Computers and Mathematics with Applications 60 (2010) 1026–1037

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Utilization of SOMA and differential evolution for robust stabilization of chaotic Logistic equation

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

Tomas Bata University in Zlin, Department of Applied Informatics, Nad Stranemi 4511, 76005 Zlin, Czech Republic

ARTICLE INFO

Keywords:
Chaos
Control
Optimization
Evolutionary algorithms
Differential evolution
SOMA

ABSTRACT

This paper deals with the utilization of two evolutionary algorithms Self-Organizing Migrating Algorithm (SOMA) and Differential Evolution (DE) for the optimization of the control of chaos. This paper is aimed at an explanation on how to use evolutionary algorithms (EAs) and how to properly define the advanced targeting cost function (CF) securing fast, precise and mainly robust stabilization of selected chaotic system on a desired state for any initial conditions. The role of EA here is as a powerful tool for an optimal tuning of control technique input parameters. As a model of deterministic chaotic system, the one-dimensional discrete Logistic equation was used. The four canonical strategies of SOMA and six canonical strategies of DE were utilized. For each EA strategy, repeated simulations were conducted to outline the effectiveness and robustness of used method and targeting CF securing robust solution. Satisfactory results obtained by both heuristic and the two proposed cost functions are compared with previous research, given by different cost function designs.

1. Introduction

The question of targeting (faster stabilization) with application to chaos control has attracted researchers since the first method for controlling of chaos was developed. The several first approaches for targeting have used special versions of OGY (Ott–Grebogi–York) control scheme [1,2] or collecting information about trajectories, which falls close to desired state [3]. Later, numerous methods were based on adaptive approach [4], center manifold targeting [5] or neural networks [6,7].

Currently, evolutionary algorithms (EAs) [8–13] are known as powerful tools for almost any difficult and complex optimization problem. But the quality of obtained results through optimization mostly depends on proper design of the used cost function, especially when the EAs are used for optimization of chaos control. The results of numerous simulations lend weight to the argument that deterministic chaos in general and also any technique to control of chaos are sensitive to parameter setting, initial conditions and in the case of optimization, they are also extremely sensitive to the construction of used cost function.

This research utilized Pyragas' delayed feedback control technique ETDAS (Extended Time Delay Auto Synchronization) [14–16]. Unlike the original OGY control method [17], it can be simply considered as a targeting and stabilizing algorithm together in one package [18]. Another big advantage of the Pyragas method is the amount of accessible control parameters. This is very advantageous for successful use of optimization of parameter setting by means of EA, leading to improvement of system behavior and better and faster stabilization to the desired periodic orbits. Some research in this field has recently been done using EAs for optimization of local control of chaos [19,20], however our approach is different.

* Corresponding author. Tel.: +420 57 603 5189; fax: +420 57 603 2716.
E-mail addresses: senkerik@fai.utb.cz (R. Senkerik), zelinka@fai.utb.cz (I. Zelinka), davendra@fai.utb.cz (D. Davendra), oplatkova@fai.utb.cz (Z. Oplatkova).

0898-1221/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2010.03.059

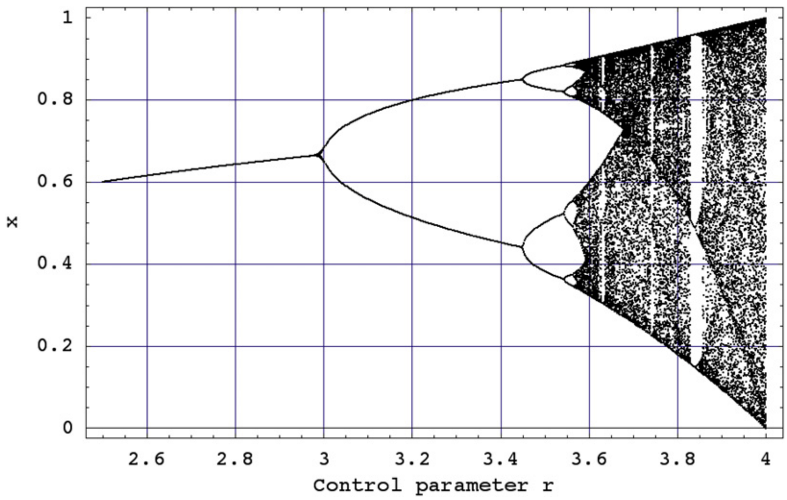

Fig. 1. Bifurcation diagram of Logistic equation.

The role of EA in this instance is as a powerful tool for an optimal tuning of control technique input parameters based on advanced targeting cost functions securing the stabilization to desired UPO (unstable periodic orbit) for any initial conditions. Numerous applications either of canonical DE or special version of DE [21–23] and SOMA [24] have proven that these heuristics are suitable for solving a difficult class of problems [25].

This work presents an accumulation of research [26] and also collates and elaborates the experiences with application of EA to chaos control [27,28] in order to reach better results and decrease the influence of negative phenomenon which can occur in such a challenging task, which is chaos control.

The main aim of this paper is not to show, which heuristic or its strategy is better or worse, but to test the selected EAs in such a challenging task, which is optimization of chaos control.

2. Problem design

2.1. Problem selection and case studies

The chosen example of a chaotic system was the one-dimensional Logistic equation as in form (1).

$$x_{n+1} = rx_n (1 - x_n) . \tag{1}$$

The Logistic equation (Logistic map) is a one-dimensional discrete-time example of how a complex chaotic behavior can arise from very simple nonlinear dynamical equation. This chaotic system was introduced and popularized by the biologist Robert May [29]. It was originally introduced as a demographic model of a typical predator–prey relationship. The chaotic behavior can be observed by varying the parameter r. When $r = 3.57$, this is the beginning of chaos, at the end of the period-doubling behavior. When $r > 3.57$, the system exhibits chaotic behavior.

The example of this behavior can be clearly seen from the bifurcation diagram in Fig. 1.

This work primarily consists of three case studies. All of them are focused on an estimation of three accessible control parameters for ETDAS control method to stabilize desired UPO, and a comparison of obtained results for used cost function. Desired UPOs are the following: p-1 (a fixed point) in the first case, p-2 (higher periodic orbit—oscillation between 2 values) in the second case and p-4 (also high periodic orbit—oscillation between 4 values) in the last case. All simulations were 50 times repeated for each EA strategy. The control method—ETDAS in the discrete form suitable for Logistic equation has the form (2).

$$\begin{aligned} x_{n+1} &= rx_n (1 - x_n) + F_n \\ F_n &= K \left[(1 - R) S_{n-m} - x_n \right] \\ S_n &= x_n + RS_{n-m} \end{aligned} \tag{2}$$

where K and R are adjustable constants, F is the perturbation, S is given by a delay equation utilizing previous states of the system and m is the period of m-periodic orbit to be stabilized. The perturbation F_n in Eqs. (2) may have an arbitrarily large value, which can cause diverging of the system outside the interval {0, 1}. Therefore, F_n should have a value between, $-F_{max}$ and F_{max}, and EA should find an appropriate value of this limitation to avoid diverging of the system.

