Tomas Bata University in Zlín
Faculty of Applied Informatics

Doctoral Thesis

Modern methods of development and modification of evolutionary computational techniques

Author: **Ing. Michal Pluháček**
Branch of study: Engineering Informatics
Supervisor: prof. Ivan Zelinka, Ph.D.
Advisor: assoc. prof. Roman Šenkeřík Ph.D.

Zlín, 2015

At this place I would like to thank all the people who helped me along the way to finish this work. My family, friends and coleagues who supported me and two exceptional men. My supervisor prof. Ivan Zelinka Ph.D. who introduced me to the world of science and amazed me with his lectures about evolutionary techniques and briliant ideas. And last but not least I would like to thank my advisor, colleague and friend assoc. prof. Roman Šenkeřík Ph.D. who was my guide during the Ph.D. study and helped me achieve many things I would never dare even think of.

"Let us do perhaps the slightest thing in the world, but let us do it best."

Tomas Bata (1876 - 1932)

ABSTRAKT

Hlavním cílem této práce je ukázat, že je možné vylepšit výkonnost evolučních výpočetních technik pro spojitou optimalizaci s jednou výstupní veličinou využitím různých metod modifikace. Je ukázáno, že s využitím relativně jednoduchým modifikací je možné zlepšit výkonnost algoritmu Rojení částic (Particle Swarm Optimization - PSO) jak pro umělé testovací funkce, tak pro reálné problémy.

Nejdříve je vysvětlena důležitost optimalizace a základní principy evoluční optimalizace. Dále jsou představeny moderní trendy v návrhu modifikací evolučních výpočetních technik spolu s oblastmi využití. Vysvětleno je též zaměření práce na algoritmus Rojení částic.

Dále jsou v práci popsány základy tzv. „Swarm Intelligence“ či inteligence hejna a významní zástupci této třídy evolučních technik. Algoritmus Rojení částic použitý v této práci je popsán detailně. Popsány jsou také využité testovací funkce.

Jelikož se významná část tohoto výzkumu zabývá užitím generátorů pseudonáhodných čísel založených na chaotických systémech, je teoretická část uzavřena detailním popisem užitých chaotických systémů včetně rovnic a grafů.

V experimentální části jsou prezentovány výsledky dlouhodobého výzkumu. Nejdříve je detailně popsán algoritmus PSO využívající chaos. Způsob implementace chaotických sekvencí jako generátorů pseudo-náhodných čísel je vysvětlen a výkonnost a chování PSO algoritmu s těmito generátory je detailně prozkoumána. Dále je prezentován ladící experiment. První část je uzavřena ukázkovou aplikací chaosem obohaceného PSO algoritmu na modelový případ návrhu PID regulátoru.

V další sekci je prezentován tzv. multi-chaotický přístup pro PSO. Jedná se o velmi slibnou metodu vyvinutou během tohoto výzkumu. V tomto je přístupu je v rámci jednoho běhu algoritmu využito více chaotických generátorů pseudonáhodných čísel. Tímto způsobem je možné vylepšit výkonnost algoritmu a upravit chování roje požadovaným způsobem. Je uvedeno i využití tohoto přístupu pro jinou evoluční výpočetní techniku – algoritmus Diferenciální evoluce.

Během výzkumu chaotického PSO byla detailně studována vnitřní dynamika

algoritmu PSO. Jako reakce na získané poznatky bylo navrženo a otestováno několik modifikací algoritmu PSO. Jako první je popsána tzv. „Multiple-choice“ strategie pro PSO. V tomto návrhu je vytvořen heterogenní roj a jednotlivé role jsou rozděleny náhodně. Jako druhý příklad úspěšné modifikace PSO algoritmu je uveden nově navržený tzv. shromažďovací (Gathering) algoritmus. V tomto algoritmu je využit tzv. lavinový efekt či efekt sněhové koule ke zdůraznění slibných regionů pomocí shromáždění množství částic. Tímto přístupem je možné vyhnout se problémům typickým pro algoritmus využívající pevný bod pro atrakci částic.

Výkonnost všech popsaných algoritmů byla testována na typicky využívaných testovacích funkcích a výsledky jsou srovnány s obyčejným PSO či zástupci nejnovějších algoritmů Výsledky výzkumu byly průběžně publikovány a presentovány na mezinárodních konferencích a byly velmi dobře přijaty.

Výsledky získané během tohoto výzkumu umožňuji tvrdit, že výkonnost evolučních výpočetních metod může být vylepšena využitím různých moderních metod, jako jsou například chaotické sekvence či modifikace vnitřních principů algoritmů.

SUMMARY

The main aim of this work is to show that it is possible to improve the performance of evolutionary computational techniques for single-objective continuous optimization problems by various modification methods. It is shown that by relatively simple modifications it is possible to improve the performance of Particle swam optimization algorithm on both artificial benchmark functions and real-world problems.

Firstly it is introduced the importance of optimization and the basic principles of evolutionary optimization. Further the modern trends in modification of evolutionary computational techniques are introduced alongside with the areas of application for these methods. Also the thesis focus on Swarm intelligence representative Particle swarm optimization algorithm is explained.

Further the basics of swarm intelligence are described alongside with notable representatives of this category of evolutionary techniques. The Particle Swarm optimization algorithm that has been used in this work is described in detail. Used benchmarks are also described. As a significant part of the research dealt with using of pseudo-random number generators based on chaotic systems, the theoretical part concludes with detailed description of used chaotic systems including equations and plots.

In the experimental part the results of long-term research are presented. Firstly the Chaos PSO is described in detail. The implementation of chaotic sequences as pseudo-random number generators is explained and the performance and behavior of PSO algorithm driven by chaotic pseudo-random number generator is investigated in detail. Further a tuning experiment is presented. The first part concludes with an example application of the chaos driven PSO algorithm on a model task of PID controller design.

In the next section it is presented the multi-chaotic approach for chaos driven PSO - promising method developed during this work. In this approach multiple chaotic pseudo-random number generators are used within one run of the algorithm and enhance its performance by changing the behavior of the swarm in a desirable way. It is also shown the utilization of this approach for another evolutionary computational technique – the Differential evolution algorithm.

During the research of chaos driven PSO the inner dynamics of the PSO algorithm were studied in detail. As a reaction several modifications of PSO algorithm were proposed and tested. As first the Multiple-choice strategy for PSO is described. In this design a heterogeneous swarm is created and the roles are randomly assigned. As a second example of successful PSO modification the newly developed Gathering algorithm is presented. In the Gathering algorithm the phenomenon known in literature as "snowball effect" is used to highlight the promising regions by gathering of multiple particles and avoid the problems common for algorithm with static attraction points.

The performance of all above mentioned algorithms was tested using common benchmark functions and the results are compared either with canonical PSO algorithm or state-of-art representatives. The research results were continuously published and presented in international conferences with great reception.

Based on results obtained during this research is possible to claim that the performance of evolutionary computation techniques can be improved by various modern methods such as chaotic sequence implementation or inner principles modifications.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ABC	Artifical Bee Colony
ACO	Ant Colony Optimization
CF	Cost Function
CFE	Cost Function Evaluations
CPRNG	Chaotic Pseudo-Random Number Generator
DE	Differential Evolution
dim	Dimension
EA	Evolutionary Algorithm
ECT	Evolutionary Computational Technique
gBest	Global Best
lBest	Local Best
NP	Number of Particles
pBest	Personal Best
PRNG	Pseudo-Random Number Generator
PSO	Particle Swarm Optimization
SI	Swarm Intelligence
SOMA	Self-Organizing Migrating Algorithm

1 Introduction

The evolutionary computation techniques (ECTs) are a class of soft-computing algorithms inspired by the Darwinian principles of natural selection as a key mechanism of evolution and Mendel's law of heredity. These non-deterministic methods are used in last decades with great success in global optimization tasks of various kinds [1]-[6]. The main advantage of these methods is their ability to find solutions of good quality in exceptionally fast times in comparison with other optimization techniques either deterministic or brutal force based and also their capability of solving problems that are completely unsolvable by other methods (usually for time restrictions or complexity reasons). The ECTs are currently finding many applications in practice [7]-[11].

1.1 The goals of the dissertation

Following five main goals of the dissertation were defined:

1. Evaluation of the current state of the research area: Evolutionary algorithms (EAs), non-deterministic pseudo-random number generators (PRNGs).

2. Definition of the field of research - finding a suitable algorithm for implementation of alternative approaches and modifications.

3. The proposal of modifications and alternative strategies. Finding alternative PRNGs and their implementation into EAs.

4. Testing and benchmarking of proposed algorithms.

5. Evaluation of results, analysis and recommendations for future works.

1.2 Optimization

Within this research the optimization is understand as a process of finding such parameters for the cost function (CF) that generate the lowest value of CF. The cost function is mathematical description of the problem, it can have and usually has multiple parameters (inputs) but only one output (usually real number). The multi-objective optimization (multiple outputs) also exists and the ECTs are successfully used there too but in this research only single objective optimization problems are used. The dimension of the problem refers to the number of CF parameters. ECTs could be used for low-dimensional problems but also for solving very high-dimensional (e.g. 1000 parameters) problems.

1.3 Basic principles of evolutionary optimization

The vast majority of evolutionary optimization techniques follow similar pattern as described below:

- At the start of optimization process an initial population of candidate solutions is randomly generated. Each candidate solution represents a combination of CF parameters.
- This initial population is transformed into new population by different mathematical and evolutionary operations (such as crossover and mutation) that are unique for each algorithm. The new population should now contain solutions of better quality (lower CF value) and serves as an initial population for the next iteration of the algorithm.
- This process repeats until certain pre-defined ending conditions are met. Typically number of algorithm iterations or result precision.

As previously mentioned ECTs are stochastic algorithms thus use random operations (realized by pseudo-random number generation). Given this nature the algorithm may produce slightly different results each run. This is one of the disadvantages of these methods

1.4 Modern trends in ECTs design

These days the most popular ECTs are widely modified and new ECTs are developed in order to achieve better results and solve more complex tasks. According to the "No free lunch theorem" [12] there will never be a single algorithm that would outperform all others on all possible optimization tasks thus the variety of algorithms is convenient. Apart from modifying the inner principles of various ECTs, one of the newly investigated enhancement approaches is the implementation of chaotic sequences [13] into the ECTs [10, 11, 14]-[17]. In this approach pseudo-random number generator (PRNG) that is based on chaotic system is used instead of basic inbuilt computer (simulation software) pseudo-random number generator. The idea is that ECTs are inspired in nature and deterministic chaos exhibits in many natural processes and systems, thus the chaotic sequences may be more natural to ECTs and may improve their performance. Main part of this thesis is focused on this matter. Recently an innovative approach for chaos driven metaheuristics was presented by author of this thesis [18]. This approach uses more than one chaotic system during the run of particular ECT and switches between these systems either manually or by adaptive approach. A variety of learning technique [19] - [22] has been recently also presented to improve the performance of ECTs.

1.5 Areas of application

The main area for application of ECTs is currently the operations research, especially: Scheduling, Time management, Supply chains, Network optimization, Allocation problems, Routing. Other real-life applications can be found e.g.: GPS navigation (GEAS), Logistics, Aerodynamics, Moulds design, Evolutionary programming, Chaos Control and many more.

1.6 Thesis focus

Even though the research involved several different ECTs over the time; the main focus of this work is the Particle Swarm Optimization algorithm (PSO) that has been studied extensively as a prominent representative of Swarm Intelligence. Several modification methods created initially for PSO have been later successfully utilized in other ECTs.

1.7 Workflow

This section presents the overview of the research. The initial impulse for research of PSO inner dynamics was the Master thesis "PSO Algorithm in the Mathematica environment" defended in 2011. Since the later 2011 (start of Ph.D. study) the PSO was the main topic of the theoretical research. In early 2012 first successful experiments with chaotic sequences implemented in PSO were undertaken and results published. During the whole 2012 the chaos driven PSO was the main topic in both theoretical and applied studies.Successful applications for the model PID controller design were published. Following the study of inner dynamic of chaos driven PSO the multi-chaotic PSO was introduced in 2013 with great results and reception from international experts. The multi-chaotic approach was later used for other ECT, which is Differential Evolution. Alongside chaos PSO the multiple choice strategy for PSO was presented as an effective alternative to original PSO core. Furthermore several modifications of PSO were created during the 2014, among them the Gathering algorithm based on PSO is first highly competitive method developed in this research. All designs successfully presented in 2014 will be extended further in 2015. For clarity the workflow is visualized in Fig. 2.1.

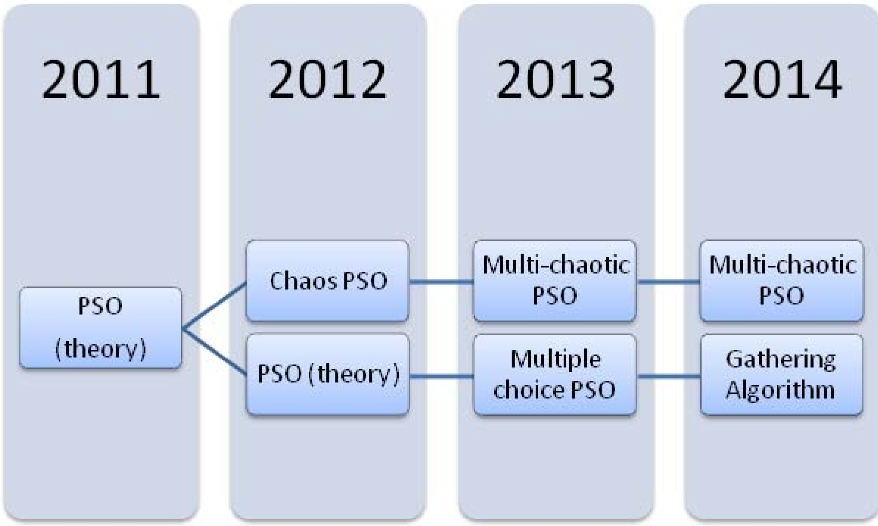

Fig. 1.1 Workflow schematic

2 Swarm Intelligence

In the theory of evolutionary computing the Swarm Intelligence [3] refers to a set of algorithms inspired by swarming behavior of insect and other animals. Also the swarm intelligence may refer to a phenomenon that very primitive individuals can solve complex tasks in organized swarms. The first and most prominent algorithms containing the swarm intelligence were the Ant Colony Optimization (ACO) [4] and Particle Swarm Optimization (PSO) [1, 2, 3]. These were followed by the SOMA algorithm [5]. Recently a vast number of new "swarm" algorithms was proposed [23]. Among many others the notable are the Artificial Bee Colony (ABC) [24], Firefly Algorithm [25], Bat Algorithm [27, 28] and Cuckoo Search [26].

2.1 Ant Colony Optimization (ACO)

The ACO mimics foraging behavior of ants in such way that the quality of a solution is given by a concentration of a pheromone [4]. The pheromone is used to mark a path to promising areas. The algorithm uses the mutation operator and specific selection rules which lead to great exploration capability but slower convergence. The ACO was one of the first SI algorithms and remains popular to these days.

2.2 Self-Organizing Migrating Algorithm (SOMA)

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 [5]. In every migration loop the best individual and called the Leader. All active individuals from the population move in the direction towards the Leader in the search space and evaluate the CF value at multiple points on the way. The SOMA and its discrete variant remain to this date among the best performing algorithms on very complex tasks.

2.3 Artificial Bee Colonoy (ABC)

The ABC algorithm took its inspiration from the behavior of bees) [24]. The population (swarm) is divided into three groups. Each group is than assigned different role. There are scouting bees, on looking bees and employed bees. Together all three groups form a heterogeneous swarm. The quality of the solution is represented by the amount of nectar (higher amount of nectar means better CF value). The onlookers and employed bees choose a food source (promising area) given their own experience (nectar amount) however the scouting bees choose their food source randomly bringing a strong heuristic-search aspect to the ABC. The discrete version of ABC is often successfully used for combinatorial problems and discrete optimization in general.

2.4 Firefly Algorithm

Firefly algorithm is based on the flashing patterns and behavior of tropical fireflies [23][25]. The innovation in this method is that the attractiveness between particles (fireflies) is introduced and used as a direction guide. Brighter fireflies (higher attractiveness) are more likely to attract other fireflies in their direction. The final movement is also partially randomized. The method manifested very impressive initial results and is now utilized by many researchers. It is claimed by the authors that the firefly algorithm can in special cases become similar to Differential evolution, Simulated annealing or Accelerated particle swarm optimization.

2.5 Cuckoo Search

This algorithm utilizes so called Lévy flights Cuckoo Search [26] and is designed as a combination of two random walks – local random walk and global random walk. Instead of typical isotropic random walks a random walk based on Lévy flights is used. Claimed to be capable of guaranteed global convergence the algorithm became very popular and achieved impressive results in many cases.

2.6 Bat Algorithm

The Bat algorithm introduced frequency tuning [23, 27, 28] (represented by loudness and pulse emission rate) as a mutation operator. Each individual in the population (bat) has the ability to change the loudness and the pulse emission rate. This helps to boost the exploitation capability of the algorithm. The Bat algorithm shares many notable similarities with other algorithms based on Swarm Intelligence e.g. PSO. The performance of the Bat algorithm nevertheless is very promising.

3 Particle Swarm Optimization Algorithm (PSO)

As has been mentioned above many Swarm intelligence bas algorithms share notable similarities therefore it is likely that performance-enhancing methods that successfully improve the performance of one representative of this category may be used to improve the performance of other members of the group. The PSO algorithm is probably the most prominent, popular and widely used representative of this category and shares notable similarities with almost all other SI methods. For these reason this algorithm has been chosen as the most suitable for this work.The PSO algorithm takes inspiration from the natural swarm behavior of birds and fish. It was firstly introduced by Eberhart and Kennedy in 1995 [1]. The membership of PSO into the ECTs is to this day widely discussed in the community [29]. Some say that the Swarm Intelligence (SI) [3] is an entirely different group of methods and others count these methods as sub-set of ECTs. For the purposes of this research the PSO is considered as one the ECTs and also SI methods.

In the PSO algorithm each particle in the population (swarm) represents a candidate solution for the optimization problem that is defined by its mathematical representation - the cost function (CF). In every iteration of the algorithm a new location (combination of CF parameters) for the particle is calculated based on its previous location and so called “velocity vector” (velocity vector contains particle velocity for each dimension of the problem).

According to the method of selection of the swarm or sub-swarm for best solution information spreading, the PSO algorithms are noted as global PSO (GPSO) or local PSO (LPSO) [30]. The advantages and disadvantages of both approaches are discussed in detail in [30]. In this research the GPSO is used (if not specified otherwise).

The new velocity of particle is given by (4.1); the velocity directly affects the position of each particle in the next iteration.

$$v_{i,j}^{t+1} = w \cdot v_{i,j}^{t} + c_1 \cdot Rand1 \cdot (pBest_{i,j}^{t} - x_{i,j}^{t}) + c_2 \cdot Rand2 \cdot (gBest_{j}^{t} - x_{i,j}^{t}) \quad (3.1)$$

Notation:

c_1, c_2 - acceleration constants typically set to 2.

Rand1, *Rand2* - random numbers from interval [0, 1]. In chaos PSO version this is the only place that chaotic pseudo-random number generators (CPRNGs) are used.

$v_{i,j}^{t+1}$ - New velocity of the *ith* particle in iteration t+1. (component j of the dimension D).

$v_{i,j}^{t}$ - Current velocity of the *ith* particle in iteration t. (component j of the dimension D).

$x_{i,j}^{t}$ - Current position of the *ith* particle in iteration t. (component j of the dimension D).

$pBest_{i,j}^{t}$ - Local (personal) best solution found by the *ith* particle in iteration t. (component j of the dimension D).

$gBest_{j}^{t}$ - Best solution found in a population in iteration t. (component j of the dimension$D$).

w – inertia weight value.

The maximum velocity is typically limited to 0.2 times the range. The new position of each particle is then given by (4.2).

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{3.2}$$

Where:

$\mathbf{X}_i^{t+1} = (x_{i,1}^{t+1},\, x_{i,2}^{t+1},\, \ldots,\, x_{i,D}^{t+1})$ - position of the i^{th} particle in iteration *t+1*.

$\mathbf{X}_i^{t} = (x_{i,1}^{t},\, x_{i,2}^{t},\, \ldots,\, x_{i,D}^{t})$ - position of the i^{th} particle in iteration *t*.

$\mathbf{V}_i^{t+1} = (v_{i,1}^{t+1},\, v_{i,2}^{t+1},\, \ldots,\, v_{i,D}^{t+1})$ - velocity of the i^{th} particle in iteration *t+1*.

Based on the method of velocity and position update it is recognized the synchronous (velocity and position updated after iteration cycle) and asynchronous (imminent velocity and position update) PSO [31]. In this study the imminently updating (asynchronous) PSO was utilized. As is discussed in [32] it may be beneficial to initialize the population with zero velocity. Therefore this advice was followed in this research.

Finally the linear decreasing inertia weight [2, 33, 34] is used in the typically referred GPSO design that was used in this research. The dynamic inertia weight is meant to slow the particles over time thus to improve the local search capability in the later phase of the optimization. The inertia weight has two control parameters w_{start} and w_{end}. A new w for each iteration is given by (4.3), where t stands for current iteration number and n stands for the total number of iterations. The values used in this *study* were w$_{start}$ = 0.9 and w$_{end}$ = 0.4.

$$w = w_{start} - \frac{((w_{start} - w_{end}) \cdot t)}{n} \tag{3.3}$$

The inertia weight is often replaced with other approaches, such as using so called "constriction factor”. Again the benefits and mutual relations of these two approaches are still discussed in the community [34]. The default setting of control parameters of the algorithm was based on most typically used PSO designs in literature. Different settings are also possible and sometimes preferable. Some of these settings are proved to lead to convergent trajectories [35].

The above described basic PSO design was used as the default version for further enhancement during all experiments (if not specified otherwise).

4 Test functions

During the research an extensive set of benchmark functions was used. Apart from the IEEE CEC 2013 Special Session and Competition on Real-Parameter Optimization benchmark set [36] that contains 28 functions an additional set of typically used benchmark functions was used. The additional set of typical benchmark functions [20] is detailed in the following Table 5.1:

Tab. 4.1 Selected typically used benchmark functions

Name	Function
Sphere	$f_{s1}(x) = \sum_{i=1}^{D} x_i^2$
Schwefel'sP2.22	$f_{s2}(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$
Rosenbrock	$f_{s3}(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2]$
Noise	$f_{s4}(x) = \sum_{i=1}^{D} x_i^4 + random[0,1)$
Schwefel's[1]	$f_{s5}(x) = 418.9829 \cdot \mathrm{D} - \sum_{i=1}^{D} x_i \sin(\sqrt{\lvert x \rvert})$
Rastrigin	$f_{s6}(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$
Ackley	$f_{s7}(x) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D} x_i^2 \cos 2\pi x_i) + 20 + e$

[1] Two variants of the Schwefel's function were used. The value of the optimum was shifted to -418.9829 ·D for the second variant. The position of the optimum and other characteristics of the function are the same.

5 Chaotic maps

This section contains the description and mathematical definitions of six discrete chaotic maps [13] used as the CPRNGs in this research.

5.1 Lozi Map

The Lozi map is a simple discrete two-dimensional chaotic map. The map equations are given in (6.1). The parameters used in this work are: a = 1.7 and b = 0.5 [13]. For these values, the system exhibits typical chaotic behavior and with this parameter setting it is used in the most research papers and other literature sources. The x,y plot of the map is given in Figure 6.1. The sample sequence produced by this map is given in Figure 6.2. Finally the distribution histogram of CPRNG constructed from this chaotic system is presented in Figure 6.3.

$$\begin{aligned} X_{n+1} &= 1 - a\left|X_n\right| + bY_n \\ Y_{n+1} &= X_n \end{aligned} \tag{5.1}$$

5.2 Dissipative Standard Map

The Dissipative Standard map is a two-dimensional chaotic map. The parameters used in this work are b = 0.6 and k = 8 [13]. The map equations are given in (6.2). The x,y plot of the map is given in Figure 6.4. The sample sequence produced by this map is given in Figure 6.5. Finally the distribution histogram of CPRNG constructed from this chaotic system is presented in Figure 6.6.

$$\begin{aligned} X_{n+1} &= X_n + Y_{n+1}(mod 2\pi) \\ Y_{n+1} &= bY_n + k\sin X_n(mod 2\pi) \end{aligned} \tag{5.2}$$

