Information Sciences

On Chaotic Dynamic Impact on the Heuristic Algorithm Performance

--Manuscript Draft--

Manuscript Number:	INS-D-21-2569
Article Type:	Full length article
Keywords:	deterministic chaos; swarm intelligence; evolutionary algorithms; algorithm dynamics; algorithm performance
Abstract:	Random mechanisms including mutations are an integral part of evolutionary algorithms. They are based on fundamental ideas of Darwin's theory of evolution as well as Mendel's theory of genetic heritage. In this participation, we debate whether random processes are needed for evolutionary algorithms or whether deterministic chaos, which is not a random process, can be used instead. We assess the mutual success of 10 evolutionary algorithms driven by chaotic dynamics and pseudorandom number generators using chaotic processes instead of random number generators. The chaos used in this competition is based on logistic equation and is used to produce N different lengths of periodical sequences, used in evolutionary algorithms. We suggest that, instead of random number generators, a specific class of deterministic or pseudo-deterministic processes can be used to improve evolutionary algorithms' efficiency. Finally, based on our findings, we propose new research questions.

Highlights

On Chaotic Dynamic Impact on the Heuristic Algorithm Performance

Ivan Zelinka, Quoc Bao Diep, Václav Snášel, Swagatam Das, Giacomo Innocenti, Alberto Tesi, Fabio Schoen, Nikolai V. Kuznetsov

- Comparing to the other research papers, this paper compares the performance of the oldest, newest, more minor and well-known algorithms on deterministic chaos generators in one massive and unique study.
- Paper show that by precision tuning, the original chaotic series convert into short N periodic time series (PTS). Thus no randomness as usually understand is there. These series are then used instead of classics pseudorandom numbers with positive impact.
- Paper reveal the clearly visible positive impact of PTS on evolutionary algorithms (EAs) dynamics, which is visible almost on all algorithms used in this paper. Compared with the same EAs with classic random generators.
- Paper open the question of whether standard random (nonchaotic) processes are really necessary for algorithm dynamics and suggest relations between randomness in EAs and noise in dynamical system control and theory.
- Paper open, sketch and suggest new ideas and strategies on how to understand algorithm dynamics as the discrete feedback dynamical systems.

On Chaotic Dynamic Impact on the Heuristic Algorithm Performance

Ivan Zelinka[a,*], Quoc Bao Diep[a], Václav Snášel[a], Swagatam Das[b], Giacomo Innocenti[c], Alberto Tesi[c], Fabio Schoen[c], Nikolai V. Kuznetsov[d,e]

[a] *Department of Computer Science, Faculty of Electrical Engineering and Computer Science VŠB-TUO, 17.listopadu 2172/15, 708 00 Ostrava-Poruba, Czech Republic*
[b] *Electronics and Communication Sciences Unit, Indian Statistical Institute, 203 B T Road, Kolkata 700108, India*
[c] *Dept. Information Engineering (DINFO) - University of Florence, via di Santa Marta 3, Firenze, Italy*
[d] *Faculty of Mathematics and Mechanics, St. Petersburg State University 198504 Peterhof, St. Petersburg, Russia*
[e] *Faculty of Information Technology, University of Jyväskylä 40014 Jyväskylä, Finland*

Abstract

Random mechanisms including mutations are an integral part of evolutionary algorithms. They are based on fundamental ideas of Darwin's theory of evolution as well as Mendel's theory of genetic heritage. In this participation, we debate whether random processes are needed for evolutionary algorithms or whether deterministic chaos, which is not a random process, can be used instead. We assess the mutual success of 10 evolutionary algorithms driven by chaotic dynamics and pseudorandom number generators using chaotic processes instead of random number generators. The chaos used in this competition is based on logistic equation and is used to produce N different lengths of periodical sequences, used in evolutionary algorithms. We suggest that, instead of random number generators, a specific class of deterministic or pseudo-deterministic processes can be used to improve evolutionary algorithms' efficiency. Finally, based on our findings, we propose new research questions.

