NAVY Research Group
Department of Computer Science
Faculty of Electrical Engineering and Computer Science VŠB-TUO
17. listopadu 15
708 33 Ostrava-Poruba
Czech Republic

Unconventional Algorithms in Control and Design of Complex Systems

ICUMT 2015

Ivan Zelinka

MBCS CIPT, www.bcs.org/
http://www.springer.com/series/10624

Department of Computer Science
Faculty of Electrical Engineering and Computer Science, VŠB-TUO
17. listopadu 15 , 708 33 Ostrava-Poruba
Czech Republic
www.ivanzelinka.eu

Topics

- Evolutionary algorithms and history overview
  - Evolutionary algorithms
  - Symbolic reression
- Case studies
  - Deterministic chaos control
  - Plasma reactor control
  - Control program synthesis
  - Nonlinear system synthesis
  - Electronic circuits synthesis
  - Evolution as a complex system
- Conclusion

FEI VŠB-TU

http://www.vsb.cz/en/

NAVY
http://navy.cs.vsb.cz

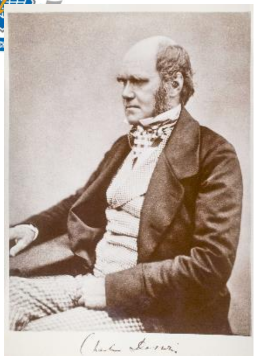
Unconventional Algorithms and Computing

Nekonvenční **a**lgoritmy a **vý**počty - **NAVY**

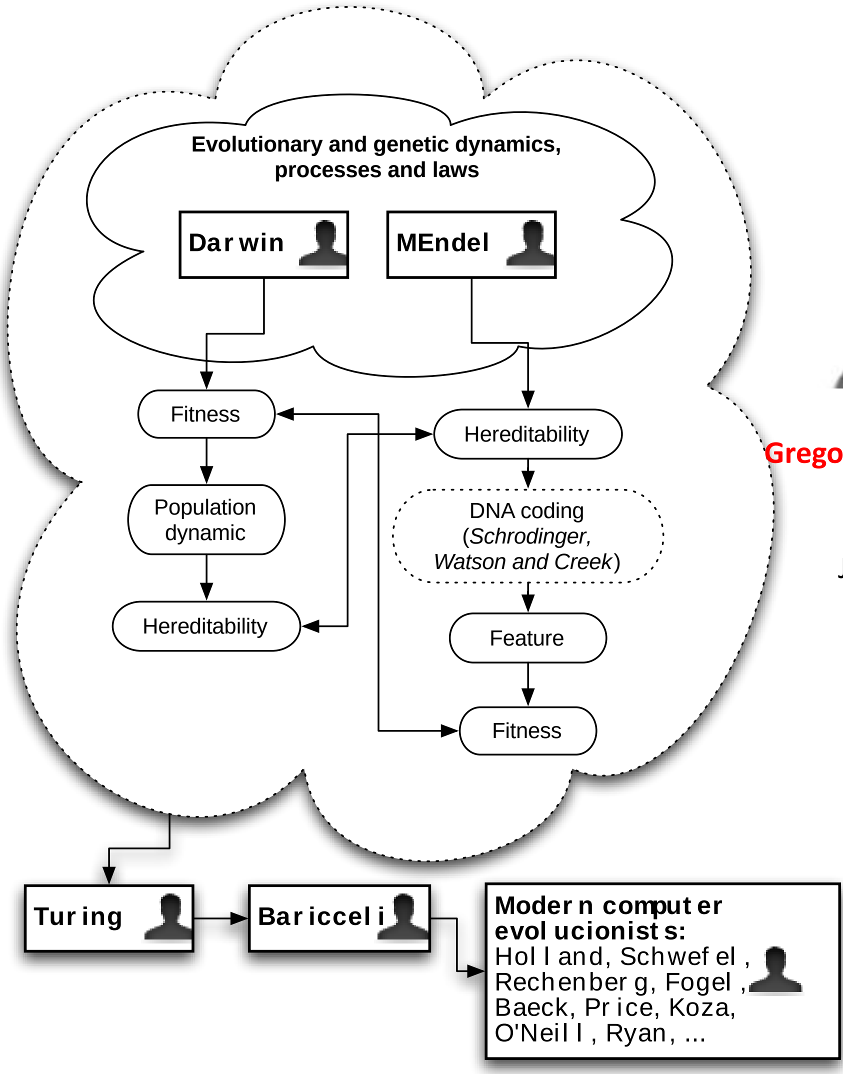
Home

- Home
- About
- Teaching
- Research
- Collaboration
- Projects
- Members
- For Students
- Contact

Homepage of research group at Faculty of Electrical Engineering and Computer Science, Department of Computer Science, VSB - Technical University of Ostrava

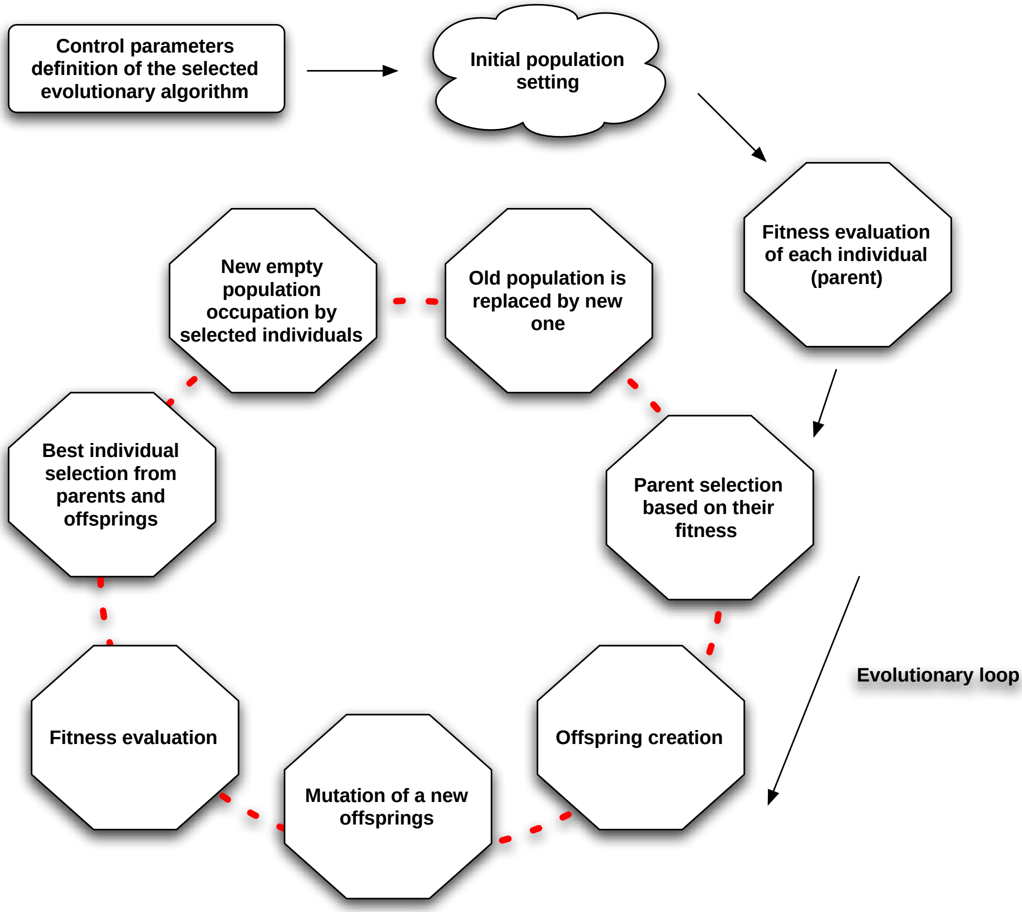


Gregor Charles Darwin 12 February 1809 – 19 April 1882.



Gregor Johann Mendel July 20, 1822 – January 6, 1884.

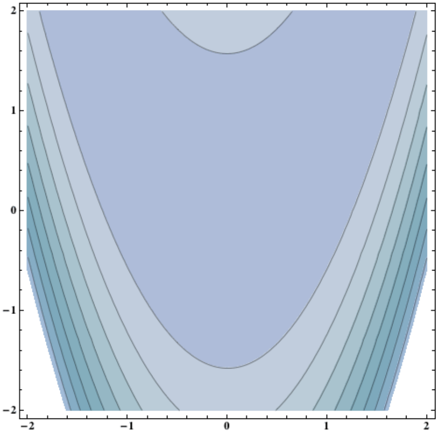
Evolution – the Central Dogma

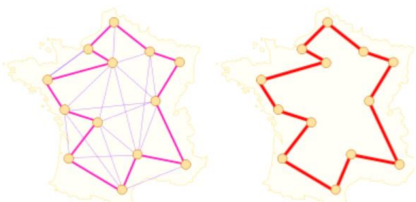


From the above mentioned main ideas of Darwin and Mendel theory of evolution, ECT uses some building blocks, see the diagram.

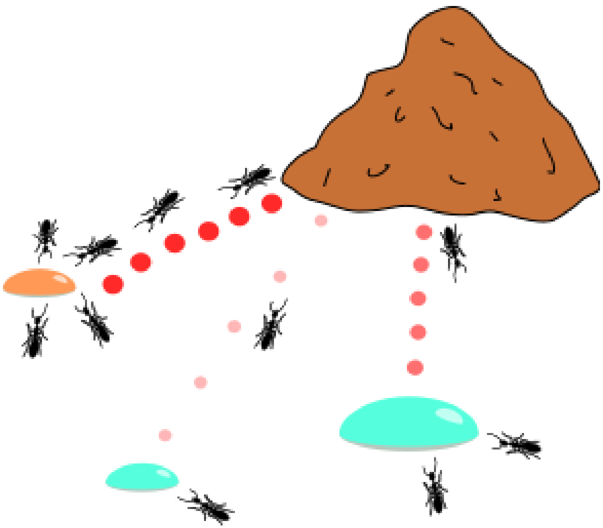
The evolutionary principles are transferred into computational methods in a simplified form that will be outlined now.

Examples





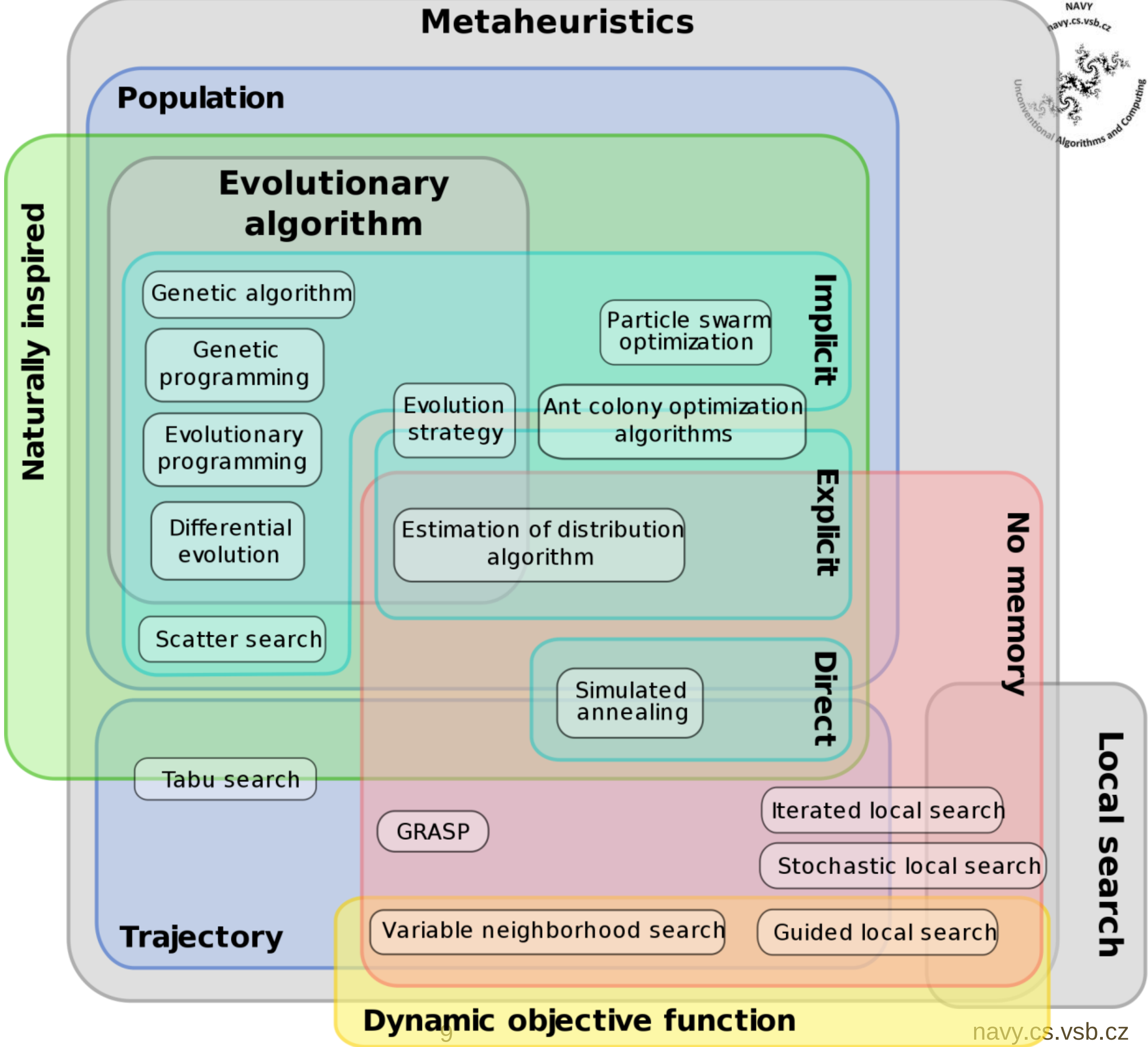
3 4



Particle Swarm

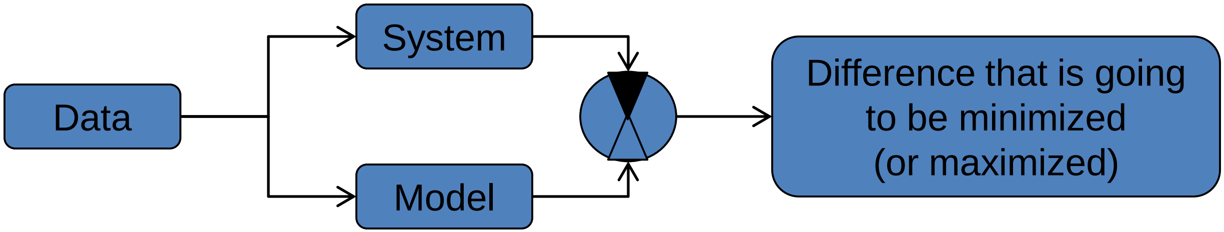
- Xboids: https://www.youtube.com/watch?v=M028vafB0l8
- Robot Swarm driven by Particle Swarm Optimization algorithm: https://www.youtube.com/watch?v=RLIA1EKfSys

Nature Inspired Computation



The Objective Function

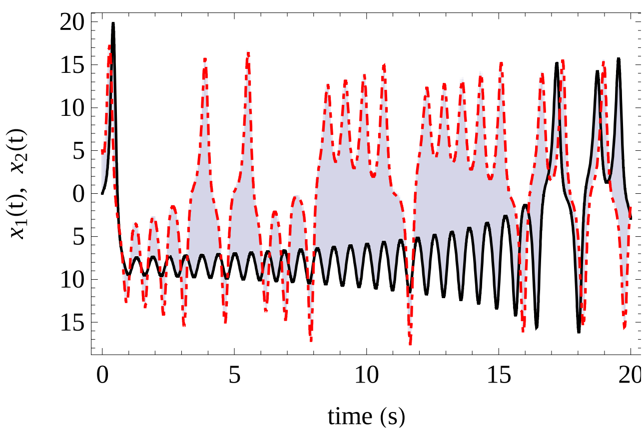
- The principle of cost function is that behavior of model has to be minimized in order to find all *N* parameters exactly.
- Model is defined by user (or by evolution).
- Data are generated artificially or better, they come from real world.