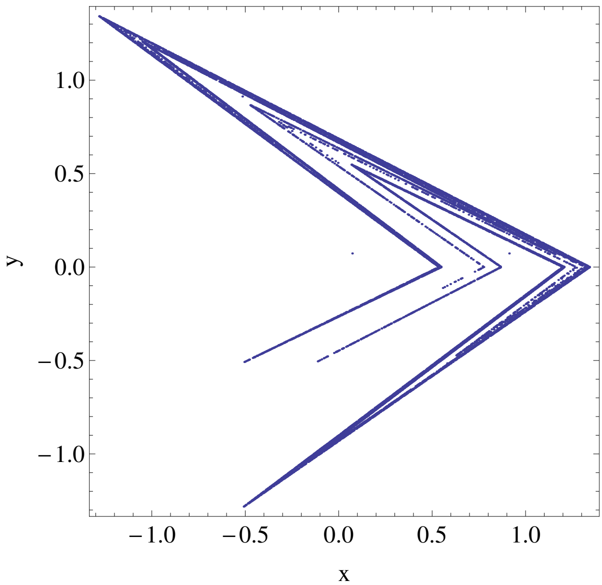

Fig. 5.1 Lozi map x,y plot

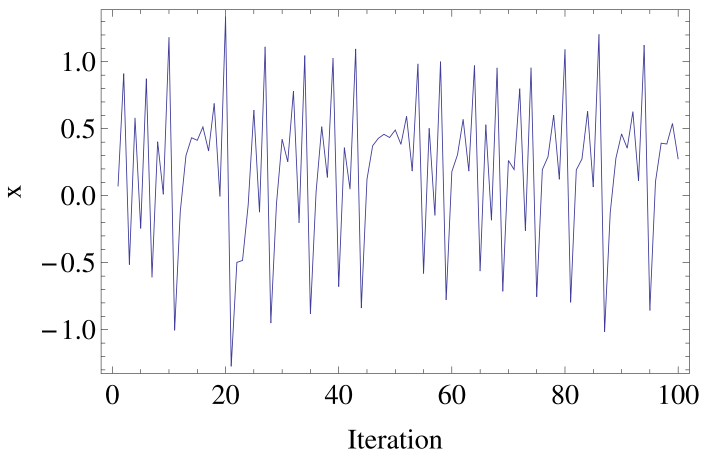

Fig. 5.2 Lozi map sequence sample

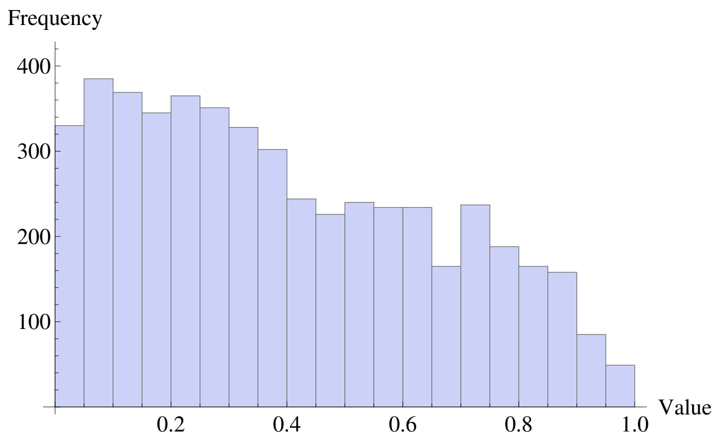

Fig. 5.3 Distribution histogram - CPRNG based on Lozi map

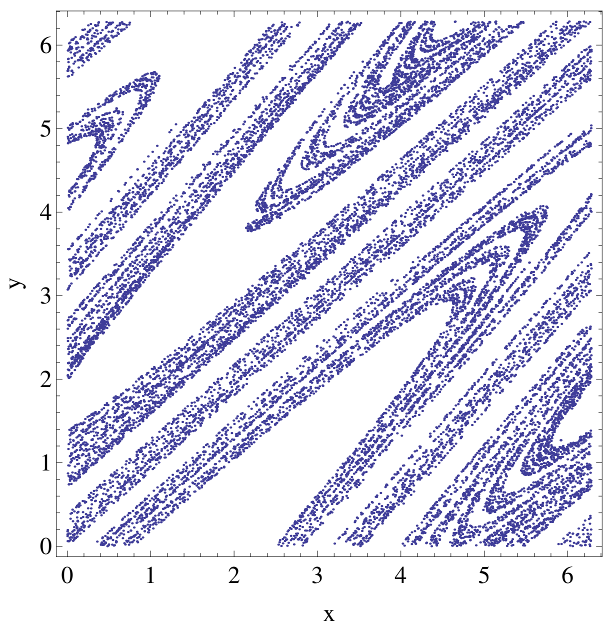

Fig. 5.4 Disipative map x,y plot

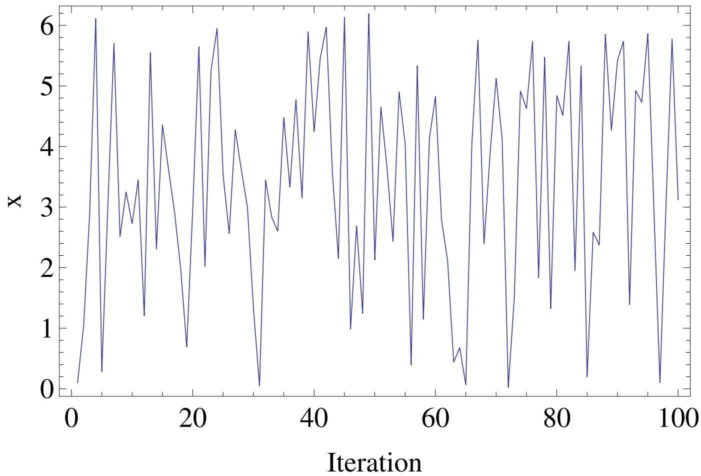

Fig. 5.5 Disipative map sequence sample

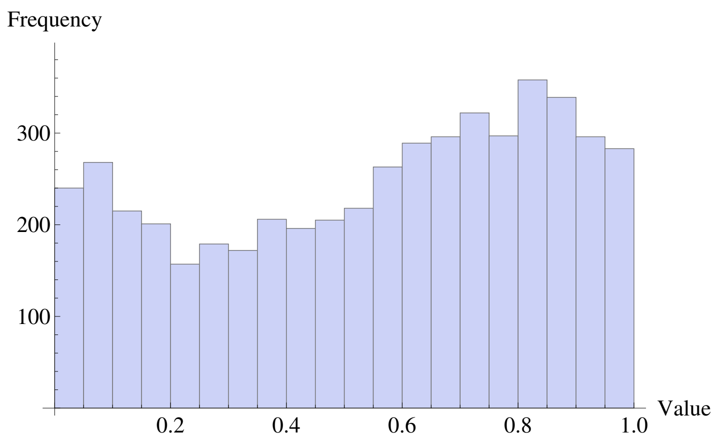

Fig. 5.6 Distribution histogram - CPRNG based on Disipative map

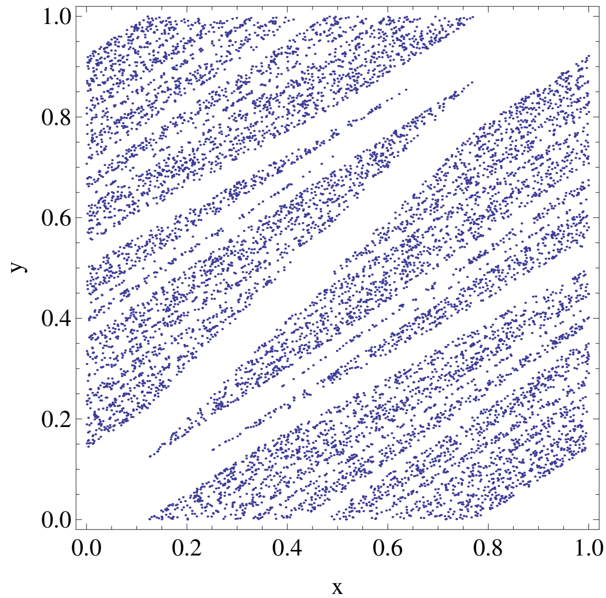

Fig. 5.7 Arnold's Cat map x,y plot

5.3 Arnold's Cat Map

The Arnold's Cat map is a simple two dimensional discrete system that stretches and folds points (x, y) to (x+y , x+2y) mod 1 in phase space. The map equations are given in (6.3). This map was used with parameter k = 0.1 [13]. The x,y plot of the map is given in Figure 6.7. The sample sequence produced by this map is given in Figure 6.8. Finally the distribution histogram of CPRNG constructed from this chaotic system is presented in Figure 6.9.

$$\begin{aligned} X_{n+1} &= X_n + Y_n(mod1) \\ Y_{n+1} &= X_n + kY_n(mod1) \end{aligned} \tag{5.3}$$

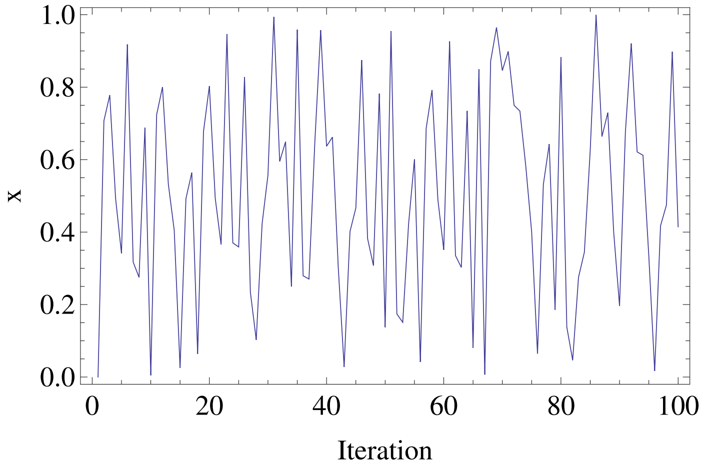

Fig. 5.8 Arnold's Cat map sequence sample

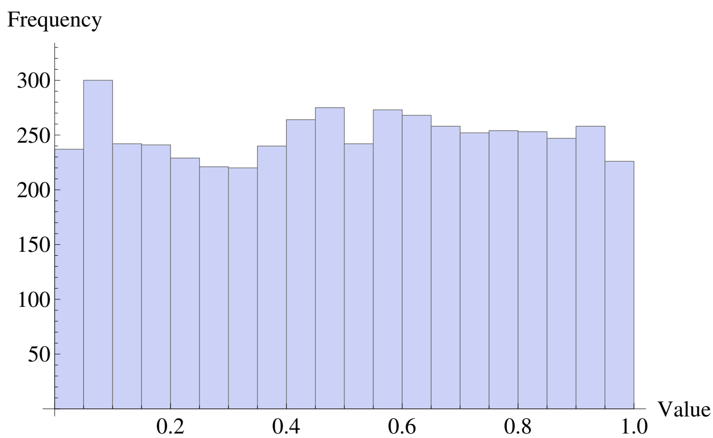

Fig. 5.9 Distribution histogram - CPRNG based on Arnold's Cat map

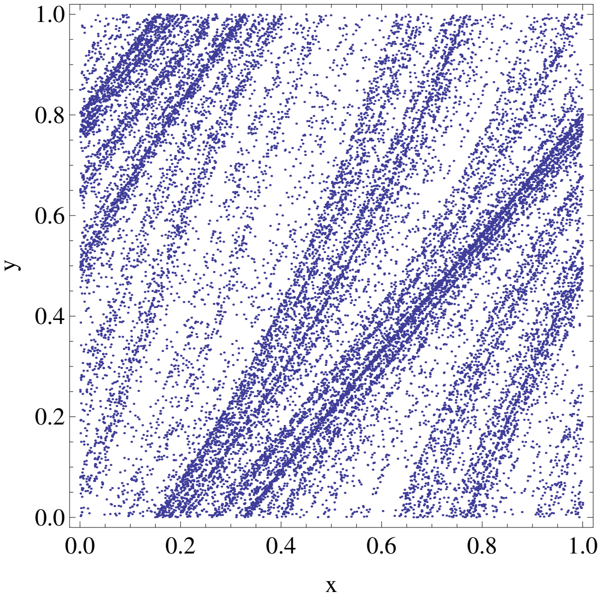

Fig. 5.10 Sinai map x,y plot

5.4 Sinai Map

The Sinai map is a simple two dimensional discrete system similar to the Arnold's Cat map. The map equations are given in (6.4). The parameter used in this work is $\delta = 0.1$ [13]. The x,y plot of the map is given in Figure 6.10. The sample sequence produced by this map is given in Figure 6.11. Finally the distribution histogram of CPRNG constructed from this chaotic system is presented in Figure 6.12.

$$\begin{aligned} X_{n+1} &= X_n + Y_n + \delta \cos 2\pi Y_n (mod 1) \\ Y_{n+1} &= X_n + 2Y_n (mod 1) \end{aligned} \tag{5.4}$$

5.5 Burgers Map

The Burgers map is a discretization of a pair of coupled differential equations The map equations are given in (6.5) with control parameters a = 0.75 and b =

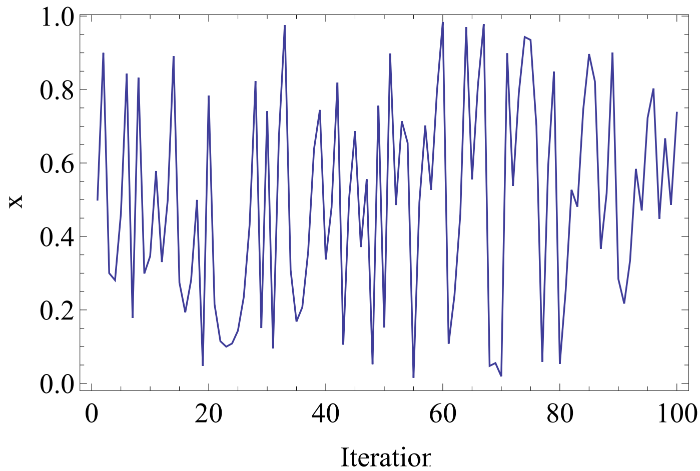

Fig. 5.11 Sinai map sequence sample

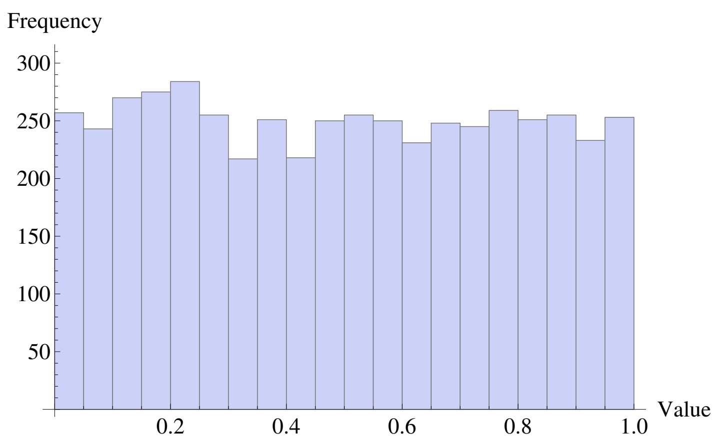

Fig. 5.12 Distribution histogram - CPRNG based on Sinai map

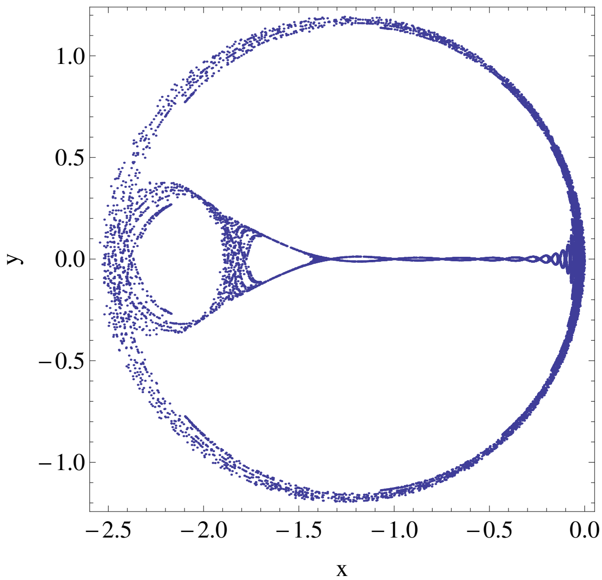

Fig. 5.13 Burgers map x,y plot

1.75 [13]. The x,y plot of the map is given in Figure 6.13. The sample sequence produced by this map is given in Figure 6.14. Finally the distribution histogram of CPRNG constructed from this chaotic system is presented in Figure 6.15.

$$
\begin{aligned}
X_{n+1} &= aX_n - Y_n^2 \\
Y_{n+1} &= bY_n + X_nY_n
\end{aligned}
\tag{5.5}
$$

5.6 Tinkerbell Map

The Tinkerbell map is a two-dimensional complex discrete-time dynamical system given by (6.6) with following control parameters: a = 0.9, b = -0.6, c = 2 and d = 0.5 [13]. The x,y plot of the map is given in Figure 6.17. The sample sequence produced by this map is given in Figure 6.18. Finally the distribution histogram of CPRNG constructed from this chaotic system is presented in Figure 6.18.

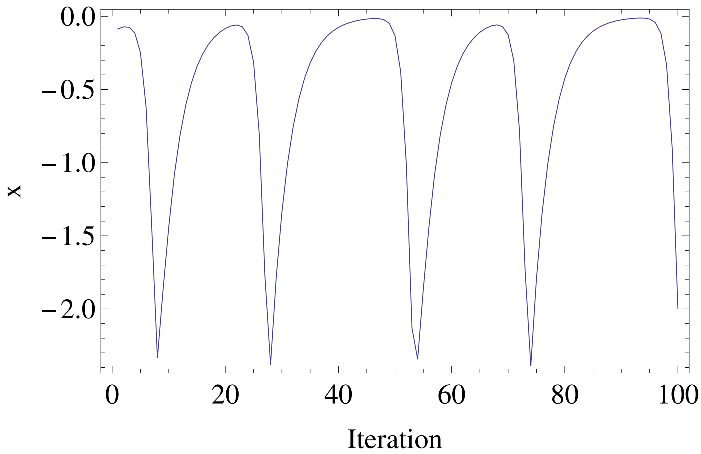

Fig. 5.14 Burgers map sequence sample

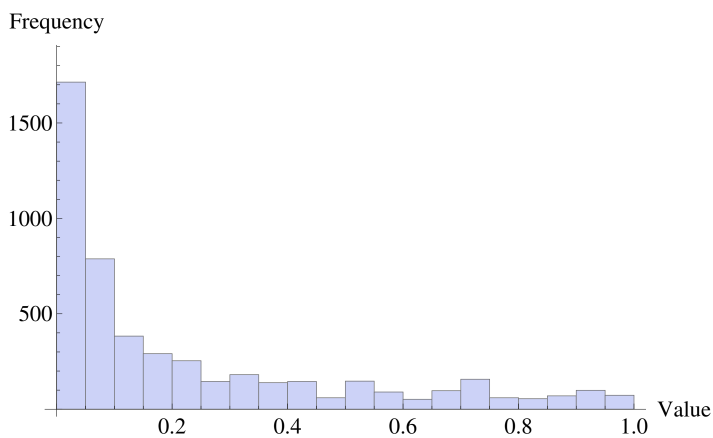

Fig. 5.15 Distribution histogram - CPRNG based on Burgers map

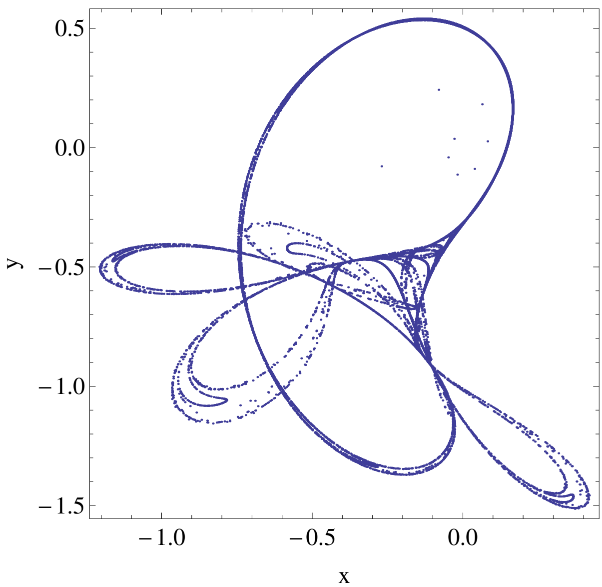

Fig. 5.16 Tinkerbell map x,y plot

$$\begin{aligned} X_{n+1} &= X_n^2 - Y_n^2 + aX_n + bY_n \\ Y_{n+1} &= 2X_nY_n + cX_n + dY_n \end{aligned} \quad (5.6)$$

6 Chaos PSO

The first and main part of the research dealt with implementation of chaotic sequences into ECTs. As the most suitable algorithm it was chosen the PSO. The initial experiments with chaotic PSO (CPSO) were presented in [15].

6.1 Chaotic pseudo-random number generator implementation

Based on the literature [15] and initial experiments the method of implementation of chaotic sequences was chosen thus that the chaotic pseudo-random numbers generator (CPRNG) was used only for generating the random numbers

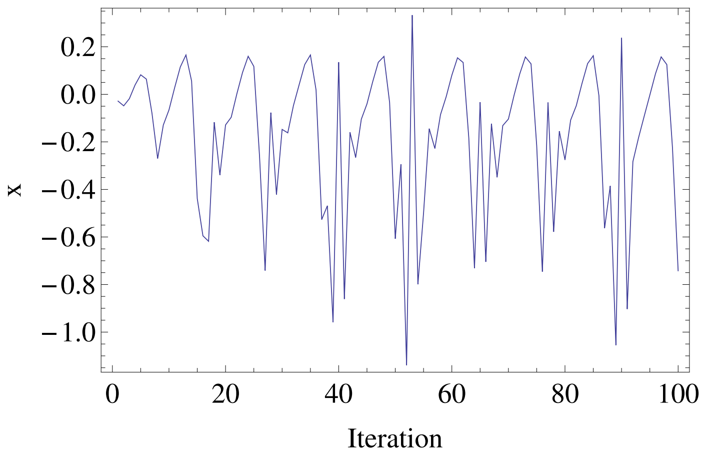

Fig. 5.17 Tinkerbell map sequence sample

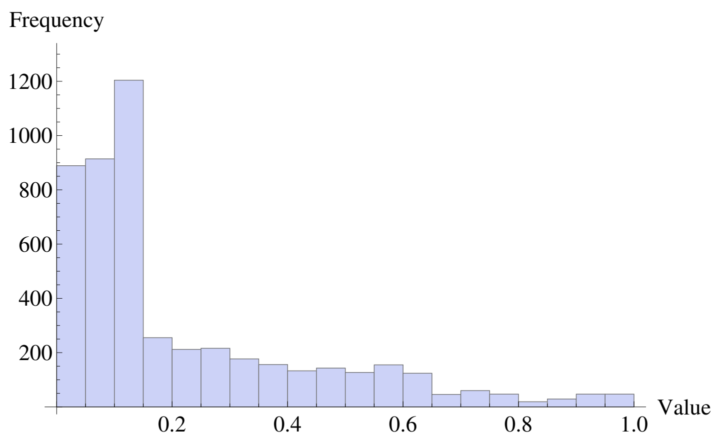

Fig. 5.18 Distribution histogram - CPRNG based on Tinkerbell map

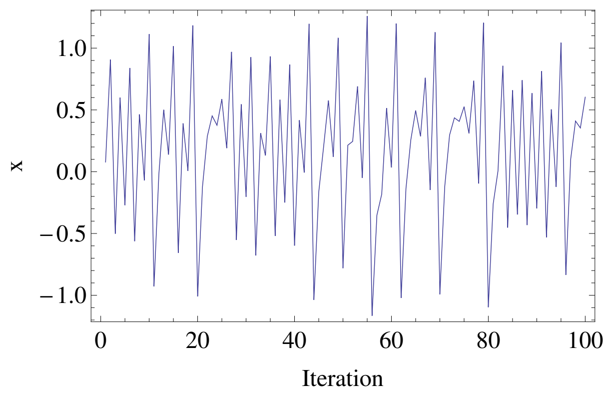

Fig. 6.1 Sample output sequence of the Lozi map (x value)

for the main PSO formula (4.1). For the needs of PSO main formula (4.1) it is necessary to generate pseudo-random numbers in the interval [0,1] however the majority of chaotic maps does not follow this restriction for the area it covers. It is therefore necessary to transform the output sequence into the required interval. In the case of two-dimensional discrete chaotic map as output sequence it can be used either the x or y value sequence given by the pair of equations that define the chaotic map (as described in previous section). Figure 7.1 gives x value sequence for the first 100 iterations of the Lozi map. Corresponding y value sequence is depicted in Figure 7.2. It can be observed that these sequences share certain similarities. However in the case of Burger's map (Figure 7.3 and 7.4) the corresponding x and y sequences are significantly more different. Further from this point the PSO versions that use the x value sequence for the CPRNG are noted with "X" and vice versa.

The performance differences of the "X" and "Y" versions is investigated later in this study. Furthermore two different approaches for transforming the sequences into the interval [0,1] were investigated in this research. In the first case the absolute value is used to transform negative numbers to positive. All numbers in the sequence are then divided by the maximum value found in the sequence (7.1). As an example the distribution of CPRNG based on the Lozi map x value sequence created this way is given in Figure 7.5.

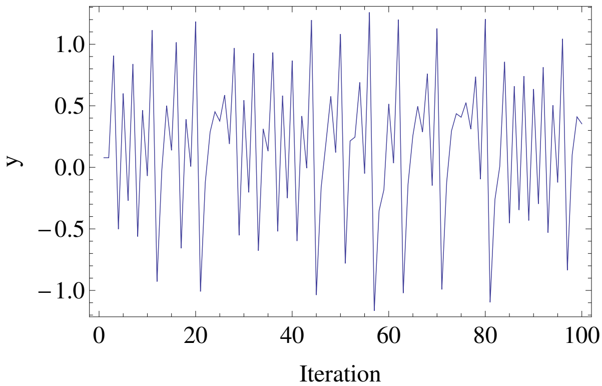

Fig. 6.2 Sample output sequence of the Lozi map (y value)

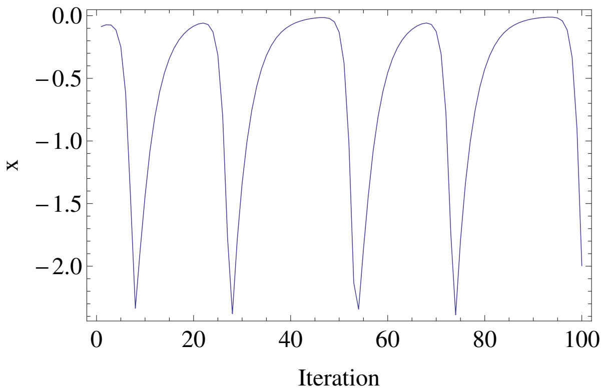

Fig. 6.3 Sample output sequence of the Burger's map (x value)

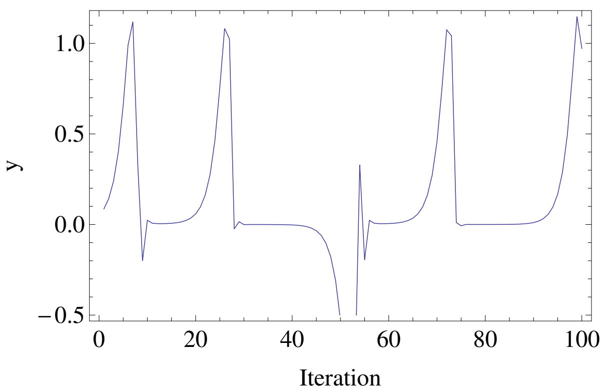

Fig. 6.4 Sample output sequence of the Burger's map (y value)

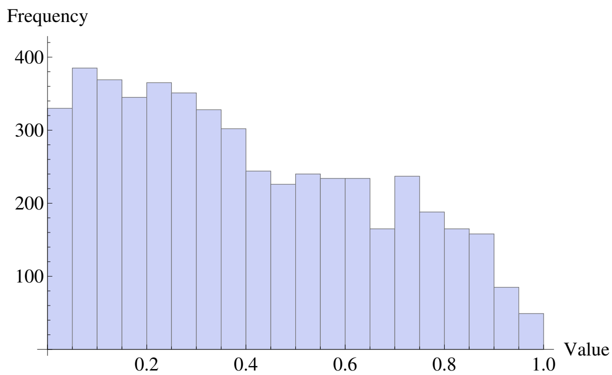

Fig. 6.5 CPRNG based on Lozi map (x sequence, absolute value applied) – distribution histogram

$$X = |X*| \, / Max(X*) \tag{6.1}$$

Where:

X – Transformed sequence

$X*$ – Original sequence

Max(X*) – Maximum number in original sequence

Alternatively it is possible to shift the whole sequence to positive numbers and interval [0,1] according to (7.2). The distribution of CPRNG based on the Lozi map x value sequence created this alternative way is given in Fig. 7.6. Versions of PSO using this approach are further noted with "s" (as shifted).

$$X = (|Min(X*)| + X*)/(|Min(X*)| + Max(X*)) \tag{6.2}$$

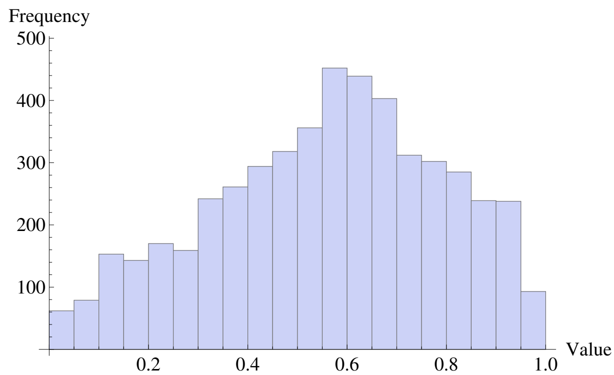

Fig. 6.6 CPRNG based on Lozi map (x sequence, shifted) – distribution histogram

Where:

Min(X)* – Minimum number in original sequence

6.2 Experiment 1

In order to evaluate the impact of different CPRNG implementations on the performance of PSO algorithm a basic experiment was designed.

6.2.1 Experiment 1 setup

The control parameters of PSO algorithm and experiment setup is given in Table 7.1.

