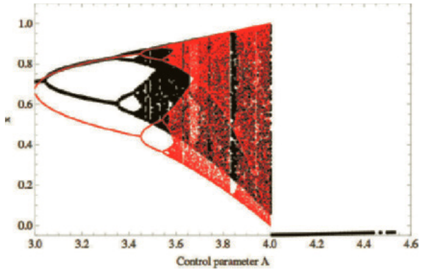

Fig. 8.12 Another similar solution.

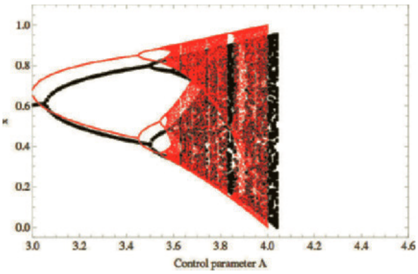

Fig. 8.13 Another similar solution.

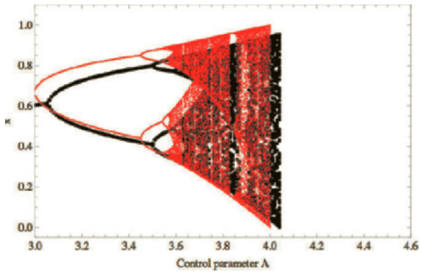

Fig. 8.14 Another similar solution.

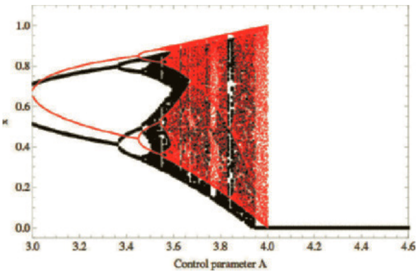

Fig. 8.15 Another similar solution.

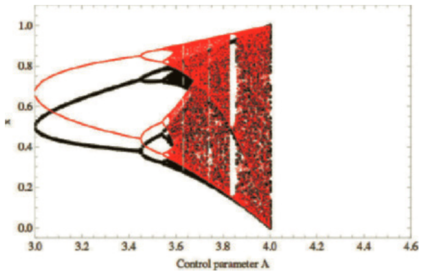

Fig. 8.16 Another similar solution.

8.4.6 Unfinished Evolution

During all simulations conducted, it been observed, that in many cases evolution would certainly need longer time to finish successfully the evolutionary reconstruction, like exact or similar reconstruction. Lets take a look on Fig. 8.17 - 8.24, or/and on eq. 8.17 - eq. 8.22. On the figures are depicted bifurcation diagrams, which are very similar to diagrams from logistic equation, they are only shifted along the x or/and y axes. It is clear that if evolution would run for a longer time, then the bifurcation diagrams (or better, the mathematical description in the background), like on

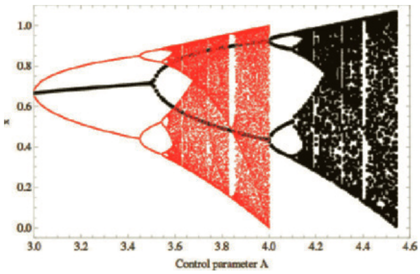

Fig. 8.17 Unfinished solution, eq. 8.17.

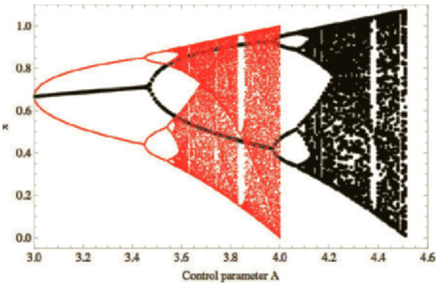

Fig. 8.18 Unfinished solution, eq. 8.18.

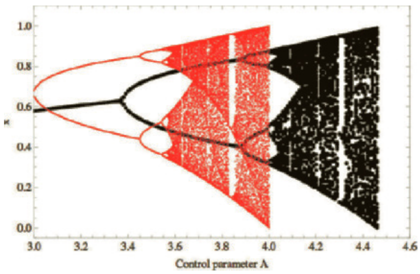

Fig. 8.19 Unfinished solution, eq. 8.19.

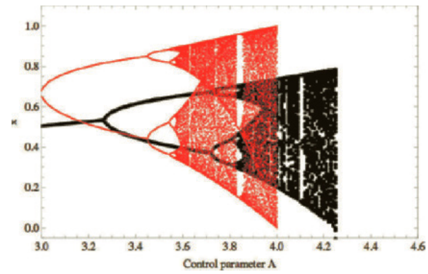

Fig. 8.20 Unfinished solution, eq. 8.20.