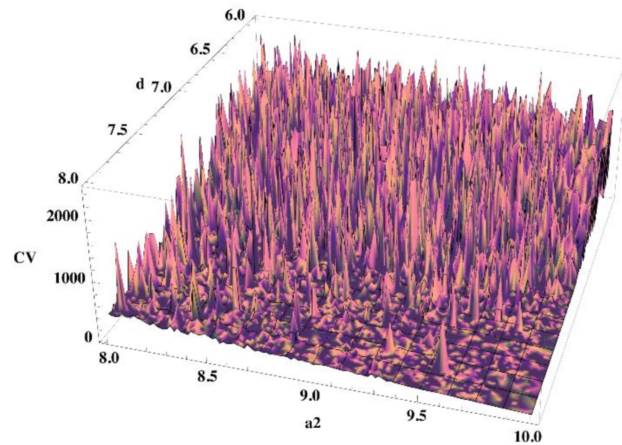


The Objective Function
Chaos Synchronization

- The cost function of the Lorenz - Lorenz (LL) synchronization is

$$CV_{LL}(a_2, d) = \int_0^{100} |x_1(t) - x_2(t)| + |y_1(t) - y_2(t)| + |z_1(t) - z_2(t)| \, \mathrm{d}t.$$

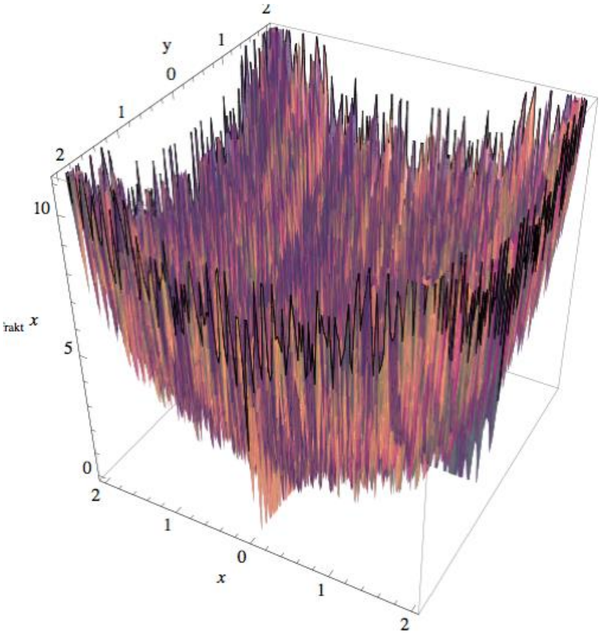




- Landscape of the synchronization cost function CV_{LL} depending on the parameters a_2 and d.

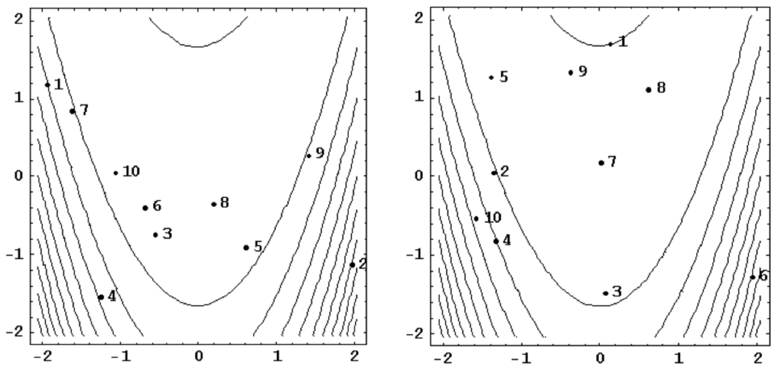
The Applicability of Evolutionary Algorithms

- The fractal function – Weierstrass non-differentiable function added on the 1st De Jong.



Population

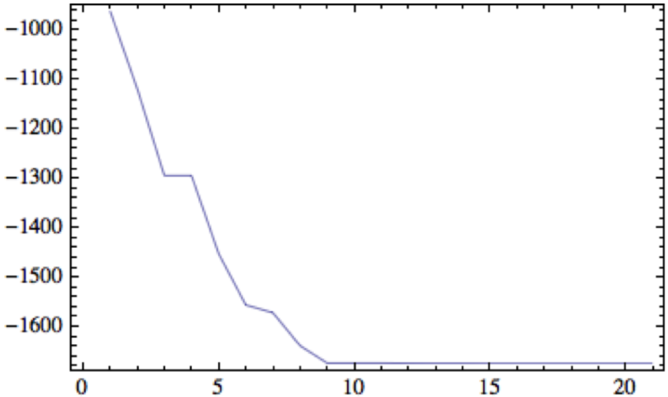
- This relationship ensures that all parameters of individuals are randomly generated within the permitted limits of the space of possible solutions.
- Sample of the population can be seen in figures below:

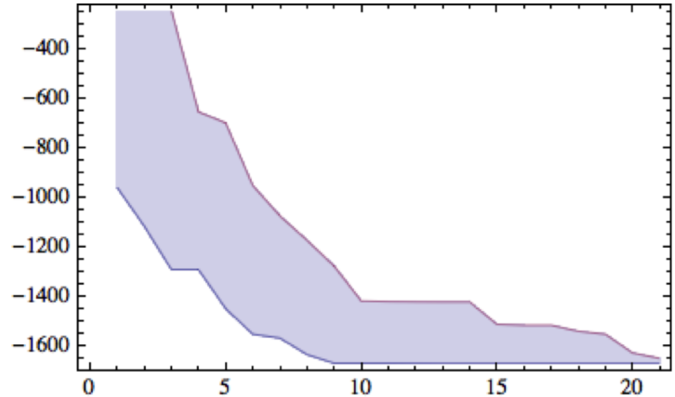


J_1	J_2	J_3	J_4	…	J_{10}
424	**104**	**53,3**	**942.9**	**…**	**178**
-1,8	-1	0,7	-1,25	…	-1,19
1,2	2	1,2	-1,5	…	0,1

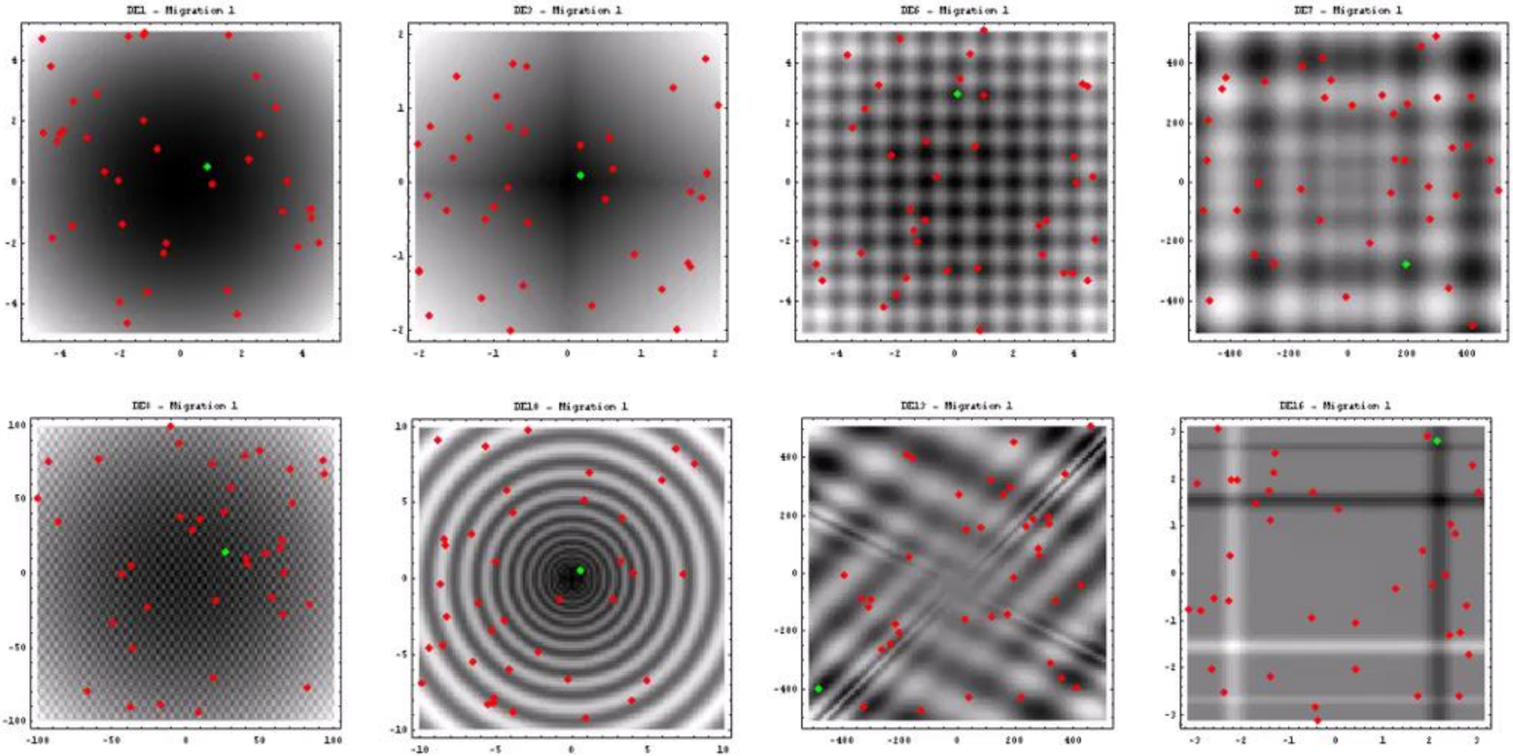
Population

- History of the best individual (or best individuals in repeated simulations)
- History of the worst individual
- Overall view of the convergence of the population

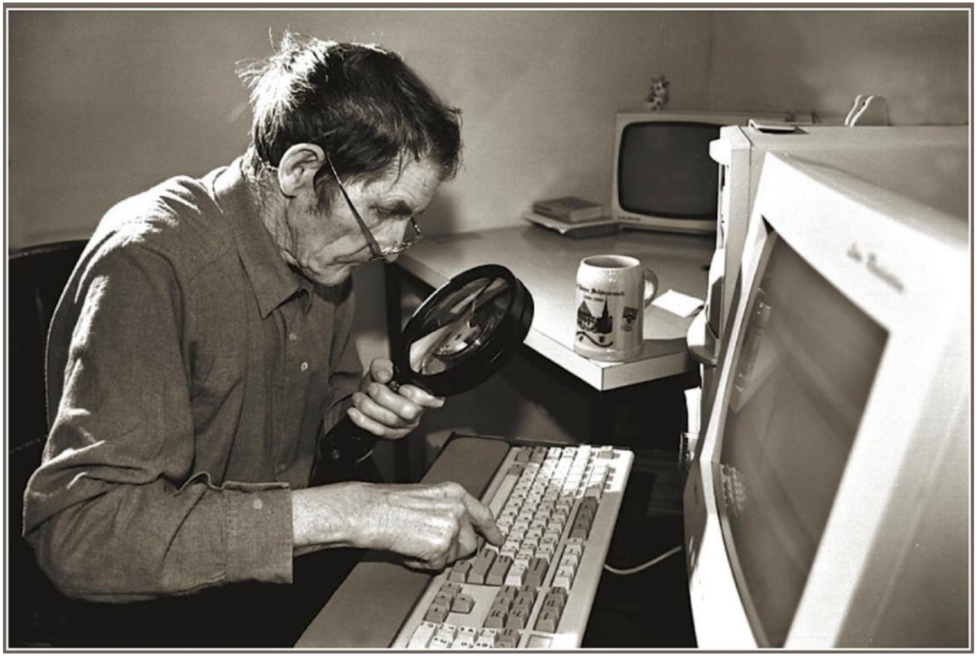




Population Dynamics



Algorithms



SOMA
Video

Differential Evolution

Pseudocode

$$v_j = x^G_{r3,j} + F\left(x^G_{r1,j} - x^G_{r2,j}\right)$$

Parameters for DE		
Dimension	D	6
Population size	NP	7
Mutation constant	F	0.8
Crossover	CR	0.5

Active individual

Three randomly chosen individuals

	Individual 1	Individual 2	Individual 3	Individual 4	Individual 5	Individual 6	Individual 7
Cost value	3.6944074	79.1015763	57.453647	3.16198009	3.5514714	12.432604	0.3474672
Parameter 1	8.0533106	71.335444	17.111268	4.14566955	13.737595	61.638486	57.332534
Parameter 2	9.2498415	5.49047646	42.776854	25.37298	65.47013	10.231425	17.186136
Parameter 3	1.1239946	6.77417004	16.048754	46.0285357	50.738214	47.074762	0.0349505
Parameter 4	10.187627	0.24863381	10.342385	29.3258786	16.036278	43.762838	17.424359
Parameter 5	9.7273059	2.31600768	0.6998136	33.5472858	34.792886	32.012036	71.870571
Parameter 6	11.294207	18.2332446	76.247148	3.24796669	5.103281	0.2021001	17.475226

-

Differential vector		Weighted differential vector		Noisy vector
-5.684284		-4.5474272		52.785107
-56.220288		-44.976231		-27.79009
-49.614219	* F	-39.691376	+	-39.65642
-5.8486509		-4.6789207		12.745438
-25.06558		-20.052464		51.818106
6.19092626		4.95274101		22.427967

	Trial vector
CV	14.9451594
Parameter 1	52.7851073
Parameter 2	-27.790094
Parameter 3	6.77417004
Parameter 4	12.7454379
Parameter 5	2.31600768
Parameter 6	18.2332446

Based on CR are parameters chosen from actual or noisy vector

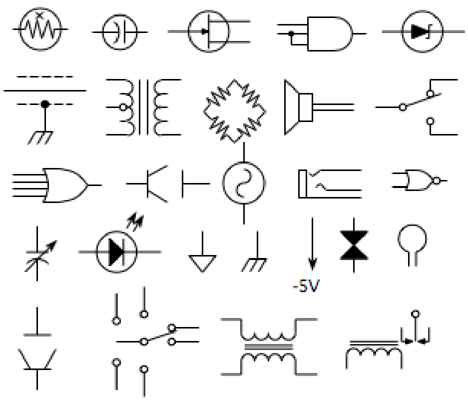
The best individual of both take place in new population

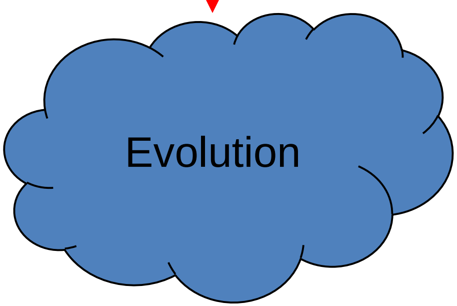
	Individual 1	Individual 2	Individual 3	Individual 4	Individual 5	Individual 6	Individual 7
CV	1.6147656	14.9451594					
Parameter 1	5.9284987	52.7851073					
Parameter 2	11.653044	-27.790094					
Parameter 3	30.56767	6.77417004					
Parameter 4	67.605951	12.7454379					
Parameter 5	45.300423	2.31600768					
Parameter 6	18.868377	18.2332446					

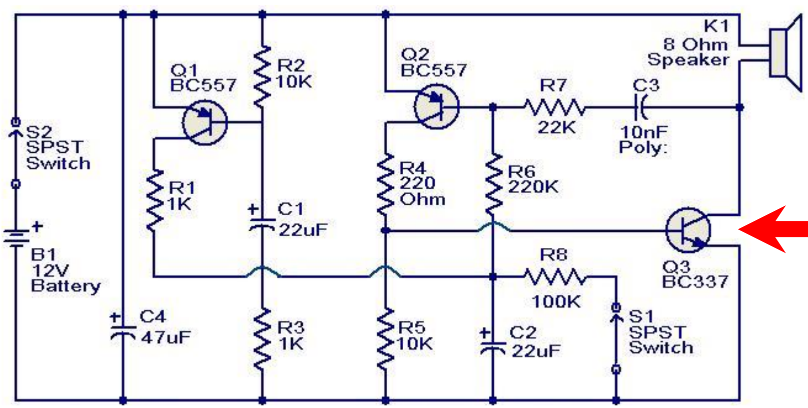
Evolution of Symbolic Structures

Brief Overview

Evolutionary manipulation with simple predefined objects essential for the synthesis of more complex structures which satisfy the predetermined conditions. As an example electronic circuit can be used.