Key words: deterministic chaos, swarm intelligence, evolutionary algorithms, algorithm dynamics, algorithm performance

*Corresponding author

Email addresses: ivan.zelinka@vsb.cz (Ivan Zelinka), diepquocbao@gmail.com (Quoc Bao Diep), vaclav.snasel@vsb.cz (Václav Snášel), swagatam.das@isical.ac.in (Swagatam Das), giacomo.innocenti@unifi.it (Giacomo Innocenti), alberto.tesi@unifi.it (Alberto Tesi), fabio.schoen@unifi.it (Fabio Schoen), nkuznetsov239@gmail.com (Nikolai V. Kuznetsov)

1. Introduction

Chaos (deterministic chaos) is a broad term that refers to a group of phenomena whose activity appears chaotic at first glance. This concept is sometimes used to describe phenomena that are strictly stochastic in nature, such as the motion of molecules in a vessel filled with gas. The discovery of the deterministic chaos hypothesis necessitated the identification of manifestations of the phenomenon in experimental results. Since chaotic systems are inherently nonlinear, traditional statistical methods, which are often linear, are inadequate for their study. A chaotic system's behavior is a random-like mechanism for external observers. Fourier spectral analysis, for example, can reveal nonzero amplitudes at all frequencies in a chaotic environment, making chaos easily mistaken for random noise. For a long time, deterministic chaos was at home only in purely theoretical physics or other natural sciences. In recent years, especially in the last decade, when computer science has developed rapidly, it has been found that chaos can be instrumental in generating pseudo-random numbers and that it is essential to understand and control the chaos. Many research papers have been published in this direction that deals with this phenomenon in both directions. We will change some of them here now.

Chaos has previously been found in a variety of processes, including evolutionary systems. Chaos has also been used to substitute pseudorandom number generators (PRNGs) in evolutionary algorithms (EAs) in recent years. Consider academic papers such as [1], one of the first use of chaos inside EAs is reported [2], [3]-[4] discussing the use of deterministic chaos inside particle swarm algorithm instead of PRNGs, [5] - [6] investigating relations between chaos and randomness or the others [7], [8], [9] and [10] using chaos with EAs in applications, amongst the others.

In [5], for example, researchers combined deterministic chaos and a pseudo-random number generator. The use of ultra-weak multidimensional coupling of p 1-dimensional dynamical systems to generate random or pseudorandom numbers is discussed there. In another article, [11] examines the logistic map as a potential pseudorandom number generator and compares it to current pseudorandom number generators. The results of logistic maps are compared to traditional pseudorandom number generation methods. The method is used to calculate the number, delay, and period of the logistic map's orbits with varying degrees of precision.

Another paper by [12] suggested a pseudorandom number generator algorithm, which combines coupled map lattice and chaotic iteration. This algorithm was also put to the test in NIST 800-22 statistical test suits, which are often used in image encryption. The authors of [13] use properties of chaotic systems to construct the CCCBG, a random bit generator in which two chaotic systems are cross-coupled with each other. The four basic tests are used to evaluate the bitstreams developed by the CCCBG: mono bit test, serial test, auto-correlation, and Poker test. The NIST suite checks, which are the most rigorous randomness tests, were also used.

Another paper [6] proposes a binary stream-cipher algorithm using dual one-

dimensional chaotic maps, with statistic properties indicating that the sequence is of strong randomness. Similar research is carried out in [14], [2], [15], and [16]. Several articles, such as academic papers, have been written in recent years that concentrate on the use of chaos in algorithms like [17], [18], [19], [20], [21], or [22].

Some of them compared the impact of a chaotic version on a CEC (Congress of Evolutionary Algorithms) test benchmark function (however older and typically one set only) to a nonchaotic version of the same algorithm. We've agreed to broaden the scope of this research to include a newer CEC collection of benchmark test functions, as well as more algorithms of various types (evolutionary vs swarm one). We also included one critical aspect of the experiment: we looked at the effect of chaos on algorithm performance as well as the impact of the "degree" of chaos on algorithm performance. This paper focuses on using deterministic chaos to produce N periodic sequences (forced by low calculation precision), which are used in evolutionary algorithms instead of pseudorandom number generators. Various EAs of different kind are used here, see table 2, do not analyze whether used pseudo-random numbers are really random one and do not use information about its randomness. Random and chaotic series are **only simply used**.

In recent years, several interesting contributions have been made dealing with the use of chaos within optimization algorithms. it is worth mentioning, for example, work [23], which also deals with the use of chaos in a newly designed algorithm, which is tested on a large number of test functions. Other, [24] is discussed investigation into the performance, scalability, convergence and robustness of chaos-enhanced evolutionary algorithms with boundary constraints and also papers like [25], [26], [27] or [28] deal with chaos application inside evolutionary algorithms. It is, therefore, clear that recently the interest of scientists has begun to be attracted by the hybridization of modern optimization algorithms and chaotic dynamics. The mentioned publications deal with applications of chaos within algorithms from different points of view, but primarily only as an improvement of one specific algorithm tested for a more significant number of functions or several algorithms that are then compared with each other. A research study dealing with how much the performance of different algorithms depends on different levels of chaotic dynamics has not yet been satisfactorily developed. For these reasons, this work was created, which aims to point out how much the performance of the evolutionary algorithm depends on the degree of chaotic dynamics, which is used inside the algorithm instead of a random number generator. This research also raises interesting research questions that put evolutionary algorithms on the level of discrete dynamic systems with feedback and thus allow the use of the theoretical mathematical apparatus of cybernetics to analyze and describe such systems. This could explain why chaos has a positive effect on the performance of evolutionary algorithms. However this is question of the future research.