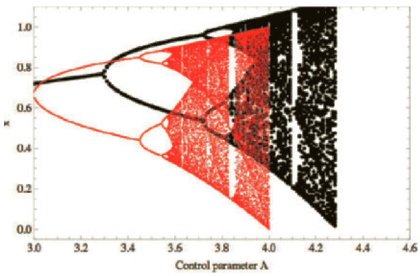

Fig. 8.21 Unfinished solution, eq. 8.21.

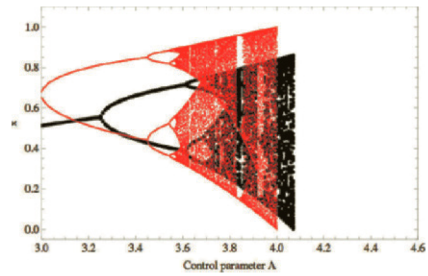

Fig. 8.22 Unfinished solution, eq. 8.22.

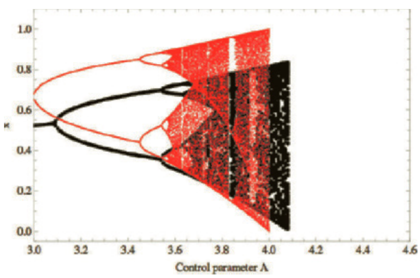

Fig. 8.23 Unfinished solution.

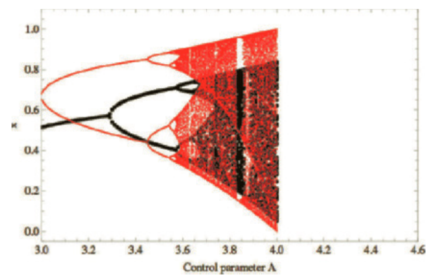

Fig. 8.24 Unfinished solution.

Fig. 8.17 - 8.24, would be better adapted to the identified one. Based on this, one can say that the above mentioned “similar” reconstruction are unfinished reconstruction with possibly very good quality (i.e. with low cost value).

$$x\left(-x(A-x)+\frac{x}{A}+A-x\right) \tag{8.17}$$

$$\frac{x\left(x-\left(\frac{A}{x}-A\right)(x-A)\right)\left(\frac{x}{(x-A)\left(-A+x^2-4x\right)}+x\right)}{A} \tag{8.18}$$

$$x\left(-x\left(-\frac{2x-Ax}{A\left(A^2-A+x\right)}+A-x\right)+A-x\right) \tag{8.19}$$

$$\frac{A\left(\frac{A}{x}+x\right)}{\frac{A\left(\frac{A}{A-x\left(\frac{x\left(\frac{x}{A}+A\right)}{2A+x}+A+x\right)}+x\right)}{x^2}+x} \tag{8.20}$$

$$-\frac{x(A+2x)\left(\frac{\frac{x}{A}-A-x}{\frac{x}{A}+A}+x\right)}{\frac{x}{A}+\frac{1}{A}+1} \tag{8.21}$$

$$x\left(\frac{\left(-Ax-\frac{x-A}{A}-A+x^2+x-1\right)(A(Ax+x)+A)}{2A^2}+A\right) \tag{8.22}$$

8.4.7 *Exotic Solutions*

Together with acceptable systems, other systems were also synthesized, which did not fit the threshold, mentioned in the section Cost function, i.e. its cost value was > 1500. This category is termed “exotic”, i.e. systems that are very different from the logistic equation (by behavior and mathematical description), however, there is still visible a similar structure to the logistic equation. An example can be the systems given by eq. (8.23) and eq. (8.24), which had been synthesized during all 1300 experiments. For behavior of systems eq. (8.23) see Fig. 8.25, and for eq. (8.23) see Fig. 8.26. Another selected example is depicted in Fig. 8.27.

$$\frac{A-x+x^2-\frac{(A-x)x^3\left(-x+x^2\right)\left(x+A\left(2A+A^2+x\right)\right)}{A\left(1-\frac{A}{x}\right)}}{A} \tag{8.23}$$

$$\frac{x\left(2A-A\left(-x-\frac{x^2}{A}\right)\left(-2x+\frac{x}{A-\frac{1+2x}{x^2}}\right)\right)}{A} \tag{8.24}$$