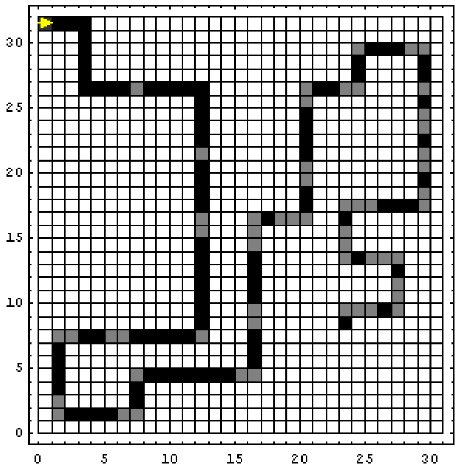


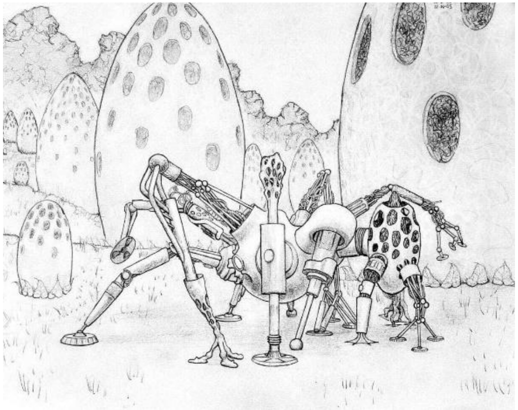


Evolution of Symbolic Structures
Brief Overview

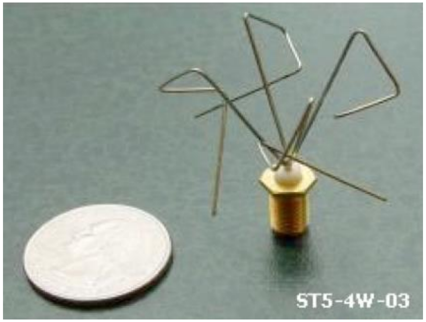
Examples:

- Robot control program
- Antenna
- Controller for feedback control
- ...





http://www.nelsonrobotics.org/evolutionary_robotics_web/links.html



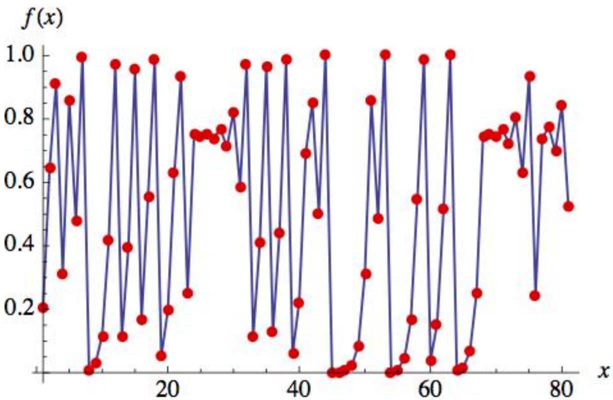
Deterministic Chaos Control
Logistic Equation

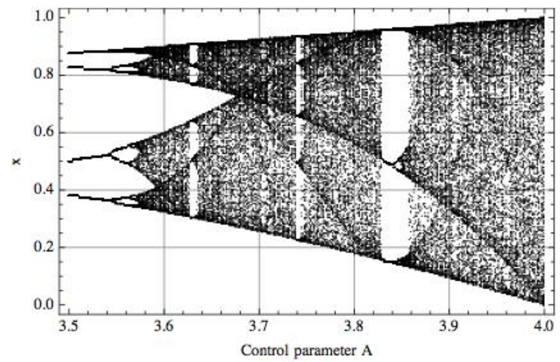
- Used logistic equation

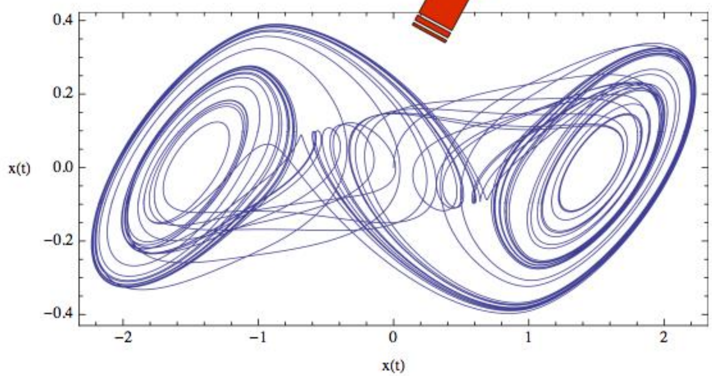
$$x_{n+1} = rx_n(1 - x_n) + p$$

- Unperturbed map (p = 0) with r = 3.8, studied UPOs:
- p-1: $x_F = 0.7384$
- p-2: $x_1 = 0.3737$, $x_2 = 0.8894$
- and higher periodic orbits: p-4, p-8

Deterministic Chaos Control
What is Deterministic Chaos?



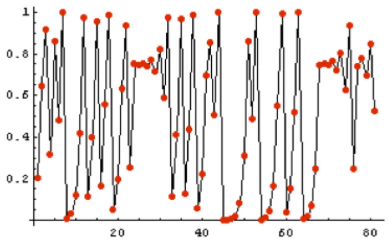




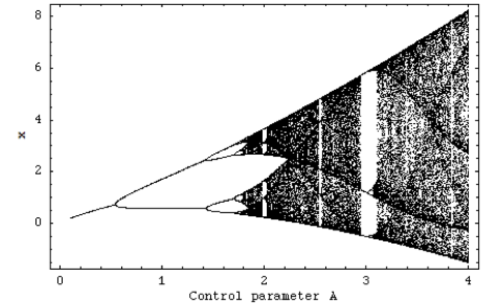
Deterministic Chaos Control

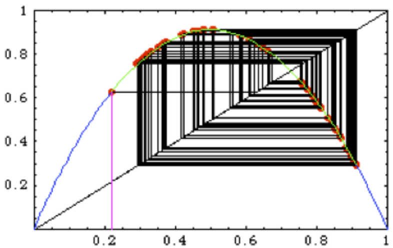
What is Deterministic Chaos - Visualization

... time view

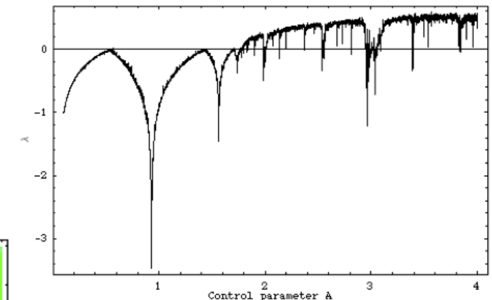


$$\frac{x-A\left(A-x-2x^2\right)}{-A-x+Ax^2-A\left(-A+\frac{A^3}{x}+2x\right)}$$

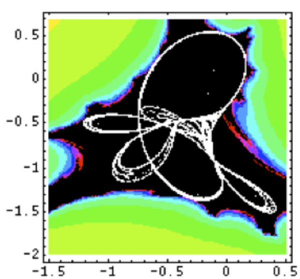


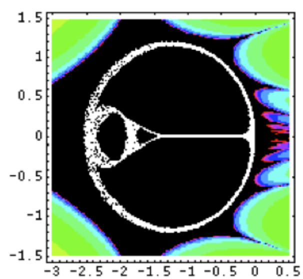


... CobWeb diagram



... Ljapunov exponent

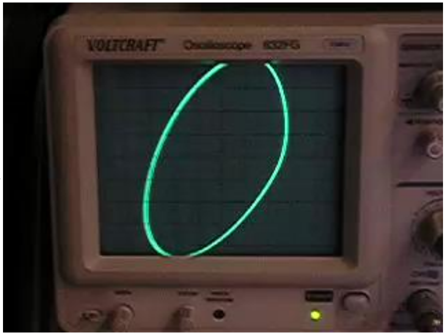




... atrtactor and basin of attraction

Deterministic Chaos Control
What is Deterministic Chaos?

- Mechanical systems (billiards, inverse pendulum ...)
- Astrophysical systems (ternary systems, plasma ...)
- Biological systems (predator-prey, DNA)
- Chemical processes (chemical oscillators - BZ reaction ...)
- Physical processes (plasma, atmospheric happening ...)
- Economic systems (stock exchange, economic cycles ...)
- Information systems (genetic algorithms, ...)
- Energetic and electronic systems (Chua's circuit, ...)
- ...



Deterministic Chaos Control
Control Method

Original Pyragas's TDAS:

$$\frac{dx}{dt} = P(x) + F(t)$$
$$F(t) = K\left[x(t-\tau) - x(t)\right]$$

Original Pyragas's ETDAS:

$$\frac{dx}{dt} = P(x) + F(t)$$
$$F(t) = K\left[(1-R)S(t-\tau) - x(t)\right]$$
$$S(t) = x(t) + RS(t-\tau)$$

Discrete modification for Henon map:

$$x_{n+1} = rx_n(1-x_n) + F_n$$
$$F_n = K\left[x_{n-m} - x_n\right]$$

$$x_{n+1} = rx_n(1-x_n) + F_n$$
$$F_n = K\left[(1-R)S_{n-m} - x_n\right]$$
$$S_n = x_n + RS_{n-m}$$

$$-F_{\max} \le F_n \le F_{\max}$$

Deterministic Chaos Control
Optimizing Algorithms

All optimizations were performed by evolutionary algorithms SOMA and DE

Used SOMA versions

Index	Algorithm / Version
1	SOMA AllToOne
2	SOMA AllToRandom
3	SOMA AllToAll
4	SOMA AllToAllAdaptive

Parameter settings

Parameter	ATO / ATR	ATA / ATAA
PathLength	3	3
Step	0.33	0.33
PRT	0.1	0.1
PopSize	25	10
Migrations	25	7
CF Evaluations (CFE)	5400	5670

Optimized EDTAS parameters: $-5 \leq K \leq 5$, $0 \leq F_{max} \leq 0.5$ and $0 \leq R \leq 1$

Deterministic Chaos Control
Cost Function – p-1 Orbit

Basic proposal

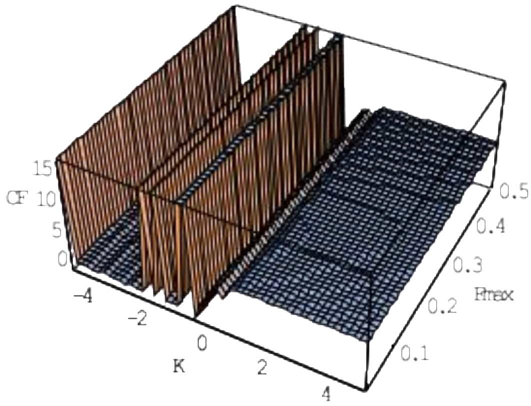
$$f_{\cos t} = \sum_{t=0}^{\tau} \left| TS_t - AS_t \right|$$

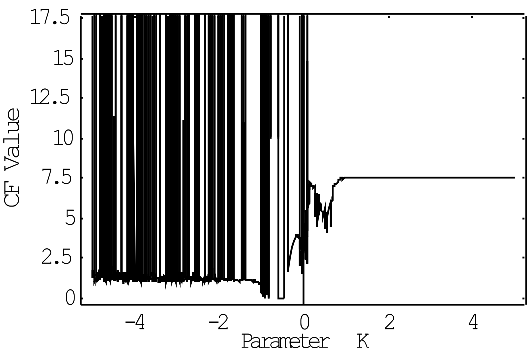
Where: TS – target state, AS – actual state, τ – simulation interval

Advanced proposal

$$f_{\cos t} = \sum_{t=\tau 1}^{\tau 2} \left| TS_t - AS_t \right|$$

Where: τ_1 – the first min. value of difference between TS and AS
τ_2 - short time interval (τ_1+ 20 iterations)





Deterministic Chaos Control
Cost Function – Higher Periodic Orbits

$$f_{\cos t} = penalization1 + \sum_{t=\tau 1}^{\tau 2} \left| TS_t - AS_t \right|$$

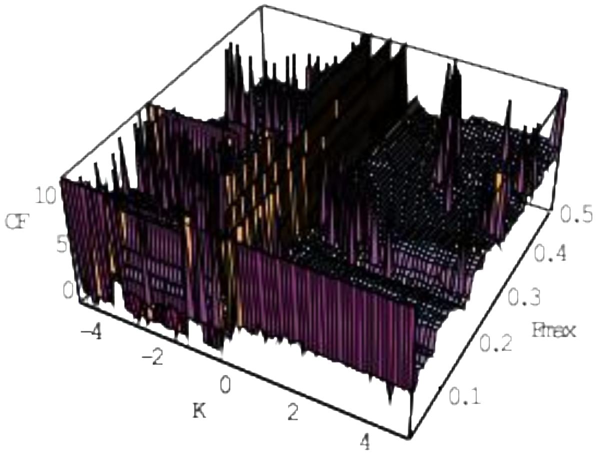
Where: TS – target state, AS – actual state

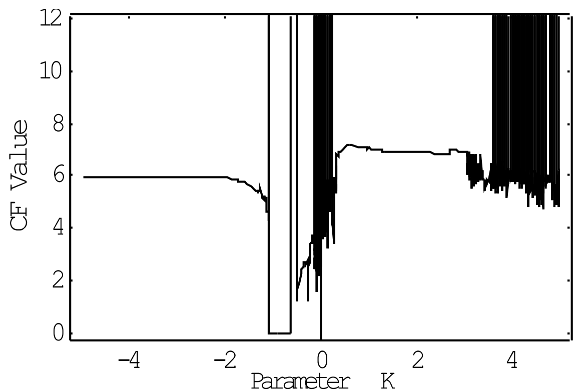
τ_1 – the first min. value of difference between TS and AS

τ_2 - short time interval τ_1+ τs (τs = 20 iterations)

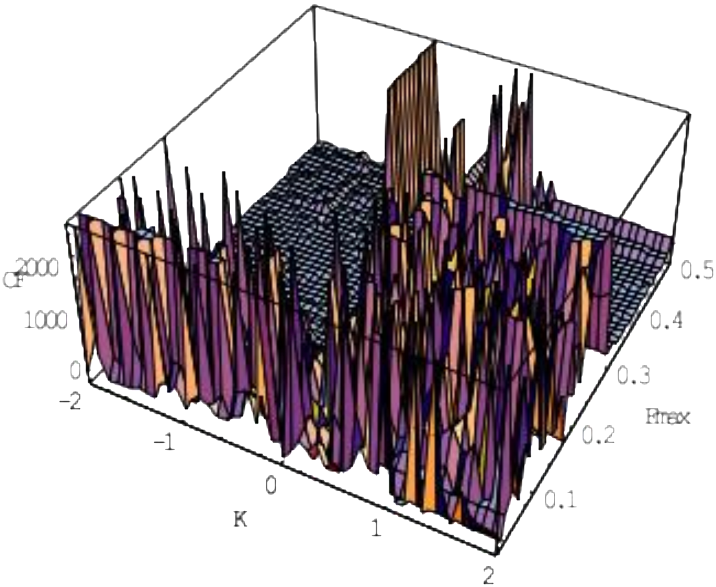
penalization1= 0 if $\tau - \tau_1 \geq$ τs