6.2.2 Notation

Several versions of original GPSO (as described in section 3) algorithm were used. The notation follows the pattern described in previous section. The version is

Tab. 6.1 Algorithm setup

Population size:	40
Iterations / Generations:	5000
c1, c2:	2
w:	Linear 0.9 -> 0.4
v_{max}:	0.2
Dim:	30
Repeated runs:	30

noted according to the chaotic map that was used (”Lozi” or ”Burger”) + the sequence that was used for CPRNG (”X” or ”Y”) + “s” if the shifting approach described in previous section was applied to move the sequence into required interval. Examples:

- GPSO Lozi Y – CPRNG based on the y value sequence of Lozi map. Transformed into the interval [0,1] using the absolute value approach.
- GPSO Burger Xs – CPRNG based on the x value sequence of Burgers map. Transformed into the interval [0,1] using the shift approach.
- Etc.

As already mentioned in previous sections the CPRNG based on Lozi map or Burger’s map was applied only for the main formula of PSO (4.1). For other purposes (generating of initial population etc.) default C language built-in pseudo-random number generator was used within all described versions of PSO.

6.2.3 Experiment 1 results

In following tables (Table 7.2 - 7.5) the numerical results of the initial experiment are summarized. The best mean result and best overall result (min. CF value) are highlighted. Furthermore the mean history of gBest value (best solution) is given in Figures 7.7 - 7.10.

Tab. 6.2 Final results comparison – Sphere function – Lozi CPRNGs

	GPSO Lozi Y	GPSO Lozi X	GPSO Lozi Ys	GPSO Lozi Xs
Mean CF Value:	**4.03E-108**	7.78E-106	1.39E-78	2.98E-78
Std. Dev.:	2.18E-107	4.26E-105	6.78E-78	1.39E-77
CF Value Median:	4.73E-112	8.74E-113	1.49E-81	1.17E-81
Max. CF Value:	1.20E-106	2.33E-104	3.72E-77	7.57E-77
Min. CF Value:	2.35E-116	**3.43E-118**	1.95E-85	7.04E-86

Tab. 6.3 Final results comparison – Schwefel's function – Lozi CPRNGs

	GPSO Lozi Y	GPSO Lozi X	GPSO Lozi Ys	GPSO Lozi Xs
Mean CF Value:	**4.08E+03**	4.09E+03	5.07E+03	5.05E+03
Std. Dev.:	4.46E+02	3.26E+02	3.93E+02	4.77E+02
CF Value Median:	4.14E+03	4.05E+03	5.07E+03	5.04E+03
Max. CF Value:	4.78E+03	5.07E+03	6.02E+03	5.98E+03
Min. CF Value:	**3.08E+03**	3.61E+03	4.44E+03	4.32E+03

Tab. 6.4 Final results comparison- Sphere function – Burger CPRNGs

	GPSO Burger Y	GPSO Burger X	GPSO Burger Ys	GPSO Burger Xs
Mean CF Value:	1.45E+02	3.87E-12	**2.32E-88**	2.93E+03
Std. Dev.:	6.47E+01	1.04E-11	1.27E-87	2.61E+02
CF Value Median:	1.30E+02	1.61E-12	2.68E-100	3.03E+03
Max. CF Value:	2.84E+02	5.80E-11	6.97E-87	3.25E+03
Min. CF Value:	2.96E+01	5.25E-14	**3.11E-111**	2.29E+03

Tab. 6.5 Results comparison – Schwefel's function Burger CPRNGs

	GPSO Burger Y	GPSO Burger X	GPSO Burger Ys	GPSO Burger Xs
Mean CF Value:	3.98E+03	**2.07E+03**	4.61E+03	8.66E+03
Std. Dev.:	6.68E+02	3.30E+02	5.28E+02	2.40E+02
CF Value Median:	3.98E+03	2.09E+03	4.71E+03	8.65E+03
Max. CF Value:	5.69E+03	2.88E+03	5.65E+03	8.98E+03
Min. CF Value:	3.07E+03	**1.54E+03**	3.57E+03	8.01E+03

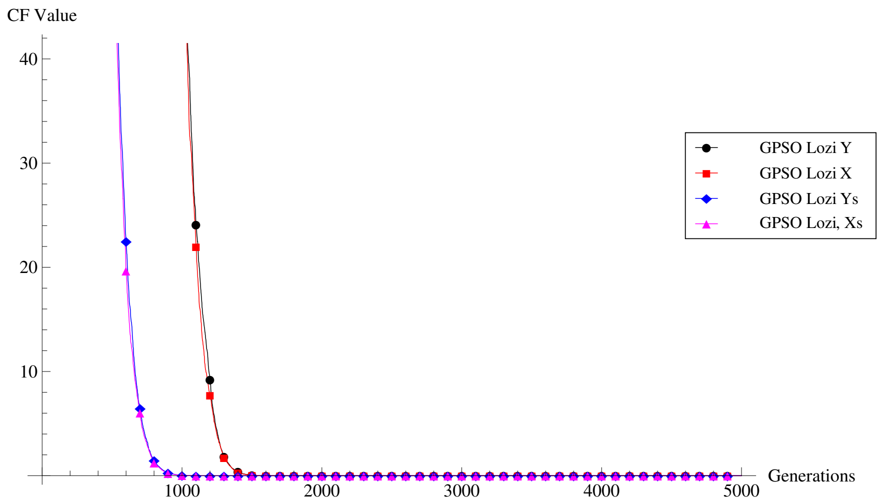

Fig. 6.7 Mean gBest history - sphere function - Lozi CPRNGs

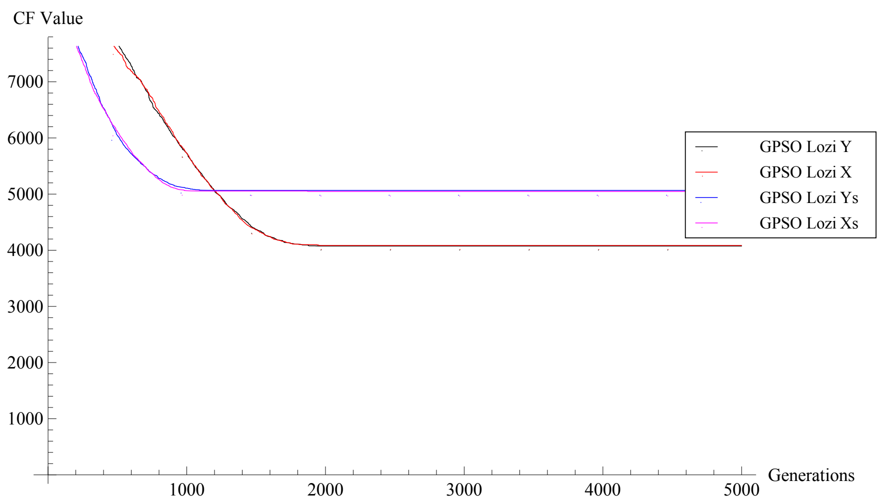

Fig. 6.8 Mean gBest history - Schwefel's function - Lozi CPRNGs

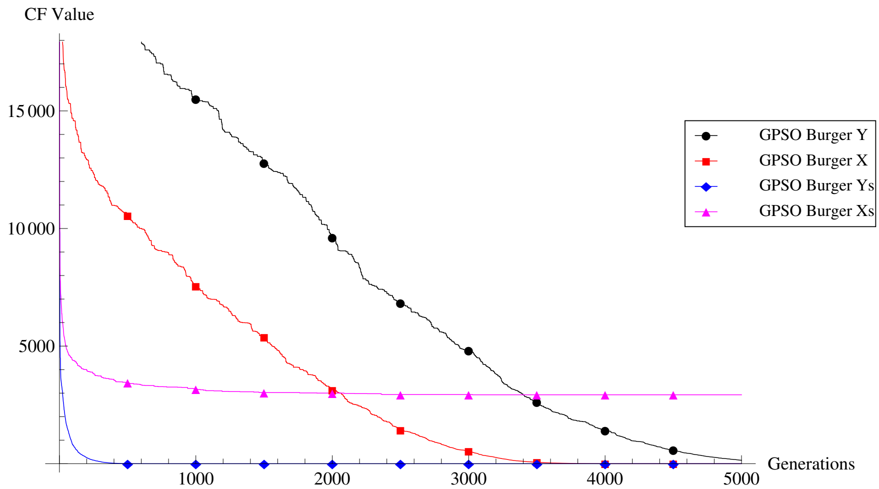

Fig. 6.9 Mean gBest history - sphere function - Burger CPRNGs

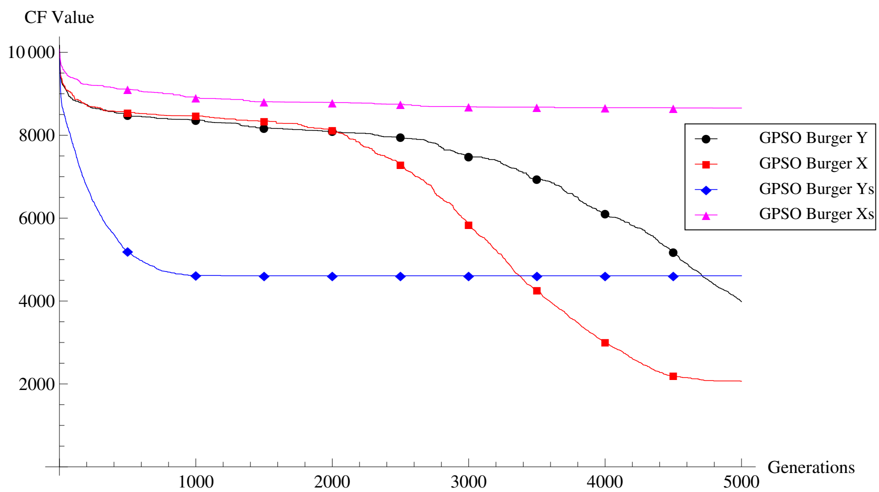

Fig. 6.10 Mean gBest history - Schwefel's function - Burger CPRNGs

Based on the results obtained in this experiment the implementation method for further chaos PSO research was chosen in such way that the 'X" equation will be used for all maps and that the transformation using absolute value will be performed.

6.3 Chaos PSO - Initial experiment

In the experiment, six chaotic maps described previously were utilized as the CPRNGs for the GPSO algorithm. The CPRNGs were used only in the process of the new velocity calculation. In order to observe the influence of different CPRNGs to the behaviour of the GPSO algorithm dealing with diverse optimization tasks, one unimodal (f_{s1}) and one multimodal (f_{s5}) benchmark function were used. Subsequently the obtained results were used as a guideline for the selection of promising CPRNGs for the next experiment with multi-chaotic CPRNGs.

The time evolutions of the mean history of *gBest* value for both used benchmark functions are presented in Figure 7.11 and Figure 7.12. Performance comparisons of all six versions of Chaos enhanced GPSO and canonical GPSO are given in Table 7.6 and Table 7.7.

Tab. 6.6 Final results comparison- Sphere function

	GPSO	GPSO Lozi	GPSO Disi	GPSO Arnold	GPSO Sinai	GPSO Burger	GPSO Tinker
Mean CF Value:	4.6721E-34	**4.3675E-113**	6.7601E-03	2.4591E-45	3.1906E-35	3.8018E-01	1.0085E-91
Std. Dev.:	1.9509E-33	1.3264E-112	5.7096E-03	7.4281E-45	1.2962E-34	1.9674E-01	5.0422E-91
CF Value Median:	4.8885E-35	1.6014E-115	6.5301E-03	3.2741E-46	2.8485E-36	3.0138E-01	1.2449E-106
Max. CF Value:	9.8184E-33	5.4880E-112	2.7976E-02	3.6716E-44	6.5272E-34	8.1530E-01	2.5211E-90
Min. CF Value:	1.0276E-37	**2.9063E-119**	5.9880E-04	1.3461E-49	2.1051E-38	8.2704E-02	3.6187E-113

As is presented above in this initial experiment it was observed very differ-

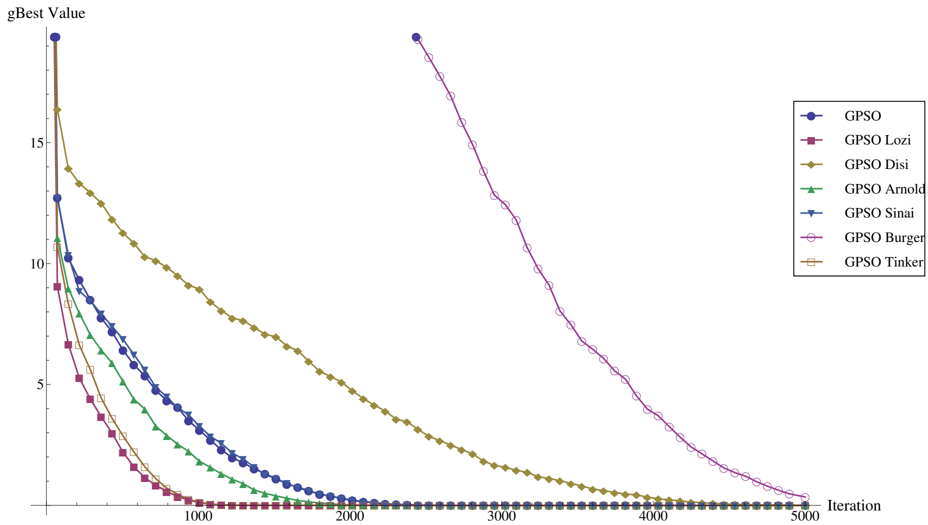

Fig. 6.11 Mean gBest value history - Sphere function – CPRNGs comparison

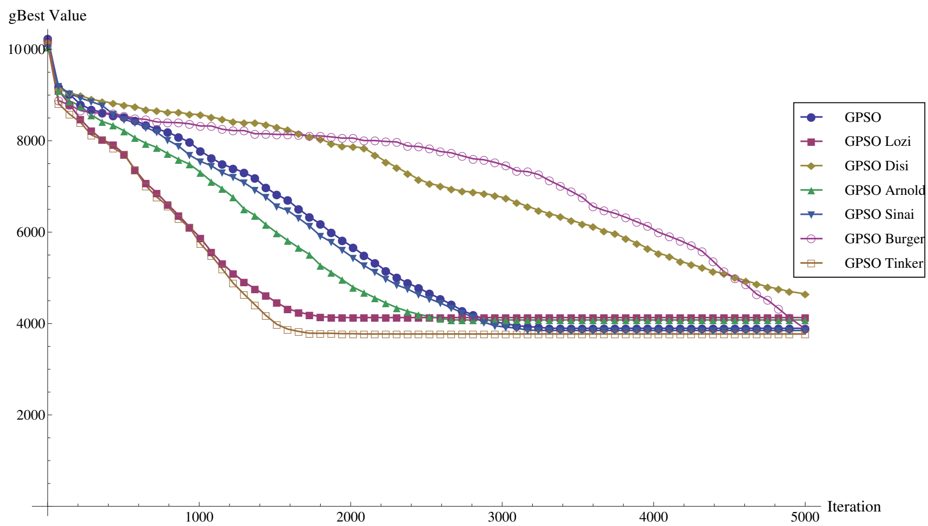

Fig. 6.12 Mean gBest value history - Schwefel's function - CPRNGs comparison

ent behavior of GPSO algorithm enhanced with different CPRNGs in terms of convergence speed, premature convergence avoidance capability and final result value. Following these findings the performance of proposed algorithms was in-

Tab. 6.7 Final results comparison- Schwefel's function

	GPSO	GPSO Lozi	GPSO Disi	GPSO Arnold	GPSO Sinai	GPSO Burger	GPSO Tinker
Mean CF Value:	3904.56	3993.01	4581.14	4220.41	3909.3	4264.07	**3741.93**
Std. Dev.:	419.763	405.379	401.721	423.804	470.136	681.092	486.467
CF Value Median:	3888.77	3947.97	4640.9	4145.4	3947.97	4172.27	3770.37
Max. CF Value:	4777.08	5152.17	5362.58	5349.56	5112.66	5509.38	4540.27
Min. CF Value:	3138.64	3079.41	3843.19	3651.89	3079.42	2854.48	**2724.17**

vestigated further using the CEC 2013 Benchmark set. The complete results overview is given in Table 7.8 and Table 7.9. The algorithms were set according to benchmark rules for dim = 10.

Based on presented results several conclusions can be made. Firstly, it is clear that different chaotic PRNGs lead to different convergence behavior of PSO. Secondly the effect on the actual performance (final result) is very problem-dependant. In some cases the performance of PSO algorithm enhanced with different CPRNGs is similar or comparable. In other cases however it is possible to significantly enhance the performance of PSO algorithm by using particular CPRNG (See Table 7.8 and Table 7.9).

After these satisfactory results with CPRNGs with default setting of control parameters a tuning experiment was conducted.

6.4 Tuning experiment

In the tuning experiment the impact of different control parameters of the chaotic system on the performance of chaos enhanced PSO was examined. The PSO enhanced with CPRNG based on Lozi map was selected for this experiment and the goal was to improve the performance on multi-modal problems. For this

Tab. 6.8 Mean final results comparison- CEC 13 Benchmark set (Part 1)

f(x)	*fmin*	GPSO	Lozi	Dissipative	Arnold's
f_1	-1400	**-1.40E+03**	**-1.40E+03**	**-1.40E+03**	**-1.40E+03**
f_2	-1300	**1.17E+05**	1.75E+05	3.10E+05	4.46E+05
f_3	-1200	2.58E+06	7.13E+06	2.50E+06	3.15E+06
f_4	-1100	-7.38E+02	**-8.15E+02**	-6.74E+02	-1.47E+02
f_5	-1000	**-1.00E+03**	**-1.00E+03**	**-1.00E+03**	**-1.00E+03**
f_6	-900	-8.89E+02	-8.90E+02	-8.93E+02	-8.93E+02
f_7	-800	**-7.97E+02**	-7.95E+02	-7.95E+02	-7.96E+02
f_8	-700	**-6.80E+02**	**-6.80E+02**	**-6.80E+02**	**-6.80E+02**
f_9	-600	**-5.97E+02**	-5.96E+02	-5.96E+02	**-5.97E+02**
f_{10}	-500	**-5.00E+02**	-4.99E+02	**-5.00E+02**	-4.99E+02
f_{11}	-400	**-3.98E+02**	-3.97E+02	**-3.98E+02**	-3.95E+02
f_{12}	-300	**-2.89E+02**	-2.88E+02	-2.87E+02	-2.79E+02
f_{13}	-200	-1.81E+02	-1.82E+02	**-1.84E+02**	-1.79E+02
f_{14}	-100	1.33E+02	1.20E+02	1.25E+02	1.59E+02
f_{15}	100	7.62E+02	7.28E+02	7.18E+02	9.79E+02
f_{16}	200	**2.01E+02**	**2.01E+02**	**2.01E+02**	**2.01E+02**
f_{17}	300	3.14E+02	**3.13E+02**	3.14E+02	3.32E+02
f_{18}	400	4.34E+02	**4.22E+02**	4.24E+02	4.43E+02
f_{19}	500	**5.01E+02**	**5.01E+02**	**5.01E+02**	5.02E+02
f_{20}	600	**6.02E+02**	6.03E+02	**6.02E+02**	6.03E+02
f_{21}	700	**1.09E+03**	1.09E+03	1.10E+03	1.10E+03
f_{22}	800	1.06E+03	1.07E+03	1.08E+03	1.04E+03
f_{23}	900	**1.60E+03**	1.66E+03	1.72E+03	1.83E+03
f_{24}	1000	1.21E+03	1.21E+03	**1.20E+03**	1.21E+03
f_{25}	1100	1.31E+03	1.31E+03	1.31E+03	1.31E+03
f_{26}	1200	**1.34E+03**	1.35E+03	1.35E+03	1.35E+03
f_{27}	1300	1.64E+03	1.64E+03	**1.63E+03**	**1.63E+03**
f_{28}	1400	1.78E+03	**1.74E+03**	1.73E+03	1.73E+03

reason the Rotated Schwefel's Function (*f15*) from CEC'13 Benchamrk set [36] was used with dimension setting 10. Number of particles was set to 40 and number of iterations was 2500. The Lozi map parameters were changed with step 0.05 in following way: parameter a: from 1.3 to 1.7; parameter b: from 0.1 to 0.6. For each setting 100 repeated runs of the algorithm were performed.

For some control parameter settings in this experiment the Lozi map no longer exhibits chaotic behavior or is reduced to two-value sequence however with respect to [37] all possible combinations in given range were tested.

Tab. 6.9 Mean final results comparison- CEC 13 Benchmark set (Part 2)

f(x)	*fmin*	Sinai	Burgers	Tinkerbell
f_1	-1400	**-1.40E+03**	**-1.40E+03**	**-1.40E+03**
f_2	-1300	1.85E+05	1.47E+05	7.11E+05
f_3	-1200	**1.70E+06**	5.92E+06	3.15E+07
f_4	-1100	-5.24E+02	-5.40E+02	7.19E+02
f_5	-1000	**-1.00E+03**	**-1.00E+03**	**-1.00E+03**
f_6	-900	**-8.95E+02**	-8.91E+02	-8.91E+02
f_7	-800	**-7.97E+02**	**-7.97E+02**	-7.93E+02
f_8	-700	**-6.80E+02**	**-6.80E+02**	**-6.80E+02**
f_9	-600	**-5.97E+02**	**-5.97E+02**	-5.96E+02
f_{10}	-500	**-5.00E+02**	**-5.00E+02**	-4.98E+02
f_{11}	-400	**-3.98E+02**	**-3.98E+02**	-3.87E+02
f_{12}	-300	**-2.89E+02**	-2.88E+02	-2.68E+02
f_{13}	-200	-1.81E+02	-1.81E+02	-1.68E+02
f_{14}	-100	**1.06E+02**	**1.06E+02**	5.30E+02
f_{15}	100	**6.90E+02**	6.94E+02	1.29E+03
f_{16}	200	**2.01E+02**	**2.01E+02**	**2.01E+02**
f_{17}	300	3.14E+02	3.15E+02	3.39E+02
f_{18}	400	4.30E+02	4.32E+02	4.50E+02
f_{19}	500	**5.01E+02**	**5.01E+02**	5.03E+02
f_{20}	600	6.03E+02	6.03E+02	6.03E+02
f_{21}	700	1.10E+03	**1.09E+03**	1.10E+03
f_{22}	800	1.02E+03	**1.00E+03**	1.41E+03
f_{23}	900	1.72E+03	1.70E+03	2.10E+03
f_{24}	1000	1.21E+03	1.21E+03	1.21E+03
f_{25}	1100	1.31E+03	**1.30E+03**	1.31E+03
f_{26}	1200	1.35E+03	1.35E+03	1.36E+03
f_{27}	1300	**1.63E+03**	**1.63E+03**	1.66E+03
f_{28}	1400	1.75E+03	1.75E+03	**1.71E+03**

The summary of mean results of the tuning experiment is given in Table 7.10. The best result and corresponding parameters values are highlighted. Following the results of the tuning experiment second CPRNG was constructed based on Lozi map with setting a = 1.5 and b = 0.45. The x,y plot of tuned Lozi map is given in Figure 7.13. The distribution of CPRNG based on tuned Lozi map is given in Figure 7.14 and sample sequence of tuned CPRNG is depicted in Figure 7.15.

This basic tuning experiment proved that by tuning the parameters of chaotic

Tab. 6.10 Tuning experiment - mean final results for*f15*;100 runs

a/b	0.1	0.15	0.2	0.25	0.3	0.35	0.4	**0.45**	0.5	0.55	0.6
1.3	911	830	704	1344	1638	1569	1383	1389	1467	1356	1402
1.35	851	820	716	665	1057	1497	1383	1437	1481	1391	1285
1.4	753	703	689	676	655	838	1397	1416	1424	1229	1325
1.45	658	715	724	713	547	617	593	1214	1373	1276	1270
1.5	703	760	757	731	671	749	701	**534**	1203	1171	1217
1.55	725	732	699	752	702	713	750	762	625	1118	1204
1.6	679	826	737	701	643	807	657	643	692	745	1066
1.65	761	680	756	681	760	798	715	706	812	708	685
1.7	615	771	752	715	696	696	774	724	747	701	690

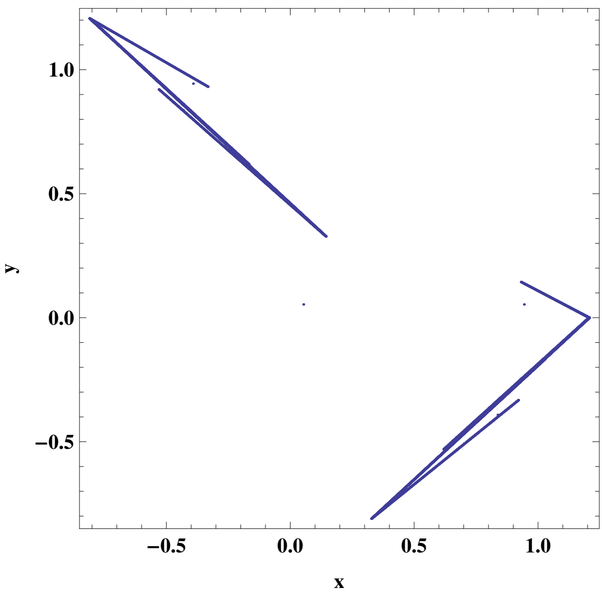

Fig. 6.13 x,y plot of Lozi map - tuned

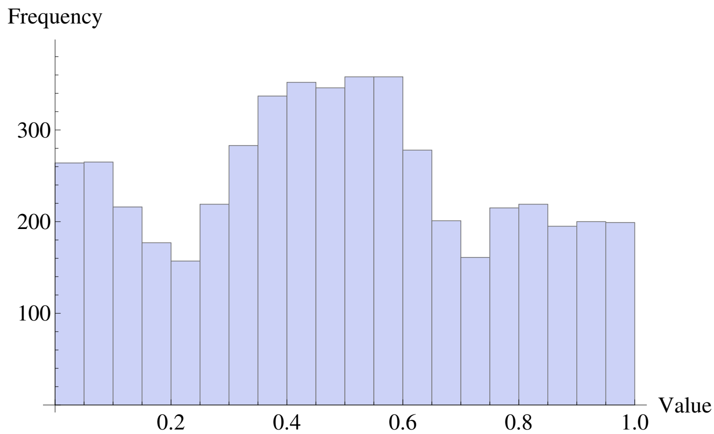

Fig. 6.14 Distribution of CPRNG - tuned

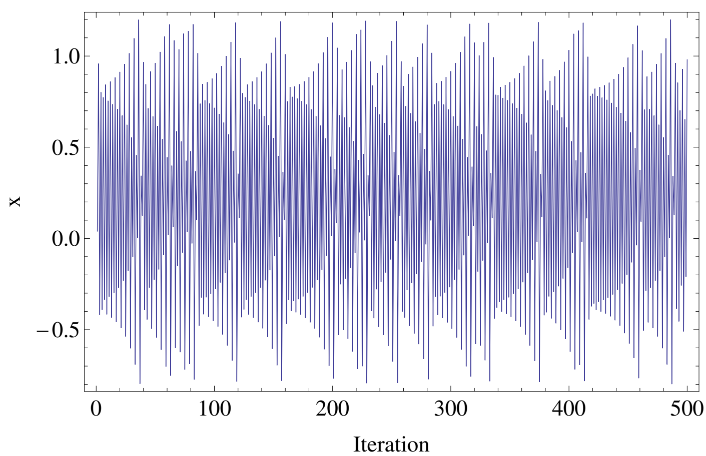

Fig. 6.15 Pseudo-random sequence sample – Lozi map based CPRNG - tuned

map it is possible to further improve the performance of chaos driven PSO algorithm on a particular problem. However the overall performance had to be re-evaluated. The performance of the tuned algorithm was compared with canonical GPSO and Lozi map enhanced PSO with default control parameters. The results of testing on the CEC 13 benchmark are given in Table 5.10 where:

- GPSO with canonical PRNG is noted GPSO
- GPSO with CPRNG based on Lozi map (a = 1.7, b = 0.5) is noted GPSO Lozi 1
- GPSO with CPRNG based on Lozi map (a = 1.5, b = 0.45) is noted GPSO Lozi 2

The bold numbers represents the best mean results. The mean results are presented alongside the total number of best results obtained. Furthermore the performance of pairs of algorithms is compared, where 1 stands for "win" of the "algorithm 1" (the first from the pair - left); number 2 stand for "win" of algorithm 2 (the second from the pair - right) and 0 stands for draw. The final score is also given in Tables 7.11 as a sum of points for wins (1 point) and draws (0.5 point).