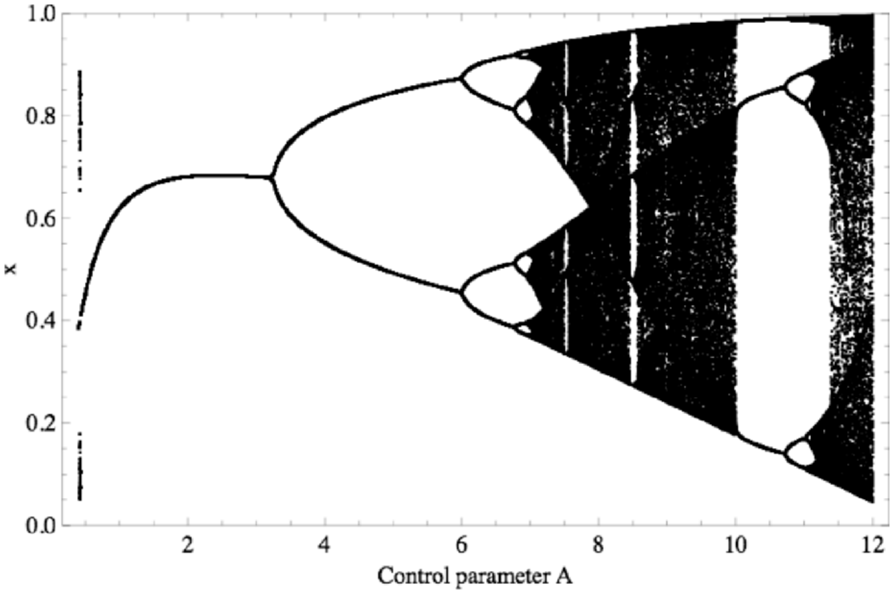

Fig. 8.25 Example of "exotic" solution, see eq. (8.23)

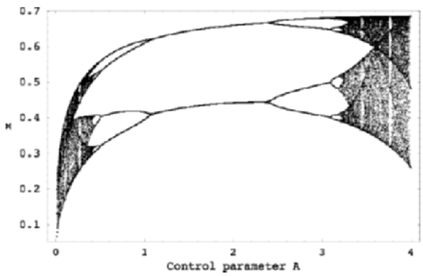

Fig. 8.26 Another "exotic" solution, see eq. (8.24).

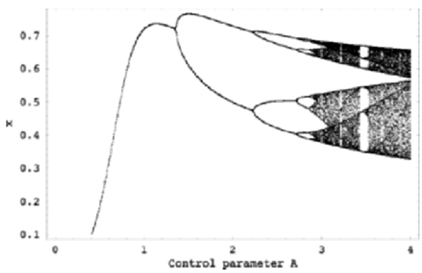

Fig. 8.27 Bifurcation diagram of Another "exotic" solution.

8.4.8 Continuous Systems: Preliminary Study

Evolutionary reconstruction of chaotic systems is certainly not restricted only to discrete systems. Methods of symbolic regression is general enough to be used also on reconstruction of continuous chaotic systems. To check this idea, the well known chaotic system has been selected - Lorenz equation, see eq. (8.25). To simplify this experiment for the first time, the third equation $\dot{z}$ has been selected to be synthesized, see eq. (8.25). Basic set of objects used in symbolic regression was $\{x(t), y(t), z(t), +, -, \times, /\}$. Total number of simulation has been set to 100 and 5 algorithms (DE, SOMA, GA, SA, ES) in all 12 versions were used in order to identify (reconstruct) by synthesis suitable solutions. In many cases the exact form of eq. $\dot{z}(t) = (x(t)y(t) - z(t))$ has been synthesized, see eq. 8.25. The remaining synthesized forms were of different form, see Table 8.18. Cost function was defined by eq. (8.26), as the difference between behavior of the original and identified system been calculated in the interval $t \in [0, 20]$ with randomly selected initial conditions. Cost value has been calculated in the interval $t \in [5, 20]$. Objective was to minimize this function to 0.

8.4.8.1 Experiment Setup

Four versions of SOMA, six versions of DE, one version of GA, and one of ES have been applied in order with AP and were used for all simulations. In Table 8.11 - Table 8.15 abbreviations of used algorithms and their setting is described. Parameters for the optimizing algorithm were set up in such a way as to reach approximately the same value of maximal cost function evaluations for all used versions. Each version of EAs has been applied 100× in order to synthesize an appropriate structure which can serve as models of the observed chaotic system.