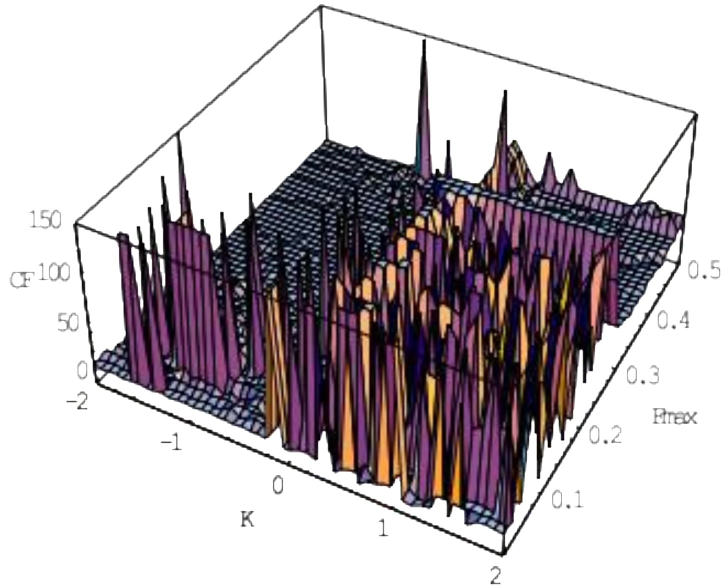
penalization1= 10*($\tau - \tau_1$) if $\tau - \tau_1 <$ τs (i.e. late stabilization)





Deterministic Chaos Control
Cost Function – Higher Periodic Orbits

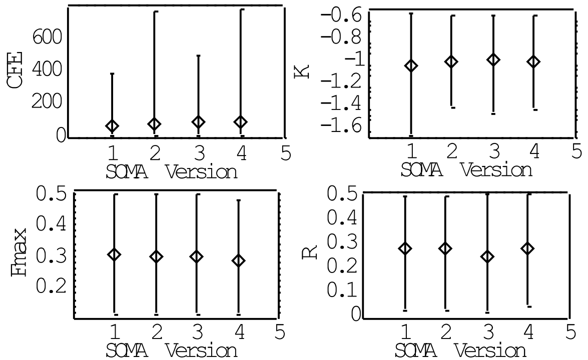


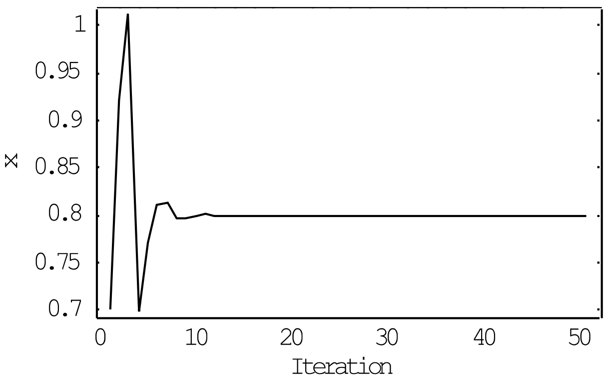


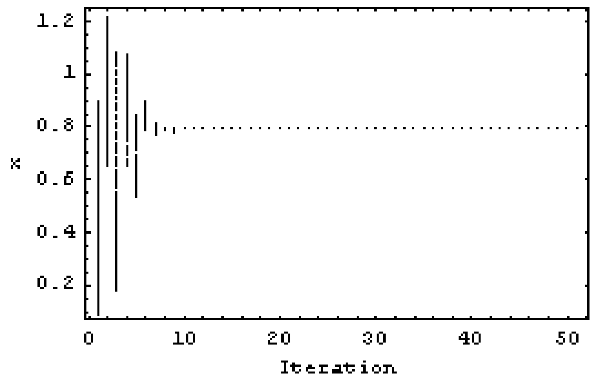
Deterministic Chaos Control

Results – Control of p-1 orbit TDAS

EA	K	F_{max}	R	CF Value
1	-1.03809	0.449163	0.326949	0
2	-1.01208	0.142598	0.294697	0
3	-1.16149	0.316863	0.256766	0
4	-1.04145	0.155919	0.345818	0

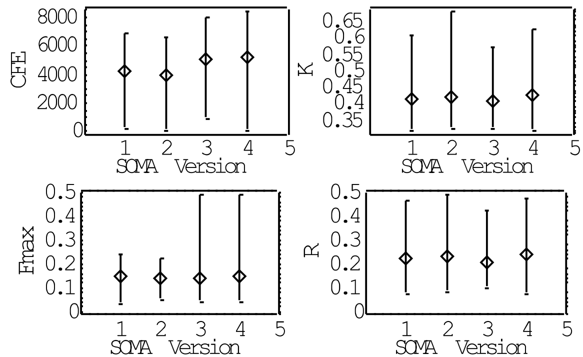


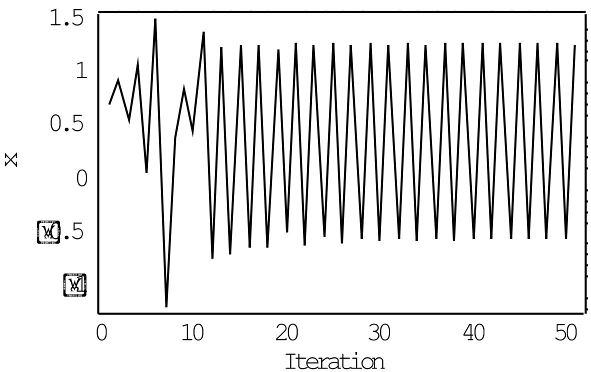


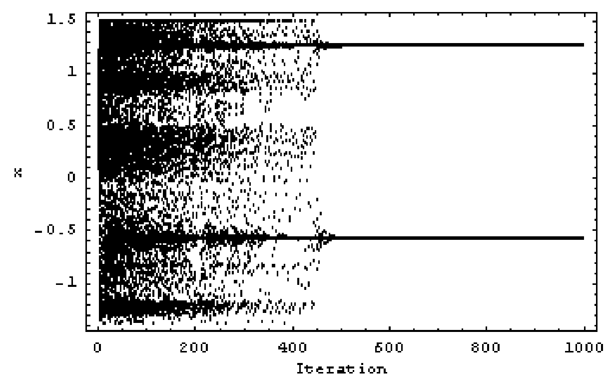


Deterministic Chaos Control
Results – Control of p-2 orbit TDAS

EA	K	F_{max}	R	CF Value
1	0.389335	0.09861	0.271823	$2.17 \cdot 10^{-7}$
2	0.472397	0.155925	0.461329	$1.60 \cdot 10^{-14}$
3	0.558013	0.15257	0.421521	$1.57 \cdot 10^{-7}$
4	0.54784	0.153437	0.43191	$1.53 \cdot 10^{-8}$

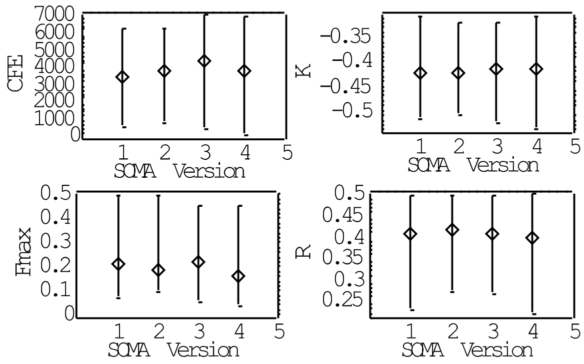


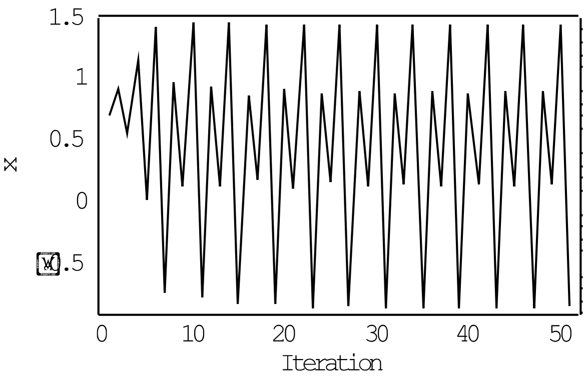


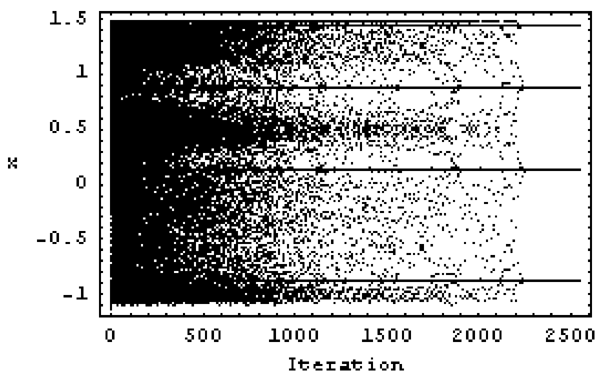


Deterministic Chaos Control
Results – Control of p-4 orbit TDAS

EA	K	F_{max}	R	CF Value
1	-0.383347	0.312323	0.436979	$9.57 \cdot 10^{-8}$
2	-0.425825	0.283005	0.457182	$5.39 \cdot 10^{-8}$
3	-0.369616	0.112991	0.410288	$9.43 \cdot 10^{-9}$
4	-0.412404	0.341886	0.467282	$9.90 \cdot 10^{-8}$

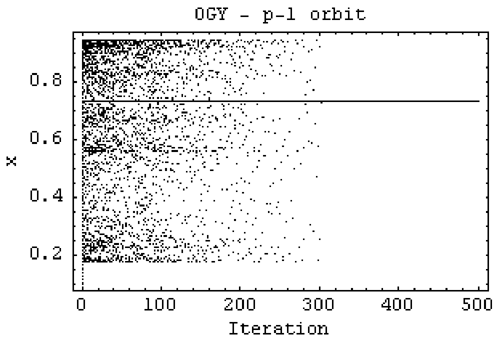


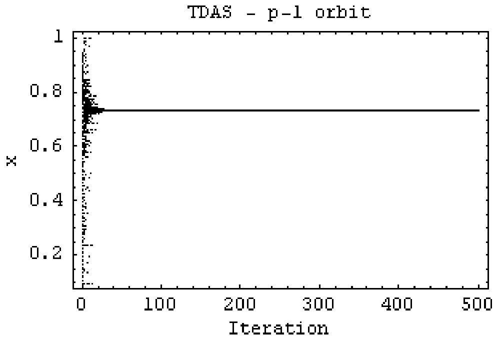




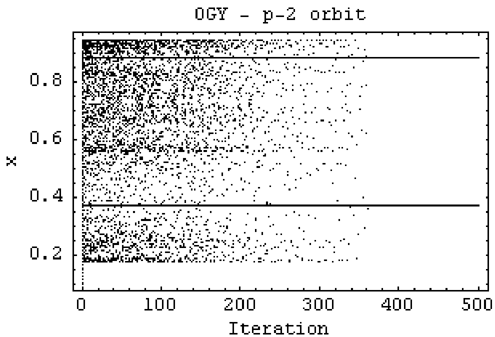
Deterministic Chaos Control
Comparison with OGY for p-1 and p-2 orbit

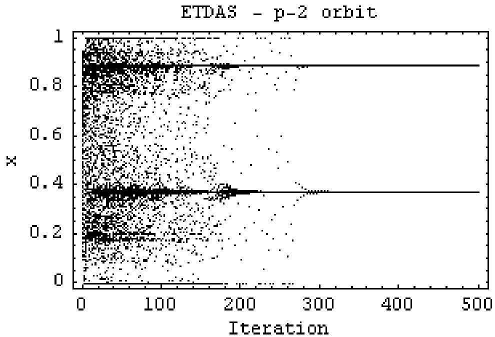
Stabilization of p-1 orbit





Stabilization of p-2 orbit





Deterministic Chaos Control

Conclusion

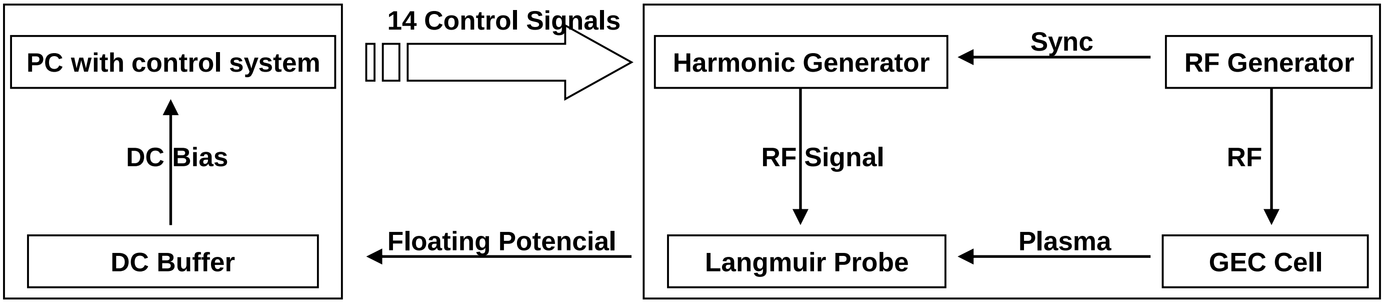
- Evolutionary algorithms are capable of solving this class of difficult problems.
- Quality of results produced by optimizations of chaos control strongly depends on the proper definition of a cost function.
- Faster stabilization when compared with classical OGY method.
- Possibilities for the future research - better settings of EA and testing more complex cost functions.

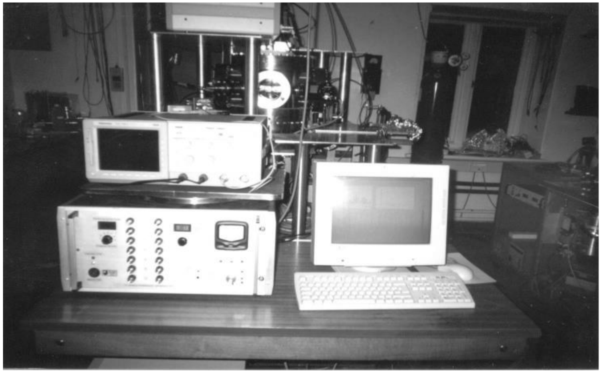
Plasma Reactor Control
Idea

- In this experiment, the performance of a self-organizing migration algorithm (SOMA) has been compared with simulated annealing (SA) and differential evolution (DE) for an engineering application.
- This application is the automated deduction of **fourteen** Fourier terms in a radio-frequency (RF) waveform to tune a Langmuir probe.
- Langmuir probes are diagnostic tools used to determine the ion density and the electron energy distribution in plasma processes. RF plasmas are inherently nonlinear, and many harmonics of the driving fundamental can be generated in the plasma.
- To improve the quality of the measurements, these RF components can be removed by an active-compensation method.
- Here, seven harmonics are used to generate the waveform applied to the probe tip. Therefore, fourteen mutually interacting parameters (seven phases and seven amplitudes) had to be tuned on-line.