2.2. The basic cost function

The proposal of the basic cost function (CF) is in general based on the simplest CF, which could be used only for the stabilization of p-1 orbit. The idea was to minimize the area created by the difference between the required state and the real system output on the whole simulation interval—τ_i.

However, another cost function had to be used for stabilizing of the higher periodic orbit. It was synthesized from the simple CF and other terms were added. In this case, it is not possible to use the simple rule of minimizing the area created by the difference between the required and actual state on the whole simulation interval—τ_i, due to many serious reasons, for example: degrading of the possible best solution by a phase shift of periodic orbit.

This CF, is in general based on the searching for desired stabilized periodic orbit and thereafter calculation of the difference between desired and found actual periodic orbit on the short time interval—τ_s (approx. 20–50 iterations) from the point, where the first minimum value of difference between desired and actual system output is found. Such a design of CF should secure the successful stabilization of p-1 and higher periodic orbit anywise phase shifted. The CF_{Basic} has the form (3).

$$CF_{Basic} = \text{penalization1} + \sum_{t=\tau_1}^{\tau_2} |TS_t - AS_t| \tag{3}$$

where:

TS—target state,
AS—actual state
τ_1—the first minimum value of difference between TS and AS
τ_2—the end of optimizing interval ($\tau_1 + \tau_s$)
penalization1 $= 0$ if $\tau_i - \tau_2 \geq \tau_s$;
penalization1 $= 10 * (\tau_i - \tau_2)$ if $\tau_i - \tau_2 < \tau_s$

(i.e. late stabilization).

2.3. The advanced targeting cost function

It was necessary to modify the definition of CF in order to decrease the average number of iteration required for the successful stabilization and avoidance of any associated problem. The CF_{Basic} is suitable for adding some term of penalization for slowly stabilizing solutions, thus it was modified for the use of all required UPOs. The CF value is multiplied by the number of iterations (NI) of the first found minimal value of difference between desired and actual system output (i.e. the beginning of fully stabilized UPO). To avoid problems associated with CF returning value 0 and to put the penalization to similar level as the non-penalized CF value, the small constant (SC) is added to CF value before penalization (multiplying by NI).

Generally, there exist two possible approaches for defining the SC value. The first one capitalizes the previous simulation results with CF basic and experiences, whereas the second approach uses the automatically computed value.

The next two proposals of CF design are based on the second approach, which should avoid any problems associated with defining the value of small constant, SC in advance (especially for stabilization of higher periodic orbit). The SC value (5) is computed with the aid of power of non-penalized basic part of CF (4).

$$\text{ExpCF} = \log_{10}\left(\sum_{t=\tau_1}^{\tau_2} |TS_t - AS_t| + 10^{-15}\right) \tag{4}$$

$$SC = 10^{\text{ExpCF}}. \tag{5}$$

In general, there exists two possible ways for applying the multiplication by number of iterations required for stabilization (NI). The first version of final design of targeting CF ($CF_{T1\text{-}ADV}$) has the form (6). Here the sum of basic part of CF and automatically computed SC is multiplied by NI. Finally, to avoid the problems with fast stabilization, only for limited range of initial conditions, the final CF value is computed as a sum of n repeated simulations for different initial conditions. Consequently, the EA should find the robust solutions securing the fast targeting into desired behavior of system for almost any initial conditions.

$$CF_{T1\text{-}Adv} = \sum_{1}^{n} NI\left(SC + \text{penalization1} + \sum_{t=\tau_1}^{\tau_2} |TS_t - AS_t|\right) \tag{6}$$

where: x_{initial} is from the range 0.05–0.95 and uses step 0.1.

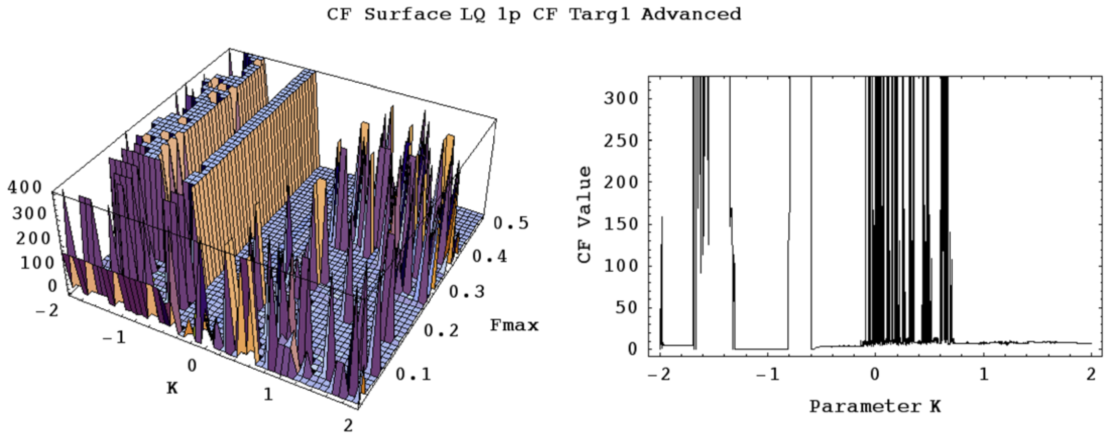

Fig. 2. Dependence of CF value on parameter K and F_{max} (left); and parameter K (right), p-1 orbit, $x_{initial} = 0.8$, $CF_{T1\text{-}ADV}$, $R = 0.50$, $F_{max} = 0.49$.

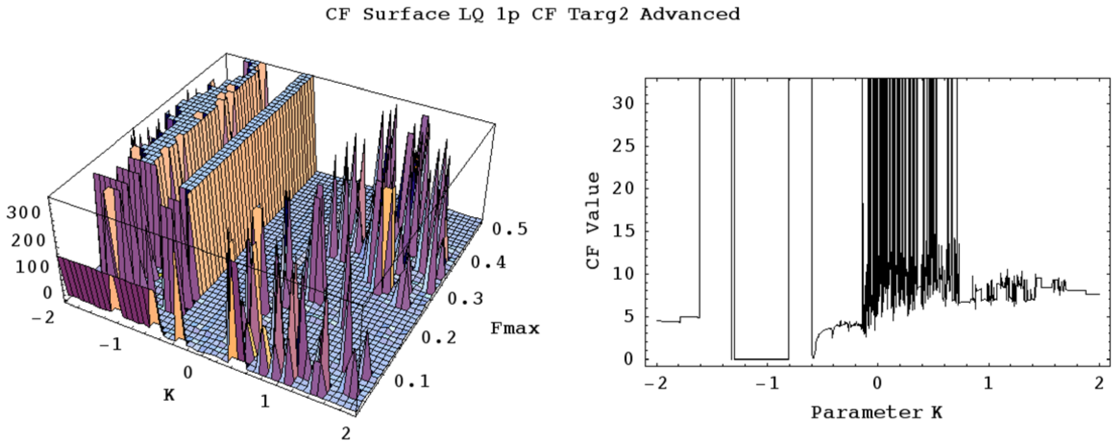

Fig. 3. Dependence of CF value on parameter K and F_{max} (left); and parameter K (right), p-1 orbit, $x_{initial} = 0.8$, $CF_{T2\text{-}ADV}$, $R = 0.50$, $F_{max} = 0.50$.