Table 8.11 Algorithms abbreviation

Algorithm	Version	Abbreviation
SOMA	AllToAllAdaptive	S1
	AllToAll	S2
	AllToOne	S3
	AllToOneRandomly	S4
Differential Evolution	DERand1Bin	D1
	DERand2Bin	D2
	DEBest2Bin	D3
	DELocalToBest	D4
	DEBest1JIter	D5
	DERand1DIter	D6
Genetic Algorithm		G
Evolutionary strategies $(\mu + \lambda)$		ES2

Table 8.12 SOMA setting for 4 basic search strategies: S1, S2, S3 and S4

Algorithm	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	100	100	100	100
Migrations	8	8	4	4
MinDiv	-0.1	-0.1	-0.1	-0.1
Individual Length	20	20	20	20

Table 8.13 DE setting for 6 basic search strategies: D1, D2, D3, D4, D5 and D6

Algorithm	D1 - D6
NP	100
F	0.9
CR	0.3
Generations	500
Individual Length	20

Table 8.14 GA setting for canonical version of GA: G

Algorithm	G
PopSize	100
Mutation	0.4
Generations	500
Individual Length	20

Table 8.15 ES setting for search strategies: ES

Algorithm	ES2
μ ,λ	100
σ	0.8
Iterations	500
Individual Length	20

8.4.8.2 Continuous Systems: Results

Results of this case study are depicted in Fig. 8.28 - 8.31 and Tables 8.16 - 8.18. Tables 8.16 - 8.17 refer to the number of cost function evaluations that has been used by EAs to obtain suitable solution. It is graphically reported in Fig. 8.28. Fig. 8.29 and 8.30 depicts two selected histograms of 15 in total to show typical performance of selected algorithms. The last figure 8.31 depict the success of used algorithms, i.e. how many times each algorithm fails or succeeds.

$$\begin{aligned} \dot{x}(t) &= -a(x(t) - y(t)) \\ \dot{y}(t) &= bx(t) - x(t)z(t) - y(t) \\ \dot{z}(t) &= identified\ part\ by\ EAs,\ see\ Table\ 8.18 \end{aligned} \tag{8.25}$$

$$CV = \sum_{t=0}^{t=20} \left|x_{t,Lorenz} - x_{t,Sythesized}\right| + \left|y_{t,Lorenz} - y_{t,Sythesized}\right| + \left|z_{t,Lorenz} - z_{t,Sythesized}\right| \tag{8.26}$$

It is clear that this approach is also usable, i.e. it can be used to synthesize continuous systems, however more extensive study is needed.

Table 8.16 Experiment summarization, continuous case, part 1.

Algorithm	D1	D2	D3	D4	D5	D6	ES2
Cost function evaluations							
see Fig. 8.28							
Minimum	25	16	14	74	35	11	32
Average	8601	8791	12010	13718	11787	11413	10516
Maximum	50870	39752	98324	57256	44956	50274	39602

Table 8.17 Experiment summarization, continuous case, part 2.

Algorithm	G	S1	S2	S3	S4
Cost function evaluations					
see Fig. 8.28					
Minimum	1	9	1048	61	59
Average	6516	11513	63907	12272	13246
Maximum	39043	79014	349478	51672	43202

Table 8.18 Exact / non-exact reconstruction.