Plasma Reactor Control
Equipment and Scheme

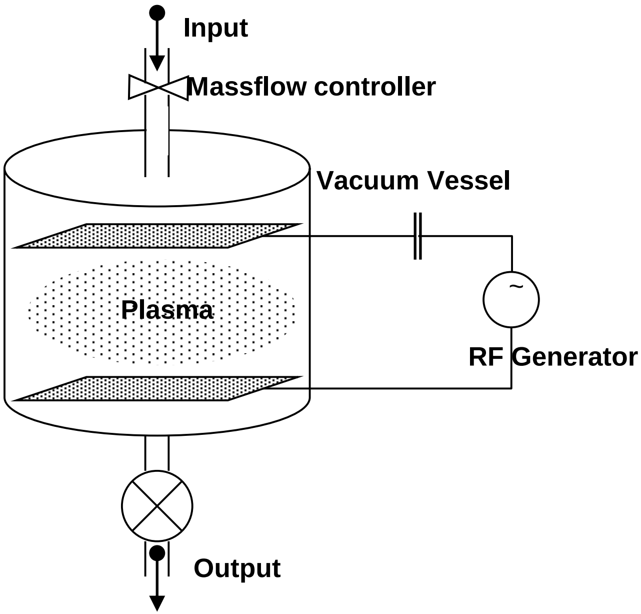
Problem scheme

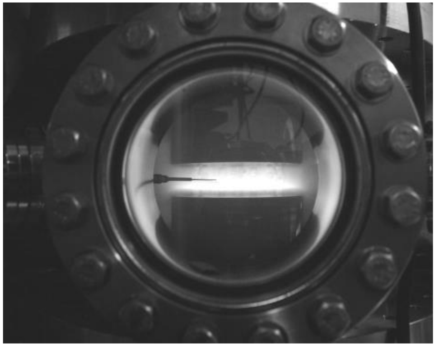




Computer with control software (right), wave synthesizer (bottom left) and oscilloscope (top left)

Plasma Reactor Control
Equipment and Scheme





Reactor Chamber

Plasma Reactor Control
Problem Definition

- The fourteen input parameters interact to some degree due to the technical realization of the synthesizer hardware and the nature of the problem.
- For example, the slightest departure from an ideal sinusoidal shape in one of the channels introduces harmonics itself.
- In practice, even after careful electronic design, it is found that there is a weak but significant coupling between amplitude control and phase and vice versa.
- Small variations in the 14 parameters caused by these interactions could lead to a large deviation from optimum tuning and hence the probe measurement itself.

Plasma Reactor Control
Problem Definition

- As a consequence of this, the number of points in the discrete search space has to be calculated as follows:

$$n = (2^b)^p$$

where:

- *n* is the number of points in search space.
- *b* is the resolution per channel in bits.
- *p* is the number of parameters to be optimized.

- The D/A and A/D converters used in this project had a resolution of **12** bits and the dimensionality of the search space was **14**.
- Hence, the search space consisted of $\boldsymbol{n = 3.7 \times 10^{50}}$ search points.
- Due to the system time constant, mapping out the entire search space would take approximately $\mathbf{10^{41}}$ years with the plasma system used.

Plasma Reactor Control
Experiment Setting

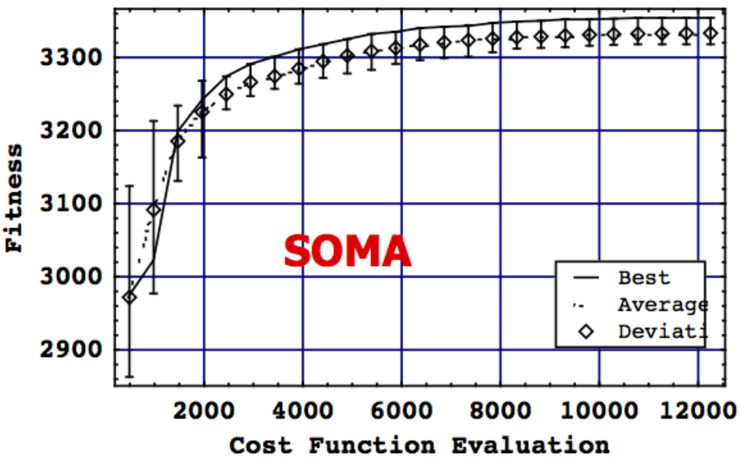
Plasma parameters used during the experiments

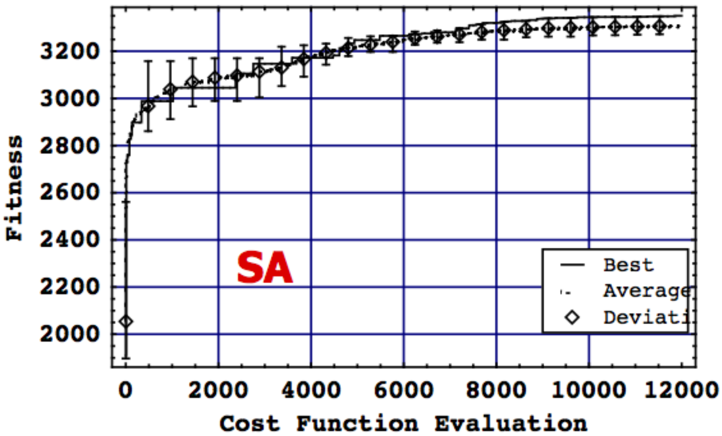
Plasma parameters	
Gas	Argon
Power	50 W
Pressure	100 mTorr
Flow rate	9.5 sccm

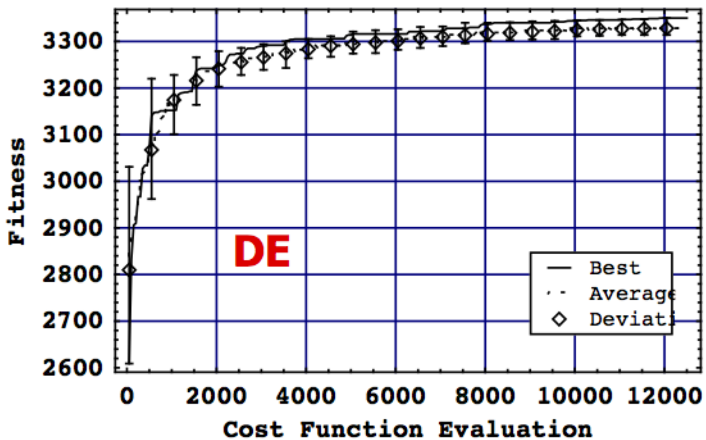
The best parameter settings used in experiments

SA			DE			SOMA	
T_{start}	25000		CR	0.5		PopSize	50
Temperature coef.	0.8		F	0.8		MinDiv	-1
Iterations per temperature	50		NP	50		Migrations	14
S_{max}	4000		Generations	250		PathLength	2
Number of particles	3					Step	0.11
Iterations	4000					PRT	0.1

Plasma Reactor Control
Results

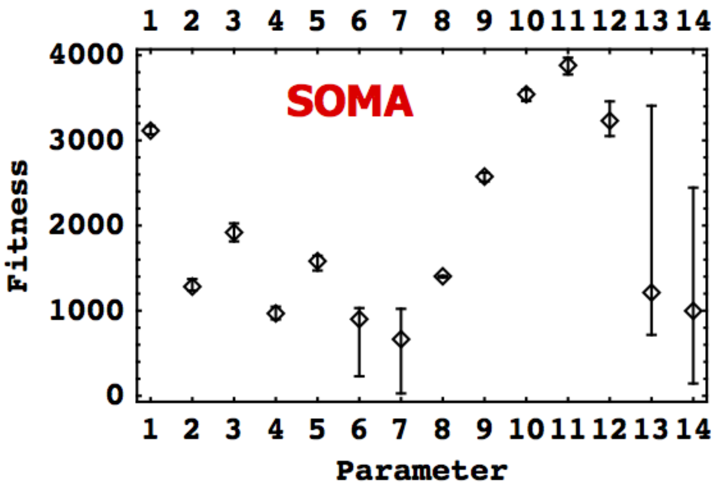


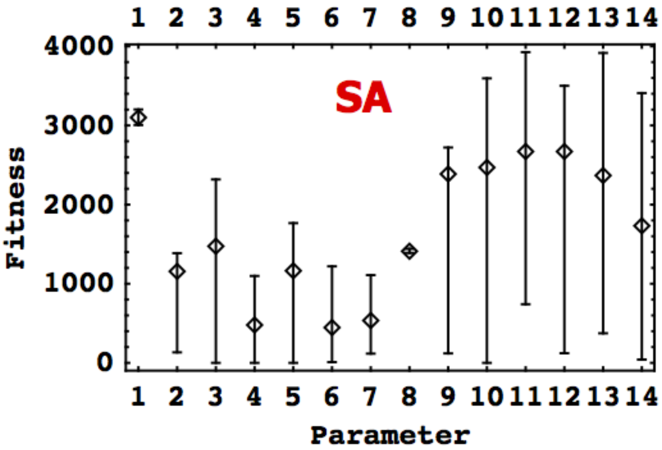


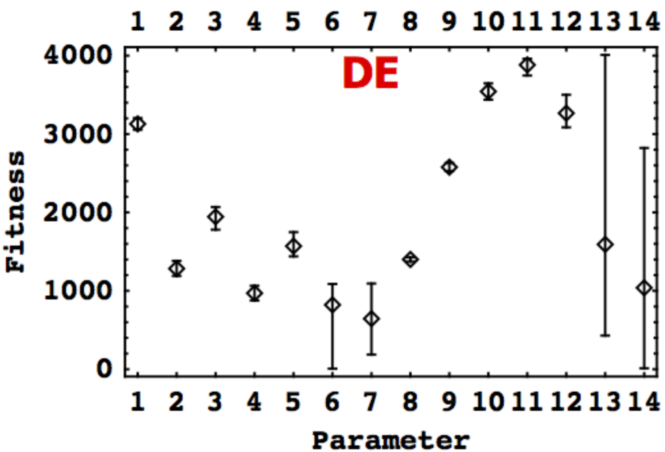


Plasma Reactor Control

Results

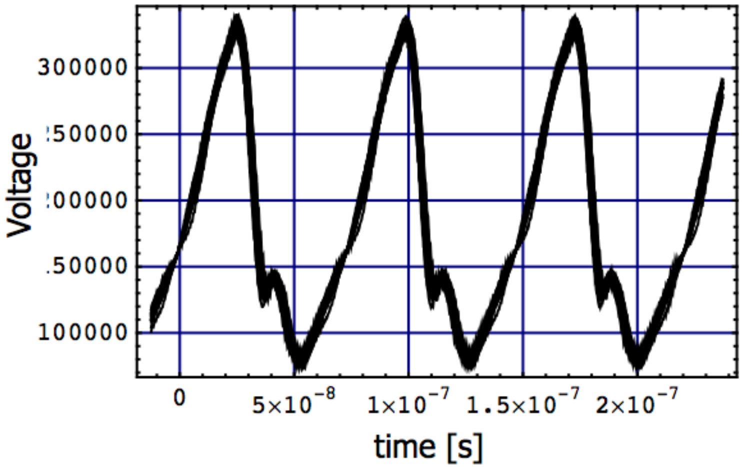




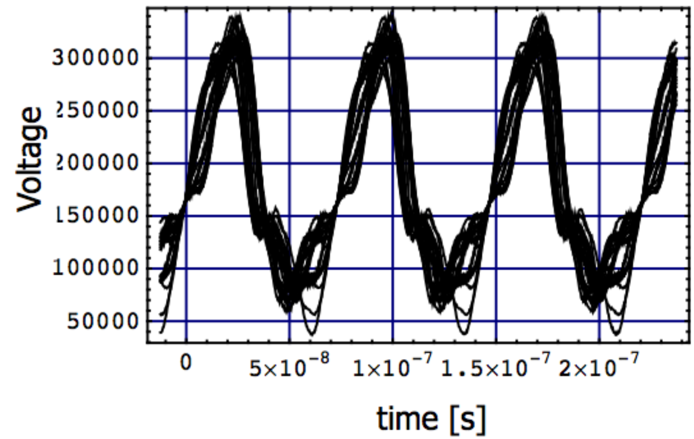


Plasma Reactor Control
Results

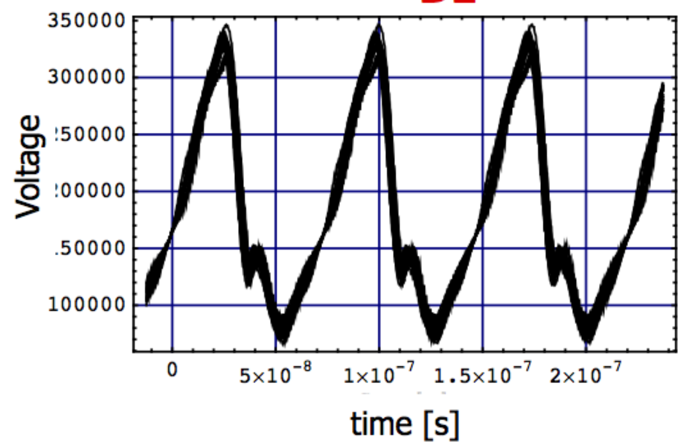
SOMA



SA



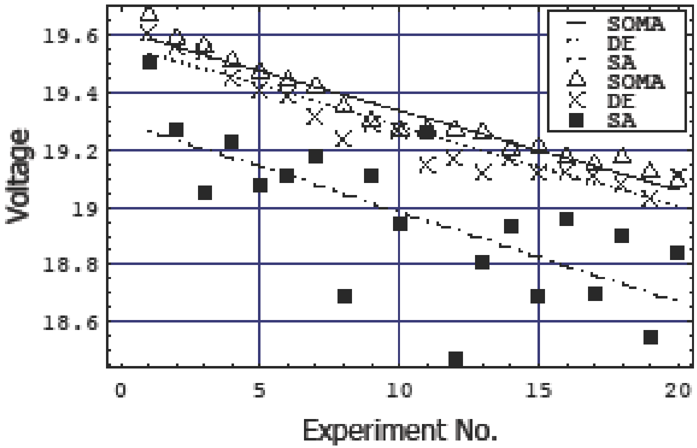
DE



Plasma Reactor Control
Results

- Experiment itself was recorded on videotape and movie shows two approaches:
  - By human operator
  - By evolutionary algorithms

- During experiments so called linear drift has been observed



Plasma Reactor Control

Conclusion

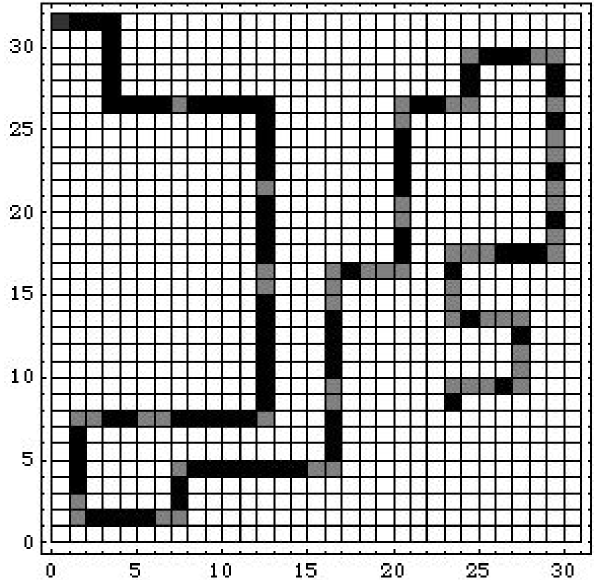
- Three stochastic optimization algorithms, SA, DE and SOMA, were used for online tuning of an actively compensated Langmuir probe system.
- These algorithms were selected because of the problem complexity.
- The experimental results demonstrate that, in general, all three algorithms were suitable for active compensation of the RF-driven plasma probe.
- However, the results also show that that SOMA and DE showed better performance compared with SA in this specific application.
- SOMA and DE performed almost three times better than SA. Bearing in mind that plasmas are highly nonlinear dynamical systems with changing properties, then the results produced by SOMA and DE are encouraging.

Robot Control Program Synthesis
Idea

- In this study sample using EAs to find a sequence of commands for robot will be demonstrated.
- More specifically, it is an artificial ant which searches for a specific sequence of commands.
- The goal is to find a sequence that ant passed through the **Santa Fe** trail with the least number of steps - commands.
- Ant and Santa Fe Trail is only a special case.
- On this problem, two methods were compared. Analytical programming and genetic programming, the results of which were taken from the literature.