In the second version of targeting CF ($CF_{T2\text{-}ADV}$), there is only a slight change in comparison with the previous proposal. Here the number of steps for stabilization (NI) multiplies only the small constant (SC) which is counted in the same way as in the previous case (5). This version of targeting CF ($CF_{T2\text{-}ADV}$) has the form (7).

$$CF_{T2\text{-}Adv} = \sum_{1}^{n} \left((\mathrm{NI} \cdot \mathrm{SC}) + \mathrm{penalization1} + \sum_{t=\tau_1}^{\tau_2} |\mathrm{TS}_t - \mathrm{AS}_t| \right). \tag{7}$$

3. Graphical overview

The difference between the proposed CFs can be clearly seen in Figs. 2–5, which shows the dependence of CF values on the adjustable parameters K and F_{max} (left part of figures — 3D diagrams) and the dependence of CF values on the adjustable parameters K (right part of figures — 2D diagrams). Remaining parameters were set at the best values reached in optimizations; consequently the two-dimensional diagram always shows the section of global minimum. From these figures, it is obvious as to how a small change in the CF design can influence the nonlinearity and unpredictability of CF surface and how the nonlinearity of CF surface increase together with growing order of UPO to be stabilized.

4. Optimization algorithms

The experimentation utilized the stochastic optimization algorithms DE [12,30] and SOMA [13]. They were chosen due to their proven ability to converge towards the global optimum.

DE is a population-based optimization method that works on real-number-coded individuals. For each individual $\vec{x}_{i,G}$ in the current generation G, DE generates a new trial individual $\vec{x}'_{i,G}$ by adding the weighted difference between two randomly

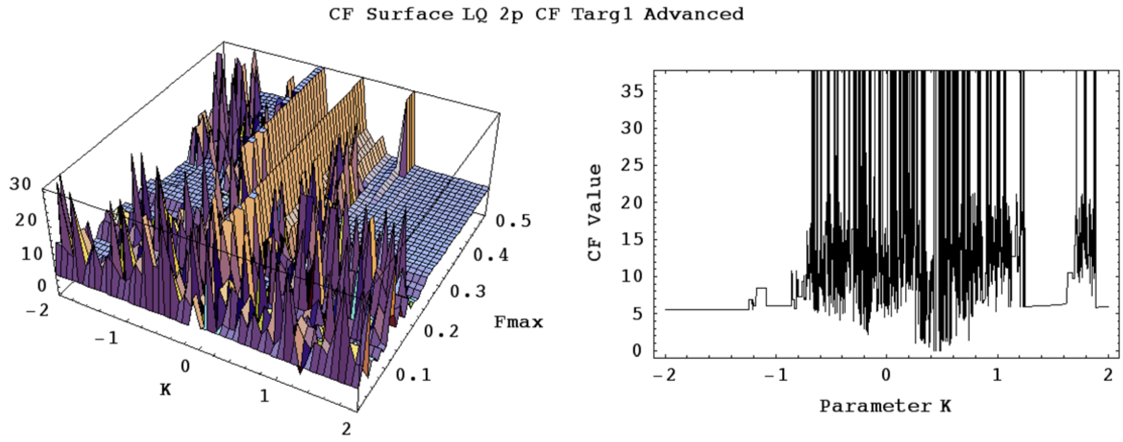

Fig. 4. Dependence of CF value on parameter K and F_{max} (left); and parameter K (right), p-2 orbit, $x_{initial} = 0.8$, $CF_{T1\text{-}ADV}$, $R = 0.24$, $F_{max} = 0.21$.

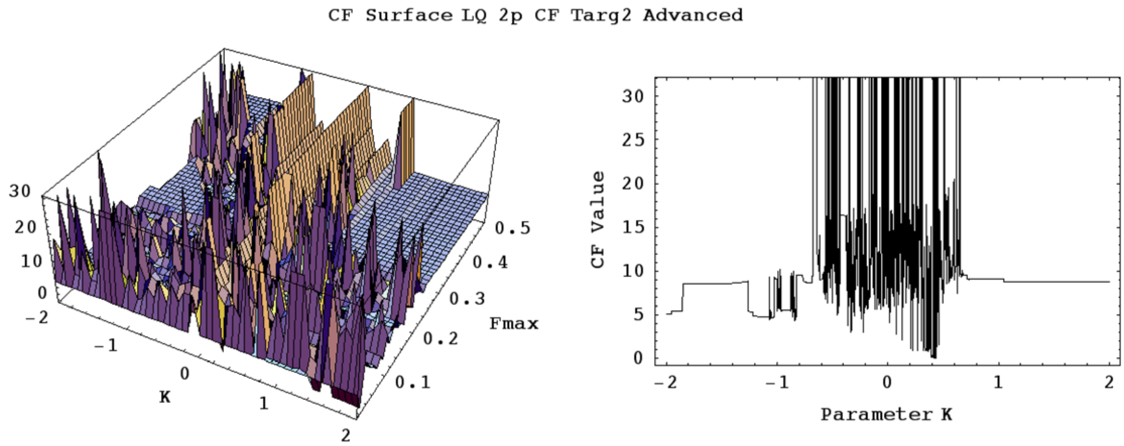

Fig. 5. Dependence of CF value on parameter K and F_{max} (left); and parameter K (right), p-2 orbit, $x_{initial} = 0.8$, $CF_{T2\text{-}ADV}$, $R = 0.19$, $F_{max} = 0.28$.

Table 1
Description of used DE strategies.

Strategy	Formulation
DERand1Bin	$u_{i,G+1} = x_{r1,G} + F \bullet (x_{r2,G} - x_{r3,G})$
DERand2Bin	$u_{i,G+1} = x_{r5,G} + F \bullet (x_{r1,G} - x_{r2,G} - x_{r3,G} - x_{r4,G})$
DEBest2Bin	$u_{i,G+1} = x_{Best,G} + F \bullet (x_{r1,G} - x_{r2,G} - x_{r3,G} - x_{r4,G})$
DELocalToBest	$u_{i,G+1} = x_{i,G} + F_{Rand}(x_{Best,G} - x_{i,G}) + F \bullet (x_{r1,G} - x_{r2,G})$
DERand1DIter	$u_{i,G+1} = x_{r1,G} + F_{NormRand} \bullet (x_{r2,G} - x_{r3,G})$
DEBest1JIter	$u_{i,G+1} = x_{Best1,G} + F_{NormRand} \bullet (x_{r1,G} - x_{r2,G})$

selected individuals $\vec{x}_{r1,G}$ and $\vec{x}_{r2,G}$ to a randomly selected third individual $\vec{x}_{r3,G}$. The resulting individual $\vec{x}'_{i,G}$ is crossed-over with the original individual $\vec{x}_{i,G}$. The fitness of the resulting individual, referred to as a perturbed vector $\vec{u}_{i,G+1}$, is then compared with the fitness of $\vec{x}_{i,G}$. If the fitness of $\vec{u}_{i,G+1}$ is greater than the fitness of $\vec{x}_{i,G}$, then $\vec{x}_{i,G}$ is replaced with $\vec{u}_{i,G+1}$; otherwise, $\vec{x}_{i,G}$ remains in the population as $\vec{x}_{i,G+1}$. DE is quite robust, fast, and effective, with global optimization ability. It does not require the objective function to be differentiable, and it works well even with noisy and time-dependent objective functions. Please refer to Table 1 for the description of six used strategies.