Solution No.	$\dot{z}(t)$ reconstruction	Exact	Non-exact
1	$x(t)y(t) - z(t)$	1090	-
2	$y(t)\left(x(t) - \frac{z(t)}{y(t)}\right)$	-	58
3	$y(t)\left(x(t) - \frac{x(t)}{y(t)}\right) + x(t) - z(t)$	-	2
4	$y(t)(x(t) + z(t)) - y(t)z(t) - z(t)$	-	4
5	$x(t)(x(t) + y(t)) - x(t)^2 - z(t)$	-	3
6	$x(t)(y(t) - x(t)) + x(t)^2 - z(t)$	-	2
7	$-y(t)\left(\frac{z(t)}{y(t)} - x(t)\right)$	-	5
8	$y(t)\left(x(t) - \frac{x(t)+z(t)}{y(t)}\right) + x(t)$	-	2
9	$-y(t)\left(\frac{z(t)-x(t)}{y(t)} - x(t)\right) - x(t)$	-	1
10	$y(t)\left(\frac{x(t)}{y(t)} + x(t)\right) - x(t) - z(t)$	-	2
11	$y(t)(x(t) - y(t)) + y(t)^2 - z(t)$	-	1
12	$(x(t) - 1)y(t) + y(t) - z(t)$	-	3
13	$x(t)(y(t) - 1) + x(t) - z(t)$	-	2
14	$-(1 - x(t))y(t) + y(t) - z(t)$	-	1
15	$y(t)\left(x(t) - \frac{y(t)+z(t)}{y(t)}\right) + y(t)$	-	1
16	$(1 - x(t))y(t) + 2x(t)y(t) - y(t) - z(t)$	-	1
17	$(x(t) + 1)y(t) - y(t) - z(t)$	-	3
18	$y(t)(x(t) - z(t)) + y(t)z(t) - z(t)$	-	1
19	$x(t)(y(t) + z(t)) - x(t)z(t) - z(t)$	-	1
20	$x(t)(y(t) + 1) - x(t) - z(t)$	-	1
21	$y(t)(x(t) - y(t) - z(t)) + y(t)(y(t) + z(t)) - z(t)$	-	1
Total		**1080**	**95**

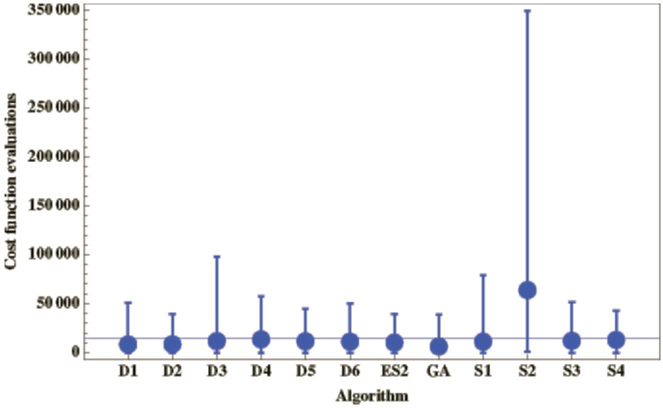

Fig. 8.28 Cost function evaluations. Thick dots are average values for each algorithm, horizontal line is average of all.

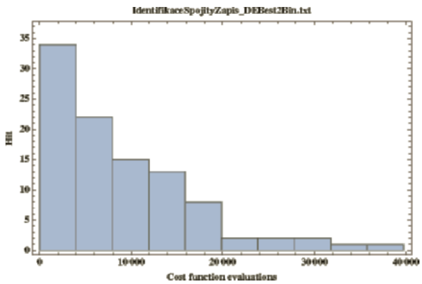

Fig. 8.29 Histogram for differential evolution algorithm, version DEBest2Bin.

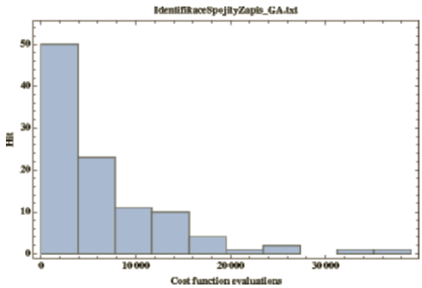

Fig. 8.30 Histogram for genetic algorithm.

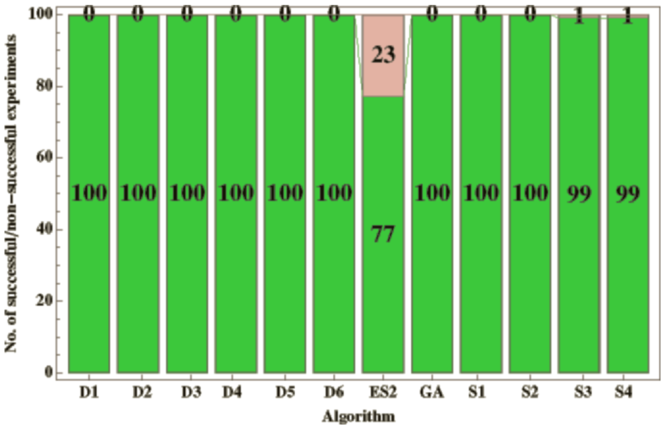

Fig. 8.31 No. of successful/non-successful reconstruction. Each bar is divided into two parts. The upper part represent number of non-successful reconstruction, the lower one successful reconstruction.