6.5 Chaos PSO - discussion

In the previous section the research of chaos driven particle swarm algorithm was presented. Following the work of other researchers [14, 15, 11] the impact of various chaotic PRNGs on the performance of PSO was studied in detail. The implementation method was experimentally selected and six different chaotic systems were utilized as CPRNGs for PSO algorithm. It has been experimentally demonstrated that using CPRNG for velocity calculation in PSO algorithm may have significant impact on the convergence behavior of the algorithm and the final result quality. Also it was observed that the effect depends on the particular system and its setting. All presented experiments lead to better understanding

Tab. 6.11 Mean final results comparison- CEC 13 Benchmark set

f(x)	GPSO	GPSO Lozi 1	GPSO Lozi 2	A1 vs. A2	A1 vs. A3	A2 vs. A3
f_1	**-1.40E+03**	**-1.40E+03**	**-1.40E+03**	0	0	0
f_2	1.59E+05	1.75E+05	**7.72E+04**	1	2	2
f_3	**1.73E+06**	7.13E+06	4.20E+06	1	1	2
f_4	-7.45E+02	**-8.15E+02**	-2.13E+02	2	1	1
f_5	**-1.00E+03**	**-1.00E+03**	**-1.00E+03**	0	0	0
f_6	-8.89E+02	-8.90E+02	**-8.92E+02**	2	2	2
f_7	**-7.97E+02**	-7.95E+02	-7.96E+02	1	1	2
f_8	**-6.80E+02**	**-6.80E+02**	**-6.80E+02**	0	0	0
f_9	**-5.97E+02**	-5.96E+02	**-5.97E+02**	1	0	2
f_{10}	**-5.00E+02**	-4.99E+02	**-5.00E+02**	1	0	2
f_{11}	**-3.98E+02**	-3.97E+02	**-3.98E+02**	1	0	2
f_{12}	-2.87E+02	-2.88E+02	**-2.91E+02**	2	2	2
f_{13}	-1.81E+02	-1.82E+02	**-1.88E+02**	2	2	2
f_{14}	1.42E+02	**1.20E+02**	1.60E+02	2	1	1
f_{15}	5.96E+02	7.28E+02	**5.79E+02**	1	2	2
f_{16}	**2.01E+02**	**2.01E+02**	**2.01E+02**	0	0	0
f_{17}	3.14E+02	**3.13E+02**	3.15E+02	2	1	1
f_{18}	4.34E+02	**4.22E+02**	4.26E+02	2	2	1
f_{19}	**5.01E+02**	**5.01E+02**	**5.01E+02**	0	0	0
f_{20}	**6.02E+02**	6.03E+02	**6.02E+02**	1	0	2
f_{21}	**1.08E+03**	1.09E+03	1.10E+03	1	1	1
f_{22}	**1.00E+03**	1.07E+03	1.04E+03	1	1	2
f_{23}	1.65E+03	1.66E+03	**1.64E+03**	1	2	2
f_{24}	**1.20E+03**	1.21E+03	1.20E+03	1	1	2
f_{25}	1.30E+03	1.30E+03	**1.30E+03**	1	2	2
f_{26}	1.35E+03	1.35E+03	**1.33E+03**	2	2	2
f_{27}	**1.63E+03**	1.63E+03	1.63E+03	1	1	0
f_{28}	1.74E+03	**1.74E+03**	1.76E+03	2	1	1
Best:	11	10	13	16.5 : 11.5	14.5 : 13.5	9.0 : 19.0

of the effect of CPRNG on the PSO algorithm and accumulated enough data for further extension of this approach as will be presented further.

6.6 Chaos PSO - application

In this section the results of application of chaotic PSO on a real-life problem is presented. In this case a PID controller was designed using evolutionary computational techniques. In this example four chaos enhanced PSO variant were used: PSO enhanced with CPRNG based on Lozi map, Dissipative map, Tinkerbell map and Burgers map.

6.6.1 Problem design

The PID controller contains three unique parts; proportional, integral and derivative controller [38, 39]. A simplified form in Laplace domain is given by (7.3)

$$G(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right) \tag{6.3}$$

The PID form most suitable for analytical calculations is given by (7.4)

$$G(s) = k_p + \frac{k_i}{s} + k_d s \tag{6.4}$$

The parameters are related to the standard form through: $k_p = K$, $k_i = K/T_i$ and $k_d = KT_d$. Acquisition of the combination of these three parameters that gives the lowest value of the test criterions was the objective of this research.

Two different systems were used in this research. First was the DC motor system given by (7.5)

$$G(s) = \frac{0.9}{0.00105s^3 + 0.2104s^2 + 0.8913s} \tag{6.5}$$

Second used system was the 4^{th} order system that is given by (7.6)

$$G(s) = \frac{1}{s^4 + 6s^3 + 11s^2 + 6s} \tag{6.6}$$

6.6.2 Cost function

Test criterion measures properties of output transfer function and can indicate quality of regulation [11, 39, 40]. Following four different integral criterions were used for the test and comparison purposes: IAE (Integral Absolute Error), ITAE (Integral Time Absolute Error), ISE (Integral Square Error) and MSE (Mean Square Error). These test criterions (given by (7.7) - (7.10) were minimized within the cost functions for the enhanced PSO algorithm.

Integral of Time multiplied by Absolute Error (ITAE)

$$I_{ITAE} = \int_0^T t|e(t)|dt \tag{6.7}$$

Integral of Absolute Magnitude of the Error (IAE)

$$I_{IAE} = \int_0^T |e(t)|dt \tag{6.8}$$

Integral of the Square of the Error (ISE)

$$I_{ISE} = \int_0^T e^2(t)dt \tag{6.9}$$

Mean of the Square of the Error (MSE)

$$I_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(e(t))^2 \tag{6.10}$$

6.6.3 Results

In following Tables 7.12 and 7.13 the results of optimization using four above described criterions as CF are presented. The results of chaos enhanced PSO variants are compared with previously published results of heuristic and non-heuristic methods [11, 40].

Furthermore examples of the step responses of the systems with PID controllers designed by different chaos-driven PSO algorithms are depicted in Figure 7.16 and 7.17.

Tab. 6.12 Results - DC motor PID controller design

Criterion	IAE	ITAE	ISE	MSE
Z-N (step response)	0.51760	3.38050	2.34670	0.01170
Kappa-Tau	0.51880	3.31130	2.25030	0.07778
Contin. cycling	0.56000	7.82000	3.20000	0.01600
EP	0.48910	0.07210	1.02770	0.00510
GA	0.77120	0.37810	1.04350	0.00520
PSO	0.91610	0.02290	1.00160	0.00500
PSO Lozi	0.21673	0.00667	**0.01839**	**0.00092**
PSO Disi	0.22306	0.00862	**0.01839**	**0.00092**
PSO Burger	0.21026	**0.00569**	**0.01839**	**0.00092**
PSO Tinker	**0.20766**	0.00899	**0.01839**	**0.00092**

In this example it was demonstrated that the chaos driven PSO algorithm is able

Tab. 6.13 Results - 4^{th} order system PID controller design

Criterion	IAE	ITAE	ISE	MSE
Z-N (step response)	34.94129	137.56500	17.84260	0.08921
DE Chaos	**12.33050**	15.38460	6.41026	**0.03203**
SOMA chaos	**12.33050**	15.38460	6.41026	**0.03203**
PSO Lozi	12.35760	16.09520	6.40521	**0.03203**
PSO Disi	12.34790	15.5334	**6.40516**	**0.03203**
PSO Burger	12.37382	16.40790	6.40538	**0.03203**
PSO Tinker	12.37004	**15.25892**	6.40517	**0.03203**

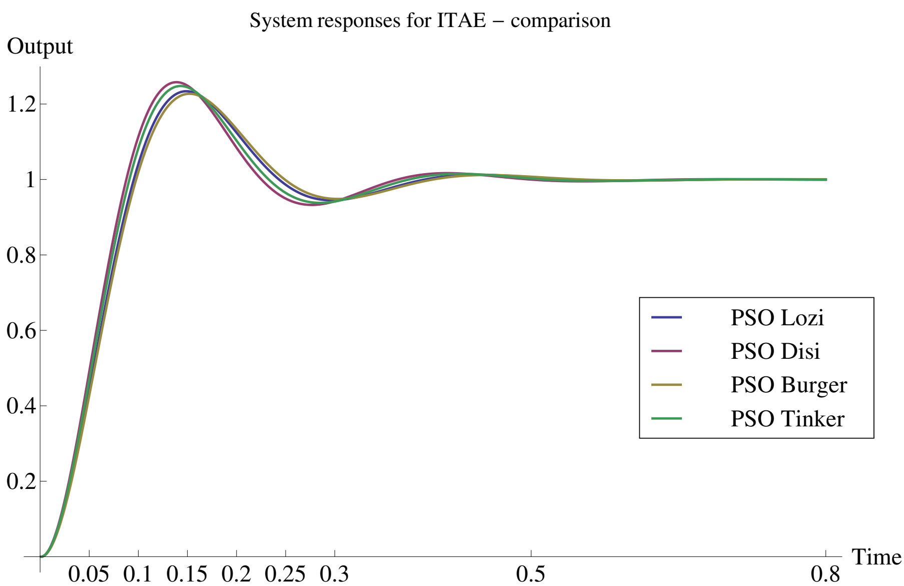

Fig. 6.16 Comparison of system responses - DC motor system - ITAE criterion

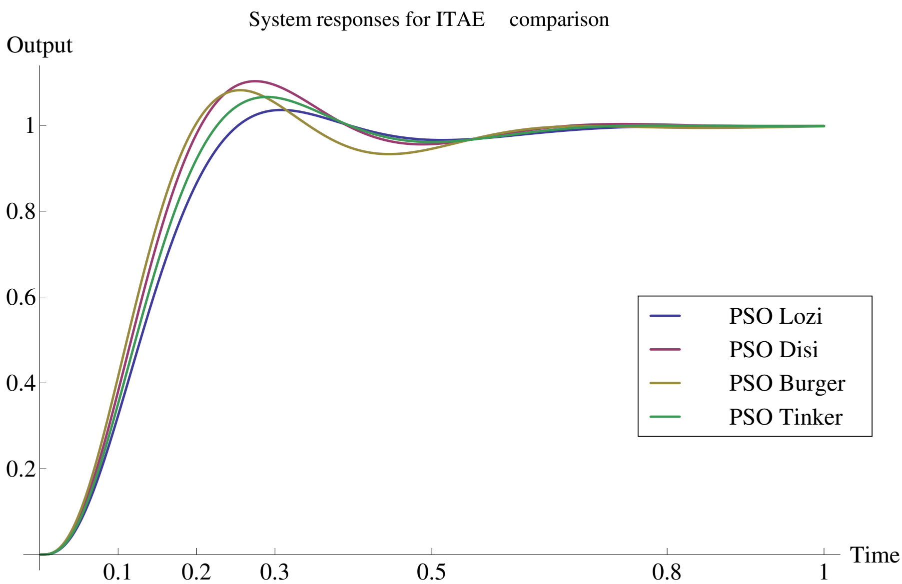

Fig. 6.17 Comparison of system responses - 4^{th} order system - ITAE criterion

to find significantly better solutions for the PID controller design problem than deterministic methods or canonical heuristics. Also it is depicted that different CPRNG leads to different solution.

7 Multi-chaos PSO

The successful experiments with chaos driven PSO algorithm and the detailed observations of inner dynamics of such algorithm led to the design of multi-chaotic PSO. In this pioneering approach multiple (usually two) chaotic PRNGs are used gradually within one run of the algorithm. The algorithm is started with one CPRNG and later the CPRNGs are switched either manually or as a result of adaptive process.

The inspiration for this method came mostly from the convergence graphs (Figure 7.11 and 7.11) of chaotic PSO. It is clear that some CPRNGs support the convergence rate at the cost of premature convergence risk and performance stagnation and vice versa. Therefore based on the data from previous experiments promising pairs of CPRNGs were selected and used in the multi-chaotic approach. The goal was to benefit from fast initial convergence provided by the first CPRNG and use second CPRNG to help the algorithm avoid premature convergence into local extreme. Later various combinations of CPRNGs were tested.

7.1 Initial design

In the initial design the Lozi map and Dissipative map were chosen as promising pair of CPRNGs. The control parameters of PSO algorithm were set in the following way:

Population size: 100; Iterations: 1000; w_{start}: 0.9; w_{end}: 0.4; Dimension: 40.

For statistical reasons, optimization for each setting was repeated 50 times.

In the following sections four versions of PSO algorithm were used. The notation is as follows:

- PSO Weight - PSO algorithm with inertia weight - non-chaotic number generator
- PSO Lozi - PSO algorithm with inertia weight - chaotic number generator - Lozi map
- PSO Disi - PSO algorithm with inertia weight - chaotic number generator - Dissipative standard map
- PSO Chaos - PSO algorithm with inertia weight - two chaotic number generators – Lozi map and Dissipative standard map (in given order).

Within this preliminary study the 'switching point" was set manually and with prior knowledge of the problem (test functions) as follows:

- Sphere function: 250 iterations
- Rosenbrock's function: 380 iterations
- Rastrigin's function: 300 iterations
- Schwefel's function: 800 iterations

The results of the experiment are presented in Tables 8.1 - 8.4 and Figures 8.1 - 8.4. Based on the evidence it may be said that it is possible to further improve the performance of chaos PSO when multi-chaotic approach is implemented.

Tab. 7.1 Results for Sphere function

	PSO Chaos	***PSO Lozi***	***PSO Disi***	***PSO Weight***
Mean CF Value:	**0.002671**	0.249126	0.0094585	0.159768
Std. Dev.:	0.00190698	0.118659	0.00573836	0.0801359
CF Value Median:	0.00188625	0.237395	0.00862576	0.155866
Max. CF Value:	0.00891579	0.569807	0.0350164	0.46442
Min. CF Value:	**0.0004576**	0.0601825	0.00180644	0.0493965

Tab. 7.2 Results for Rosenbrock's function

	PSO Chaos	***PSO Lozi***	***PSO Disi***	***PSO Weight***
Mean CF Value:	**37.3217**	49.7668	38.0615	42.6178
Std. Dev.:	1.21635	8.87105	0.912696	2.4834
CF Value Median:	37.4701	46.2212	38.1459	41.8757
Max. CF Value:	39.3733	83.8619	39.6336	48.6175
Min. CF Value:	**33.0365**	41.1448	35.4239	39.285

7.2 Adaptive approach

After the initial experiments with multi-chaotic PSO the need for an adaptive approach was addressed. Based on the results presented in the previous sections two promising combinations of chaotic PRNGs were selected. The first

Tab. 7.3 Results for Rastrigin's function

	PSO Chaos	***PSO Lozi***	***PSO Disi***	***PSO Weight***
Mean CF Value:	**28.1264**	72.5002	34.9056	61.6047
Std. Dev.:	7.17803	17.6654	10.5036	14.9645
CF Value Median:	26.5129	73.3243	30.6565	58.826
Max. CF Value:	47.6554	114.168	61.038	102.034
Min. CF Value:	**13.2115**	40.724	23.6007	37.218

Tab. 7.4 Results for Schwefel's function

	PSO Chaos	***PSO Lozi***	***PSO Disi***	***PSO Weight***
Mean CF Value:	-7286.05	**-7509.91**	-6853.9	-7435.39
Std. Dev.:	549.805	537.325	740.796	661.692
CF Value Median:	-7270.66	-7571.91	-6853.54	-7311.83
Max. CF Value:	-6013.95	-6069.7	-5267.61	-6133.73
Min. CF Value:	-8523.44	-8383.6	-8284.54	**-8927.9**

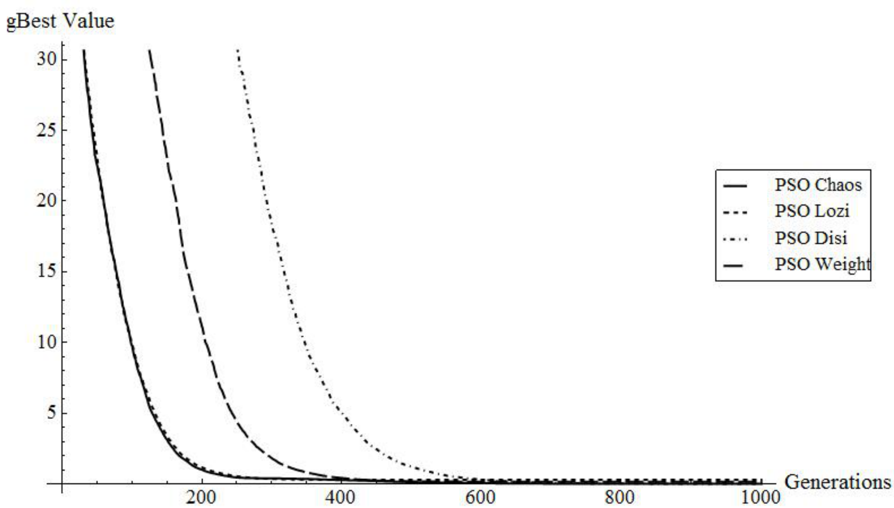

Fig. 7.1 History of mean *gBest* value for Sphere function

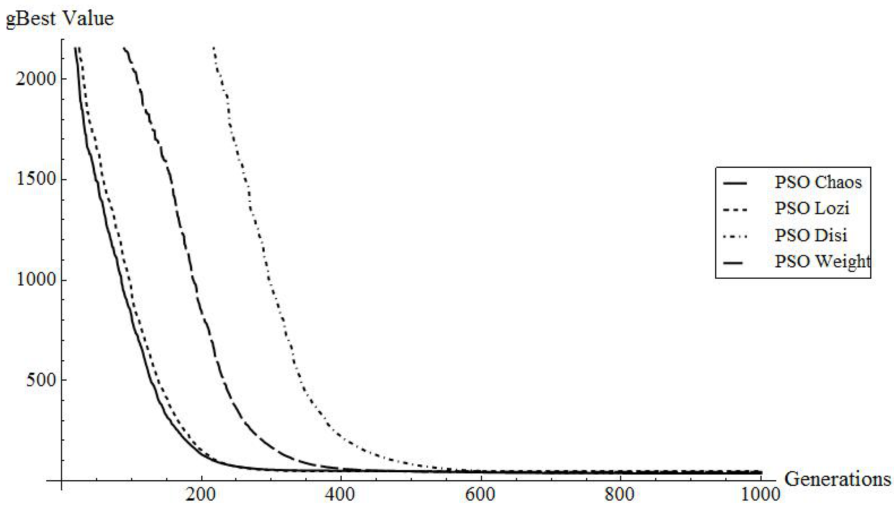

Fig. 7.2 History of mean *gBest* value for Rosenbrock's function

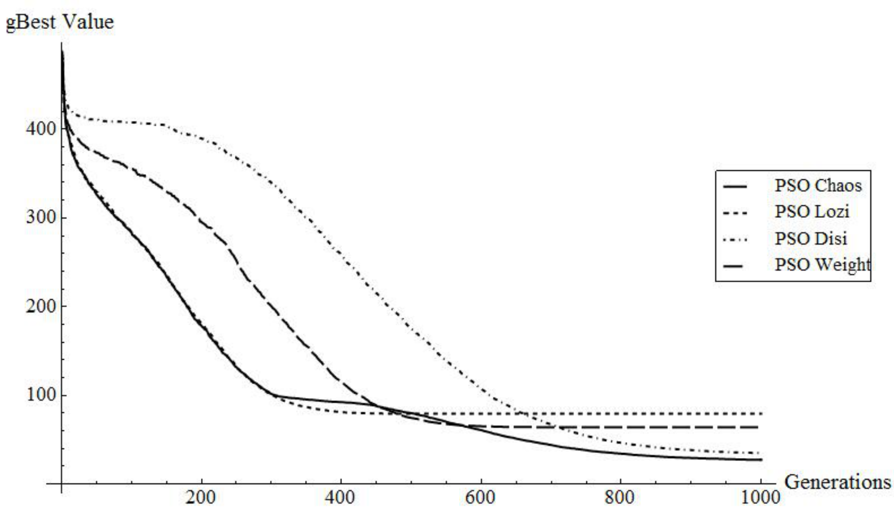

Fig. 7.3 History of mean gBest value for Rastrigin's function

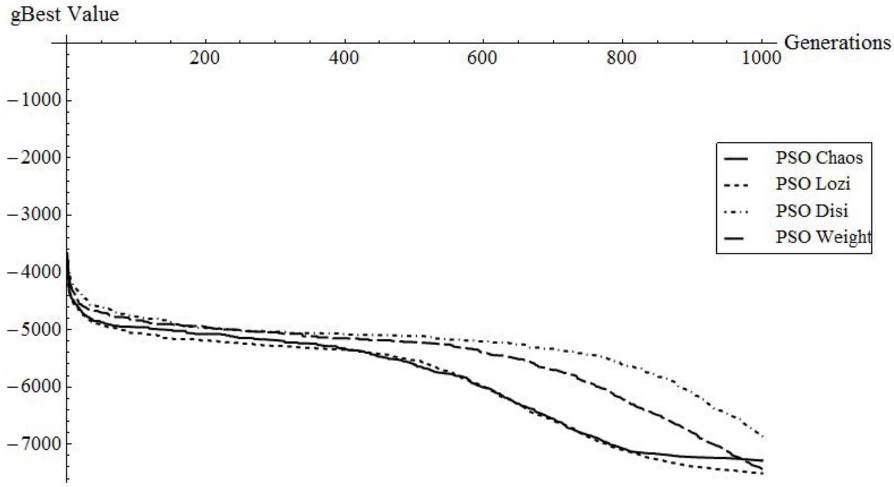

Fig. 7.4 History of mean gBest value for Schwefel's function

combination (noted GPSO Chaos 1) utilized Tinkerbell and Burgers maps. The second one (noted GPSO Chaos 2) combined together Lozi and Burgers maps. As presented previously (Figure 7.11 and 7.12) the GPSO algorithm driven by either Lozi or Tinkerbell map based CPRNGs has manifested very fast speed of initial convergence in comparison with the canonical GPSO. To represent the opposite trend, i.e. permanent and slow trend towards the global extreme, the Burgers map based CPRNG was used for the second phase of the optimization.

The exact moment of the CPRNGs switch is determined dynamically thus may be different for each run of the algorithm. The :improvement" of the *gBest* value over the time is continuously determined and the CPRNGs are switched over when the algorithm falls into stagnation of "significant duration" in terms of the *gBest* value. In the presented experiments, the "significant duration" is considered as a 1/10 of the total number of iterations. In the experiment presented here, the population size was set to 40 and the number of iterations was 5000. To acquire the statistical indicators all experiments were repeated 25 times

The detailed statistical overview of the performance of newly designed PSO Chaos 1 and PSO Chaos 2 versions is presented in Tables 8.5 and 8.6. The overall comparison with the both original (canonical) GPSO algorithm and the OLPSO-G [20] is given in table 8.7. Furthermore for better understanding of the multi-chaos CPRNG influence to the PSO behavior, it is presented the mean history of *gBest* value in the case of Schwefel's (f_{s2}) function 8.5.

Tab. 7.5 Statistical overview - GPSO Chaos 1

GPSO Chaos 1	f_{s1}	f_{s2}	f_{s3}	f_{s4}	f_{s5}	f_{s6}	f_{s7}
Mean CF Value:	6.86 E-92	1.34 E-10	9.35 E+00	3.77 E-03	3.01 E+03	2.31 E+01	6.98 E-15
Std. Dev.:	3.43 E-91	6.68 E-10	1.35 E+01	2.03 E-03	4.99 E+02	4.77 E+00	1.33 E-15
CF Value Median:	1.64 E-104	8.34 E-16	8.11 E+00	3.46 E-03	2.93 E+03	2.24 E+01	7.55 E-15
Max. CF Value:	1.72 E-90	3.34 E-09	6.84 E+01	1.02 E-02	3.86 E+03	3.10 E+01	7.55 E-15
Min. CF Value:	5.52 E-112	9.00 E-22	1.43 E-05	9.60 E-04	2.32 E+03	1.23 E+01	4.00 E-15

Tab. 7.6 Statistical overview - GPSO Chaos 2

GPSO Chaos 2	f_{s1}	f_{s2}	f_{s3}	f_{s4}	f_{s5}	f_{s6}	f_{s7}
Mean CF Value:	1.59 E-108	2.04 E-32	1.21 E+01	4.43 E-03	3.34 E+03	2.49 E+01	7.12 E-15
Std. Dev.:	7.93 E-108	9.73 E-32	1.41 E+01	4.02 E-03	5.98 E+02	6.46 E+00	7.11 E-16
CF Value Median:	3.73 E-113	1.22 E-39	1.28 E+01	3.44 E-03	3.28 E+03	2.36 E+01	7.55 E-15
Max. CF Value:	3.97 E-107	4.87 E-31	7.02 E+01	2.17 E-02	4.69 E+03	4.23 E+01	7.55 E-15
Min. CF Value:	4.41 E-117	1.34 E-50	2.36 E-02	9.18 E-04	2.25 E+03	1.66 E+01	4.00 E-15

Presented evidence strongly supports the proposed method. The performance of multi-chaotic PSO variants is comparable and often better than the performance of original PSO and a state-of-art method OLPSO-G.