Scaling vector F is replaced with a randomly generated vector F_{Rand} in the DELocalToBest strategy and with a randomly generated vector $F_{NormRand}$ with normal distribution in strategies DEBest1JIter and DERand1DIter.

SOMA works with groups of individuals (population) whose behavior can be described as a competitive–cooperative strategy. The construction of a new population of individuals is not based on evolutionary principles (two parents produce offspring) but on the behavior of social group, e.g. a herd of animals looking for food. This algorithm can be classified as an algorithm of a social environment. To the same group of algorithms, Particle Swarm Optimization (PSO) algorithm can also be put in, sometimes called swarm intelligence. In the case of SOMA, there is no velocity vector as in PSO, only the position of individuals in the search space is changed during one generation, here called 'migration loop'.

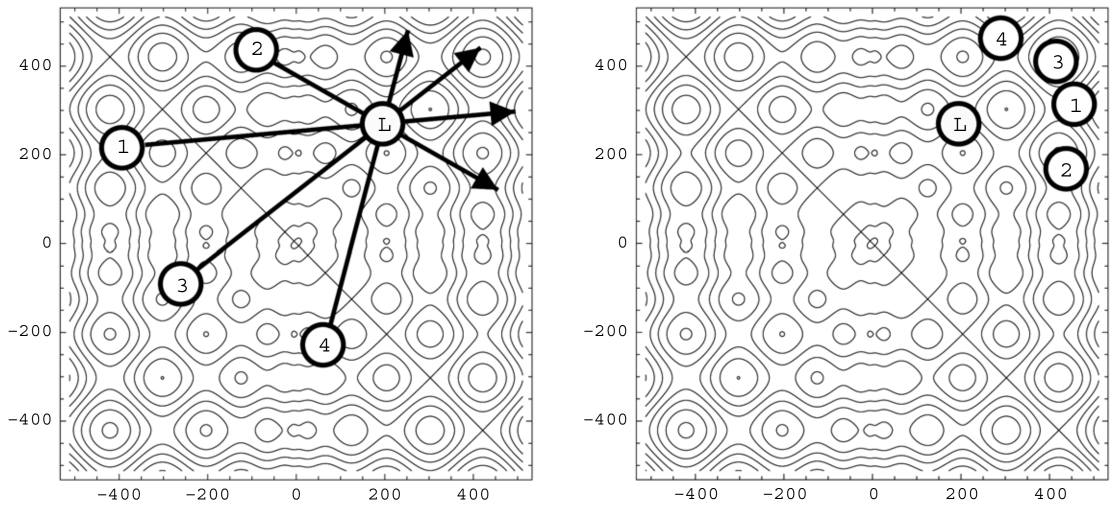

Fig. 6. Basic principle of SOMA.

Table 2
Used versions of SOMA.

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

The rules are as follows: In every migration loop the best individual is chosen, i.e. individual with the minimum cost value, which is called the Leader. An active individual from the population moves in the direction towards the Leader in the search space. At the end of the crossover, the position of the individual with minimum cost value is chosen. If the cost value of the new position is better than the cost value of an individual from the old population, the new one appears in new population. Otherwise the old one remains there. The crossover is described by Eq. (8) and the main principle is depicted in Figs. 6 and 7.

$$x_{i,j}^{\mathrm{ML}+1} = x_{i,j,\mathrm{START}}^{\mathrm{ML}} + (x_{L,j}^{\mathrm{ML}} - x_{i,j,\mathrm{START}}^{\mathrm{ML}}) * t * \mathrm{PRTVector}_j \tag{8}$$

where

$x_{i,j}^{\mathrm{ML}+1}$—value of i-individual's j-parameter, in step t in migration loop ML + 1
$x_{i,j,\mathrm{START}}^{\mathrm{ML}}$—value of i-individual's j-parameter, Start position in actual migration loop
$x_{L,j}^{\mathrm{ML}}$—value of Leader's j-parameter in migration loop ML
t-step ∈ ⟨0, by Step to, PathLength⟩
PRTVector—is vector of ones and zeros dependent on PRT. If random number from interval ⟨0, 1⟩ is less than PRT, then 1 is saved to PRTVector, otherwise it is 0.

5. Experimental results

Four strategies of SOMA and six strategies of DE were used for all simulations. (See Tables 2 and 3). See also Tables 4 and 5 for parameter settings of the used EAs. The parameter setup for both heuristic was based on the previous numerous experiments with chaos and nonlinear systems and on the recommendations proposed by the authors of these evolutionary algorithms.

Parameters for the optimizing algorithm were set up in such a way in order to reach the same value of maximal CF evaluations for all used strategies. Each SOMA and DE strategy has been applied 50 times in order to find the actual optimum.

Both evolutionary algorithms are terminated after reaching the maximal value of CFE—Cost Function Evaluation. This value is evaluated from the number of generations and population size (DE); and number of migration loops, population size, step size and PathLength (SOMA).

Here is the list of desired UPOs for $r = 3.8$:
p-1 (fixed point): $x_F = 0.73842$.
p-2 orbit: $x_1 = 0.3737, x_2 = 0.8894$.
p-4 orbit: $x_1 = 0.3038, x_2 = 0.8037, x_3 = 0.5995, x_4 = 0.9124$.