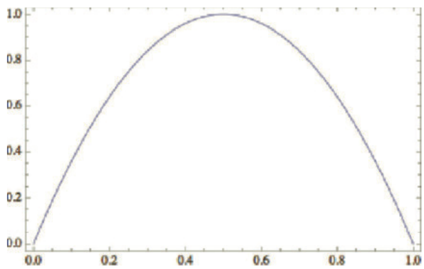

Fig. 8.32 Graph of the first part of eq. (8.27).

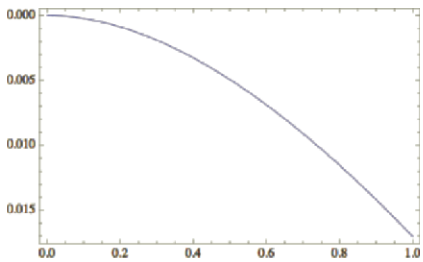

Fig. 8.33 Graph of the second part of eq. (8.27).

8.5 Conclusion

Based on recorded data and results, it may be stated that the simulations provided promising results, which shows that EAs are capable of model reconstruction of chaotic systems. In this chapter, five evolutionary algorithms in 13 (12 for continuous case) versions were used and tested. Exact descriptions of the identified systems (logistic equation, Lorenz system) as well as its variations have been identified from the results (see for example eq. (8.13), or Table 8.18). The question is why such complex equation like eq. (8.13) have similar behavior to eq. (8.4). The answer is simple. After expansion of eq. (8.13), equation (8.27) is obtained. The first part is basically the logistic equation (see Fig. 8.32). The remaining part participates on the final behavior without significant impact (see Fig. 8.33).

$$Ax - Ax^2 + \frac{x^2}{A^2\,(2A - x - Ax)} - \frac{x^2}{A\,(2A - x - Ax)} \tag{8.27}$$

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

- **Experiment overview.** The cost function (8.5) 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 1300 experiments were conducted, 39 000 000 data-points and 13 000 000 of these points were used for the evaluation of all synthesized systems. The continuous case has similar behavior, see 8.26. This cost function calculates the difference between original behavior of Lorenz system and the just identified one in the time interval [0,20].
- **Number of successful reconstruction.** The results were divided into three categories for discrete system: exact, similar and exotic reconstructions. The representation is as follows: exact implies that logistic equation has been recovered in its canonical version (or its algorithmic variations), similar means that behavior 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 eq. (8.13), (8.14), (8.15) and eq. (8.16)). Exotic

reconstruction 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 73 times (see Table 8.8). Number of synthesized similar systems was 186. For general overview see Table 8.8.

In Table 8.18 results from continuous case are displayed. Exact description has been identified 1090× (see solution 1), another, say equivalent descriptions 58× (solution 2) etc. It is important to note that the behavior of Lorenz attractor was identical only in the interval $t \in [0,20]$. Outside this interval behavior of synthesized "Lorenz system" has been less or more divergent, which is of course obvious.

- **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 8.2). It is caused by the different internal algorithm structure for new individuals calculations. Due to this fact, condition b) has been kept with the highest priority for the chaotic discrete system identification. In the case of the Lorenz system we were interested mainly whether this idea will work or not.
- **Behavior preciseness.** It should be noted here that reconstruction has not been focused on exact behavior reconstruction for each time development of logistic equation, but on similarity of behavior via data used later for bifurcation diagrams, i.e. difference between bifurcation diagrams has been calculated for discrete systems. 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, trajectories of synthesized systems were 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. 8.9 - 8.16, it is also visible that a little bit longer time is needed for better estimation of system description. For some identified systems it has been observed that while $A \in [3,4]$ the behavior is identical or very similar with that of logistic equation was produced, whereas other values of A (for example negative) other chaotic behavior were generated, see for example $-A+(-(1/A)+A/x-2x)x+(A-x)x$ with $A \in [0,1]$. Concerning to the identified Lorenz system, as mentioned before, in the time interval $t \in [0,20]$ the difference between original and identified Lorenz was minimal.
- **Problem complexity and algorithm performance.** Lets take into consideration only discrete system. 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 the 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 259 synthesized solutions (from 1300 in total) represents $8.51 \times 10^{-61}\%$ of such defined searched space. If we will follow maximal allowed number of cost function evaluations (see Table 8.2 -

Table 8.6, 534 808 cost function evaluations, i.e. tested solutions) then evolution searched maximally $1.74 \times 10^{-57}\%$ 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 the expected lifespan of our universe. 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 chapter, the so-called analytic programming has been used, however we have to say that another and more well known techniques like genetic programming, see [17], [18] or grammatical evolution, see [22], should give similar results as reported here.