Tab. 7.7 Mean results and std. dev. comparison for GPSO, GPSO Chaos 1, GPSO Chaos 2 and OLPSO-G

f(x)	**GPSO**	**GPSO Chaos 1**	**GPSO Chaos 2**	**OLPSO-G**
f_{s1}	9.90E-35± 2.08-E34	6.86E-92 ± 3.43E-91	**1.58E-108 ± 7.93E-108**	4.12E-54 ± 6.34E-54
f_{s2}	1.57E-21 ± 5.46E-21	1.34E-10 ± 6.67E-10	**2.03E-32 ± 9.73E-32**	9.85E-30 ± 1.01E-29
f_{s3}	42.61 ± 27.67	**9.34 ± 13.47**	12.06 ± 14.06	21.52 ± 29.92
f_{s4}	7.01E-03 ± 2.67E-03	**3.76E-3 ± 2.02E-3**	4.43E-3 ± 4.01E-3	1.16E-2 ± 4.10E-3
f_{s5}	3789.28 ± 449.18	3007.47 ± 498.70	3338.27 ± 597.90	**384 ± 217**
f_{s6}	26.50 ± 8.95	23.08 ± 4.77	24.88 ± 6.45	**1.07 ± 0.99**
f_{s7}	1.13E-14 ± 3.69E-15	**6.98E-15 ± 1.32E-15**	7.12E-15 ± 7.10E-16	7.98E-15 ± 2.03E-15

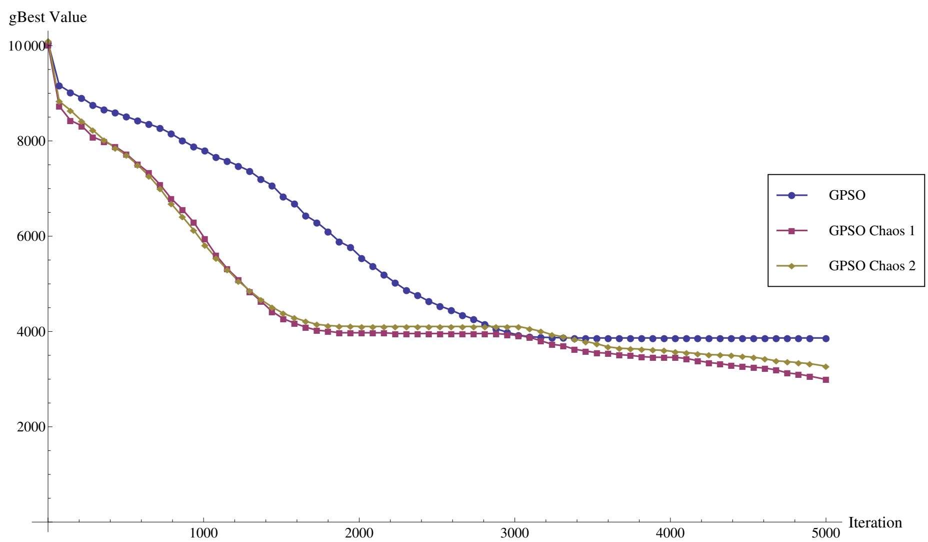

Fig. 7.5 History of mean gBest value for Schwefel's function

8 Multi-chaotic Differential evolution

In reaction to very positive results of multi-chaos PSO the concept was embedded into another popular evolutionary computational technique - the Differential Evolution.

8.1 Differential Evolution (DE)

DE is a population-based optimization method that works on real-number-coded individuals [6]. 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 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 [1] for the detailed description of the used DERand1Bin strategy (9.1) (both for ChaosDE and Canonical DE).

$$u_{i,G+1} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G}) \tag{8.1}$$

8.2 The Concept of ChaosDE

As two different types of numbers are required in ChaosDE; real and integers, the use of modulo operators is used to obtain values between the specified ranges, as given in the following equations (9.2) and (9.3):

$$rndreal = mod(abs(rndChaos), 1.0) \tag{8.2}$$

$$rndint = mod(abs(rndChaos), 1.0) \cdot Range + 1 \tag{8.3}$$

Where *abs* refers to the absolute portion of the chaotic map generated number *rndChaos*, and *mod* is the modulo operator. *Range* specifies the value (inclusive) till where the number is to be scaled.

8.3 Results

The novelty of this approach represents the utilization of discrete chaotic maps as the multi-chaotic pseudo random number generator for the DE. In this paper, the canonical DE strategy DERand1Bin and the Multi-Chaos DERand1Bin strategy driven alternately by two different chaotic maps (ChaosDE) were used.

The previous research [41, 42] showed that through utilization of Burgers and Tinkerbelt map the unique properties with connection to DE were achieved: strong progress towards global extreme, but weak overall statistical results, like average (benchmark function) Cost Function (CF) value and std. dev. Whereas through the utilization of the Lozi and Delayed Logistic map the continuously stable and very satisfactory performance of ChaosDE was achieved. The idea is then to connect these two different influences to the performance of DE into the one novel multi-chaotic concept. The moment of manual switching over between two chaotic maps as well as the parameter settings for both canonical DE and ChaosDE were obtained analytically based on numerous experiments and simulations (see Table 9.1)

Within this initial research, one type of experiment was performed. It utilizes

Tab. 8.1 Parameter set up for canonical DE and ChaosDE

DE Parameter	Value
Popsize	75
F	0.8
Cr	0.8
Dimensions	30
Generations	100D = 3000
CFE limit	225000

the maximum number of generations fixed at 3000 generations. This allowed the possibility to analyze the progress of DE within a limited number of generations and cost function evaluations. The optimum of the cost functions was shifted randomly so the position of the optimum for E_n: $(x_1, x_2 \ldots x_n) = \boldsymbol{s}$,

Where s_i is a random number from the 50% range of function interval; $\boldsymbol{s}$ vector is randomly generated before each run of the optimization process.

The statistical results of the experiments are shown in Tables 9.2, 9.4, 9.6, which represent the simple statistics for cost function values, e.g. average, median, maximum values, standard deviations and minimum values representing the best individual solution for all 50 repeated runs of canonical DE and several versions of ChaosDE and Multi-ChaosDE.

Tables 9.3, 9.5 and 9.7 compare the progress of several versions of ChaosDE, Multi-ChaosDE and Canonical DE. These tables contain the average CF values for the generation No. 750, 1500, 2250 and 3000 from all 50 runs. The bold values within the all Tables 9.2 - 9.7 depict the best obtained result. Furthemore it is presented in Fig. 9.1 - 9.3 the mean progression of CF Value. The impact of CPRNG switching on the convergence behavior can be clearly observed. Following versions of Multi-ChaosDE were studied:

Burgers-Lozi-Switch-500: Start with Burgers map CPRNG, switch to the Lozi map CPRNG after 500 generations.

Lozi-Burgers-Switch-1500: Start with Lozi map CPRNG, switch to the Burgers map CPRNG after 1500 generations.

Tab. 8.2 Simple results statistics for the shifted Sphere function – 30D

DE Version	Avg CF	Median CF	Max CF	Min CF	StdDev
Canonical DE	5.929778	5.435726	11.69084	2.53501	2.432546
Lozi-No-Switch	3.73E-05	2.27E-05	0.000222	1.54E-06	4.17E-05
Burger-No-Switch	**1.02E-14**	**2.88E-15**	**5.73E-14**	**5.98E-17**	**1.49E-14**
Burger-Lozi-Switch-500	1.78E-06	4.2E-07	2.95E-05	1.59E-08	4.61E-06
Lozi-Burger-Switch-1500	8.34E-10	2.75E-10	1.2E-08	2.81E-11	1.76E-09

Tab. 8.3 Comparison of progress towards the minimum for the shifted Sphere function

DE Version	Generation No.750	Generation No.1500	Generation No. 2250	Generation No. 3000
Canonical DE	482.4017	114.3075	26.34619	5.929778
Lozi-No-Switch	90.40304	0.74516	0.004854	3.73E-05
Burger-No-Switch	**0.531726**	**1.33E-05**	**4.32E-10**	**1.02E-14**
Burger-Lozi-Switch-500	2.764289	0.022319	0.000201	1.78E-06
Lozi-Burger-Switch-1500	87.49406	0.709014	3.15E-05	8.34E-10

Obtained numerical results support the claim that all Multi-Chaos/ChaosDE versions have given better overall results in comparison with the canonical DE version. Although the shifted benchmark functions were utilized, from the presented data for the unimodal Sphere function it follows, that Multi-Chaos DE versions driven by Lozi/Burgers Map have given very satisfactory results, nevertheless the single-chaos concept of original ChaosDE has given the best overall results. High sensitivity of the differential evolution on the selection, settings and internal dynamics of the chaotic PRNG is fully manifested in the case of multi-modal functions.

Tab. 8.4 Simple results statistics for the shifted Ackley's function – 30D

DE Version	Avg CF	Median CF	Max CF	Min CF	StdDev
Canonical DE	3.791676	3.841045	4.518592	2.95934	0.341008
Lozi-No-Switch	0.005533	0.00452	0.014929	0.00154	0.003334
Burger-No-Switch	0.067287	**6.34E-08**	1.501747	**5.47E-09**	0.27714
Burger-Lozi-Switch-500	8.04E-04	7.24E-04	0.002667	1.85 E-04	4.84E-04
Lozi-Burger-Switch-1500	**1.77E-05**	1.03E-05	**7.6E-05**	1.95E-06	**1.46E-05**

Tab. 8.5 Comparison of progress towards the min. for the shifted Ackley's function

DE Version	Generation No.750	Generation No.1500	Generation No. 2250	Generation No. 3000
Canonical DE	13.16276	8.511778	5.506989	3.791676
Lozi-No-Switch	8.199525	1.79389	0.081797	0.005533
Burger-No-Switch	**1.548046**	**0.071167**	0.067307	0.067287
Burger-Lozi-Switch-500	2.797948	0.168713	0.009855	8.04E-04
Lozi-Burger-Switch-1500	7.852258	1.621723	**0.003654**	**1.77E-05**

Tab. 8.6 Simple results statistics for the shifted Rastrigin's function – 30D

DE Version	Avg CF	Median CF	Max CF	Min CF	StdDev
Canonical DE	270.9612	273.2324	305.4176	234.2218	15.99992
Lozi-No-Switch	50.68194	45.70853	110.4599	21.52906	21.71585
Burger-No-Switch	44.36785	43.06218	80.76961	16.9143	15.17985
Burger-Lozi-Switch-500	**38.67436**	**36.68143**	82.46749	**16.18175**	**11.82373**
Lozi-Burger-Switch-1500	42.94927	43.09553	**72.74598**	20.8219	13.53718

Tab. 8.7 Comparison of progress towards the minimum for the shifted Rastrigin‘s function

DE Version	Generation No.750	Generation No.1500	Generation No. 2250	Generation No. 3000
Canonical DE	790.1378	404.8734	308.3072	270.9612
Lozi-No-Switch	370.954	177.9286	93.68944	50.68194
Burger-No-Switch	**189.6604**	**55.04461**	**44.56468**	44.36785
Burger-Lozi-Switch-500	221.8914	116.8081	60.55444	**38.67436**
Lozi-Burger-Switch-1500	365.1778	171.6624	57.50722	42.94927

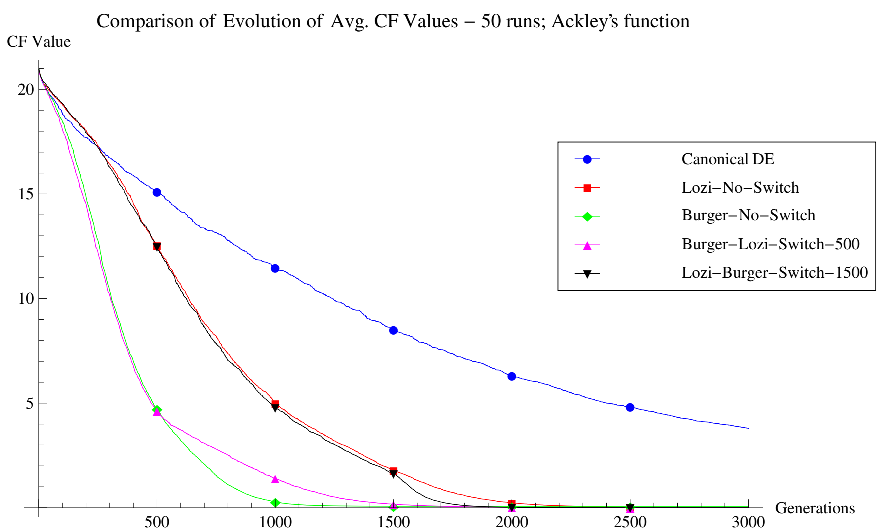

Fig. 8.1 Comparison of Evolution of Avg. CF Values - runs; Ackley's function

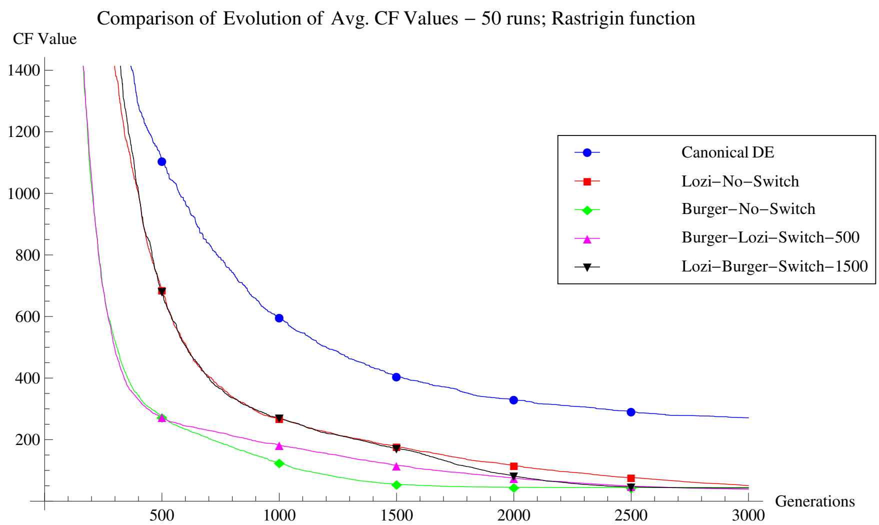

Fig. 8.2 Comparison of Evolution of Avg. CF Values - runs; Rastrigin function

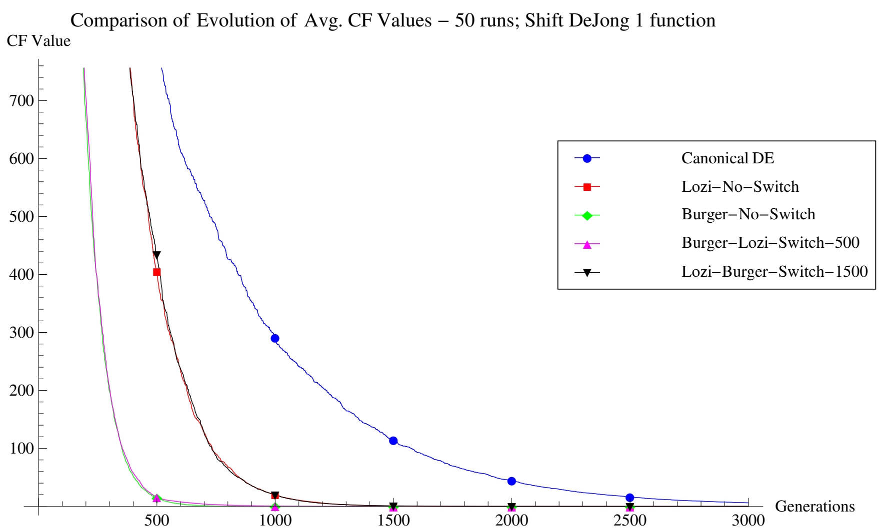

Fig. 8.3 Comparison of Evolution of Avg. CF Values - runs; Shifted Dejong 1 (Sphere) function

For the *Burgers-Lozi-Switch-500* version the progressive Burgers map CPRNG secured the faster approaching towards the global extreme from the very beginning of evolutionary process. The very fast switch over to the Lozi map based CPRNG helped to avoid the Burgers map based CPRNG weak spots, which are the weak overall statistical results, like average CF value and std. dev.; and tendency to stagnation.

Through the utilization of *Lozi-Burgers-Switch-1500* version, the strong progress towards global extreme given by Burgers map CPRNG helped to the evolutionary process driven from the start by mans of Lozi map CPRNG to achieve almost the best avg. CF and median CF values.

9 Multiple-Choice strategy for PSO

During the experiment witch chaos driven PSO and multi-chaos PSO the inner dynamic of the PSO algorithm was investigated very closely. As a reaction to several weaknesses of the original PSO design a novel Multiple-Choice strategy for PSO (MC-PSO) was developed.

This strategy alters the original way (4.1) of calculating the particle velocity in next generation. At first, three numbers b_1, b_2 and b_3 are defined at the start of algorithm. These numbers represent border values for different rules, so they should follow the pattern: $b_1 < b_2 < b_3$. In this study following values were used: $b_1 = 0.2$, $b_2 = 0.4$, $b_3 = 0.7$. Afterwards during the calculation of new velocity of each particle a random number r is generated from the interval <0, 1>. Than the new velocity is calculated following these four rules:

If $r < b_1$ the new velocity of particle is given by (10.1).

$$v(t+1) = 0 \tag{9.1}$$

If $b_1 < r < b_2$ the new velocity of particle is given by (10.2).

$$v(t+1) = w \cdot v(t) + c \cdot Rand \cdot (x_r(t) - x(t)) \tag{9.2}$$

Where $x_{r(t)}$ is the position of randomly chosen particle.

If $b_2 < r < b_3$ the new velocity of particle is given by (10.3).

$$v(t+1) = w \cdot v(t) + c \cdot Rand \cdot (pBest - x(t)) \tag{9.3}$$

If $b_3 < r$ the new velocity of particle is given by (10.4).

$$v(t+1) = w \cdot v(t) + c \cdot Rand \cdot (gBest - x(t)) \tag{9.4}$$

The priority factors c_1 and c_2 from original PSO formula (4.1) are replaced within this novel approach with single parameter c. Within this new strategy parameter c defines not the priority (which is given by b_1, b_2 and b_3 setting) but the overstep value. Within this research, c was set to 2.

9.1 MC-PSO Experiment

In the experiment it was investigated the performance of MC-PSO on large-scale (high dimensional) problems. The algorithm was set up accordingly (see Table 10.1) and four previously described benchmark functions were used: Sphere function, Rosenbrock's function, Rastrigin's function and Schwefel's function.

Two versions of PSO algorithm were used. The notation is as follows:

PSO Weight – PSO algorithm with linear decreasing inertia weight

MC-PSO – Multiple choice PSO algorithm with linear decreasing inertia weight

The results for each test function and two versions of PSO algorithms are summarized in the statistical overview given in Tables 10.2 - 10.5 . The best results (cost function values) and the best mean results are highlighted by bold numbers within these Tables 10.2 - 10.5. Also the history of *gBest* values for dimension

= 1000 was tracked and it is depicted in Figures 10.1 - 10.4.

Tab. 9.1 Experiment setup

Population size:	200
Iterations:	1000
c1, c2, c:	2
w:	Linear 0.9 -> 0.4
v_{max}	0.2
Dim:	250, 500, 750, 1000
Repeated runs:	30

Tab. 9.2 Results for Sphere function

Dimension:	250		500		750		1000	
PSO Version:	PSO Weight	MC-PSO	PSO Weight	MC-PSO	PSO Weight	MC-PSO	PSO Weight	MC-PSO
Mean Value:	38.5933	**3.59432**	275.316	**24.2015**	275.316	**24.2015**	430.702	**36.7424**
Std. Dev.:	4.436	0.539777	24.67	1.92985	24.67	1.92985	43.5069	2.79294
Median:	38.9674	3.60291	270.794	24.1434	270.794	24.1434	427.56	36.4451
Worst result:	47.6534	4.69833	350.85	28.9283	350.85	28.9283	542.402	43.9094
Best result:	30.6938	**2.61684**	223.634	**20.9344**	223.634	**20.9344**	345.921	**31.715**

Tab. 9.3 Results for Schwefel's function

Dimension:	250		500		750		1000	
PSO Version:	PSO Weight	MC-PSO	PSO Weight	MC-PSO	PSO Weight	MC-PSO	PSO Weight	MC-PSO
Mean Value:	-21107	**-35402.5**	-29678.9	**-53198.6**	-37137.8	**-68255.5**	-41384.9	**-80269.4**
Std. Dev.:	1634.78	2674.73	2580.65	3322.55	3277.72	4314.53	4009.01	4922.38
Median:	-21323.9	-34573.3	-29390.3	-53151.5	-37604.3	-67788.2	-41412.2	-81508
Worst result:	-17224.9	-29375.1	-22540.4	-46185.3	-28237.1	-59119	-32664	-66661.7
Best result:	-24627.7	**-41099.7**	-34761.6	**-59739.5**	-45600.7	**-77153.4**	-51770.9	**-89149.9**

Tab. 9.4 Results for Rastrigin's function

Dimension:	250		500		750		1000	
PSO Version:	PSO Weight	MC-PSO	PSO Weight	MC-PSO	PSO Weight	MC-PSO	PSO Weight	MC-PSO
Mean Value:	1534.4	**936.438**	3816.35	**2713.02**	6222.51	**4665.34**	8681.53	**6766.28**
Std. Dev.:	105.157	94.4946	124.55	162.63	198.081	210.484	196.055	253.688
Median:	1526.65	932.761	3812.5	2702.53	6232.57	4642.29	8730.42	6752.56
Worst result:	1793.88	1196.54	4076.89	3095.76	6578.13	5086.7	8973.59	7549.17
Best result:	1283.73	**746.202**	3593.85	**2336.36**	5673.58	**4199.21**	8217.97	**6338.89**

Tab. 9.5 Results for Rosenbrock's function

Dimension:	250		500		750		1000	
PSO Version:	PSO Weight	MC-PSO	PSO Weight	MC-PSO	PSO Weight	MC-PSO	PSO Weight	MC-PSO
Mean Value:	8315.57	**741.837**	38871.7	**2050.54**	79568.8	**3403.36**	131934	**4977.79**
Std. Dev.:	1506.54	76.9678	7066.53	143.774	15812.2	292.3	21813.9	385.606
Median:	7906.02	727.182	38170.1	2040.09	77810.5	3339.45	132155	4899.54
Worst result:	11901.1	973.58	53198.3	2340.47	123287	4240.53	191451	6229.1
Best result:	5250.74	**629.908**	23867.5	**1761.37**	53897.2	**2902.29**	73943.6	**4320.05**

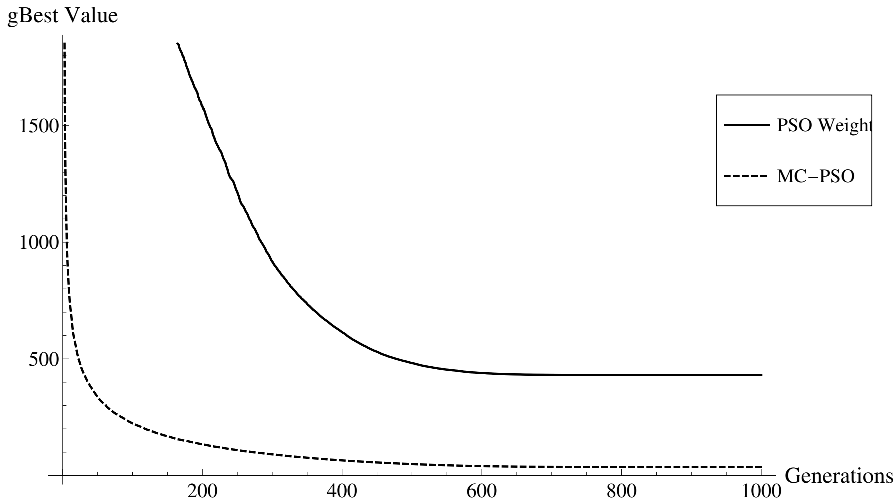

Fig. 9.1 History of mean *gBest* value for Sphere function, Dim = 1000

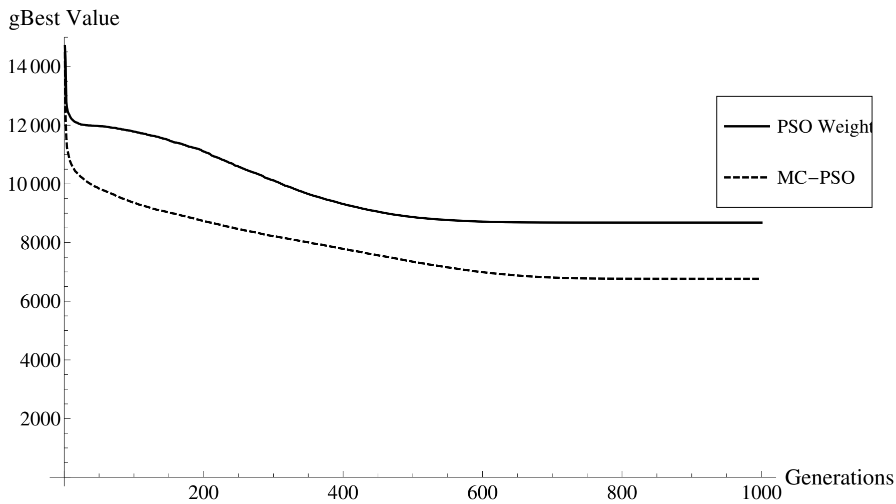

Fig. 9.2 History of mean *gBest* value for Rosenbrock's function, Dim = 1000

The results presented here strongly support the claim that the multiple choice strategy for PSO algorithm as presented in this paper seems to have significant positive effect on the performance of the PSO algorithm in the task of large

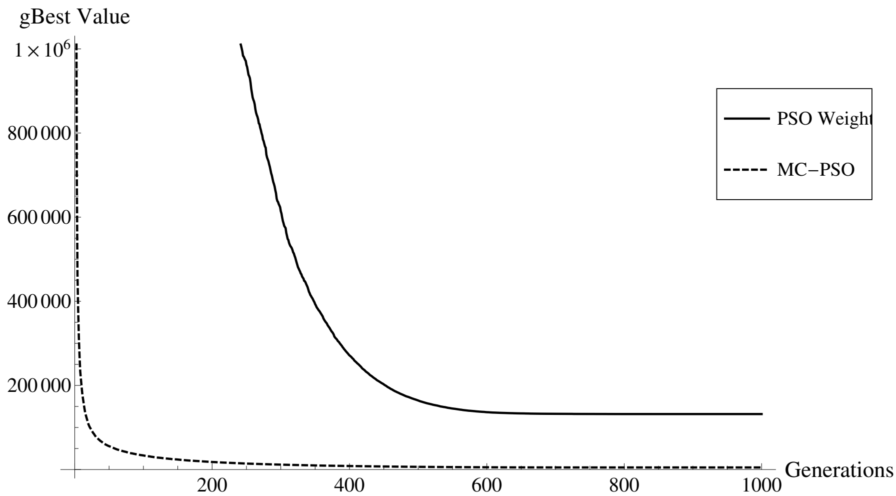

Fig. 9.3 History of mean *gBest* value for Rastrigin's function, Dim = 1000

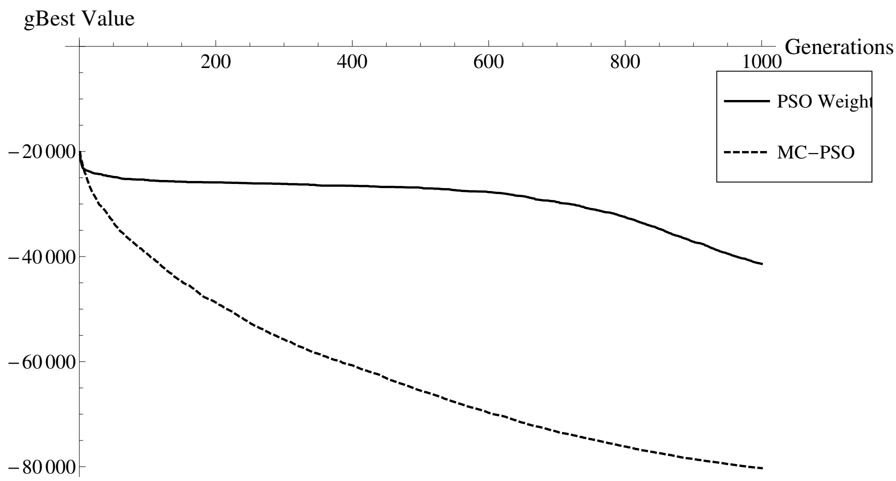

Fig. 9.4 History of mean *gBest* value for Schwefel's function, Dim = 1000

scale optimization problems. The newly proposed algorithm seems to exhibit much faster convergence and better overall performance on the used set of four basic test functions over the given number of iterations. In all the experiments performed the newly proposed MC-PSO algorithm managed to obtain better results.