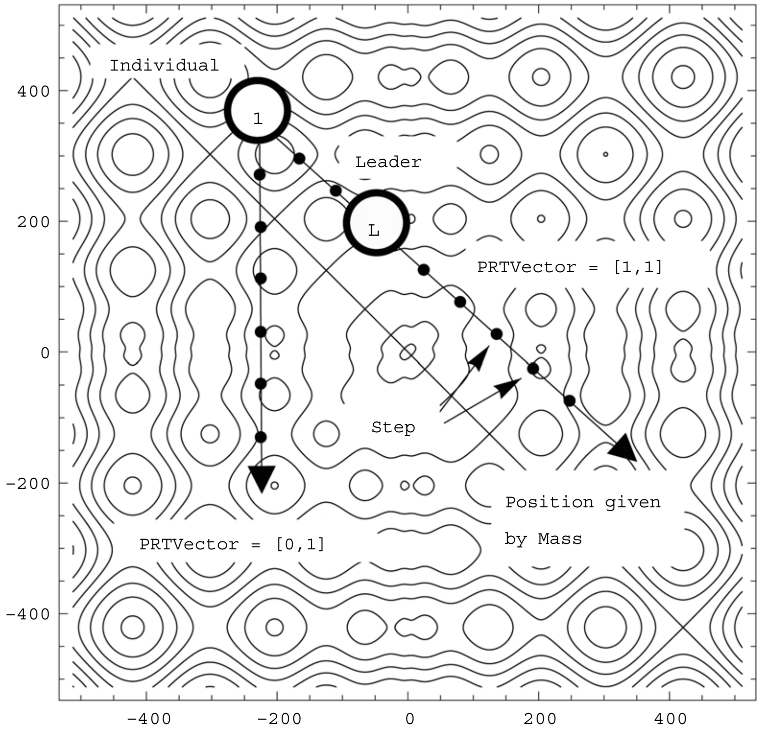

Fig. 7. Basic principle of crossover in SOMA, PathLength is replaced here by mass.

Table 3
Used strategies of DE.

Index	Algorithm/Version
5	DERand1Bin
6	DERand2Bin
7	DEBest2Bin
8	DELocalToBest
9	DERand1DIter
10	DEBest1JIter

Table 4
Parameter settings for SOMA.

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

Table 5
Parameter settings for DE.

Parameter	Values
F	0.9
Cr	0.2
PopSize	25
Generations	215
Max. CF Evaluations (CFE)	5375

The optimization interval for p-1 orbit was $\tau_i = 100$ iterations, for higher periodic orbits (p-2 and p-4) it was mostly $\tau_i = 150$ iterations.

Table 6
Results for p-1 orbit, $CF_{T1\text{-}ADV}$ and $CF_{T2\text{-}ADV}$.

CF version	$CF_{T1\text{-}ADV}$		$CF_{T2\text{-}ADV}$	
EA version	SOMA	DE	SOMA	DE
K	−0.9351	−0.9323	−0.9336	−0.9307
F_{max}	0.4888	0.4842	0.4957	0.4995
R	0.4990	0.4985	0.4994	0.4973
CF val.	2.57×10^{-14}	$\mathbf{2.56 \times 10^{-14}}$	$\mathbf{2.54 \times 10^{-14}}$	2.56×10^{-14}
Avg. IStab	34	33	32	32

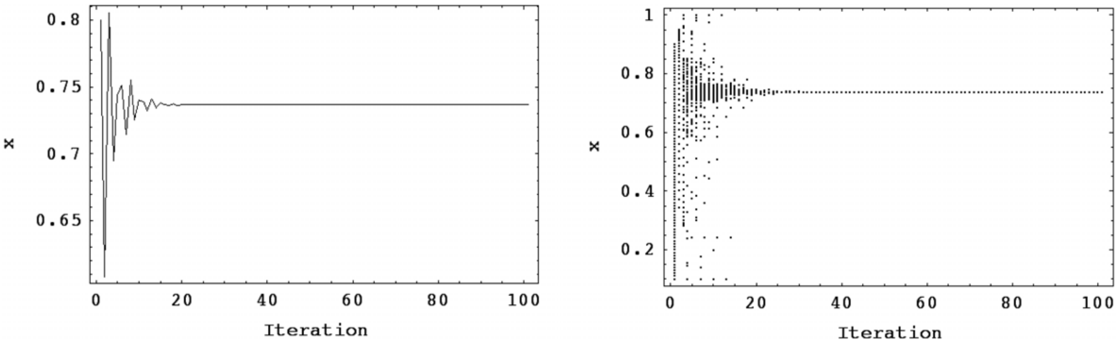

Fig. 8. Best solution: p-1 orbit, CF_{Basic}, DERand1Bin.

The ranges of all estimated parameters were these:

$$-2 \leq K \leq 2, \qquad 0 \leq F_{max} \leq 0.5 \quad \text{and} \quad 0 \leq R \leq 0.99.$$

The best solution for each strategy of SOMA and DE and both versions of CF are shown in Tables 6–8. Presented results in these three tables include estimated optimal values of three parameters for ETDAS method (K, F_{max}, R) and final CF Value for the best individual solution together with other important optimization result like average number of iterations required for successful stabilization for 50 repeated simulations (Avg. IStab). The best individual solutions are highlighted by bold numbers.

The Figs. 8–16 show simulation of chaos control by means of ETDAS method adjusted using the best individual solution given either by SOMA or DE. These figures consists of two parts—simulation with identical initial conditions for variables x of the logistic equation as for the optimization (left part of the image) and the complex simulation of the best individual solution with distributed initial conditions in the range $0 < x_{initial} < 1$ (100 samples — right part of the image), which gives weight to the argument that presented CFs (6) and (7) are able to give optimal parameter setup for robust control of chaotic system for any initial condition x.

Each section dealing with the different target state of stabilized chaotic system (different order of UPO) consists of series of three figures. The first one shows the results from previous research given by CF_{Basic} (4) [19]. It is followed by two figures depicting the simulation output by means of optimal parameter setup given by EAs and presented targeting CFs (6) and (7). From these three figures, it is obvious the positive influence of presented targeting cost functions to the faster stabilization of chaotic system and, of course, also the influence of a small change in CF design to the quality of stabilization.

5.1. Control of chaos, p-1 orbit

For the best individual solutions given by $CF_{T1\text{-}ADV}$ (DELocalToBest) and $CF_{T2\text{-}ADV}$ (SOMA ATO), please refer to Table 6. Each EA strategy gave almost identical result of CF value for the best solution. From the series of complex simulations depicted in Figs. 8–10, it is obvious, that the control parameters estimated in the optimizations ensured very fast and precise reaching of a desired state for any initial conditions.

5.2. Control of chaos, p-2 orbit

This case is focused on the stabilization of p-2 orbit.

The best results given by both evolutionary algorithms are shown in Table 7. The outputs of simulations are depicted in Figs. 11–13.

The results given by $CF_{T1\text{-}ADV}$ show the following attributes: rapid achievement of desired UPO in comparison with CF_{Basic} (see Table 9)—only 13 iterations were required for the stabilization of the best solution, together with very poor performance of EA, i.e. the proportion of the solutions with either perfect stabilization or temporary or possibly none at all. Also, relatively considerable period doubling or oscillating in the close neighborhood of desired UPO arose (Fig. 12).