Conclusions about preliminary and simplified study of reconstruction of Lorenz system is:

- **Number of successful reconstruction.** The number of the same reconstructed form of the $\dot{z}(t)$ is reported in the Table 8.18 and is quite large. It shows that EAs were able to reconstruct $\dot{z}(t)$ in its exact form. On the other side, in Table 8.18 it is visible that EAs also has found another similar solutions, which, in the interval $t \in [0, 20]$ has fit the behavior of reconstructed system very well. Behind this interval, the trajectories of such systems usually runs out of attractor domain.
- **Simplifications.** For preliminary study on continuous system, such simplification that only $\dot{z}(t)$ has to be reconstructed has been used. Based on the performance of used algorithms in this and other chapters, it is logical to expect, that EAs should be able to identify all three part of Lorenz system. To confirm such statement, it is however necessary to do more extensive research.

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 8.7, Table 8.8) the cost function or/and algorithm settings should be improved.

Acknowledgements. 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 102/09/1680.

References

1. Back, T., Fogel, D., Michalewicz, Z.: Handbook of Evolutionary Computation. Institute of Physics, London (1997)
2. Beyer, H.: Theory of Evolution Strategies. Springer, New York (2001)
3. Broomhead, D., King, G.: Extracting qualitative dynamics from experimental data. Physica D 20, 217 (1986)

4. Cerny, V.: Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. J. Opt. Theory Appl. 45(1), 41–51 (1985)
5. Das, S., Konar, A.: A swarm intelligence approach to the synthesis of two-dimensional IIR filters. Eng. Appl. Artif. Intell. 20(8), 1086–1096 (2007)
6. Dashora, Y., Sanjeev, K., Nagesh, S., Tiwarid, M.: Improved and generalized learning strategies for dynamically fast and statistically robust evolutionary algorithms. Eng. Appl. Artif. Intell (2007), doi:10.1016/j.engappai.2007.06.005
7. Drazin, P., Kind, G. (eds.): Interpretation of time series from nonlinear Systems. Special issue of Physica D 58 (1992)
8. Eckmann, J., Procaccia, I.: Fluctuation of Dynamical Scaling Indices in Non-Linear Systems. Phys. Rev. 34A, 659 (1986)
9. Fraedrich, K.: Estimating the dimension of weather and climate attractors. J. Atmom. Sci. 43, 419 (1986)
10. Galka, A.: Topics in nonlinear time series analysis with implications for EEG analysis. World Scientific, Singapore (2000)
11. Hwang, G.-H., Kim, D.-W., Lee, J.-H., An, Y.-J.: Design of fuzzy power system stabilizer using adaptive evolutionary algorithm. Eng. Appl. Artif. Intell. 21(1), 86–96 (2007)
12. Halsey, T., Jensen, M., Kadanoff, L., Procaccia, I., Schraiman, B.: Fractal Measures and Their Singularities: the Characterization of Strange Sets. Phys. Rev. 33A, 1141 (1986)
13. Holland, J.: Adaptation in Natural and Artificial Systems. Univ. Michigan Press, Ann Arbor (1975)
14. Johnson, C.: Artificial immune systems programming for symbolic regression. In: Ryan, C., Soule, T., Keijzer, M., Tsang, E., Poli, R., Costa, E. (eds.) EuroGP 2003. LNCS, vol. 2610, pp. 345–353. Springer, Heidelberg (2003)
15. Kantz, H., Schreiber, T.: Nonlinear time series analysis. Cambridge University Press, Cambridge (1997)
16. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science 220(4598), 671–680 (1983)
17. Koza, J.: Genetic Programming II. MIT Press, Cambridge (1998)
18. Koza, J., Bennet, F., Andre, D., Keane, M.: Genetic Programming III. Morgan Kaufmann, San Francisco (1999)
19. Koza, J., Keane, M., Streeter, M.: Evolving Inventions. Sci. Am., 40–47 (2003)
20. Liu, L., Wenxin, L., David, A.: Particle swarm optimization-based parameter identification applied to permanent magnet synchronous motors. Eng. Appl. Artif. Intell. (2007), doi:10.1016/j.engappai.2007.10.002
21. Lorenz, E.: Deterministic non-periodic flow. J. Atmos. Sci. 20, 20 (1963)
22. O'Neill, M., Ryan, C.: Grammatical Evolution. In: Evolutionary Automatic Programming in an Arbitrary Language. Kluwer Academic Publishers, Dordrecht (2002)
23. O'Sullivan, J., Conor, R.: An Investigation into the Use of Different Search Strategies with Grammatical Evolution. In: Foster, J.A., Lutton, E., Miller, J., Ryan, C., Tettamanzi, A.G.B. (eds.) EuroGP 2002. LNCS, vol. 2278, pp. 268–277. Springer, Heidelberg (2002)
24. Packard, N., Crutchfield, J., Farmer, D., Shaw, R.: Geometry from a time series. Phys. Rev. Lett. 45, 712 (1980)
25. Price, K.: An Introduction to Differential Evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 79–108. McGraw-Hill, London (1999)
26. He, Q., Wang, L.: An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng. Appl. Artif. Intell. 20(1), 89–99 (2007)
27. Richter, H.: An evolutionary algorithm for controlling chaos: The use of multi-objective fitness functions. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 308–317. Springer, Heidelberg (2002)