Moreover during the detailed results analysis it was found out that the new MC-PSO version seems to be also faster in terms of total time demands required for all repeated 50 runs. Example of these results is presented in following Table 10.6 where the average times in seconds are listed for the case study of dim = 1000. The bold number represents the best value.

Tab. 9.6 Time Demands Comparison

Dim = 1000	PSO Weight (s)	MC-PSO (s)
Sphere	439	**270**
Rosenbrock	506	**334**
Rastrigin	639	**489**
Schwefel	1387	**925**

10 The Gathering Algorithm

Within the previous experiments with Multiple-Choice strategy for PSO, it has been found out that it may be very beneficial to make some particles stop for a given number of iterations in order to prevent premature convergence and improve the exploration capability of PSO. In our new design the velocity calculation formula is altered significantly. One random particle from the swarm is selected and the difference between the random particle and active particle is multiplied by a random number from the interval [-0.2,1.2] as given in (11.1). The interval bounds were acquired by tuning with multiple benchmark sets. The maximum velocity is not limited in this proposed design.

$$v_i^{t+1} = Rand(-0.2, 1.2) \cdot (x_{random}^t - x_i^t) \tag{10.1}$$

It is clear from (11.1) that the position of pBest or gBest is not used in the velocity calculations, even though the gBest and pBest values and positions are recorded.

During the initialization phase, each particle is assigned a counter that is set to zero. As long as the counter value for active particle remains zero, the velocity and new position of the active particle is calculated by (11.1). If new pBest or gBest is encountered, the counter value of the corresponding particle is set to 10 (pBest) or random integer number from interval [50, 70] (gBest).The aforementioned values were also set by means of extensive tuning with multiple benchmark sets and combinations of settings.

If the counter for active particle is not zero, the particle becomes a "stationary" particle. In the stationary mode, two different dimensional mutation operations (type A or type B) are randomly selected. Firstly the dimension index is selected randomly. A copy of active particle is created and noted as a trial particle. Thereafter, the type of mutation operation is randomly chosen (50% : 50% chance).

Mutation type A: random number from interval defined by the lower and upper bounds specified for that dimension is generated and stored into the trial particle in corresponding dimension index (11.2). This mutation operation improves exploration.

$$x_{trialj} = Rand(low_j, high_j) \tag{10.2}$$

where:

j – Randomly selected dimension.

x_{trialj} – j^{th} dimension component of trial particle.

low_j, $high_j$ – Lower and upper bounds of j^{th} dimension.

Mutation type B: the current value of active particle in the selected dimension is multiplied by a random real number from interval [0.99, 1.01] and the result is stored into the trial particles corresponding to the dimension index (11.3).

$$x_{trialj} = x_{ij}^{t} \cdot Rand(0.99, 1.01) \tag{10.3}$$

If the trial particle CF value is better than the pBest of the active particle, the counter is increased by 5. If a new gBest is found by the trial particle, the counter is increased even further by 10. In both cases, the active particle is replaced by the trial particle. Otherwise the active particle remains intact. At the end of the iteration loop, the counter value for the active particle is decreased by 1.

For better clarity, the full pseudo-code of the main part of the proposed algorithm is given in Figure 11.1.

10.1 Initial experiment

For the initial performance evaluation and for better understanding of functionality and dimension dependency, the commonly used Schwefel's benchmark function was used with different dimension settings. The dimension of the problem was set to the following values: 2, 5, 10, 20, 30 and 50.

The performance was compared with the original PSO [1, 2] with global topology, linear decreasing inertia weight and velocity limited to 0.2 of the range (noted GPSO). The mean results for 100 runs are summarized in Table 11.1. It may be clearly observed that in comparison with GPSO, the performance of

```
For each iteration (t)
    For each particle (i)
        random = RandomInteger(1,popsize)
        if (counter_i) == 0
            For each dimension (j)
                v_ij^{t+1} = RandomReal(-0.2, 1.2) · (x_randomj^t - x_ij^t)
            x_i^{t+1} = x_i^t + v_i^{t+1}
            If(new pBest) counter_i = 10;
            If(new gBest) counter_i = RandomInteger(50, 70);
        if (counter_i) != 0
            j = RandomInteger(1, Dim)
            if (RandomReal(0, 1) ≤ 0.5)
                x_trialj = RandomReal(low_j, high_j)
            else x_trialj = x_i^t · RandomReal(0.99, 1.01)
            if(CF(x_trial) < pBest_i)
                x_i^{t+1} = x_trial
                counter_i += 5
            if(CF(x_trial) < gBest_i)
                counter_i += 10
            counter_i --
```

Fig. 10.1 Pseudocode of the main part of the proposed algorithm

the Gathering algorithm (noted hereafter as GATHER in Tables and Figures) was promising in terms of dependency of the results on the dimensionality of the problem. This trend is highlighted in Figure 11.2 where the results of Gathering algorithm and GPSO are compared graphically. This hints that the proposed design is very resistant to the notorious "two step forward, one step back" problem of PSO.

Tab. 10.1 Mean results comparison - 100 runs

DIM	**2**	**5**	**10**	**20**	**30**	**50**
GPSO	2.55 E-05	2.19 E+02	6.56 E+02	2.25 E+03	4.19 E+03	8.49 E+03
GATHER	1.09 E-04	2.50 E-03	1.43 E-02	6.81 E-02	8.92 E+00	6.51 E+02

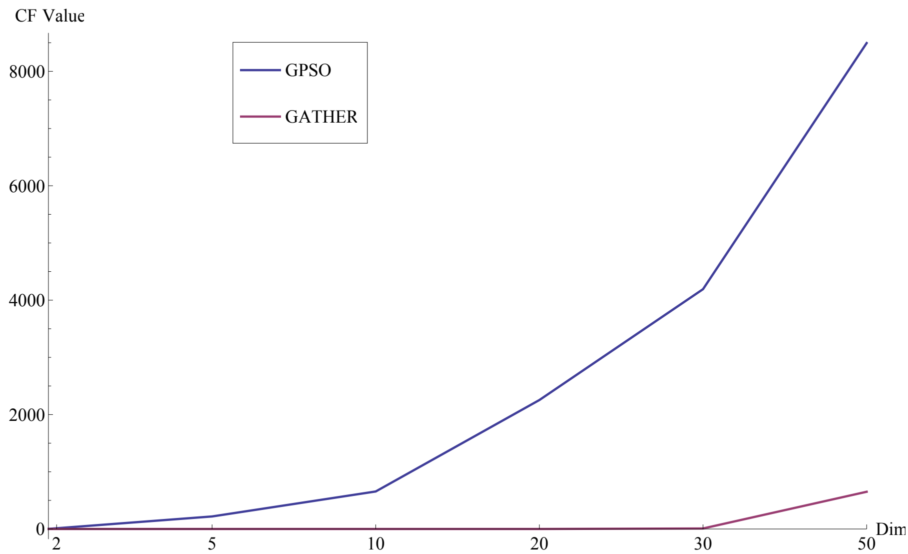

Fig. 10.2 Mean result comparison for different dimension settings - 100 runs

10.2 Benchmark results

In this section, the results of performance testing for the Gathering algorithm on the CEC´13 benchmark set are presented. The statistical overview for all three different dimensional setting (10, 30 and 50) is given in Table 11.2, 11.3 and 11.4. The unimodal functions are denoted with u basic multimodal functions are denoted with m and the composition functions are denoted with c. Furthermore, in Tables 11.5, 11.6 and 11.7, the mean results are compared with two PSO based algorithms participating in the CEC´13 competition: the self-adaptive heterogeneous PSO for real-parameter optimization [43] noted fk-PSO and the Particle Swarm Optimization and Artificial Bee Colony Hybrid algorithm noted ABS-SPSO [44]. The best mean results are highlighted by bold numbers. The analysis of the results follows afterwards. Presented results indicate encouraging performance of the Gathering algorithm for complex and multimodal optimization problems. The lack of performance in the case of unimodal functions hints that the local-search capability should be improved in the future modification. In the case of dim = 10, (See Table 11.5) the Gathering algorithm managed to obtain solo best mean results for six functions (f6, f17, f21, f24, f25, f28). In the case of dim = 30 (See Table 11.6) the Gathering algorithm outperformed other compared algorithm on five functions (f14, f21, f22, f26, f27). With dim = 50, the proposed algorithm managed to obtain individual best results for three functions (f14, f21, f26) (See Table 11.7).

Tab. 10.2 Results for D = 10

f_x	**Best**	**Worst**	**Median**	**Mean**	**Std**
$f_1{}^u$	-1.40E+03	-1.40E+03	-1.40E+03	-1.40E+03	3.12E-05
$f_2{}^u$	3.81E+05	4.64E+06	1.57E+06	1.87E+06	1.35E+06
$f_3{}^u$	8.05E+06	5.67E+08	8.78E+07	1.26E+08	1.15E+08
$f_4{}^u$	7.35E+03	3.20E+04	1.32E+04	1.49E+04	5.05E+03
$f_5{}^u$	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	5.29E-04
$f_6{}^m$	-9.00E+02	-8.98E+02	-9.00E+02	-8.99E+02	5.05E-01
$f_7{}^m$	-7.84E+02	-7.37E+02	-7.58E+02	-7.59E+02	1.16E+01
$f_8{}^m$	-6.80E+02	-6.79E+02	-6.79E+02	-6.79E+02	1.12E-01
$f_9{}^m$	-5.98E+02	-5.92E+02	-5.95E+02	-5.95E+02	9.95E-01
$f_{10}{}^m$	-4.97E+02	-4.90E+02	-4.94E+02	-4.94E+02	1.74E+00
$f_{11}{}^m$	-4.00E+02	-3.98E+02	-4.00E+02	-4.00E+02	4.76E-01
$f_{12}{}^m$	-2.89E+02	-2.65E+02	-2.79E+02	-2.78E+02	6.96E+00
$f_{13}{}^m$	-1.87E+02	-1.53E+02	-1.64E+02	-1.66E+02	8.02E+00
$f_{14}{}^m$	-9.99E+01	-9.63E+01	-9.97E+01	-9.92E+01	1.23E+00
$f_{15}{}^m$	4.13E+02	1.16E+03	8.10E+02	8.29E+02	1.64E+02
$f_{16}{}^m$	2.00E+02	2.01E+02	2.00E+02	2.00E+02	1.85E-01
$f_{17}{}^m$	3.01E+02	3.13E+02	3.09E+02	3.09E+02	3.58E+00
$f_{18}{}^m$	4.25E+02	4.56E+02	4.41E+02	4.41E+02	7.25E+00
$f_{19}{}^m$	5.00E+02	5.01E+02	5.00E+02	5.00E+02	1.71E-01
$f_{20}{}^m$	6.02E+02	6.05E+02	6.03E+02	6.03E+02	6.62E-01
$f_{21}{}^c$	7.00E+02	9.01E+02	8.00E+02	8.30E+02	5.39E+01
$f_{22}{}^c$	8.01E+02	8.41E+02	8.16E+02	8.16E+02	6.86E+00
$f_{23}{}^c$	1.48E+03	2.21E+03	1.94E+03	1.92E+03	1.64E+02
$f_{24}{}^c$	1.11E+03	1.18E+03	1.13E+03	1.13E+03	1.27E+01
$f_{25}{}^c$	1.21E+03	1.32E+03	1.31E+03	1.28E+03	4.00E+01
$f_{26}{}^c$	1.31E+03	1.34E+03	1.32E+03	1.33E+03	6.27E+00
$f_{27}{}^c$	1.61E+03	1.70E+03	1.70E+03	1.70E+03	1.36E+01
$f_{28}{}^c$	1.50E+03	1.70E+03	1.50E+03	1.54E+03	8.02E+01

Tab. 10.3 Results for D = 30

f_x	**Best**	**Worst**	**Median**	**Mean**	**Std**
$f_1{}^u$	-1.40E+03	-1.40E+03	-1.40E+03	-1.40E+03	1.63E-04
$f_2{}^u$	6.28E+06	2.31E+07	1.48E+07	1.48E+07	4.33E+06
$f_3{}^u$	8.29E+08	7.30E+09	3.92E+09	3.87E+09	1.77E+09
$f_4{}^u$	4.64E+04	1.23E+05	9.05E+04	8.96E+04	1.73E+04
$f_5{}^u$	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	1.77E-03
$f_6{}^m$	-8.92E+02	-8.73E+02	-8.79E+02	-8.79E+02	4.43E+00
$f_7{}^m$	-7.28E+02	-6.19E+02	-6.78E+02	-6.79E+02	2.02E+01
$f_8{}^m$	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02	7.71E-02
$f_9{}^m$	-5.76E+02	-5.64E+02	-5.70E+02	-5.70E+02	2.86E+00
$f_{10}{}^m$	-4.94E+02	-4.57E+02	-4.75E+02	-4.75E+02	8.89E+00
$f_{11}{}^m$	-3.99E+02	-3.94E+02	-3.96E+02	-3.96E+02	1.21E+00
$f_{12}{}^m$	-1.71E+02	-5.14E+01	-1.01E+02	-1.04E+02	2.93E+01
$f_{13}{}^m$	-3.03E+01	1.27E+02	8.29E+01	7.55E+01	3.57E+01
$f_{14}{}^m$	-9.43E+01	-7.91E+01	-8.85E+01	-8.80E+01	3.87E+00
$f_{15}{}^m$	2.04E+03	4.51E+03	3.79E+03	3.81E+03	4.51E+02
$f_{16}{}^m$	2.00E+02	2.02E+02	2.01E+02	2.01E+02	3.66E-01
$f_{17}{}^m$	3.18E+02	3.50E+02	3.41E+02	3.41E+02	4.99E+00
$f_{18}{}^m$	5.93E+02	8.40E+02	7.35E+02	7.31E+02	5.47E+01
$f_{19}{}^m$	5.01E+02	5.03E+02	5.02E+02	5.02E+02	4.90E-01
$f_{20}{}^m$	6.12E+02	6.15E+02	6.15E+02	6.15E+02	5.79E-01
$f_{21}{}^c$	8.07E+02	9.23E+02	9.01E+02	8.83E+02	2.92E+01
$f_{22}{}^c$	8.20E+02	9.40E+02	8.40E+02	8.52E+02	3.34E+01
$f_{23}{}^c$	4.16E+03	7.14E+03	5.71E+03	5.67E+03	6.25E+02
$f_{24}{}^c$	1.24E+03	1.31E+03	1.29E+03	1.29E+03	1.09E+01
$f_{25}{}^c$	1.38E+03	1.42E+03	1.41E+03	1.41E+03	9.75E+00
$f_{26}{}^c$	1.40E+03	1.40E+03	1.40E+03	1.40E+03	2.90E-01
$f_{27}{}^c$	1.70E+03	2.57E+03	1.70E+03	1.99E+03	3.84E+02
$f_{28}{}^c$	1.53E+03	2.26E+03	1.92E+03	1.85E+03	2.14E+02

Tab. 10.4 Results for D = 50

f_x	**Best**	**Worst**	**Median**	**Mean**	**Std**
$f_1{}^u$	-1.40E+03	-1.40E+03	-1.40E+03	-1.40E+03	5.08E-03
$f_2{}^u$	1.25E+07	4.51E+07	2.68E+07	2.62E+07	6.18E+06
$f_3{}^u$	3.94E+09	2.37E+10	1.72E+10	1.67E+10	3.98E+09
$f_4{}^u$	1.06E+05	2.09E+05	1.54E+05	1.58E+05	2.09E+04
$f_5{}^u$	-1.00E+03	-1.00E+03	-1.00E+03	-1.00E+03	5.53E-03
$f_6{}^m$	-8.59E+02	-8.51E+02	-8.54E+02	-8.54E+02	1.46E+00
$f_7{}^m$	-6.87E+02	-6.11E+02	-6.51E+02	-6.48E+02	1.86E+01
$f_8{}^m$	-6.79E+02	-6.79E+02	-6.79E+02	-6.79E+02	5.09E-02
$f_9{}^m$	-5.50E+02	-5.35E+02	-5.42E+02	-5.42E+02	4.07E+00
$f_{10}{}^m$	-4.50E+02	-3.87E+02	-4.18E+02	-4.19E+02	1.47E+01
$f_{11}{}^m$	-3.95E+02	-3.89E+02	-3.92E+02	-3.92E+02	1.64E+00
$f_{12}{}^m$	5.18E+01	3.81E+02	2.07E+02	2.10E+02	6.94E+01
$f_{13}{}^m$	3.47E+02	5.99E+02	4.98E+02	4.92E+02	5.02E+01
$f_{14}{}^m$	-8.73E+01	-6.50E+01	-7.53E+01	-7.50E+01	5.53E+00
$f_{15}{}^m$	5.78E+03	9.26E+03	7.69E+03	7.68E+03	6.48E+02
$f_{16}{}^m$	2.01E+02	2.02E+02	2.02E+02	2.02E+02	3.94E-01
$f_{17}{}^m$	3.47E+02	3.83E+02	3.72E+02	3.71E+02	7.17E+00
$f_{18}{}^m$	1.01E+03	1.41E+03	1.24E+03	1.24E+03	1.05E+02
$f_{19}{}^m$	5.03E+02	5.06E+02	5.05E+02	5.05E+02	6.96E-01
$f_{20}{}^m$	6.23E+02	6.25E+02	6.25E+02	6.25E+02	4.18E-01
$f_{21}{}^c$	8.20E+02	9.40E+02	9.09E+02	9.04E+02	2.35E+01
$f_{22}{}^c$	8.34E+02	8.71E+02	8.49E+02	8.51E+02	8.74E+00
$f_{23}{}^c$	9.24E+03	1.29E+04	1.10E+04	1.09E+04	8.22E+02
$f_{24}{}^c$	1.32E+03	1.41E+03	1.37E+03	1.37E+03	1.38E+01
$f_{25}{}^c$	1.48E+03	1.55E+03	1.51E+03	1.51E+03	1.27E+01
$f_{26}{}^c$	1.40E+03	1.40E+03	1.40E+03	1.40E+03	7.10E-01
$f_{27}{}^c$	1.70E+03	3.46E+03	3.22E+03	3.12E+03	4.30E+02
$f_{28}{}^c$	1.80E+03	1.81E+03	1.81E+03	1.81E+03	2.83E+00

Tab. 10.5 Mean Results Comparison for D = 10

f_x	f_{min}	fk-PSO	GATHER	ABS-SPSO
$f_1{}^u$	-1400	**-1.40E+03**	**-1.40E+03**	**-1.40E+03**
$f_2{}^u$	-1300	**1.43E+05**	1.87E+06	1.48E+05
$f_3{}^u$	-1200	6.74E+05	1.26E+08	**1.27E+05**
$f_4{}^u$	-1100	**-6.84E+02**	1.49E+04	1.30E+03
$f_5{}^u$	-1000	-1.00E+03	-1.00E+03	-1.00E+03
$f_6{}^m$	-900	-8.97E+02	**-8.99E+02**	-8.95E+02
$f_7{}^m$	-800	-7.98E+02	-7.59E+02	**-8.00E+02**
$f_8{}^m$	-700	**-6.80E+02**	-6.79E+02	**-6.80E+02**
$f_9{}^m$	-600	**-5.97E+02**	-5.95E+02	-5.96E+02
$f_{10}{}^m$	-500	-4.99E+02	-4.94E+02	**-5.00E+02**
$f_{11}{}^m$	-400	**-4.00E+02**	**-4.00E+02**	**-4.00E+02**
$f_{12}{}^m$	-300	-2.93E+02	-2.78E+02	**-2.94E+02**
$f_{13}{}^m$	-200	-1.89E+02	-1.66E+02	**-1.94E+02**
$f_{14}{}^m$	-100	-6.22E+01	-9.92E+01	**-9.96E+01**
$f_{15}{}^m$	100	**5.54E+02**	8.29E+02	5.96E+02
$f_{16}{}^m$	200	**2.00E+02**	**2.00E+02**	**2.00E+02**
$f_{17}{}^m$	300	3.11E+02	**3.09E+02**	3.10E+02
$f_{18}{}^m$	400	**4.16E+02**	4.41E+02	4.17E+02
$f_{19}{}^m$	500	5.01E+02	**5.00E+02**	**5.00E+02**
$f_{20}{}^m$	600	6.03E+02	6.03E+02	**6.02E+02**
$f_{21}{}^c$	700	1.08E+03	**8.30E+02**	1.10E+03
$f_{22}{}^c$	800	9.22E+02	8.16E+02	**8.13E+02**
$f_{23}{}^c$	900	**1.42E+03**	1.92E+03	1.50E+03
$f_{24}{}^c$	1000	1.20E+03	**1.13E+03**	1.20E+03
$f_{25}{}^c$	1100	1.31E+03	**1.28E+03**	1.30E+03
$f_{26}{}^c$	1200	1.39E+03	**1.33E+03**	**1.33E+03**
$f_{27}{}^c$	1300	1.67E+03	1.70E+03	**1.65E+03**
$f_{28}{}^c$	1400	1.73E+03	**1.54E+03**	1.69E+03

Tab. 10.6 Mean Results Comparison for D = 30

f_x	f_{min}	fk-PSO	GATHER	ABS-SPSO
$f_1{}^u$	-1400	**-1.40E+03**	**-1.40E+03**	**-1.40E+03**
$f_2{}^u$	-1300	1.59E+06	1.48E+07	**8.77E+05**
$f_3{}^u$	-1200	2.40E+08	3.87E+09	**5.16E+07**
$f_4{}^u$	-1100	**-6.22E+02**	8.96E+04	4.92E+03
$f_5{}^u$	-1000	**-1.00E+03**	**-1.00E+03**	**-1.00E+03**
$f_6{}^m$	-900	-8.70E+02	-8.79E+02	**-8.89E+02**
$f_7{}^m$	-800	-7.36E+02	-6.79E+02	**-7.49E+02**
$f_8{}^m$	-700	**-6.79E+02**	**-6.79E+02**	**-6.79E+02**
$f_9{}^m$	-600	**-5.82E+02**	-5.70E+02	-5.71E+02
$f_{10}{}^m$	-500	**-5.00E+02**	-4.75E+02	**-5.00E+02**
$f_{11}{}^m$	-400	-3.76E+02	-3.96E+02	**-4.00E+02**
$f_{12}{}^m$	-300	**-2.44E+02**	-1.04E+02	-2.36E+02
$f_{13}{}^m$	-200	-7.70E+01	7.55E+01	**-8.53E+01**
$f_{14}{}^m$	-100	6.04E+02	**-8.80E+01**	-8.45E+01
$f_{15}{}^m$	100	**3.52E+03**	3.81E+03	3.65E+03
$f_{16}{}^m$	200	**2.01E+02**	**2.01E+02**	**2.01E+02**
$f_{17}{}^m$	300	3.53E+02	3.41E+02	**3.31E+02**
$f_{18}{}^m$	400	**4.68E+02**	7.31E+02	4.90E+02
$f_{19}{}^m$	500	5.03E+02	**5.02E+02**	**5.02E+02**
$f_{20}{}^m$	600	6.12E+02	6.15E+02	**6.11E+02**
$f_{21}{}^c$	700	1.01E+03	**8.83E+02**	1.02E+03
$f_{22}{}^c$	800	1.66E+03	**8.52E+02**	8.84E+02
$f_{23}{}^c$	900	**4.47E+03**	5.67E+03	5.08E+03
$f_{24}{}^c$	1000	**1.25E+03**	1.29E+03	**1.25E+03**
$f_{25}{}^c$	1100	**1.35E+03**	1.41E+03	1.38E+03
$f_{26}{}^c$	1200	1.50E+03	**1.40E+03**	1.46E+03
$f_{27}{}^c$	1300	2.08E+03	**1.99E+03**	2.21E+03
$f_{28}{}^c$	1400	1.80E+03	1.85E+03	**1.73E+03**

Tab. 10.7 Mean Results Comparison for D = 50

f_x	f_{min}	fk-PSO	GATHER	ABS-SPSO
$f_1{}^u$	-1400	**-1.40E+03**	**-1.40E+03**	**-1.40E+03**
$f_2{}^u$	-1300	2.76E+06	2.62E+07	**4.94E+05**
$f_3{}^u$	-1200	9.68E+08	1.67E+10	**1.21E+08**
$f_4{}^u$	-1100	**-5.75E+02**	1.58E+05	3.78E+03
$f_5{}^u$	-1000	**-1.00E+03**	**-1.00E+03**	**-1.00E+03**
$f_6{}^m$	-900	-8.45E+02	-8.54E+02	**-8.59E+02**
$f_7{}^m$	-800	-7.22E+02	-6.48E+02	**-7.27E+02**
$f_8{}^m$	-700	**-6.79E+02**	**-6.79E+02**	**-6.79E+02**
$f_9{}^m$	-600	**-5.62E+02**	-5.42E+02	-5.42E+02
$f_{10}{}^m$	-500	**-5.00E+02**	-4.19E+02	**-5.00E+02**
$f_{11}{}^m$	-400	-3.14E+02	-3.92E+02	**-4.00E+02**
$f_{12}{}^m$	-300	**-1.55E+02**	2.10E+02	-1.27E+02
$f_{13}{}^m$	-200	**7.40E+01**	4.92E+02	8.65E+01
$f_{14}{}^m$	-100	1.86E+03	**-7.50E+01**	-7.36E+01
$f_{15}{}^m$	100	**6.73E+03**	7.68E+03	7.52E+03
$f_{16}{}^m$	200	**2.01E+02**	2.02E+02	**2.01E+02**
$f_{17}{}^m$	300	4.16E+02	3.71E+02	**3.52E+02**
$f_{18}{}^m$	400	**5.32E+02**	1.24E+03	6.16E+02
$f_{19}{}^m$	500	5.08E+02	**5.05E+02**	**5.05E+02**
$f_{20}{}^m$	600	6.21E+02	6.25E+02	**6.20E+02**
$f_{21}{}^c$	700	1.53E+03	**9.04E+02**	1.60E+03
$f_{22}{}^c$	800	3.02E+03	**8.51E+02**	**8.51E+02**
$f_{23}{}^c$	900	**8.30E+03**	1.09E+04	9.94E+03
$f_{24}{}^c$	1000	**1.30E+03**	1.37E+03	1.31E+03
$f_{25}{}^c$	1100	**1.40E+03**	1.51E+03	1.46E+03
$f_{26}{}^c$	1200	1.59E+03	**1.40E+03**	1.60E+03
$f_{27}{}^c$	1300	**2.62E+03**	3.12E+03	2.93E+03
$f_{28}{}^c$	1400	**1.80E+03**	1.81E+03	2.25E+03

11 Conclusions and discussions

In this work there have been intensively studied various approaches for modification and performance enhancement of evolutionary computational techniques, mainly the Particle swam optimization algorithm. The PSO was chosen as the most prominent representative of Swarm intelligence based group of algorithms. As many SI algorithms share certain similarities it seems likely that approaches that manage to enhance the performance of PSO algorithm may also enhance the performance of other SI algorithms.

The main research direction dealt with the possibility of incorporation chaotic pseudo-random number generators instead of canonical PRNGs (that are typically used for ECTs). The general idea for chaotic PRNG is that for nature-inspired methods such as the ECTs it may be more convenient to use nature-based PRNGs. This research was motivated by previous successful experiments conducted by other researchers.