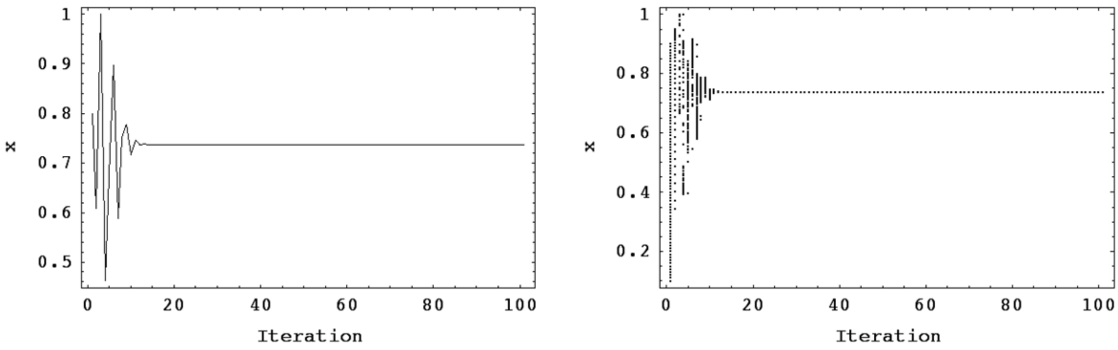

Fig. 9. Best solution: p-1 orbit, $CF_{T1\text{-}ADV}$, DELocalToBest.

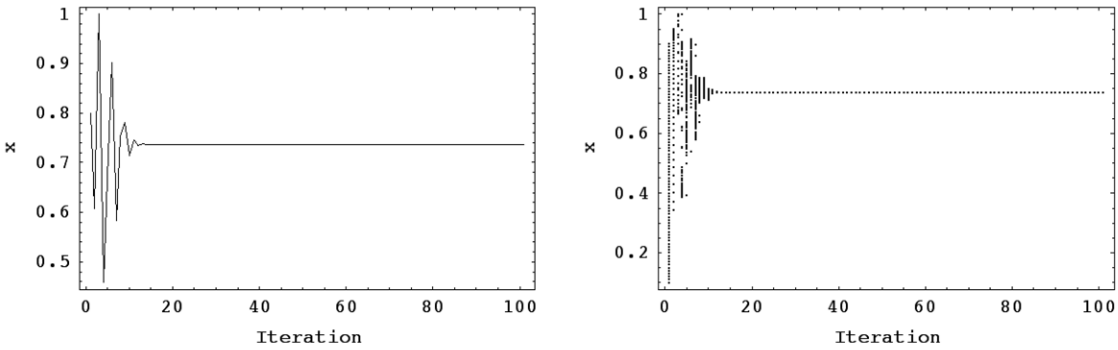

Fig. 10. Best solution: p-1 orbit, $CF_{T2\text{-}ADV}$, SOMA ATO.

Table 7
Results for p-2 orbit, $CF_{T1\text{-}ADV}$ and $CF_{T2\text{-}ADV}$.

CF version	$CF_{T1\text{-}ADV}$		$CF_{T2\text{-}ADV}$	
EA version	SOMA	DE	SOMA	DE
K	0.4236	0.4047	0.4093	0.4092
F_{max}	0.2086	0.4271	0.2824	0.2794
R	0.2421	0.1819	0.1949	0.1847
CF val.	**15.0221**	19.7431	0.6392	**0.2353**
Avg. IStab	60	56	31	38

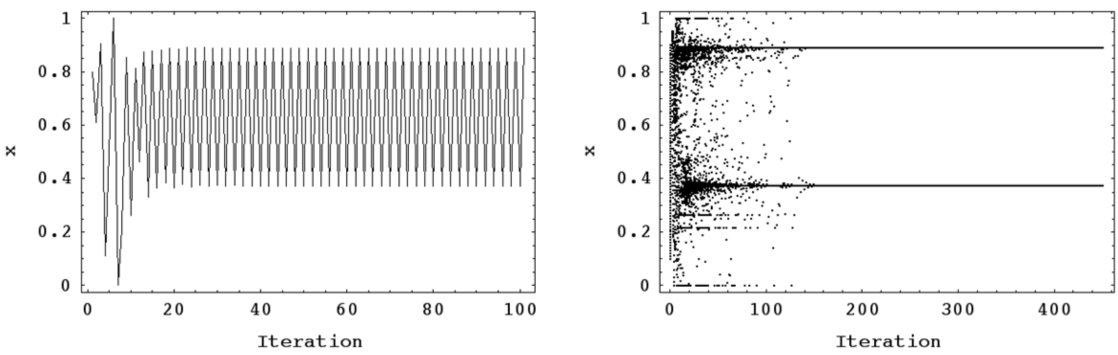

Fig. 11. Best solution: p-2 orbit, CF_{Basic}, SOMA ATR.

In the case of $CF_{T2\text{-}ADV}$, the two main above mentioned problems with period doubling (i.e. low-quality stabilization) and very poor performance of EAs in finding the stabilizing securing solutions were noticeably suppressed.

5.3. Control of chaos, p-4 orbit

See Table 8 for the results of this optimization. The simulations of the best individual solutions are depicted in Figs. 14–16.

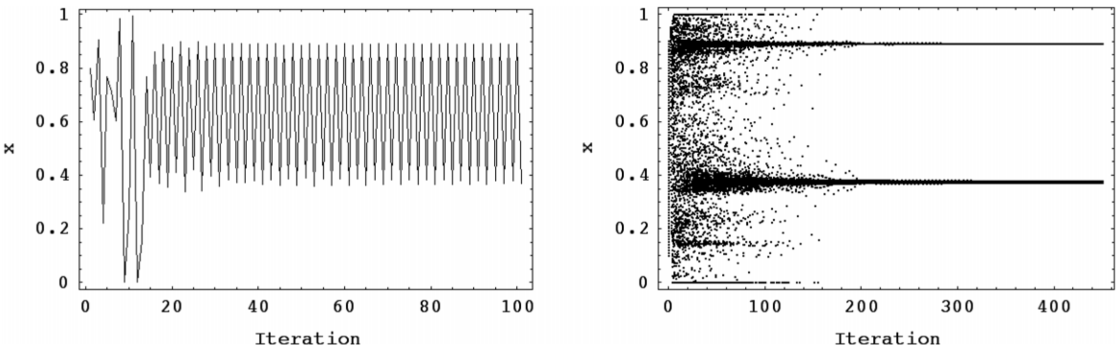

Fig. 12. Best solution: p-2 orbit, $CF_{T1\text{-}ADV}$, SOMA ATA.

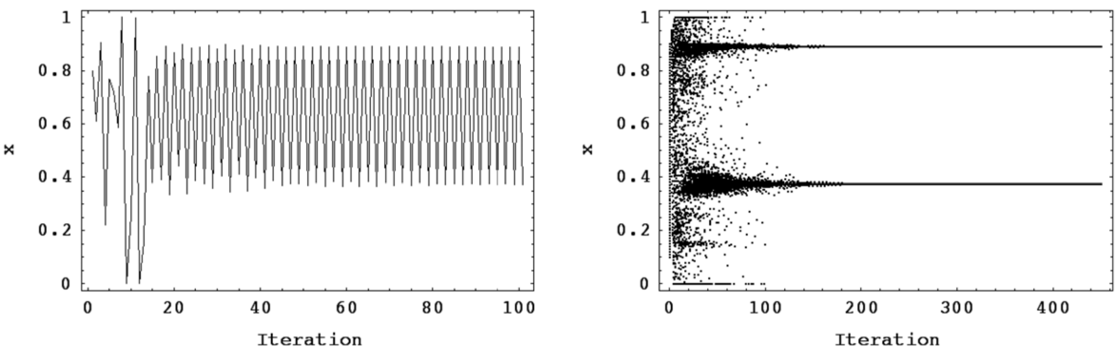

Fig. 13. Best solution: p-2 orbit, $CF_{T2\text{-}ADV}$, DERand1DIter.