28. Richter, H.: A study of dynamic severity in chaotic fitness landscapes, Evolutionary Computation, 2005. The IEEE Congress 3(2), 2824–2831 (2005)
29. Richter, H.: Evolutionary Optimization in Spatio-temporal Fitness Landscapes. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN 2006. LNCS, vol. 4193, pp. 1–10. Springer, Heidelberg (2006)
30. Richter, H., Reinschke, K.: Optimization of local control of chaos by an evolutionary algorithm. Physica D 144, 309–334 (2000)
31. Ryan, C., Collins, J.J., Neill, M.O.: Grammatical Evolution: Evolving Programs for an Arbitrary Language. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds.) EuroGP 1998. LNCS, vol. 1391, p. 83. Springer, Heidelberg (1998)
32. Sauer, T., Yorke, J., Casdagli, M.: Embedology. J. Stat. Phys. 65, 579 (1991)
33. Sauer, T., Yorke, J.: How many delay coordinates do you need? Int. J. Bifurcat Chaos 3, 737 (1993)
34. Senkerik, R., Zelinka, I., Navratil, E.: Optimization of feedback control of chaos by evolutionary algorithms. In: 1st IFAC Conference on Analysis and Control of Chaotic Systems, Reims, France (2006)
35. Takens, F.: Detecting strange attractors in turbulence. Lect. Notes Math., pp. 366–381 (1981)
36. Zelinka, I.: Analytic programming by Means of Soma Algorithm. In: Proc. 8th International Conference on Soft Computing Mendel 2002, Brno, Czech Republic, pp. 93–101 (2002a)
37. Zelinka, I.: Analytic programming by Means of Soma Algorithm. In: ICICIS 2002, First International Conference on Intelligent Computing and Information Systems, Egypt, Cairo (2002b)
38. Zelinka, I.: SOMA - Self Organizing Migrating Algorithm. In: Babu, B., Onwubolu, G. (eds.) New Optimization Techniques in Engineering. Springer, Heidelberg (2004)
39. Zelinka, I.: Investigation on Realtime Deterministic Chaos Control by Means of Evolutionary Algorithms. In: 1st IFAC Conference on Analysis and Control of Chaotic Systems, Reims, France (2006)
40. Zelinka, I.: Real-time deterministic chaos control by means of selected evolutionary algorithms. Eng. Appl. Artif. Intell. (2008), doi:10.1016/j.engappai.2008.07.008
41. Zelinka, I., Oplatkova, Z.: Analytic programming - Comparative Study. In: The second International Conference on Computational Intelligence, Robotics, and Autonomous Systems, CIRAS 2003, Singapore (2003)
42. Zelinka, I., Oplatkova, Z.: Boolean Parity Function Synthesis by Means of Arbitrarry Evolutionary Algorithms - Comparative Study. In: 8th World Multiconference on Systemics, Cybernetics and Informatics, SCI 2004, Orlando, USA (2004)
43. Zelinka, I., Chen, G., Celikovsky, S.: Chaos Synthesis by Means of Evolutionary Algorithms. Int. J. Bifurcat Chaos Appl. Sci. Eng. 18(4), 911–942 (2008)
44. Zelinka, I., Senkerik, R., Navratil, E.: Investigation on Evolutionary Optimization of Chaos Control. Chaos, Solitons & Fractals (2007), doi:10.1016/j.chaos.2007.07.045