Robot Control Program Synthesis
Problem Description

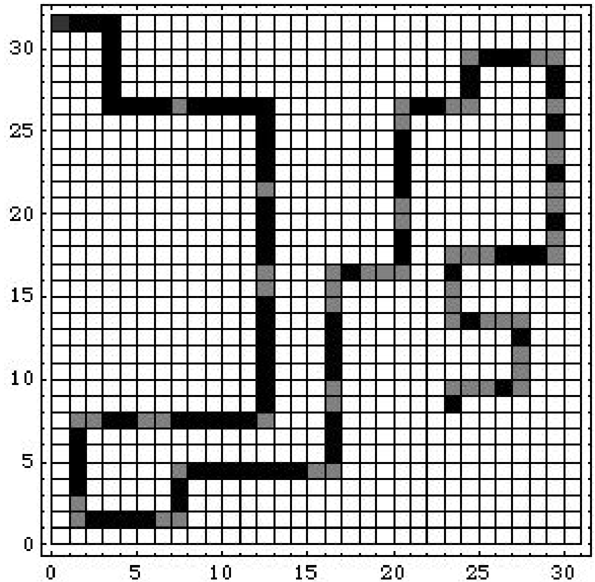
- Santa Fe trail for an artificial ant, defined by (Koza, 1998) in simulations with genetic programming.
- The role of artificial ant is engaged in movement in space, collecting food on a path and avoiding obstacles.
- This approach can be applied to a robot that has to move in a certain area.
- Santa Fe Trail is defined as an array of 31 × 32 boxes, see figure.



cz

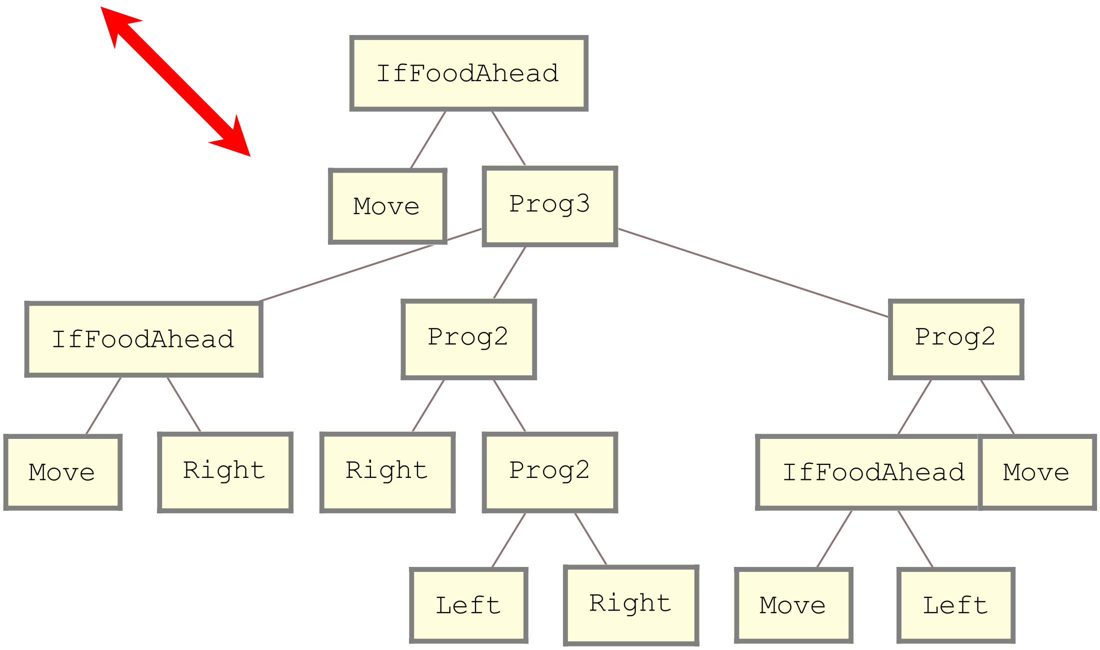
Robot Control Program Synthesis
Problem Description

- Black boxes in this figure represent food in the terminology of the Santa Fe problem.
- Food has to be collected. White and gray boxes do not contain food.
- Gray cells were chosen to illustrate the resulting paths that the ant passes. Standard ant could actually go along the path as follows:
  "... Look ahead when the next meal box, move to the field and pick up the food, otherwise turn clockwise and start from the beginning ...".

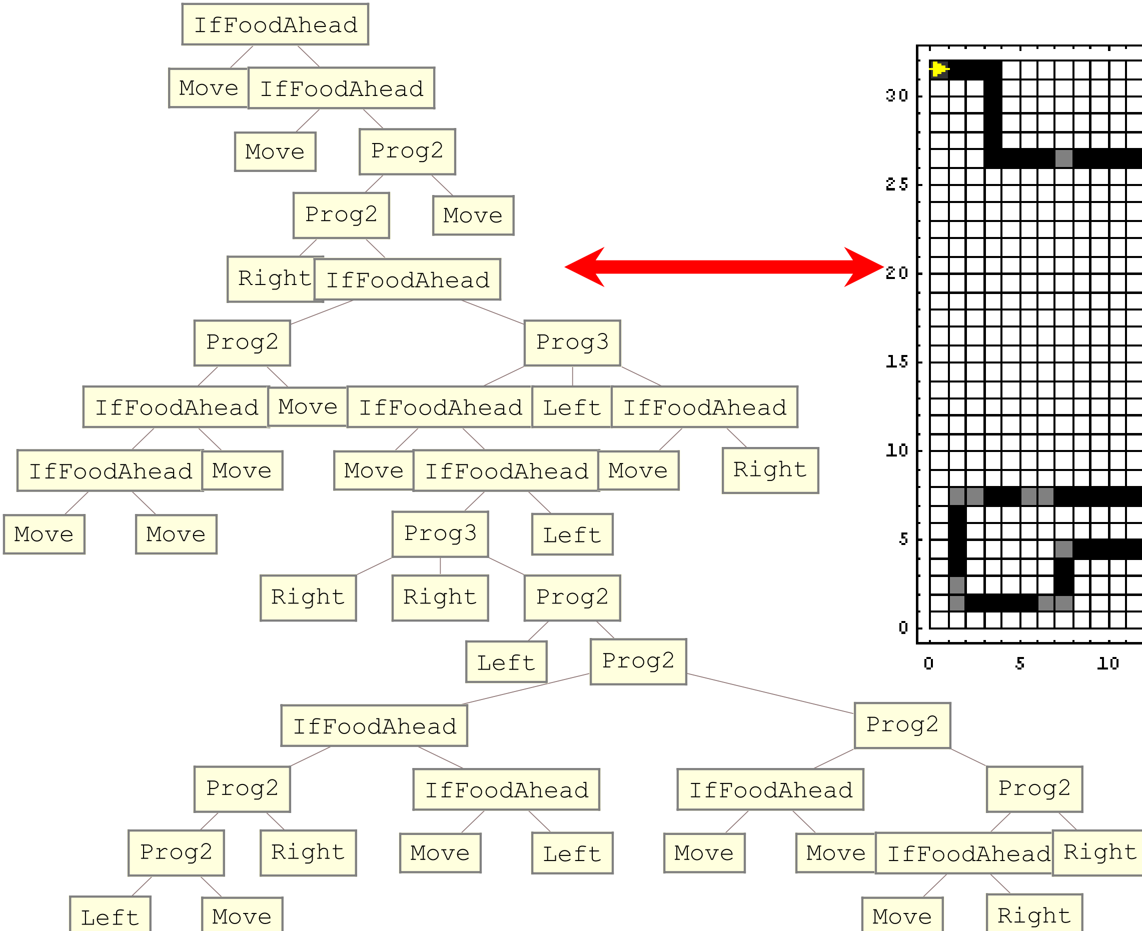


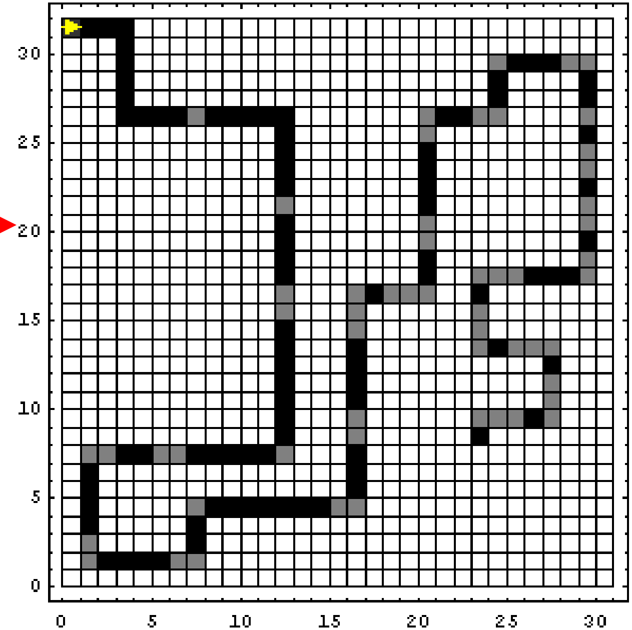
Robot Control Program Synthesis
Problem Description

IfFoodAhead[Move, Prog3[IfFoodAhead [Move, Right],
Prog2[Right, **Prog2[Left, Right]**],
Prog2[IfFoodAhead[Move, Left], Move]]]



Robot Control Program Synthesis
Program Dynamics





Deterministic Chaos Synthesis
Idea

- This experiment introduces the notion of chaos synthesis by means of evolutionary algorithms and develops a new method for chaotic systems synthesis.
- Used method (AP) is similar to genetic programming and grammatical evolution and is being applied along with three evolutionary algorithms: **differential evolution, self-organizing migration and genetic algorithm**.
- The aim of this investigation is to synthesize new and “simple” chaotic systems based on some elements contained in a selected existing chaotic system and a properly defined cost function.
- 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 in this report.

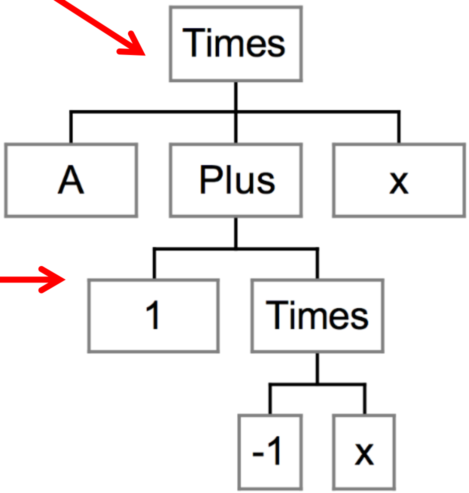
Deterministic Chaos Synthesis
Used System

- Logistic equation as a system used for decomposition was selected

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

- This equation can be viewed as a tree structure.

Times[A, Plus[1, Times[−1, x]], x]



- Building “bricks” from that tree were used as basic objects for symbolic regression.

Deterministic Chaos Synthesis
Used Algorithms

Algorithms

Algorithm	Version	Abbreviation
SOMA	AllToOne	A
	AllToOneRandomly	B
	AllToAll	C
	AllToAllAdaptive	D
Diferential Evolution	DERand1Bin	E
	DERand2Bin	F
	DEBest2Bin	G
	DELocalToBest	H
	DEBest1JIter	I
	DERand1DIter	J
Genetic Algorithm		K

SOMA

Algorithm	A	B	C	D
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	200	200
Migrations	10	10	10	10
MinDiv	−0.1	−0.1	−0.1	−0.1
Individual Length	50	50	50	50

DE

Algorithm	E	F	G	H	I	J
NP	200	200	200	200	200	200
F	0.9	0.9	0.9	0.9	0.9	0.9
CR	0.3	0.3	0.3	0.3	0.3	0.3
Generations	200	200	200	200	200	200
Individual Length	50	50	50	50	50	50

GA

Algorithm	K
PopSize	200
Mutation	0.4
Generations	100
Individual Length	50

Deterministic Chaos Synthesis
The Cost Function

- The cost function used for chaos synthesis, comparing with other problems like chaos control or black-box optimization, is quite a complex structure which cannot be easily described by a few simple mathematical equations. Instead, it is described by the following procedure:

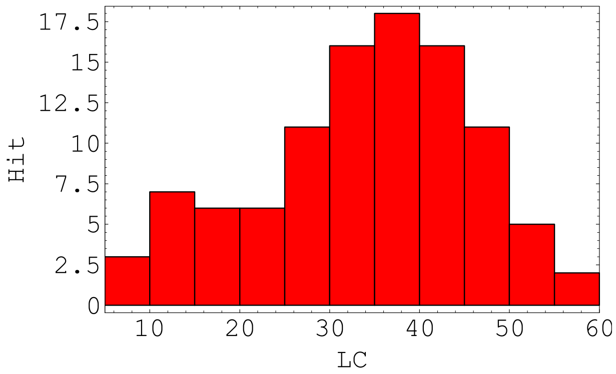
1. Take a synthesized function and evaluate it for 500 iterations with $A \in [0, 4]$ with a sampling step of $\Delta A = 0.1$.
2. Check if each value of A for all 500 iterations is unique or if some data are repeated in the series (the first check for chaos, indirectly). If the data is not unique, then go to step 5 else go to step 3.
3. Take the last 200 values, and for each value of A, calculate its Lyapunov exponent.

Deterministic Chaos Synthesis
The Cost Function

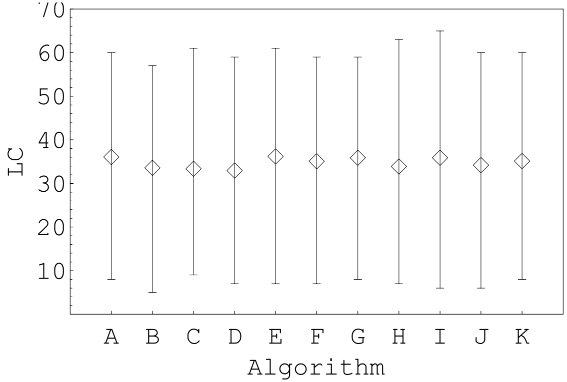
4. Check the Lyapunov exponent: if the Lyapunov exponent is positive, write all important data (synthesized functions, number of cost function evaluations, etc.) into a file. Then, repeat the simulation for another independent new one by going to step 1.
5. If the data is not unique, i.e. if the Lyapunov exponent is not positive, return an individual fitness, and sum all values whose occurrences in the dataset from step 1 are more than 1 (simply, it returns the occurrences of periodicity. Periodicity – higher penalization of an individual in the evolution).

Deterministic Chaos Synthesis
Results Evaluation

- For results evaluation was used a few criteria:
  - Histogram of leaf count (LC).



  - Program size in the form of LC for each algorithm.