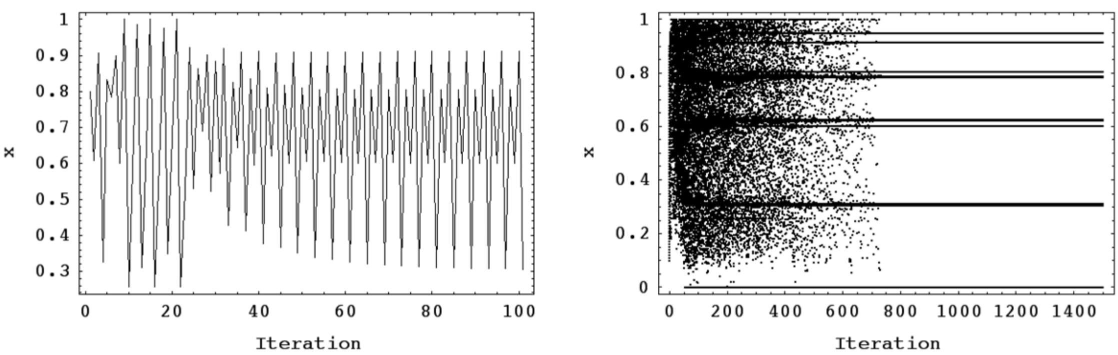

Fig. 14. Best solution: p-4 orbit, CF_{Basic}, SOMA ATO.

Table 8
Results for p-4 orbit, $CF_{T1\text{-}ADV}$ and $CF_{T2\text{-}ADV}$.

CF version	$CF_{T1\text{-}ADV}$		$CF_{T2\text{-}ADV}$	
EA version	SOMA	DE	SOMA	DE
K	−0.6394	−0.6176	−0.5132	−0.5355
F_{max}	0.1107	0.0906	0.1249	0.1267
R	0.7255	0.7049	0.5960	0.6508
CF val.	**0.1229**	0.6302	2.1906	**0.0031**
Avg. IStab	450	463	194	199

As a conclusion of this case study, it is possible to say that also in the case of p-4 orbit and optimizations by means of $CF_{T1\text{-}ADV}$, the phenomenon of faster targeting of desired UPO (only 31 iterations for the best individual solution) for a wide range of initial conditions occurs at the cost of very poor performance of both DE and SOMA (only 2% of given solutions led to stabilization).

In the case of $CF_{T2\text{-}ADV}$, the presented results show positive features as in the case of p-2 orbit and from the comparison with CF_{Basic} (Fig. 14—not successful stabilization), it follows that stabilization was reached precisely.

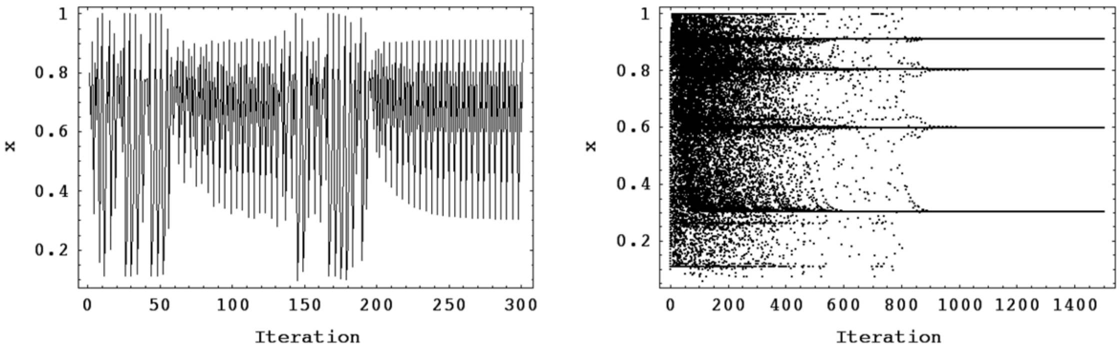

Fig. 15. Best solution: p-4 orbit, $CF_{T1\text{-}ADV}$, SOMA ATR.

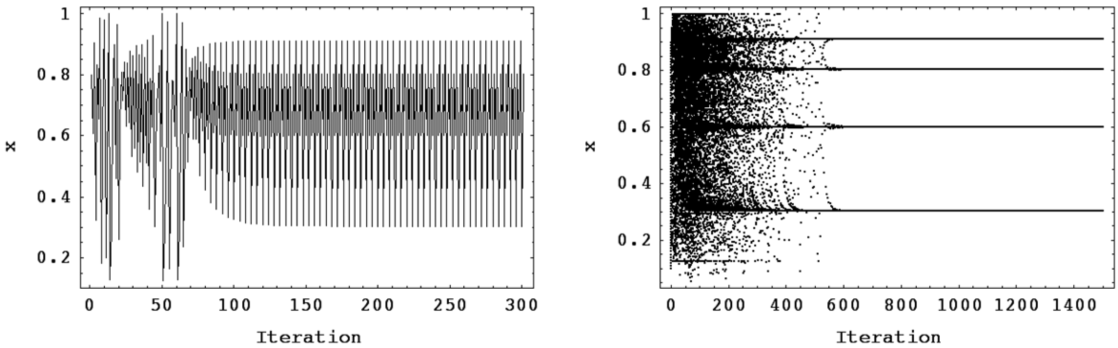

Fig. 16. Best solution: p-4 orbit, $CF_{T2\text{-}ADV}$, DERand1Bin.

Table 9
Comparison of results for five CFs.

UPO	p-1	p-2	p-4
CF Basic	97	123	∞(208)
CF Targ1	32	112	∞(205)
CF Targ2	31	112	∞(190)
CF Targ1 Adv.	36	58	396 (450)
CF Targ2 Adv.	33	64	196 (248)

The results for $CF_{T1\text{-}ADV}$ were in the first instance better (Figs. 15 and 16) however, in the case of $CF_{T2\text{-}ADV}$, more than 50% of the samples were stabilized within the first 150 iterations. This is apparent from the notable difference of Avg. IStab values.

6. Investigation on results of chaos control

This section presents the combination of research [26–28] and the results presented here. Please refer to Table 9 for the comparison of average number of iterations required for stabilization, which was elaborated for five CFs and all desired UPOs. The infinity value and the number in braces represent the case, where the stabilization was reached only for the best solution and this solution failed within complex simulation. Here, a gradual decrease of average IStab value together with development and testing of complex targeting cost functions can be clearly seen.