The second research direction focused on altering the inner principles of PSO in such way that the performance was improved. In this research direction several promising modifications were proposed and tested on common benchmark problems.

11.1 Chaos PSO summary

As has been mentioned in above sections the mutual interaction of CPRNGs and ECTs (even the PSO) has been studied by several researchers previously. However the studies concentrated mostly only on the performance of chaos enhanced ECTs and many questions remained unanswered. One the main goal of this work was to answer (at least partially) these questions.

Firstly it was investigated in detail the way of implementation of chaotic system as a CPRNG and the incorporation into PSO. Six chaotic maps in total were used during this research and based on the literature it was decided to use the CPRNG only for the velocity calculation formula. Several methods of CPRNG creation were proposed and tested. After this detailed study is was concluded that:

- Different ways of implementation of chaotic map as CPRNG have significant effect on the performance of chaos enhanced PSO algorithm.
- Using absolute value to transform negative numbers to positive seems like a promising method in comparison to common "shift" approach.

After the method for CPRNG creation was selected it was further tested the effect of different chaotic systems used as CPRNGs for PSO on the performance and the convergence behavior of the algorithm. The set of commonly used benchmark problems and CEC'13 Benchmark suite were used during this performance and behavior investigation. There have been many tests with various algorithm settings performed and the results and observations led to following conclusions:

- When Lozi chaotic map is employed as a CPRNG for PSO algorithm the convergence of the algorithm is typically very fast. This may improve the performance significantly on low-dimensional and unimodal or basic multi-modal problems. Also this may prove very helpful for strictly time-restricted optimization or real-time optimization. The fast convergence however typically leads to premature convergence into local extreme and therefore Lozi CPRNG enhanced PSO lacks performance on complex and high-dimensional optimization task.
- The performance and convergence behavior of PSO algorithm enhanced with CPRNG based on Tinkerbell map is very similar to the Chaos PSO enhanced by Lozi chaotic map based CPRNG. The avoidance of premature convergence into locals is slightly better and therefore the performance may differ in certain cases but overall the performance of these two versions is mostly comparable.

- The Sinai chaotic map does not seem to have a significant effect on the performance and behavior of PSO algorithm. The numerical results are mostly the same or very similar as are those of PSO with canonical PRNG. From the convergence lines it is clear that the convergence behavior is also almost the same.

- CPRNG based on Arnold's Cat map also seems to have only marginal effect. The performance is very comparable to canonical PSO and does not bring any significant advantage against other compared versions.

- Dissipative standard map as CPRNG significantly decreases the convergence speed of PSO algorithm. The slower convergence speed improves (to certain degree) the ability of PSO to avoid local extremes and therefore this particular version is very promising for very complex optimization tasks and large-scale optimization. The disadvantage is that the time demands of the algorithm are significantly higher. This method therefore is not feasible for simple real-time optimization.

- The last chaotic system that was tested as candidate CPRNG for PSO was the Burgers map. The PSO enhanced with Burgers map based CPRNG exhibits very promising overall performance. The initial convergence speed is very typically comparable with that of Dissipative map enhanced PSO but the behavior is very problem-dependant. That said the Burgers map based CPRNG enhanced PSO seems like a good overall initial optimizer for such tasks as black box optimization

After the analysis of different CPRNGs for PSO algorithm the research had to deal with a question: "Is the difference in performance and behavior given only by the different distributions of used CPRNGs? " To answer this question and uncover more about the relation of particular chaotic system and the inner dynamics of PSO an extensive tuning experiment was performed. The Lozi map based CPRNG was tuned extensively using the controlling parameters of the chaotic map. After detailed analysis and visualizations of the results it is possible to conclude the following points:

- It is possible to improve the performance of PSO algorithm driven by CPRNG on a particular optimization task by tuning the control parameters of the chaotic system.
- Tuning the chaotic system may worsen the performance of the PSO on other types of optimization problems.
- It is possible to tune the chaotic system in such way that the distribution of CPRNG is very different but the performance of PSO is very similar. Therefore it seems likely that the distribution of the CPRNG does not have overly prominent or dominant role.
- It seems very likely that the sequencing of numbers given by the inner mechanism of the chaotic system has at least similar role in affecting the behavior of PSO algorithm as the distribution of generated numbers by the CPRNG.

The theoretical research of Chaotic PSO was supplemented by an example of application. It was demonstrated that the differences between PSO algorithms driven by different CPRNGs are maintained even in real-world optimization tasks such as the PID controller design. It was also demonstrated that using chaos enhanced PSO it is possible to acquire better performing PID controller designs than by deterministic methods or canonical metaheuristics

As a main result of the research the innovative multi-chaotic approach for PSO (and ECTs. In general) was developed.

11.2 Multi-chaos PSO summary

The multi-chaotic approach for PSO was developed as a reaction to the disadvantages of particular CPRNGs. The main goal was to combine the advantages of multiple CPRNGs and achieve such behavior that cannot be otherwise achieved without extensive modification of the algorithm. This approach uses more than one chaotic system during the run of ECT and switches between these systems either manually or by adaptive approach. By the author's best knowledge such approach has never been proposed or tested and may be considered pioneering in the area of chaos enhanced metaheuristics.

In the initial experiment two systems with opposite convergence behavior were selected. The Lozi map and Dissipative map based CPRNGs were manually switched during the optimization. The goal was to achieve fast initial convergence (as exhibited by PSO with Lozi map based CPRNG) but avoid premature convergence into locals (typical for Dissipative map based CPRNG). The experiments and results may be concluded into these points:

- It is possible to manually switch two different CPRNGs during one run of the PSO algorithm and achieve better performance on various problems.
- This technique however requires good knowledge of the problem and its characteristics (modality, complexity etc.).
- The best switching point is different for each problem and typically requires either extensive tuning or setting by an expert.
- Overall the manual approach proved that it is possible to combine two CPRNGs and gain advantages of both but it is not usable for routine optimization of various tasks especially dynamic optimization.

Following the encouraging results of the above described manual approach for multi-chaotic PSO an adaptive method was proposed. For the adaptive switching of CPRNGs it was tracked the change of the best solution. In such way it is possible to detect premature convergence or stagnation and use the second CPRNG to continue the optimization process. Several combinations of CPRNGs were selected and tested. The research concluded into these findings:

- The combinations of Tinkerbell map and Burgers map based CPRNGs and Lozi map and Burgers map based CPRNGs seem most promising.

- It is possible to successfully use the adaptive approach over variety of different optimization problems without need of any further knowledge about the problems.

- The multi-chaotic PSO typically achieves better results than PSO driven either by single CPRNG or canonical PRNG.

- The basic PSO with multi-chaotic PRNG is able to compete with more complex state of art methods (based on PSO).

- It is also possible to outperform the state of art methods in some cases.

- The adaptive approach improves the overall performance of chaotic PSO across different optimization tasks and seems very promising for black-box optimization, large scale optimization and also real-time optimization.

After these successful results the multi-chaotic approach was embedded into another chaos enhanced metaheuristic the Differential Evolution. The initial results presented here hint that this approach may be beneficial for DE as well as PSO and in extension to other ECTs. During the experiments witch chaotic and multi-chaotic PSO the inner dynamics of the algorithm were studied in detail and as a reaction to uncovered inner dynamics and disadvantages several modifications were proposed with different purposes.

11.3 Summary of proposed modifications of PSO

The first proposed modification named the Multiple-choice strategy for PSO (MC-PSO) was intended to reduce or possibly remove of the notorious disadvantages of PSO – the premature convergence into locals. The goal was to allow particles different behavior patterns in such way that the whole swarm would never converge into single area. The experimental part provided data for these conclusions:

- The MC-PSO outperforms the original PSO on variety of different optimization problems.
- The MC-PSO significantly outperforms the original PSO in the task of large-scale optimization.
- The convergence speed of MC-PSO is significantly faster but the algorithm still manages to avoid premature convergence significantly longer than the original PSO.
- The MC-PSO is up to 40% faster than the original PSO.
- Based on these findings the MC-PSO seems very promising for the needs of fast and real-time optimization.

The second proposed modification that is presented here is the Gathering algorithm. It is and extensive modification of the PSO that alters some of the fundamental inner principles of PSO and its variants. The key points and conclusions from the testing are summarized here:

- It is possible to find high quality solution without the sharing of gBest position within the swarm.
- The "snowball effect" can be used as way to measure the quality of the solution and its neighborhood and also as a way of distribution of particles among several promising regions.
- The performance o Gathering algorithm is comparable or better than the performance of the best performing PSO based methods on the given benchmark.
- The increasing dimensionality of the problem does not significantly worsen the performance of the algorithm in comparison with original PSO.
- Te design is almost immune to premature convergence because of the "heuristic search" component in the form of mutation of stationary particles.

The Gathering algorithm hints that swarms may not need a central point of attraction to work effectively.

11.4 Recommendation for future works

Based on all published and presented results and analysis the author summarizes the following recommendation for future work:

- Using PRNGs based on chaotic systems is a very simple but effective way to improve the performance of ECTs. It can be used as a simple plug-in for any existing ECT and authors of ECTs should be encouraged to do so by this work.
- Very basic and simple ECTs can compete with more complex state of art method when proper CPNRG is implemented.
- It is possible to combine two CPRNGs to achieve demanded behavior of the algorithm. Combination of more than 2 CPRNGs is one of the open topics for future research.
- The exact influence and mutual interaction of CPRNGs and ECTs remains an open question. However strong support has been given here for the research of the sequencing of pseudo-random numbers for ECTs.
- Heterogeneous swarms seem to be a very promising direction for SI algorithms.
- SI algorithms may use various attraction mechanisms and successfully find high quality solutions.

11.5 The goals of the dissertation

All five main goals of the dissertation are addressed here:

1. **Evaluation of the current state of the research area: Evolutionary algorithms (EAs), non-deterministic pseudo-random number generators (PRNGs).** The stare of the research was investigated during the first part of the research. The most typically used PRNG for EAs from literature seemed to be the Mersenne Twister. It was afterwards utilized as a PRNG for the "canonical" versions of ECTs. The most dynamically developing group of EAs are the algorithms based on Swarm Intelligence (SI). Among them the PSO remains the most prominent. Another highly significant EA is the Differential Evolution (DE).

2. **Definition of the field of research - finding a suitable algorithm for implementation of alternative approaches and modifications.** After the literature review and given the author's previous experiences the PSO algorithm has been chosen as the most suitable algorithm for the research. The PSO is for a long time one the most popular and widely used ECTs and shares notable similarities with many other ECTs especially from the SI category.

3. **The proposal of modifications and alternative strategies. Finding alternative PRNGs and their implementation into EAs.** Six alternative PRNGs based on six different chaotic systems were proposed and implemented. The way of implementation was selected both experimentally and based on literature review. Furthermore a multi-chaotic PRNG was designed and implemented. Several modifications of PSO algorithm were proposed. The two most notable are the Multiple-choice strategy for PSO (MC-PSO) and the Gathering algorithm.

4. **Testing and benchmarking of proposed algorithms.** All proposed algorithms and modifications were extensively tested using typically used benchmark problems. As a benchmark it was used the IEEE CEC'13 Benchmark set. The performance of most promising methods was benchmarked against state of the art methods based on PSO.

5. **Evaluation of results, analysis and recommendations for future works.** The results were regularly published in journals and conference

proceedings alongside with analysis. The summary of the results and analysis is given in this work. The recommendations for future work were presented.

It may be stated that all main goals of the dissertation were successfully fulfilled.

REFERENCES

[1] Kennedy J., Eberhart R., Particle Swarm Optimization, Proceedings of IEEE International Conference on Neural Networks. IV., 1995, pp. 1942–1948.

[2] Shi Y.H., Eberhart R.C., A modified particle swarm optimizer, IEEE International Conference on Evolutionary Computation, Anchorage Alaska, 1998, pp. 69–73.

[3] Eberhart R., J. Kennedy, Swarm Intelligence, The Morgan Kaufmann Series in Artificial Intelligence, Morgan Kaufmann, 2001.

[4] Dorigo M., Ant Colony Optimization and Swarm Intelligence, Springer, 2006.

[5] Zelinka I., SOMA - self organizing migrating algorithm, in: B.V. Babu, G. Onwubolu (Eds.), New Optimization Techniques in Engineering, vol. 33, Springer-Verlag, ISBN: 3-540-20167X, 2004, (Chapter 7).

[6] Price K., Storn R. M., Lampinen J. A. 2005: Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series) Springer; 1 edition , 2005, ISBN: 3540209506.

[7] Belfiore, N.P.; Rudas, I.J., "Applications of computational intelligence to mechanical engineering," Computational Intelligence and Informatics (CINTI), 2014 IEEE 15th International Symposium on , vol., no., pp.351,368, 19-21 Nov. 2014

[8] Ahmed, Q.Z.; Ahmed, S.; Alouini, M.-S.; Aissa, S. "Minimizing the Symbol-Error-Rate for Amplify-and-Forward Relaying Systems Using Evolutionary Algorithms", Communications, IEEE Transactions on, On page(s): 390 - 400 Volume: 63, Issue: 2, Feb. 2015

[9] Javani, M.; Faez, K.; Aghlmandi, D. "Clustering and feature selection via PSO algorithm", Artificial Intelligence and Signal Processing (AISP), 2011 International Symposium on, On page(s): 71 - 76

[10] Senkerik R., Davendra D., Zelinka I. Oplatkova Z., Pluhacek M., Optimization of the Batch Reactor by Means of Chaos Driven Differential Evolution, In Proceedings of SOCO 2012: Soft Computing Models in Industrial and Environmental Applications, Springer Series: "Advances in Intelligent Systems and Computing", Vol. 188, 2012, pp. 93 – 102, ISBN: 978-3-642-32922-7.

[11] Davendra D., Zelinka I., Senkerik R., Chaos driven evolutionary algorithms for the task of PID control, Computers & Mathematics with Applications, Volume 60, Issue 4, 2010, pp 1088-1104, ISSN 0898-1221

[12] Ho, Y.-C.; Pepyne, D.L., "Simple explanation of the no free lunch theorem of optimization," Decision and Control, 2001. Proceedings of the 40th IEEE Conference on , vol.5, no., pp.4409,4414 vol.5, 2001

[13] Sprott J. C., Chaos and Time-Series Analysis, Oxford University Press, 2003.

[14] Caponetto R., Fortuna L., Fazzino S., Xibilia M.G., Chaotic sequences to improve the performance of evolutionary algorithms, Evolutionary Computation, IEEE Transactions on , vol.7, no.3, pp. 289- 304, June 2003

[15] B. Alatas, E. Akin, B. A. Ozer, Chaos embedded particle swarm optimization algorithms, Chaos, Solitons & Fractals, Volume 40, Issue 4, 30 May 2009, Pages 1715-1734, ISSN 0960-0779.

[16] Pluhacek M., Senkerik R., Zelinka I., IMPACT OF VARIOUS CHAOTIC MAPS ON THE PERFORMANCE OF CHAOS ENHANCED PSO ALGORITHM WITH INERTIA WEIGHT – AN INITIAL STUDY, Advances in Intelligent Systems and Computing, 1, Volume 192, Nostradamus: Modern Methods of Prediction, Modeling and Analysis of Nonlinear Systems, pp. 153-166, 2012, ISBN 978-3-642-33226-5.

[17] Pluhacek M., Senkerik R., Davendra D., Kominkova Oplatkova Z., Zelinka I.„ On the behavior and performance of chaos driven PSO algorithm with inertia weight, Computers and Mathematics with Applications, 2013, doi: 10.1016/j.camwa.2013.01.016, ISSN 0898-1221

[18] Pluhacek M., Senkerik R. and Zelinka I., Particle swarm optimization algorithm driven by multichaotic number generator. Soft Computing, 2014/04/01 2014, 18(4), 631-639.

[19] Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. Evolutionary Computation, IEEE Transactions on 10(3), 281-295 (2006).

[20] Zhi-Hui, Z., Jun, Z., Yun, L., Yu-hui, S.: Orthogonal Learning Particle Swarm Optimization. IEEE Transactions on Evolutionary Computation 15(6), 832-847 (2011)

[21] Mallipeddi R., Suganthan P. N., "Differential Evolution Algorithm with Ensemble of Parameters and Mutation and Crossover Strategies", Swarm Evolutionary and Memetic Computing Conference, Chennai, India 2010, pp. 71-78, Vol. 6466.

[22] Mallipeddi R., Suganthan P. N., "Differential Evolution Algorithm with Ensemble of populations for Global Numerical optimization", OPSEARCH, vol. 46, no. 2, pp. 184-213, June 2009, Springer.

[23] Yang, X.S. Swarm intelligence based algorithms: a critical analysis. Evolutionary Intelligence, 2014/04/01 2014, 7(1), 17-28.

[24] Karaboga, D. An idea based on honey bee swarm for numerical optimization, Technical Report TR06, Erciyes University, Turkey, 2005

[25] Yang, X. S., Firefly algorithm, stochastic test functions and design optimisation, Int.J. Bio-inspired Computation, 2(2), 78–84,2010.

[26] Yang, X. S. and Deb, S., Cuckoo search via L´evy flights, Proceeings of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), IEEE Publications, USA, pp. 210–214, 2009.

[27] Yang, X. S., (2010). A New Metaheuristic Bat-Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization (Eds. Cruz, C.;

Gonz´alez, J. R.; Pelta, D. A.; Terrazas, G), Studies in Computational Intelligence Vol. 284, Springer Berlin, pp. 65–74, 2010.

[28] Yang, X. S., Gandomi, A. H., Bat algorithm: a novel approach for global engineering optimization, Engineering Computations, 29(5), 1–18, 2012.

[29] Engelbrecht A., "Particle Swarm Optimization: Where does It Belong?" in Proceedings of the IEEE Swarm Intelligence Symposium, 2006.

[30] Engelbrecht A., "Particle Swarm Optimization: Global Best or Local Best?" in submitted to BRICS-CCI, 2013.

[31] Engelbrecht A., "Particle Swarm Optimization: Iteration Strategies Revisited," in Proceedings of the BRICS Conference on omputational Intelligence, 2013.

[32] Engelbrecht A., "Particle Swarm Optimization: Velocity Initialization," in Proceedings of the IEEE Congress on Evolutionary Computation, 2012.

[33] Nickabadi A., Ebadzadeh M. M., Safabakhsh R., A novel particle swarm optimization algorithm with adaptive inertia weight, Applied Soft Computing, Volume 11, Issue 4, June 2011, Pages 3658-3670, ISSN 1568-4946.

[34] Eberhart R. and Shi Y., "Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization," in Proceedings of the IEEE Congress on Evolutionary Computation, vol. 1, July 2000, pp. 84–88.

[35] van den Bergh, F., Engelbrecht, A.P.: A study of particle swarm optimization particle trajectories. Information Sciences 176(8), 937-971 (2006).

[36] Liang J.J., Qu B-Y., Suganthan P.N., Hernández-Díaz A.G. (2013) Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session and Competition on Real-Parameter Optimization , Technical Report 201212, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore.

[37] Zelinka, I.; Senkerik, R.; Pluhacek, M., "Do evolutionary algorithms indeed require randomness?," Evolutionary Computation (CEC), 2013 IEEE

Congress on , vol., no., pp.2283,2289,2 0-23, June 2013, doi: 10.1109/CEC.2 013.6557841

[38] Astrom, K., "Control System Design". Santa Barbra, California: University of California, 2002

[39] Landau, Y., "Digital Control Systems". Springer, London, 2006

[40] Nagraj, B.,Subha, S.,Rampriya, B., "Tuning algorithms for pid controller using soft computing techniques", International Journal of Computer Science and Network Security, 8, 2008, pp.278-281.3

[41] Senkerik R, Pluhacek M, Zelinka I, Oplatkova Z, Vala R, Jasek R (2014) Performance of Chaos Driven Differential Evolution on Shifted Benchmark Functions Set. In: Herrero Á, Baruque B, Klett F et al. (eds) International Joint Conference SOCO'13-CISIS'13-ICEUTE'13, vol 239. Advances in Intelligent Systems and Computing. Springer Interna-tional Publishing, pp 41-50.

[42] Senkerik R, Davendra D, Zelinka I, Pluhacek M, Kominkova Oplatkova Z On the Diffe-rential Evolution Drivan by Selected Discrete Chaotic Systems: Extended Study. In: 19th International Conference on Soft Computing, MENDEL 2013, 2013. pp 137-144.

[43] Nepomuceno F. and Engelbrecht A. , "A self-adaptive heterogeneous pso for real-parameter optimization," in Proceedings of the IEEE International Conference on Evolutionary Computation, 2013.

[44] El-Abd M.. Testing a Particle Swarm Optimization and Artifi cial Bee Colony Hybrid algorithm on the CEC13 benchmarks. In IEEE Congress on Evolutionary Computation , pages 2215–2220, 2013.

PUBLICATIONS OF THE AUTHOR

[P.1] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan. Particle swarm optimization algorithm driven by multichaotic number generator. Soft computing, 2014, roč. 18, č. 4, s. 631-639. ISSN 1432-7643.

[P.2] Pluhacek, Michal, Senkerik, Roman, Davendra, Donald, Kominkova Oplatkova, Zuzana, Zelinka, Ivan. On the behavior and performance of chaos driven PSO algorithm with inertia weight. Computers & Mathematics with Applications, 2013, roč. 66, č. 2, s. 122-134. ISSN 0898-1221.

[P.3] Senkerik, Roman, Kominkova Oplatkova, Zuzana, Zelinka, Ivan, Chramcov, Bronislav, Davendra, Donald, Pluhacek, Michal. Utilization of analytic programming for the evolutionary synthesis of the robust multi-chaotic controller for selected sets of discrete chaotic systems. Soft computing, 2014, roč. 18, č. 4, s. 651-668. ISSN 1432-7643.

[P.4] Kominkova Oplatkova, Zuzana, Senkerik, Roman, Zelinka, Ivan, Pluhacek, Michal. Analytic programming in the task of evolutionary synthesis of a controller for high order oscillations stabilization of discrete chaotic systems. Computers & Mathematics with Applications, 2013, roč. 66, č. 2, s. 177-189. ISSN 0898-1221.

[P.5] Davendra, Donald, Senkerik, Roman, Zelinka, Ivan, Pluhacek, Michal, Bialic-Davendra, Magdalena Lucyna. Utilising the chaos-induced discrete self organising migrating algorithm to solve the lot-streaming flowshop scheduling problem with setup time. Soft computing, 2014, roč. 18, č. 4, s. 669-681. ISSN 1432-7643.

[P.6] Senkerik, Roman, Zelinka, Ivan, Pluhacek, Michal, Davendra, Donald, Kominkova Oplatkova, Zuzana. Chaos Enhanced Differential Evolution in the Task of Evolutionary Control of Selected Set of Discrete Chaotic Systems. The Scintific World Journal, 2014, roč. 2014, č. 836484, s. 1-13. ISSN 2356-6140.

[P.7] Zelinka, Ivan, Davendra, Donald, Senkerik, Roman. Pluhacek, Michal, Investigation on evolutionary predictive control of chemical reactor.

Journal of Applied Logic, 2015, 13(2, Part A), s. 156-166, ISSN: 1570-8683

[P.8] Pluhacek, Michal, Budikova, Vera, Senkerik, Roman, Oplatkova, Zuzana, Zelinka, Ivan. ON THE PERFORMANCE OF ENHANCED PSO ALGORITHM WITH LOZI CHAOTIC MAP – AN INITIAL STUDY. In 18th International Conference on Soft Computing. Brno : VUT Brno, 2012, s. 40-45. ISSN 1803-3814. ISBN 978-80-214-4540-6.

[P.9] Pluhacek, Michal, Senkerik, Roman, Davendra, Donald, Zelinka, Ivan. PID CONTROLLER DESIGN FOR 4TH ORDER SYSTEM BY MEANS OF ENHANCED PSO ALGORITHM WITH LOZI CHAOTIC MAP. In 18th International Conference on Soft Computing. Brno : VUT Brno, 2012, s. 35-39. ISSN 1803-3814. ISBN 978-80-214-4540-6.

[P.10] Pluhacek, Michal, Senkerik, Roman, Davendra, Donald, Zelinka, Ivan. DESIGNING PID CONTROLLER FOR DC MOTOR SYSTEM BY MEANS OF ENHANCED PSO ALGORITHM WITH DISCRETE CHAOTIC LOZI MAP. In 26th European Conference on Modelling and Simulation. Nicosia : ECMS, 2012, s. 405-409. ISBN 978-0-9564944-4-3.

[P.11] Pluhacek, Michal, Senkerik, Roman, Davendra, Donald, Zelinka, Ivan. DESIGNING PID CONTROLLER FOR 4TH ORDER SYSTEM BY MEANS OF ENHANCED PSO ALGORITHM WITH DISCRETE CHAOTIC DISSIPATIVE STANDARD MAP. In The 9th International Multidisciplinary Modeling & Simulation Multiconference. Vienna : IEEE, 2012, s. 396-400. ISBN 978-88-97999-09-6.

[P.12] Pluhacek, Michal, Senkerik, Roman, Davendra, Donald, Zelinka, Ivan. Designing PID controller for DC motor by means of enhanced PSO algorithm with dissipative chaotic map. In Soft Computing Models in Industrial and Environmental Applications. Heidelberg : Springer-Verlag Berlin, 2013, s. 475-483. ISSN 2194-5357. ISBN 978-3-642-32921-0.

[P.13] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan. Impact of Various Chaotic Maps on the Performance of Chaos Enhanced PSO Algorithm with Inertia Weight – an Initial Study. In Nostradamus: Modern Methods of Prediction, Modeling and Analysis of Nonlinear Systems. Heidelberg : Springer-Verlag Berlin, 2013, s. 153-166. ISSN 2194-5357. ISBN 978-3-642-33226-5.

[P.14] Pluhacek, Michal, Budikova, Věra, Senkerik, Roman, Kominkova Oplatkova, Zuzana, Zelinka, Ivan. Extended Initial Study on the Performance of Enhanced PSO Algorithm with Lozi Chaotic Map. In Nostradamus: Modern Methods of Prediction, Modeling and Analysis of Nonlinear Systems. Heidelberg : Springer-Verlag Berlin, 2013, s. 167-177. ISSN 2194-5357. ISBN 978-3-642-33226-5.

[P.15] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, Davendra, Donald. Gathering algorithm: A new concept of PSO based metaheuristic with dimensional mutation. In 2014 IEEE Symposium on Swarm Intelligence Proceedings. New Jersey, Piscataway : IEEE, 2014, s. 42-47. ISBN 978-1-4799-4459-0.

[P.16] Davendra, Donald, Zelinka, Ivan, Metlicka, Magdalena, Šenkeřík, Roman, Pluháček, Michal, Complex Network Analysis of Differential Evolution Algorithm applied to Flowshop with No-Wait Problem. In 2014 IEEE Symposium on Differential Evolution. New Jersey, Piscataway : IEEE, 2014, s. 1-8. ISBN 978-1-4799-4462-0.

[P.17] Senkerík, Roman, Pluhacek, Michal, Kominkova Oplatkova, Zuzana, Davendra, Donald, On the parameter settings for the chaotic dynamics embedded differential evolution, in Evolutionary Computation (CEC), 2015 IEEE Congress on , vol., no., pp.1410-1417, 2015

[P.18] Pluhacek, Michal, Senkerik, Roman, Davendra, Donald, Multiple choice strategy with dimensional mutation for PSO algorithm enhanced with chaotic dissipative standard map, in Evolutionary Computation (CEC), 2015 IEEE Congress on , vol., no., pp.1404-1409, 2015

[P.19] Pluhacek, Michal, Senkerik, Roman. COMPARING PERFORMANCE OF PSO ALGORITHM WITH AND WITHOUT INERTIA WEIGHT

ON UNIMODAL AND MULTIMODAL FUNCTION. In Conference Proceedings XIV. Annual Internetional Conference, Internet, Competitiveness and Organizational Security. Zlín : Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2012, s. 122.125. ISBN 978-80-7454-142-1.