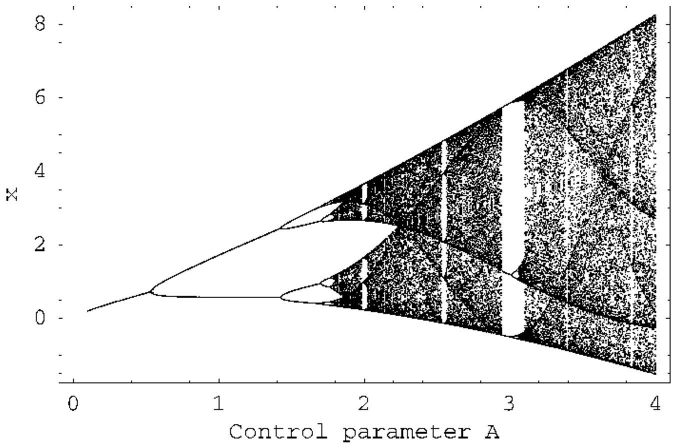


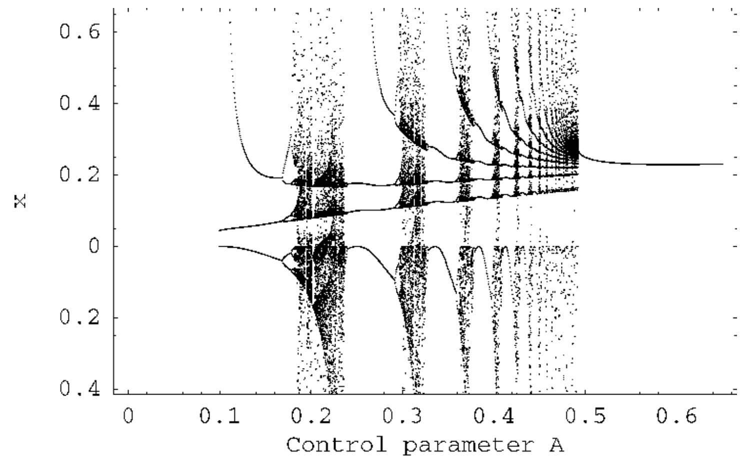
Deterministic Chaos Synthesis
Results

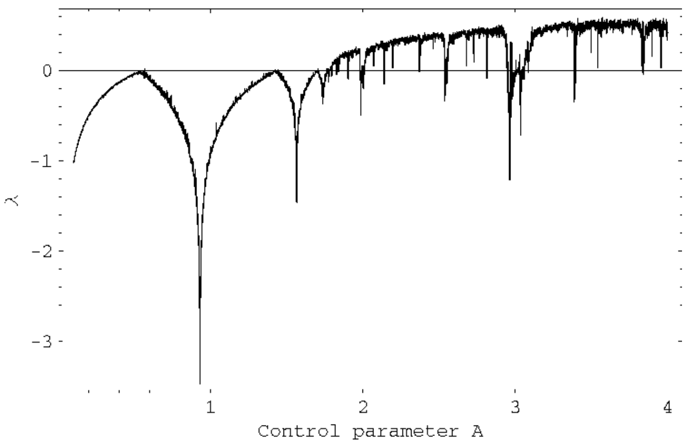
Equation	Bifurcations and Chaos Observable in the Interval
$A - x\left(-A + \frac{x\left(-\frac{A}{x} + x + Ax\right)}{A + x^2}\right)$	$\langle 0, 4 \rangle$
$\frac{A(2A - 2x^2 - 3x(A - x + Ax))}{-A + x - x^2}$	$\langle 0.1, 0.13 \rangle$ $\langle 0.8, 1.2 \rangle$
$-x - \frac{1 - 2A + 2x + 2A^2 x}{1 - A + \frac{A^2 - x}{x} + x}$	$\langle 0.3, 0.5 \rangle$
$\frac{x - A(A - x - 2x^2)}{-A - x + Ax^2 - A\left(-A + \frac{A^3}{x} + 2x\right)}$	$\langle 0.4, 0.5 \rangle$
$\frac{2A(-2A + 2x)}{x + \frac{1 + A^2 + x}{x}}$	$\langle 0, 4 \rangle$
$\frac{x}{(3A + 2x)\left(-1 - A - x + \frac{x(A + 2x)}{A^2 + x}\right)}$	$\langle 0.12, 0.23 \rangle$ $\langle 0.3, 0.36 \rangle$
$\frac{(1 - x)(-2A + x)}{-2A + \frac{A}{x} + x + x(A + x)}$	$\langle 0, 4 \rangle$

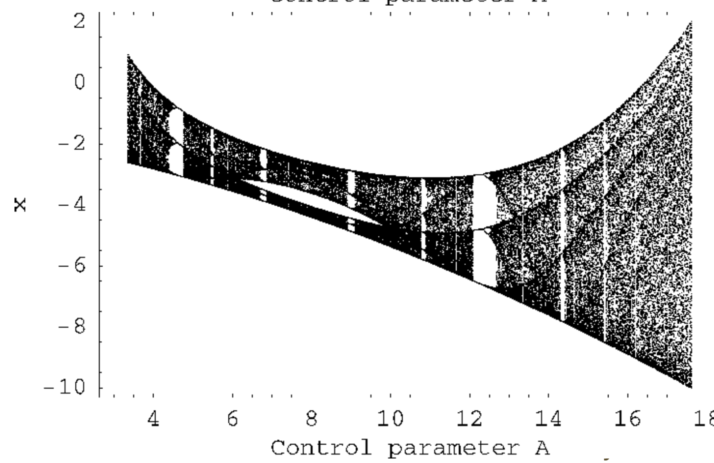
Equation	Bifurcations and Chaos Observable in the Interval
$\frac{1}{\left(\frac{A^2 - x}{x} + x\right)\left(-x + \frac{A - x + x^2}{A(A - x)}\right)}$	$\langle 0.7, 1.6 \rangle$
$\frac{x}{-x + \frac{x}{x - \frac{x}{A(A + A^2)} + \frac{2Ax}{(1 + A - x)(1 + x)}}}$	$\langle 0.44, 0.475 \rangle$
$\frac{x(2 + A - 2x + x(A + x))}{-A + \frac{A + 3x}{-\frac{A}{x} + x^2}}$	$\langle 0.6, 1.1 \rangle$
$\frac{A(-A + x)}{2 + A + x - x^2 + 2x(A + x^2) + \frac{x}{x + x^2}}$	$\langle 0.5, 2.6 \rangle$
$\frac{A}{-\frac{1}{x} + x - Ax + Ax^2 - \frac{x}{2x - \frac{x}{A}}}$	$\langle 0.8, 4 \rangle$
$\frac{(A - x)(A - x + Ax)}{-A + x - x^2}$	$\langle 1.5, 6 \rangle$
$-x + \frac{2A + \frac{A}{x} - 2x}{\frac{A}{x} + x + x^3}$	$\langle 0, 4 \rangle$

Deterministic Chaos Synthesis
Results

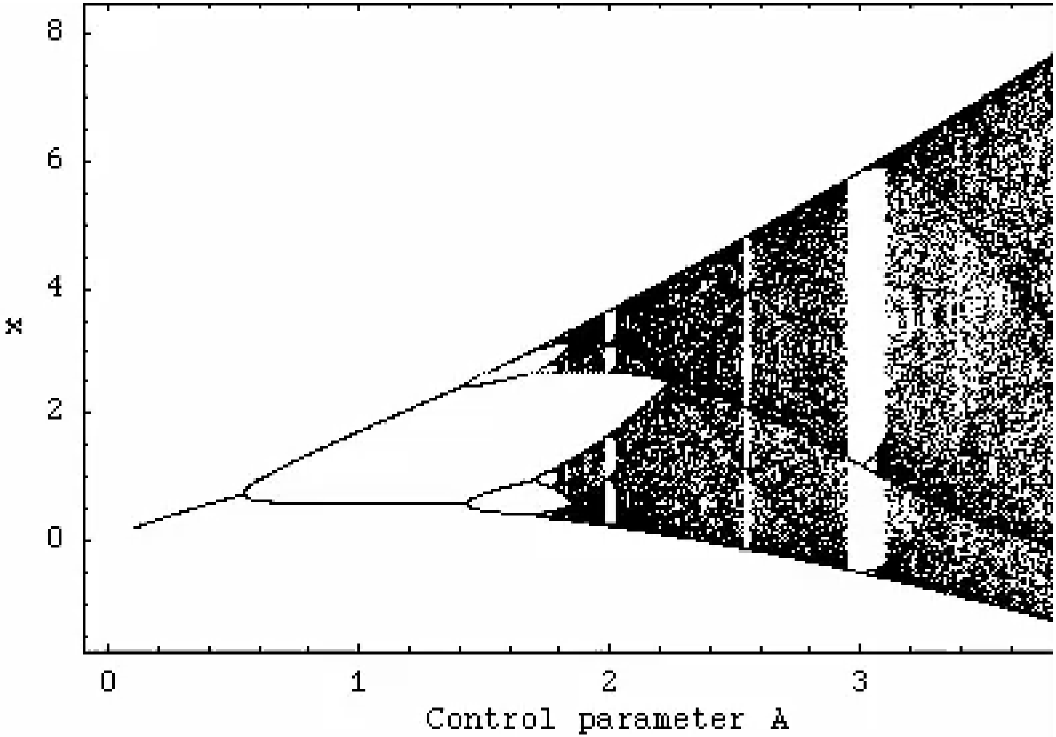








Deterministic Chaos Synthesis
Results (video)



Deterministic Chaos Synthesis
Conclusion

- The aim of this presentation 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 statistics for overall performance evaluation of evolutionary chaos synthesis. The results look quite promising and convincing.
- For comparative studies, three algorithms were used — DE [Price, 1999], SOMA [Zelinka, 2004] and GA [Holland, 1975; Davis, 1996].
- They were chosen to show that evolutionary synthesis of chaos by AP can be implemented via any evolutionary algorithm and that they all give reasonable results.

Logic (Electronics) Circuits Synthesis
Idea

- In this experiments three electronic circuits were experimentally synthesized.
- The main point of this application was to confirm, that EA’s are capable of successful designing electronics circuits.
- In experiments the synthesis of
  - Boolean problems
  - Three electronic circuits for
    - traffic light control
    - heat control
    - train station control

  are described.
- In all three experiments (50 times repeated) AP has been observed to be capable of electronic circuit synthesis.

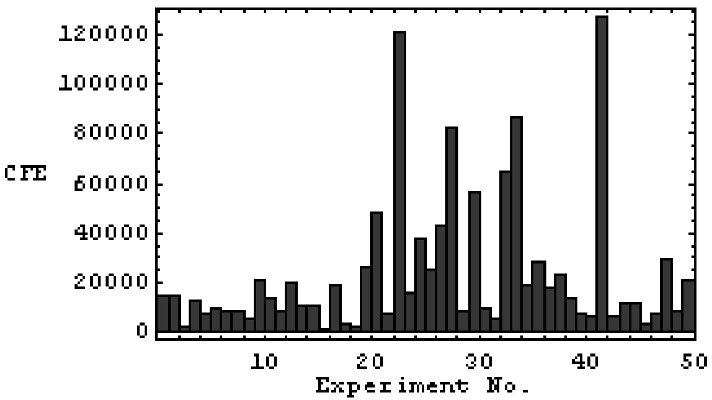
Logic (Electronics) Circuits Synthesis
Boolean Problem

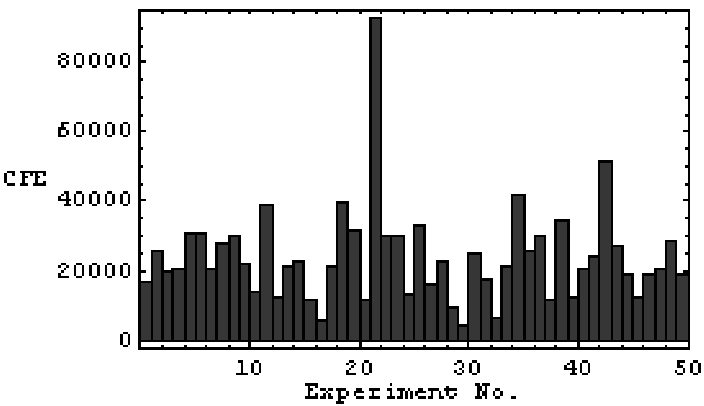
Input 1	Input 2	Input 3	Output
True	True	True	False
True	True	False	True
True	False	True	True
False	True	True	True
True	False	False	False
False	True	False	False
False	False	True	False
False	False	False	True

$$f_{\cos t} = \sum_{i=1}^{2^n} \left| TT_i - P_i \right|$$

TT_i - i^{th} output of the truth table

P_i - i^{th} output of the synthetized program





Solutions of the problem of Boolean even-3-parity using SA.

	CFE	Program length
Minimum	1460	710
Average	23387	4203
Maximum	127468	14731

Solutions of the problem of Boolean even-3-parity using GA.

	CFE	Program length
Minimum	2923	220
Average	35386	2910
Maximum	130069	10157

Solutions of the problem of Boolean even-3-parity using SOMA.

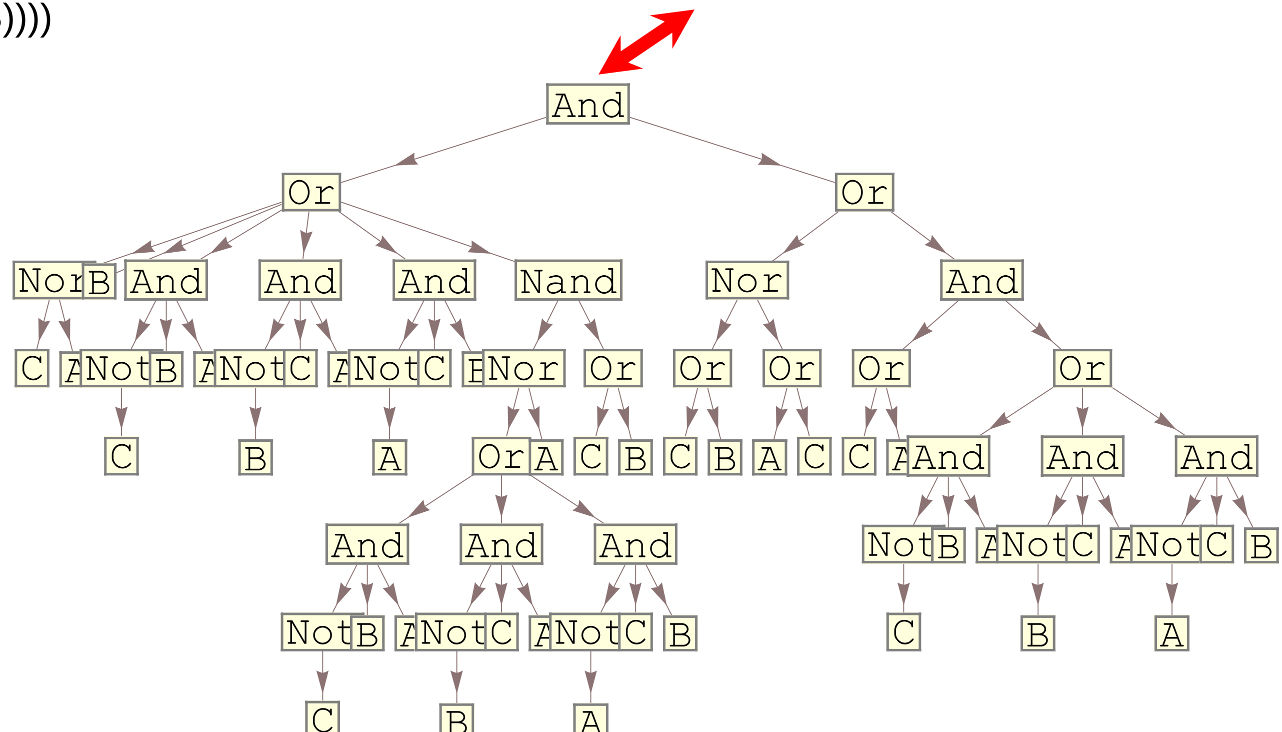
	CFE	Program length
Minimum	4256	508
Average	23861	2950
Maximum	92542	10770

Solutions of the problem of Boolean even-3-parity using DE.