In this paper, we suggest the idea that evolutions and swarm systems are multiagent systems with intense internal interactions (discrete) dynamical systems for which already exist rich mathematical apparatus in control theory. Applying this pre-existing mathematical apparatus to evolutionary and swarm

intelligence algorithms shall help shed light on the many unanswered questions in the algorithm community today and provide exciting and powerful mathematical tools for researching these algorithms.

EAs with deterministic chaos systems (DCHS), as demonstrated here, produce the same or better results, as seen in the results. Since DCHS generates periodic series (due to the final numerical precision - the location of the decimal behind zero), it is obvious that the output of EAs with DCHS should be comparable to that of classical EAs with randomness. As a result, the article's structure is a s follows. We will clarify the relevance of research in this direction and point out the unanswered questions in the use of chaos in evolutionary algorithms in the following section, titled motivation. The experiment design section follows, which explains the conditions under which our experiments were conducted. We also explore the impact of accuracy on the chaotic course's periodicity, as well as the nature of our own experiments. The Congress on Evolutionary Algorithms (CEC) test functions were used to test ten selected evolutionary algorithms. The results are discussed and summarized in the conclusions.

2. Motivation

The experiments detailed in this article were motivated by the current state, which was detailed in the preceding section, as well as the widespread perception that operations like mutations and other operations that need random operations cannot be performed without random numbers. It's evident from the description of evolutionary algorithms that they're discrete dynamic systems, which means they're feedback systems that process their output, in this case the population with fitness. Which individual will be used in the following iteration is determined by fitness (generation, migration, etc.). As a result, if we consider evolutionary or swarm algorithms to be dynamic systems, the assumption that successful control of these systems requires random processes is at odds with what the evolutionary algorithm community normally expects (remind that in control theory randomness usually represent a destructive force, that has to be eliminated).

As a result, our tests were meant to see if the random numbers created by the traditional type of random number generator (Mersenne - Twister like) are actually necessary for the performance of evolutionary algorithms, or if alternative sorts of processes, such as deterministic chaos processes, can be used instead.It is possible to create chaotic series processes of various lengths. Chaos as such, as well as pure randomness, do not exist in computers. It's only pseudo-chaotic or pseudorandom processes here. Chaos, unlike pseudorandom processes generated by other algorithms, may alter the length of its period by altering the precision of its calculation. In other words, if we employ, for example, a logistic equation 1 with the parameter $A = 4$ and let this equation loop in a computer, we have a theoretically quasi-random series (for the external observer) that never repeats and, in fact, belongs to the world of deterministic chaos due to the high accuracy with which today's computers work. When the precision of the calculation is reduced, however, rounding happens, resulting in the effect

that this otherwise potentially endlessly long series of chaotic numbers begins to recur. As shown below, the amount of repeat is dependent on the extent of the computation accuracy.

Our tests are not only based on determining the impact of chaotic dynamics as random numbers on the algorithm's performance, but also on determining if modifying the computation accuracy or shortening the chaotic period will alter the EAs' performance quality. It is also debated whether truly random processes are required or whether short pseudo-chaotic processes generated using a deterministic method are sufficient. This raises a number of new research problems, such as whether we actually require evolutionary algorithms for random processes to work, and whether there is a link between the type of test function (which typically contains nonlinear elements and is a nonlinear system interacting with a specific algorithm) and algorithm performance. All of this is demonstrated by our findings. In the conclusion section, further possible questions and research ideas are presented.

3. Experiment Design

This study's experiments can be classified into two categories. The first examines how the existence of periodicity generated by deterministic chaos systems is influenced by computation precision, while the second employs N periodical time series generated by chaotic systems inside EAs and compared with the same EAs powered by PRNGs.

Our experiments were designed according to the following facts. To verify/demonstrate our hypothesis about chaos usability firstly must be remembered that there are a large number of algorithms and a large number of test problems, a large number of repeated simulations from different initial conditions (necessary to demonstrate the viability of the results). All this must then be statistically evaluated, and it is clear that it is not possible to cover all possible scenarios. We, therefore, focused on a narrower selection of more well-known and less well-known algorithms, both in the field of both classical evolutionary algorithms and in the field of swarm intelligence. These were algorithms such as differential evolution (DE) of particles swarm (PSO), grey wolf algorithm (GWA), SOMA algorithm, and many others. These algorithms were tested regularly on selected test functions used at Congress of evolutionary algorithms (CEC) conferences. Regular statistical tests and evaluations were then performed on a huge amount of results, which in the end allowed us to evaluate the impact of the chaos use with varying degrees of accuracy on the performance of these algorithms.