[P.20] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan. An Overview on the Chaotic Lozi Map Enhanced PSO Algorithm with Inertia Weight. Trilobit, 2012, elektronický časopis.

[P.21] Senkerik, Roman, Davendra, Donald, Zelinka, Ivan, Pluhacek, Michal, Oplatkova, Zuzana. An Investigation on the Chaos Driven Differential Evolution – An Initial Study. In Proceedings of the Fifth International Conference on Bioinspired Optimization Methods and their Applications. Ljubljana : Jožef Stefan Institute, 2012, s. 185-194. ISBN 978-961-264-043-9.

[P.22] Senkerik, Roman, Davendra, Donald, Zelinka, Ivan, Pluhacek, Michal, Oplatkova, Zuzana. An Investigation on the Differential Evolution Driven by Selected Discrete Chaotic Systems. In 18th International Conference on Soft Computing. Brno : VUT Brno, 2012, s. 157-162. ISSN 1803-3814. ISBN 978-80-214-4540-6.

[P.23] Senkerik, Roman, Davendra, Donald, Zelinka, Ivan, Kominkova Oplatkova, Zuzana, Pluhacek, Michal. Optimization of the Batch Reactor by means of Chaos Driven Differential Evolution. In Soft Computing Models in Industrial and Environmental Applications. Heidelberg : Springer-Verlag Berlin, 2013, s. 93-102. ISSN 2194-5357. ISBN 978-3-642-32921-0.

[P.24] Senkerik, Roman, Oplatkova, Zuzana, Zelinka, Ivan, Davendra, Donald, Pluhacek, Michal. Utilization of Analytic Programming for the Stabilization of High Order Oscillations of Chaotic Logistic Equation. In The 9th International Multidisciplinary Modeling & Simulation Multiconference. Vienna : IEEE, 2012, s. 458-464. ISBN 978-88-97999-09-6.

[P.25] Zelinka, Ivan, CHADLI, Mohammed, Davendra, Donald, Senkerik, Roman, Pluhacek, Michal, LAMPINEN, Jouni. Hidden Periodicity - Chaos Dependance on Numerical Precision. In Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems. Heidelberg : Springer-Verlag Berlin, 2013, s. 47-59. ISSN 2194-5357. ISBN 978-3-319-00541-6.

[P.26] Zelinka, Ivan, CHADLI, Mohammed, Davendra, Donald, Senkerik, Roman, Pluhacek, Michal, LAMPINEN, Jouni. Do Evolutionary Algorithms Indeed Require Random Numbers? Extended Study. In Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems. Heidelberg : Springer-Verlag Berlin, 2013, s. 61-75. ISSN 2194-5357. ISBN 978-3-319-00541-6.

[P.27] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, Davendra, Donald. New Adaptive Approach for Chaos PSO Algorithm Driven Alternately by Two Different Chaotic Maps - An Initial Study. In Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems. Heidelberg : Springer-Verlag Berlin, 2013, s. 77-87. ISSN 2194-5357. ISBN 978-3-319-00541-6.

[P.28] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, Davendra, Donald. MC-PSO/DE Hybrid with Repulsive Strategy – Initial Study, In: Hybrid Artificial Intelligent Systems, Volume 9121 of the series Lecture Notes in Computer Science pp 213-220, 2015

[P.29] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, Davendra, Multiple Choice Strategy for PSO Algorithm Enhanced with Dimensional Mutation, In: Artificial Intelligence and Soft Computing Volume 9119 of the series Lecture Notes in Computer Science pp 370-378, 2015

[P.30] Davendra, Donald, Senkerik, Roman, Zelinka, Ivan, Pluhacek, Michal, BIALIC-Davendra, Magdalena Lucyna. Utilising the Chaos-Induced Discrete Self Organising Migrating Algorithm to Schedule the Lot-Streaming Flowshop Scheduling Problem with Setup Time. In Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems. Heidelberg : Springer-Verlag Berlin, 2013, s. 31-45. ISSN 2194-5357. ISBN 978-3-319-00541-6.

[P.31] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, The Initial Study on the Potential of Super-Sized Swarm in PSO, In: Mendel 2015 Volume 378 of the series Advances in Intelligent Systems and Computing pp 127-135, 2015

[P.32] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, Davendra, Donald. On the Performance of Enhanced PSO Algorithm with Dissipative Chaotic Map in the Task of High Dimensional Optimization Problems. In Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems. Heidelberg : Springer-Verlag Berlin, 2013, s. 89-99. ISSN 2194-5357. ISBN 978-3-319-00541-6.

[P.33] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, Chaos Driven Particle Swarm Optimization with Basic Particle Performance Evaluation – An Initial Study. In Proceedings of the 13th IFIP TC8 International Conference. Heidelberg : Springer-Verlag Berlin, 2014, s. 445-454. ISSN 0302-9743. ISBN 978-3-662-45236-3.

[P.34] Senkerik, Roman, Pluhacek, Michal, Zelinka, Ivan, Davendra, Donald, Kominkova Oplatkova, Zuzana, Janostik, Jakub, Hybridization of Adaptivity and Chaotic Dynamics for Differential Evolution, In: Mendel 2015 Volume 378 of the series Advances in Intelligent Systems and Computing pp 149-158, 2015

[P.35] Senkerik, Roman, Kominkova Oplatkova, Zuzana, Zelinka, Ivan, Pluhacek, Michal. Utilization of Analytic Programming for Evolutionary Synthesis of the Robust Controller for Set of Chaotic Systems. In Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems. Heidelberg : Springer-Verlag Berlin, 2013, s. 101-110. ISSN 2194-5357. ISBN 978-3-319-00541-6.

[P.36] Senkerik, Roman, Pluhacek, Michal, Zelinka, Ivan, Davendra, Donald; Janostik, Jakub, New Adaptive Approach for Multi-chaotic Differential

Evolution Concept, In: Hybrid Artificial Intelligent Systems, Volume 9121 of the series Lecture Notes in Computer Science pp 234-243, 2015

[P.37] Senkerik, Roman, Pluhacek, Michal, Davendra, Donald, Zelinka, Ivan, Kominkova Oplatkova, Zuzana. Chaos driven evolutionary algorithm: A new approach for evolutionary optimization. International Journal of Mathematics and Computers in Simulations, 2013, roč. 7, č. 4, s. 363-368. ISSN 1998-0159.

[P.38] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan. Investigation on the Performance of a New Multiple Choice Strategy for PSO Algorithm in the task of Large Scale Optimization Problems. In 2013 IEEE Congress on Evolutionary Computation, CEC 2013. Boston : IEEE, 2013, s. 2007-2011. ISBN 978-1-4799-0454-9.

[P.39] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, Davendra, Donald. Chaos PSO Algorithm Driven Alternately by two Different Chaotic Maps - an Initial Study. In 2013 IEEE Congress on Evolutionary Computation, CEC 2013. Boston : IEEE, 2013, s. 2444-2449. ISBN 978-1-4799-0454-9.

[P.40] Senkerik, Roman, Pluhacek, Michal, Kominkova Oplatkova, Zuzana, Davendra, Donald, Zelinka, Ivan. Investigation on the Differential Evolution Driven by Selected Six Chaotic Systems in the Task of Reactor Geometry Optimization. In 2013 IEEE Congress on Evolutionary Computation, CEC 2013. Boston : IEEE, 2013, s. 3087-3094. ISBN 978-1-4799-0454-9.

[P.41] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan. Multiple Choice Strategy - A Novel Approach for Particle Swarm Optimization - Preliminary Study. In Artificial Intelligence and Soft Computing Part II. Heidelberg : Springer-Verlag Berlin, 2013, s. 36-45. ISSN 0302-9743. ISBN 978-3-642-38609-1.

[P.42] Senkerik, Roman, Davendra, Donald, Zelinka, Ivan, Pluhacek, Michal, Kominkova Oplatkova, Zuzana. Chaos Driven Differential Evolution with Lozi Map in the Task of Chemical Reactor Optimization. In Artificial Intelligence and Soft Computing Part II. Heidelberg : Springer-Verlag Berlin, 2013, s. 56-66. ISSN 0302-9743. ISBN 978-3-642-38609-1.

[P.43] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, Davendra, Donald. Designing PID Controllers by Means of PSO Algorithm Enhanced by Various Chaotic Maps. In EUROSIM 2013. Cardiff : The Institute of Electrical and Electronics Engineers, 2013, s. 19-23. ISBN 978-0-7695-5073-2.

[P.44] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan. Multiple Choice Strategy Based PSO Algorithm with Chaotic Decision Making – A Preliminary Study. In International Joint Conference SOCO´13 - CISIS´13 - ICEUTE´13. Heidelberg : Springer-Verlag Berlin, 2014, s. 21-30. ISSN 2194-5357. ISBN 978-3-319-01853-9.

[P.45] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, DONALD, Davendra. Multiple Choice Strategy For PSO Algorithm - Performance Analysis On Shifted Test Functions. In 27th European Conference on Modelling and Simulation. Alesund : ECMS, 2013, s. 393-397. ISBN 978-0-9564944-6-7.

[P.46] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, DONALD, Davendra. EXTENSIVE STUDY ON THE PERFORMANCE OF PSO ALGORITHM ENHANCED WITH BURGERS CHAOTIC MAP. In MENDEL 2013 19th International Conference on Soft Computing. Brno : Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2013, s. 161-166. ISSN 1803-3814. ISBN 978-80-214-4755-4.

[P.47] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan. DESIGNING PID CONTROLLER FOR DC MOTOR BY MEANS OF CHAOS ENHANCED PSO ALGORITHM. In Proceeding of Seventh Global Conference of Power Control and Optimization. Melville : American Institute of Physics, 2013, s. nestrankovano. ISBN 9789834448363.

[P.48] Pluhacek, Michal, Svejda, Jaromír, Talandova, Hana, Zak, Roman, JAŠEK, Roman. Artificial intelligence in biom etrical identification systems. In Bezpečnostní technologie, systémy a management 2013. Zlín : Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2013, s. nestrankovano. ISBN 978-80-7454-289-3.

[P.49] Davendra, Donald, BIALIC-Davendra, Magdalena Lucyna, Senkerik, Roman, Pluhacek, Michal. Scheduling the Flow Shop with Blocking Problem with the Chaos-Induced Discrete Self Organising Migration Algorithm. In 27th European Conference on Modelling and Simulation. Alesund : ECMS, 2013, s. 386-392. ISBN 978-0-9564944-6-7.

[P.50] Senkerik, Roman, Kominkova Oplatkova, Zuzana, Zelinka, Ivan, Pluhacek, Michal. Analytic Programming In The Task Of Evolutionary Synthesis Of The Robust Controller For Selected Discrete Chaotic Systems. In 27th European Conference on Modelling and Simulation. Alesund : ECMS, 2013, s. 398-404. ISBN 978-0-9564944-6-7.

[P.51] Zelinka, Ivan, Senkerik, Roman, Pluhacek, Michal. Do Evolutionary Algorithms Indeed Require Randomness?. In 2013 IEEE Congress on Evolutionary Computation, CEC 2013. Boston : IEEE, 2013, s. 2283-2289. ISBN 978-1-4799-0454-9.

[P.52] Senkerik, Roman, Davendra, Donald, Zelinka, Ivan, Pluhacek, Michal, Kominkova Oplatkova, Zuzana. On The Differential Evolution Driven By Selected Discrete Chaotic Systems: Extended Study. In MENDEL 2013 19th International Conference on Soft Computing. Brno : Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2013, s. 137-144. ISSN 1803-3814. ISBN 978-80-214-4755-4.

[P.53] Zelinka, Ivan, Senkerik, Roman, Pluhacek, Michal. Evolutionary Algorithms Powered by Nonrandom Processes. In MENDEL 2013 19th International Conference on Soft Computing. Brno : Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2013, s. 145-152. ISSN 1803-3814. ISBN 978-80-214-4755-4.

[P.54] Senkerik, Roman, Kominkova Oplatkova, Zuzana, Pluhacek, Michal. Analytic Programming – a Novel Tool for Synthesis of Controller for

Chaotic Lozi Map. In Recent Advances in Systems, Control, Signal Processing and Informatics. Craiova : Europment, 2013, s. 201-206. ISSN 1790-5117. ISBN 978-1-61804-204-0.

[P.55] Senkerik, Roman, Pluhacek, Michal, Kominkova Oplatkova, Zuzana. Chaos Driven Evolutionary Algorithm: a Novel Approach for Optimization. In Recent Advances in Systems, Control, Signal Processing and Informatics. Craiova : Europment, 2013, s. 222-226. ISSN 1790-5117. ISBN 978-1-61804-204-0.

[P.56] Senkerik, Roman, Pluhacek, Michal, Davendra, Donald, Zelinka, Ivan, Kominkova Oplatkova, Zuzana. Chaos Driven Evolutionary Algorithm: a New Approach for Evolutionary Optimization. In Recent Advances in Systems, Control and Informatics. Craiova : Europment, 2013, s. 117-122. ISSN 1790-5117. ISBN 978-1-61804-206-4.

[P.57] Senkerik, Roman, Pluhacek, Michal, Zelinka, Ivan, Davendra, Donald, Kominkova Oplatkova, Zuzana, JAŠEK, Roman. Chaos Driven Differential Evolution In The Task Of Evolutionary Control Of Delayed Logistic Chaotic System. In Proceeding of Seventh Global Conference of Power Control and Optimization. Melville : American Institute of Physics, 2013, s. nestrankovano. ISBN 9789834448363.

[P.58] Senkerik, Roman, Pluhacek, Michal, Zelinka, Ivan, Davendra, Donald, Kominkova Oplatkova, Zuzana, JAŠEK, Roman. Evolutionary Control of Chaotic Lozi Map by Means of Chaos Driven Differential Evolution. In AETA 2013: Recent Advances in Electrical Engineering and Related Sciences. Heidelberg : Springer-Verlag Berlin, 2014, s. 371-380. ISSN 1876-1100. ISBN 978-3-642-41967-6.

[P.59] Senkerik, Roman, Pluhacek, Michal, Zelinka, Ivan, Kominkova Oplatkova, Zuzana, VALA, Radek, JAŠEK, Roman. Performance of Chaos Driven Differential Evolution on Shifted Benchmark Functions Set. In International Joint Conference SOCO´13 - CISIS´13 - ICEUTE´13. Heidelberg : Springer-Verlag Berlin, 2014, s. 41-50. ISSN 2194-5357. ISBN 978-3-319-01853-9.

[P.60] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, Davendra, Donald. Performance comparison of evolutionary techniques enhanced by Lozi chaotic map in the task of reactor geometry optimization. In 28th European Conference on Modelling and Simulation. Nottingham : ECMS, 2014, s. 371-376. ISBN 978-0-9564944-8-1.

[P.61] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan. Preliminary Study on the Particle Swarm Optimization with the Particle Performance Evaluation. In Artificial Intelligence and Soft Computing. Heidelberg : Springer-Verlag Berlin, 2014, s. 395-405. ISSN 0302-9743. ISBN 978-3-319-07172-5.

[P.62] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, Davendra, Donald. Chaos Driven PSO with Ensemble of Priority Factors. In Advances in Intelligent Systems and Computing. 289. Heidelberg : Springer-Verlag Berlin, 2014, s. 89-97. ISSN 2194-5357. ISBN 978-3-319-07400-9.

[P.63] Lapkova, Dora, Pluhacek, Michal, Kominkova Oplatkova, Zuzana, ADÁMEK, Milan. USING ARTIFICIAL NEURAL NETWORK FOR THE KICK TECHNIQUES CLASSIFICATION – AN INITIAL STUDY. In 28th European Conference on Modelling and Simulation. Nottingham : ECMS, 2014, s. 382-386. ISBN 978-0-9564944-8-1.

[P.64] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan. Tuning the Lozi Map in Chaos Driven PSO Inspired by the Multi-chaotic Approach. In Advances in Intelligent Systems and Computing. 289. Heidelberg : Springer-Verlag Berlin, 2014, s. 79-88. ISSN 2194-5357. ISBN 978-3-319-07400-9.

[P.65] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, Davendra, Donald. Investigation on the Dynamics of PSO Algorithm Enhanced with Chaotic Lozi Map. In Emergence, Complexity and Computation ECC. Heidelberg : Springer-Verlag Berlin, 2014, s. 357-367. ISSN 2194-7287. ISBN 978-3-642-45437-0.

[P.66] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan, Davendra, Donald.

Chaos Driven PSO – On the Influence of Various CPRNG Implementations – An Initial Study. In ISCS 2014: Interdisciplinary Symposium on complex Systems. Heidelberg : Springer-Verlag Berlin, 2014, s. 225-237. ISSN 2194-7287. ISBN 978-3-319-10758-5.

[P.67] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan. MULTI-SWARM APROACH FOR CHAOTIC PSO ALGORITHM – AN INITIAL STUDY. In MENDEL 2014 20 th International conference on soft Computing. Brno : Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014, s. 141-146. ISSN 1803-3814. ISBN 978-80-214-4984-8.

[P.68] Lapkova, Dora, Pluhacek, Michal, Kominkova Oplatkova, Zuzana, Senkerik, Roman, ADÁMEK, Milan. Application of Neural Networks for the Classification of Gender from Kick Force Profile – A Small Scale Study. In Proceedings of the Fifth International Cnference on Innovations in Bio-Inspired Computing and Applications. Heidelberg : Springer-Verlag Berlin, 2014, s. 429-438. ISSN 2194-5357. ISBN 978-3-319-08155-7.

[P.69] Lapkova, Dora, Pluhacek, Michal, ADÁMEK, Milan. Computer Aided Analysis of Direct Punch Force Using the Tensometric Sensor. In Advances in Intelligent Systems and Computing. 285. Heidelberg : Springer-Verlag Berlin, 2014, s. 507-514. ISSN 2194-5357. ISBN 978-3-319-06739-1.

[P.70] Senkerik, Roman, Zelinka, Ivan, Pluhacek, Michal, Davendra, Donald, Kominkova Oplatkova, Zuzana. Chaos Enhanced Differential Evolution in the Task of Evolutionary Control of Selected Set of Discrete Chaotic Systems. The Scintific World Journal, 2014, roč. 2014, č. 836484, s. 1-13. ISSN 2356-6140.

[P.71] Senkerik, Roman, Zelinka, Ivan, Pluhacek, Michal, Kominkova Oplatkova, Zuzana. Evolutionary Control of Chaotic Burgers Map by means of Chaos Enhanced Differential Evolution. International Journal of Mathematics and Computers in Simulations, 2014, roč. 2014, č. 8, s. 39-45. ISSN 1998-0159.

[P.72] Zelinka, Ivan, Davendra, Donald, LAMPINEN, Jouni, Senkerik, Roman, Pluhacek, Michal. Evolutionary Algorithms Dynamics and its Hidden Complex Network Structures. In 2014 IEEE congress on Evolutionary Computation. New Jersey, Piscataway : IEEE, 2014, s. 3246-3251. ISBN 978-1-4799-1487-6.

[P.73] Davendra, Donald, Senkerik, Roman, Zelinka, Ivan, Pluhacek, Michal. Scatter Search Algorithm with Chaos based Stochasticity. In 2014 IEEE congress on Evolutionary Computation. New Jersey, Piscataway : IEEE, 2014, s. 860-866. ISBN 978-1-4799-1487-6.

[P.74] Senkerik, Roman, Pluhacek, Michal, Zelinka, Ivan, Kominkova Oplatkova, Zuzana. Utilization of the Discrete Chaotic Systems as the Pseudo Random Number Generators. In Advances in Intelligent Systems and Computing. 285. Heidelberg : Springer-Verlag Berlin, 2014, s. 155-164. ISSN 2194-5357. ISBN 978-3-319-06739-1.

[P.75] Senkerik, Roman, Pluhacek, Michal, Davendra, Donald, Zelinka, Ivan, Kominkova Oplatkova, Zuzana. SIMULATION OF THE DIFFERENTIAL EVOLUTION PERFORMANCE DEPENDENCY ON SWITCHING OF THE DRIVING CHAOTIC SYSTEMS. In 28th European Conference on Modelling and Simulation. Nottingham : ECMS, 2014, s. 393-398. ISBN 978-0-9564944-8-1.

[P.76] Senkerik, Roman, Pluhacek, Michal, Zelinka, Ivan, Davendra, Donald, Kominkova Oplatkova, Zuzana. Multi-chaotic Differential Evolution: A Preliminary Study. In Artificial Intelligence and Soft Computing. Heidelberg : Springer-Verlag Berlin, 2014, s. 416-427. ISSN 0302-9743. ISBN 978-3-319-07172-5.

[P.77] Senkerik, Roman, Pluhacek, Michal, Davendra, Donald, Zelinka, Ivan, Kominkova Oplatkova, Zuzana. Performance Testing of Multi-Chaotic Differential Evolution Concept on Shifted Benchmark Functions. In Hybrid Artificial Intelligence Systems. Heidelberg : Springer-Verlag Berlin, 2014, s. 306-317. ISSN 0302-9743. ISBN 978-3-319-07616-4.

[P.78] Senkerik, Roman, Pluhacek, Michal, Davendra, Donald, Zelinka, Ivan, Kominkova Oplatkova, Zuzana. An Initial Study on the Multi-Chaotic Approach for the Differential Evolution. In MENDEL 2014 20 th International conference on soft Computing. Brno : Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014, s. 89-96. ISSN 1803-3814. ISBN 978-80-214-4984-8.

[P.79] Zelinka, Ivan, Senkerik, Roman, Pluhacek, Michal. Nonrandom Evolutionary Algorithms. In MENDEL 2014 20 th International conference on soft Computing. Brno : Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014, s. 123-130. ISSN 1803-3814. ISBN 978-80-214-4984-8.

[P.80] Senkerik, Roman, Pluhacek, Michal, Kominkova Oplatkova, Zuzana. Chaos enhanced differential evolution with the dissipative map for the PID tuning problem. In Latest Trends on Systems. Volume II. Rhodes : Eurorpment, 2014, s. 516-520. ISSN 1790-5117. ISBN 978-1-61804-244-6.

[P.81] Senkerik, Roman, Pluhacek, Michal, Kominkova Oplatkova, Zuzana. Simulation of time-continuous chaotic systems for the generating of random numbers. In Latest Trends on Systems. Volume II. Rhodes : Europment, 2014, s. 557-561. ISSN 1790-5117. ISBN 978-1-61804-244-6.

[P.82] Senkerik, Roman, Pluhacek, Michal, Zelinka, Ivan, Davendra, Donald, Kominkova Oplatkova, Zuzana. Comparison of Chaos Driven PSO and Differential Evolution on the Selected PID Tuning Problem. In Proceedings of the 13th IFIP TC8 International Conference. Heidelberg : Springer-Verlag Berlin, 2014, s. 67-76. ISSN 0302-9743. ISBN 978-3-662-45236-3.

[P.83] Pluhacek, Michal, Senkerik, Roman, Zelinka, Ivan. Chaos Driven Particle Swarm Optimization with Basic Particle Performance Evaluation – An Initial Study. In Proceedings of the 13th IFIP TC8 International Conference. Heidelberg : Springer-Verlag Berlin, 2014, s. 445-454. ISSN 0302-9743. ISBN 978-3-662-45236-3.

[P.84] Zelinka, Ivan, Davendra, Donald, Senkerik, Roman, Pluhacek, Michal, Kominkova Oplatkova, Zuzana. On Convergence of Evolutionary Algorithms Powered by Non-random Generators. In Artificial Intelligence and Soft Computing. Heidelberg : Springer-Verlag Berlin, 2014, s. 492-502. ISSN 0302-9743. ISBN 978-3-319-07172-5.

[P.85] Senkerik, Roman, Pluhacek, Michal, Zelinka, Ivan, Davendra, Donald, Kominkova Oplatkova, Zuzana. Multi-chaotic Differential Evolution: Determining the Switching Time. In Advances in Intelligent Systems and Computing. 289. Heidelberg : Springer-Verlag Berlin, 2014, s. 99-110. ISSN 2194-5357. ISBN 978-3-319-07400-9.

[P.86] Davendra, Donald, Zelinka, Ivan, Senkerik, Roman, Pluhacek, Michal. Complex Network Analysis of Discrete Self-organising Migrating Algorithm. In Advances in Intelligent Systems and Computing. 289. Heidelberg : Springer-Verlag Berlin, 2014, s. 161-174. ISSN 2194-5357. ISBN 978-3-319-07400-9.

[P.87] Senkerik, Roman, Kominkova Oplatkova, Zuzana, Pluhacek, Michal, Zelinka, Ivan. Analytic Programming—A New Tool for Synthesis of Controller for Discrete Chaotic Lozi Map. In Computational Problems in Engineering. Heidelberg : Springer-Verlag Berlin, 2014, s. 137-151. ISBN 978-3-319-03966-4.

[P.88] Lapkova, Dora, Pluhacek, Michal, ADÁMEK, Milan. Using Artificial Neural Network for Force Profile Analysis in Professional Defence. In Proceedings of the 2014 International Conference on Mathematics and Computers in Sciences and Industry (MCSI 2014). New Jersey, Piscataway : IEEE, 2014, s. 0-5. ISBN 978-1-4799-4744-7.

[P.89] Davendra, Donald, Zelinka, Ivan, METLICKA, Magdalena, Senkerik, Roman, Pluhacek, Michal. Complex Network Analysis of Differential Evolution Algorithm applied to Flowshop with No-Wait Problem. In 2014 IEEE Symposium on Differential Evolution. New Jersey, Piscataway : IEEE, 2014, s. 1-8. ISBN 978-1-4799-4462-0.

[P.90] Davendra, Donald, Zelinka, Ivan, Senkerik, Roman, Pluhacek, Michal. Complex Network Analysis of Evolutionary Algorithms Applied to Combinatorial Optimisation Problem. In Proceedings of the Fifth International Cnference on Innovations in Bio-Inspired Computing and Applications. Heidelberg : Springer-Verlag Berlin, 2014, s. 141-150. ISSN 2194-5357. ISBN 978-3-319-08155-7.

CURRICULUM VITAE

Personal Information:		
	Name	Michal Pluháček
	Date of birth	5 May 1986
	Present address	Drahy 1023 696 42 Vracov
	Marital status	Single
	Contact	phone: +420 723 632 709 email: pluhacek@fai.utb.cz
Education:		
	2001 – 2005	Secondary School, Strážnice, Engineering - Computer specialization.
	2005 – 2009	Tomas Bata University in Zlín, Faculty of Applied Informatics, Information Technologies (Bc.)
	2009 – 2011	Tomas Bata University in Zlín, Faculty of Applied Informatics, Information Technologies (Ing.)
	2011 – present	Tomas Bata University in Zlín, Faculty of Applied Informatics, Engineering Informatics (Ph.D.)
Language knowledge:		
	Czech	Native
	English	Intermediate (Doctoral exam)
Professional achievements:		
	Best paper awards	Nostradamus 2013 Nostradamus 2014 CISIM 2014
	Others	Nominee for Joseph Fourier Prize 2013 Nominee for Danubius Young Scientist Award 2014
	Grant activities	Grant Agency of the Czech Republic - GACR P103/15/06700S (Member of the team)