The first CF Basic gives satisfactory results and can be used wherever the good quality of stabilization is expected and the speed of stabilization and "universality of this solution" for a wider range of initial conditions are not decisive.

In case of targeting cost functions, the results for p-1 orbit are significantly better, however, on the other hand the slightly better results for higher periodic orbits were achieved at the cost of arising of problem with worse performance of EAs and obtaining of solutions with only temporary stabilization or none at all. This negative phenomenon culminates in the case of $CF_{T1\text{-}ADV}$. Finally, $CF_{T2\text{-}ADV}$ design suppresses all mentioned problems and gives excellent performance from the point of view of quickness and quality of stabilization for any initial conditions.

7. Conclusion

From the optimization results it follows, that they are extremely sensitive to the construction of used CF and any small change in the design of CF can cause radical improvement of the system behavior (as in case of CF Targ2 Advanced), but of course, on the other hand can cause a worsening of observed parameters and behavior of chaotic system as well.

When comparing both used heuristics, it is obvious, that the performance of four SOMA strategies and six canonical strategies of DE is very similar and satisfactory.

The results for CF Targ1 were for the first view satisfactory, but two very momentous problems arose—period doubling and very poor performance of EAs. These problems uncovered hidden non-optimal structure of CF Targ1.

In the last proposal of CF Targ2, there were only slight changes in CF design, but from the presented results it can be seen, how such a small change can influence the performance of a controlled system, especially when it is an extremely sensitive chaotic system. CF Targ2 advanced gives excellent results for simulations with a wide range of initial conditions and seem to be the choice for the task of finding of ''universal and robust solution''. The problems with poor EA performance and period doubling were mostly suppressed here. The only disadvantage of this proposal is the relatively big computational-time demands.

According to all results presented here, it is planned that the main activities will be focused on testing of evolutionary deterministic chaos control in continuous-time and high-order systems and finally testing of evolutionary real-time chaos control.

Acknowledgements

This work was supported by grant No. MSM 7088352101 of the Ministry of Education of the Czech Republic and by grants of Grant Agency of Czech Republic GACR 102/09/1680.

References

[1] E.J. Kostelich, E. Ott, C. Grebogi, J.A. Yorke, Higher-dimensional targeting, Physical Review E 47 (1) (1993) 305–310.
[2] E.M. Bollt, E.J. Kostelich, Optimal targeting of chaos, Physics Letters A 245 (1998) 399–406.
[3] C.M. Bird, P.J. Aston, Targeting in the presence of noise, Chaos, Solitons & Fractals 9 (1–2) (1998) 251–259.
[4] R. Ramaswamy, S. Sinha, N. Gupte, Targeting chaos through adaptive control, Physical Review E 57 (3) (1998) 2507–2510.
[5] J. Starrett, Time-optimal chaos control by center manifold targeting, Physical Review E 66 (4) (2002) 6206–6211.
[6] S. Iplikci, Y. Denizhan, Control of chaotic systems using targeting by extended control regions method, Physica D: Nonlinear Phenomena 150 (3–4) (2001) 163–176.
[7] S. Iplikci, Y. Denizhan, An improved neural network based targeting method for chaotic dynamics, Chaos, Solitons & Fractals 17 (2–3) (2003) 523–529.
[8] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company Inc., 1989.
[9] R. Haupt, S. Haupt, Practical Genetic Algorithms, 2nd ed., John Wiley & Sons, USA, 2004.
[10] M. Clerc, Particle Swarm Optimization, ISTE Publishing Company, 2009.
[11] G. Onwubolu, B. Babu, New Optimization Techniques in Engineering, Springer, New York, 2004.
[12] K. Price, An introduction to differential evolution, in: D. Corne, M. Dorigo, F. Glover (Eds.), New Ideas in Optimization, McGraw-Hill, London, UK, ISBN: 007-709506-5, 1999, pp. 79–108.
[13] I. Zelinka, SOMA—self organizing migrating algorithm, in: B.V. Babu, G. Onwubolu (Eds.), New Optimization Techniques in Engineering, vol. 33, Springer-Verlag, 2004 (Chapter 7).
[14] W. Just, Principles of time delayed feedback control, in: H.G. Schuster (Ed.), Handbook of Chaos Control, Wiley-Vch, ISBN: 3-527-29436-8, 1999.
[15] K. Pyragas, Continuous control of chaos by self-controlling feedback, Physics Letters A 170 (1992) 421–428.
[16] K. Pyragas, Control of chaos via extended delay feedback, Physics Letters A 206 (1995) 323–330.
[17] E. Ott, C. Greboki, J.A. Yorke, Controlling chaos, Physical Review Letters 64 (1990) 1196–1199.
[18] O.J. Kwon, Targeting and stabilizing chaotic trajectories in the standard map, Physics Letters A 258 (1999) 229–236.
[19] H. Richter, K.J. Reinschke, Optimization of local control of chaos by an evolutionary algorithm, Physica D 144 (2000) 309–334.
[20] H. Richter, An evolutionary algorithm for controlling chaos: the use of multi-objective fitness function, in: Lecture Notes in Computer Science, vol. 2439, 2002, pp. 308–320.
[21] G. Onwubolu, D. Davendra (Eds.), Differential evolution: A Handbook for Permutation-Based Combinatorial Optimization, Springer, Germany, 2009.
[22] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Transactions on Evolutionary Computation 13 (2) (2009) 398–417.
[23] X. Yuan, Y. Zhang, L. Wang, Y. Yuan, An enhanced differential evolution algorithm for daily optimal hydro generation scheduling, Computers & Mathematics with Applications 55 (11) (2008) 2458–2468.
[24] L. Coelho, Self-organizing migration algorithm applied to machining allocation of clutch assembly, Mathematics and Computers in Simulation 80 (2) (2009) 427–435.
[25] I. Zelinka, S. Celikovsky, H. Richter, G. Chen (Eds.), Evolutionary Algorithms and Chaotic Systems, Springer, Germany, 2010.
[26] I. Zelinka, R. Senkerik, E. Navratil, Investigation on evolutionary optimitazion of chaos control, Chaos, Solitons & Fractals 40 (2009) 111–129.
[27] R. Senkerik, I. Zelinka, E. Navratil, Cost function design for evolutionary optimization of chaos control, in: Proc. 9th European Control Conference 2007, ECC'07, Kos, Greece, 2–5 July 2007, pp. 1682–1687.
[28] R. Senkerik, I. Zelinka, E. Navratil, Design of targeting cost function for evolutionary optimization of chaos control, in: Proc. ECMS 2007, Prague, Czech Republic, June 2007, pp. 345–350.
[29] R.M. May, Stability and Complexity in Model Ecosystems, Princeton University Press, ISBN: 0-691-08861-6, 2001.
[30] K. Price, R. Storn, Differential evolution homepage, 2001 [online 27.2.2009]. Available at: http://www.icsi.berkeley.edu/~storn/code.html.