	CFE	Program length
Minimum	1371	58
Average	9456	126
Maximum	20789	186

Logic (Electronics) Circuits Synthesis
Boolean Problem

((Nor[C, A]) || B || (! C && B && A) || (! B && C && A) || (! A && C && B) || (Nand[Nor[(! C && B && A) || (! B && C && A) || (! A && C && B), A], C || B])) && ((Nor[C || B, A || C]) || ((C || A) && ((! C && B && A) || (! B && C && A) || (! A && C && B))))

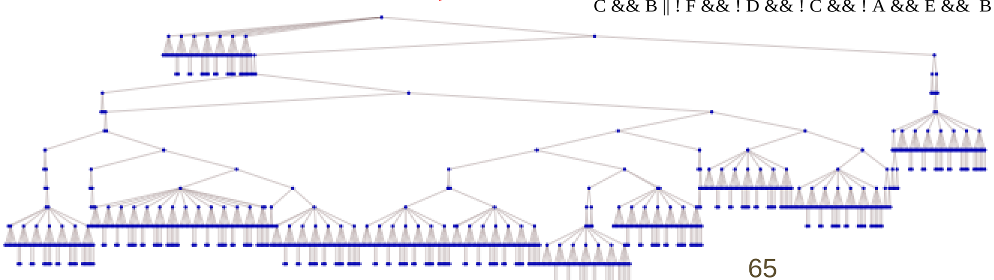


Logic (Electronics) Circuits Synthesis

Boolean Problem

((C && D) || (! C && ! D)) && ((B && ã) || (! B && ! ã)) && ((F && A) || (! F && ! A))

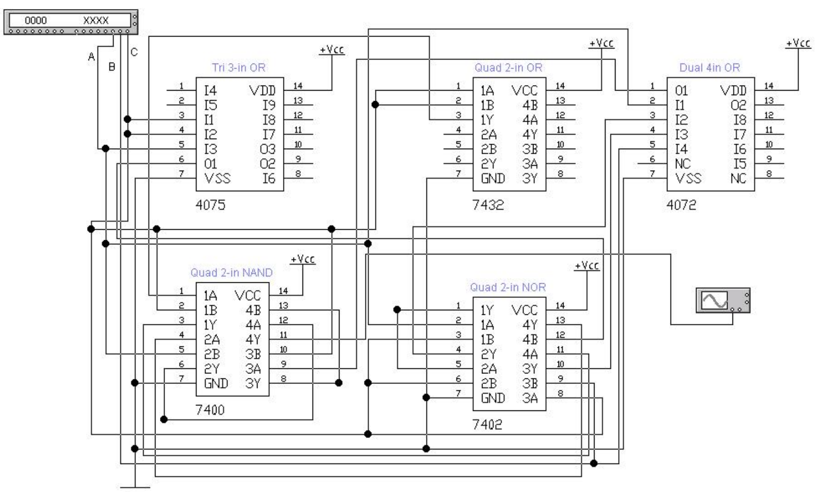
F && E && D && C && B && A || ! D && ! C && F && E && B && A || ! E && ! B && F && D && C && A ||
! E && ! D && ! C && ! B && F && A || ! F && ! A && E && D && C && B || ! F && ! D && ! C && ! A &&
E && B || ! F && ! E && ! D && ! C && ! B && ! A || (Nor[Nand[D, (Nor[F, E]) && ((Nor[(Nor[B, Nand[E,
Nand[E, A || F && E && D && C && B && A || ! D && ! C && F && E && B && A || ! E && ! B && F && D
&& C && A || ! E && ! D && ! C && ! B && F && A || ! F && ! A && E && D && C && B || ! F && ! D && !
C && ! A && E && B || ! F && ! E && ! D && ! C && ! B && ! A]]]) || (Nor[Nor[D, F && B], Nand[F && E &&
D && C && B && A || ! D && ! C && F && E && B && A || ! E && ! B && F && D && C && A || ! E && ! D
&& ! C && ! B && F && A || ! F && ! A && E && D && C && B || ! F && ! D && ! C && ! A && E && B || ! F
&& ! E && ! D && ! C && ! B && ! A || F && E && D && C && B && A || ! D && ! C && F && E && B &&
A || ! E && ! B && F && D && C && A || ! E && ! D && ! C && ! B && F && A || ! F && ! A && E && D &&
C && B || ! F && ! D && ! C && ! A && E && B || ! F && ! E && ! D && ! C && ! B && ! A || B, Nand[Nor[C,
F], E || F && E && D && C && B && A || ! D && ! C && F && E && B && A || ! E && ! B && F && D && C
&& A || ! E && ! D && ! C && ! B && F && A || ! F && ! A && E && D && C && B || ! F && ! D && ! C && !
A && E && B || ! F && ! E && ! D && ! C && ! B && ! A]]]), A]) || (Nor[Nand[C && (Nor[F && E && D &&
C && B && A || ! D && ! C && F && E && B && A || ! E && ! B && F && D && C && A || ! E && ! D && ! C
&& ! B && F && A || ! F && ! A && E && D && C && B || ! F && ! D && ! C && ! A && E && B || ! F && !
E && ! D && ! C && ! B && ! A, F || F && E && D && C && B && A || ! D && ! C && F && E && B && A || !
E && ! B && F && D && C && A || ! E && ! D && ! C && ! B && F && A || ! F && ! A && E && D && C &&
B || ! F && ! D && ! C && ! A && E && B || ! F && ! E && ! D && ! C && ! B && ! A]), ((F && E && D &&
C && B && A || ! D && ! C && F && E && B && A || ! E && ! B && F && D && C && A || ! E && ! D && ! C
&& ! B && F && A || ! F && ! A && E && D && C && B || ! F && ! D && ! C && ! A && E && B || ! F && !
E && ! D && ! C && ! B && ! A) && D || A && E) && (F && E && D && C && B && A || ! D && ! C && F
&& E && B && A || ! E && ! B && F && D && C && A || ! E && ! D && ! C && ! B && F && A || ! F && ! A
&& E && D && C && B || ! F && ! D && ! C && ! A && E && B || ! F && ! E && ! D && ! C && ! B && ! A)
&& B], Nor[C, Nor[Nand[D, A], C]]]) && (Nand[F && E && D && C && B && A || ! D && ! C && F && E &&
B && A || ! E && ! B && F && D && C && A || ! E && ! D && ! C && ! B && F && A || ! F && ! A && E &&
D && C && B || ! F && ! D && ! C && ! A && E && B || ! F && ! E && ! D && ! C && ! B && ! A, Nor[F &&
E && D && C && B && A || ! D && ! C && F && E && B && A || ! E && ! B && F && D && C && A || ! E
&& ! D && ! C && ! B && F && A || ! F && ! A && E && D && C && B || ! F && ! D && ! C && ! A && E &&
B || ! F && ! E && ! D && ! C && ! B && ! A, Nand[A, E] || (Nor[D, F])]]]))], Nand[Nor[F, F], Nor[D &&
(Nand[C, (Nand[E, A]) && (Nor[A, E])] || F && E && D && C && B && A || ! D && ! C && F && E && B &&
A || ! E && ! B && F && D && C && A || ! E && ! D && ! C && ! B && F && A || ! F && ! A && E && D &&
C && B || ! F && ! D && ! C && ! A && E && B || ! F && ! E && ! D && ! C && ! B && ! A), A]]])

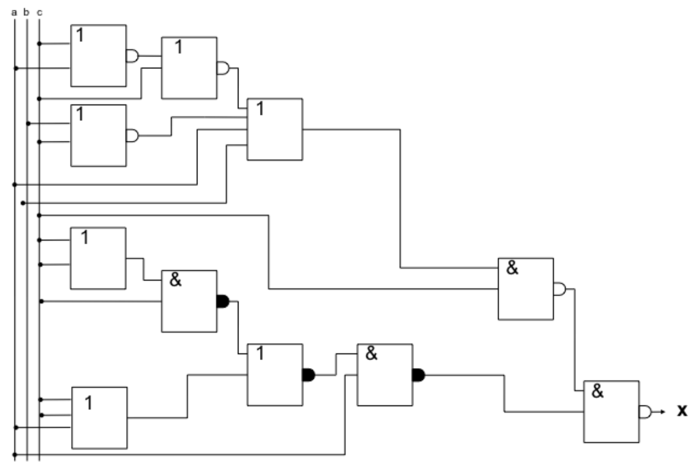


Logic (Electronics) Circuits Synthesis

Light Control

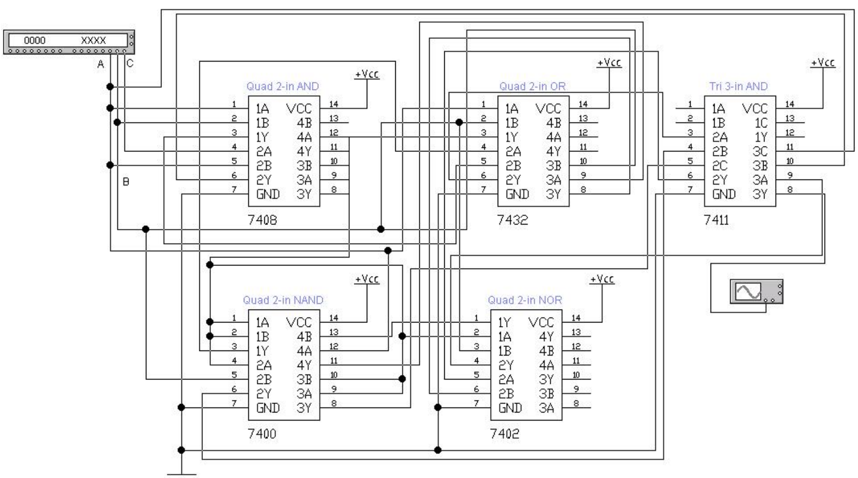
Input 1	Input 2	Input 3	Output
False	False	False	False
False	False	True	False
False	True	False	False
False	True	True	True
True	False	False	False
True	False	True	True
True	True	False	False

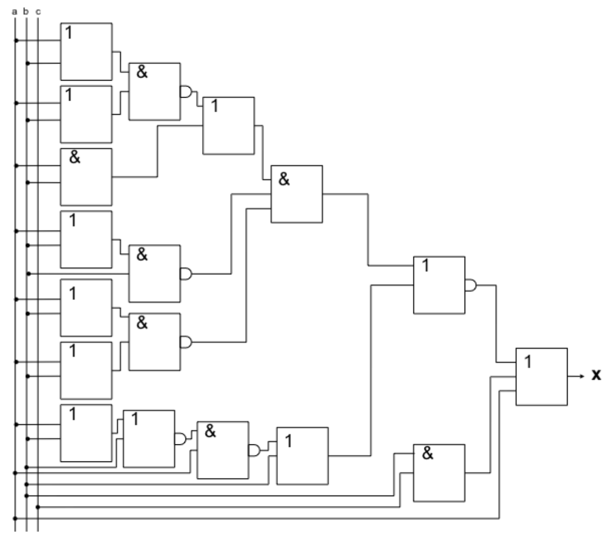




Logic (Electronics) Circuits Synthesis
Heat Control

Input 1	Input 2	Input 3	Output
False	False	False	False
False	False	True	False
False	True	False	False
False	True	True	True
True	False	False	True
True	False	True	True
True	True	False	True

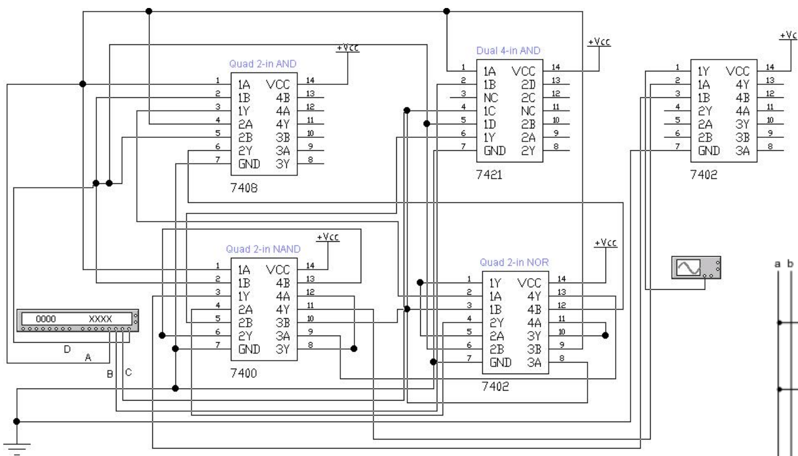


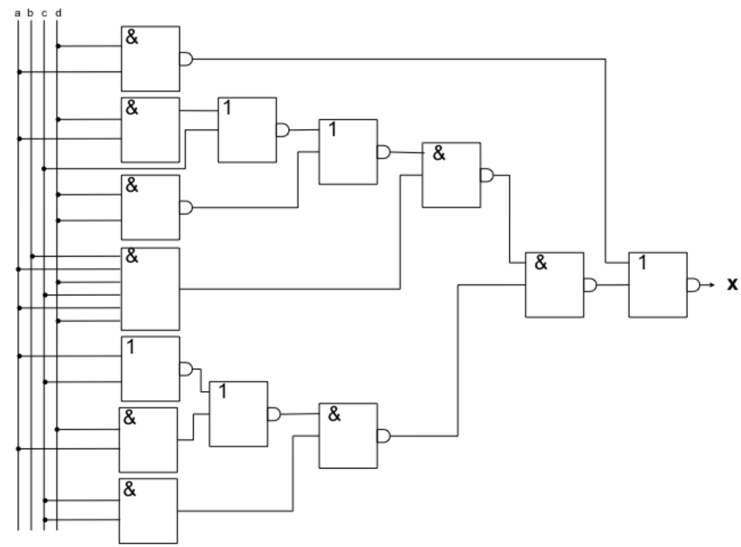


Logic (Electronics) Circuits Synthesis
Train Traffic Control

Input 1	Input 2	Input 3	Input 4	Output
False	False	False	False	False
False	False	False	True	False
False	False	True	False	False
False	False	True	True	False
False	True	False	False	False
False	True	False	True	False
False	True	True	False	False
False	True	True	True	False
True	False	False	False	False
True	False	False	True	True
True	False	True	False	False
True	False	True	True	True
True	True	False	False	False
True	True	False	True	True
True	True	True	False	False
True	True	True	True	False

Logic (Electronics) Circuits Synthesis
Train Traffic Control





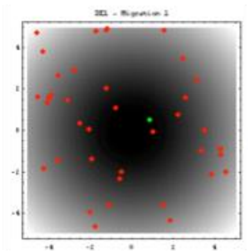
Logic (Electronics) Circuits Synthesis
Conclusion

- Within this sample studies there were synthesized logical expressions and circuits by means of selected evolutionary techniques.
- Some results have been presented as a logical formula, other schemes as logic circuits.
- Not only from relatively trivial results presented in this study, but also a great number of results published in the international literature is seen that the evolutionary synthesis using genetic programming, grammatical evolution and similar techniques is promising in the future applications and results.
- “Proof” of this is now patented electronic device designed in full evolution process…

Frontiers (?)
Evolution as a Complex Network

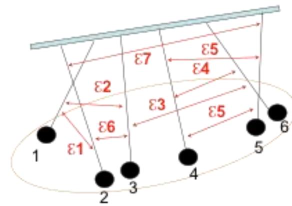
Chain control (Loop Control)

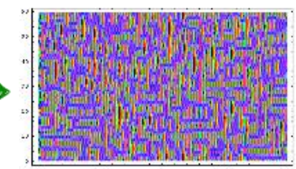
Evolutionary Algorithms ➡ Complex Network ➡ CML ➡ CML Control ➡ Complex Network Control ➡ Dynamics of Evolutionary Algorithms Control



EAs has dynamics, that can be described ...

... like complex network with increasing edges and its weights, ...

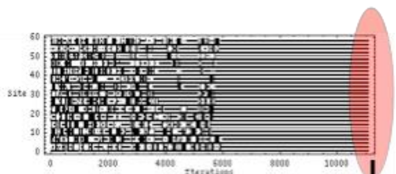


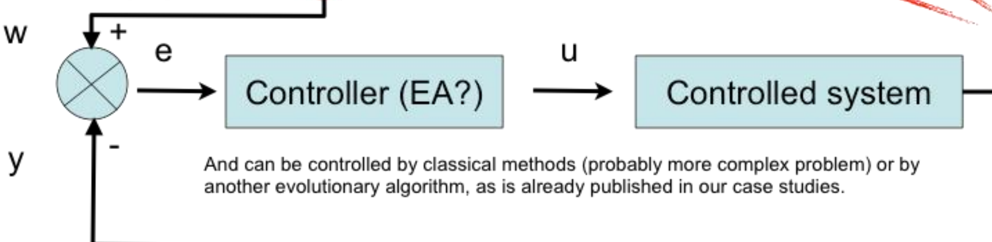


... and converted to the CML system as shown by example with row of pendulums. Each row of CML represent one pendulum and structure of ε connections is more complex than in classic CML.

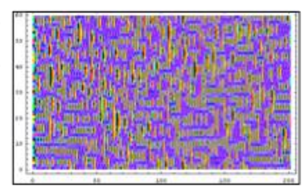
If we control CML, then, in fact, we control dynamics of CML and EA.

CML is controllable system.





And can be controlled by classical methods (probably more complex problem) or by another evolutionary algorithm, as is already published in our case studies.



Conclusions

- EA has been discussed
- Basic ideas of symbolic regression
- Case studies
- Frontiers

Want to know more?

- Contact us :)

THANK YOU FOR YOUR ATTENTION

ivan.zelinka@ieee.org

www.ivanzelinka.eu

Copyright