The experiments were performed on a standard PC with a classic i7 processor and 8 GB of memory, as well as partly on a supercomputer of the national supercomputer center IT4 (https://www.it4i.cz/en) to speed up the calculation and shorten the time. There was no other reason to use a supercomputer - so these experiments can be repeated on any fast enough computer. Only the logistic equation 1 was used to generate chaos because it is a well-known basic system generating chaos. The truth is, that our study can be extended by a larger class of chaotic systems. However, it is clear from the results we obtained that

extending a larger class of chaotic systems would only yield more simulations, but the results would probably not be fundamentally different.

Let's clarify a bit a few preliminaries now.

4. Chaos or Periodicity

The so-called logistic equation (Eq. 1) was used to demonstrate the reliance of pseudo-chaos periodicity on numerical precision. Several experiments were carried out, and some figures displaying the results were created, as shown in see Fig. 1 - 7. Other chaotic generators of various mathematical descriptions, such as Lozi, Henon, Ikeda, or others, such as those experimentally manufactured and published in [1], can also be employed.

In the chaotic dynamics can be also observed the existence of pseudo-patterns, that is, the situation where a time series does not repeat itself exactly, but it periodically displays a very similar behavior. This is a feature of true chaos, which has an inner structure made of a "basic recurring shape", which is persistently approached by the system trajectory. Each chaos, i.e., each pseudo-pattern, has its unique features, which make a number no more uncorrelated with the previous ones. As a consequence, the probability density function is no more evenly distributed among different sequences of numbers, which is a big difference from white noises. Therefore, using chaos for random numbers can make the generator "biased", as well as the algorithms which exploit it. However, to observe this, shorter parts of "clear" chaotic series shall be taken into consideration where these pseudo-patterns are dominant (i.e. take a major part of chaotic series). In our experimentation we have used max 100 000 cost function evaluations, so that means $X\times$ more chaotic numbers has been used for one algorithm run (remind, a "randomness" is generally needed more than once, to build new offspring for one cost function evaluation) so thus the existence of pseudo-patterns shall have an insignificant impact on algorithm performance if occurred in there. Because of divergence of nearby trajectories, such patterns shall shortly end if appears in chaotic series.

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

The logistic equation has been employed in several stages of research, i.e.

- Identification of a N periodic orbit with numerical precision as a criterion, see Fig. 1 - 7.
- A global picture of the behavior of a logistic equation with a fixed parameter A=4, see Fig. 2 and 3.
- A broad view of the behavior of logistic equations considering a variety of parameters $A \in [3.4, -4]$ and numerical precision $\in [1, 20]$, logistic equation has been iterated for 1 000, 10 000, 100 000 iterations, see Tab. 1.
- Investigate and visualize why and when n periodic activity occurs, see Fig. 2 - 3.

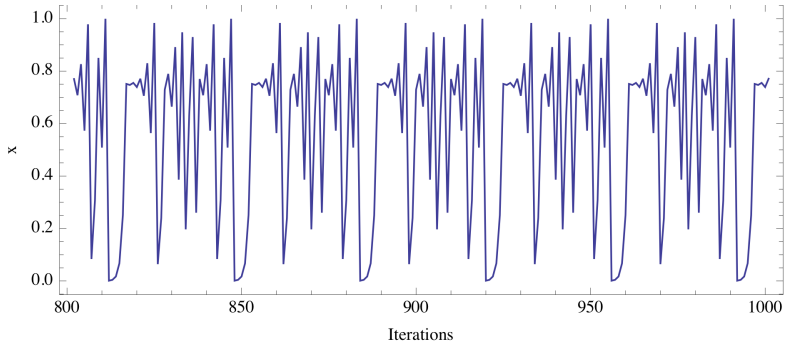

Figure 1: The period 36 (precision = 4) based on Eq. 1 for $A = 4$, see Tab. 1

- The analysis of the dynamics of logistic equation has been done. For parameter $A = 4$, numerical precision $\in [1, 13]$, initial conditions $x_{start} \in [0.01, 0.99]$ (x was incremented periodically by $= 0.01$) and 1 000 000 iterations for each combination of this parameters, see Table 1.

The impact of the precision on the dynamics of logistic equation is depicted on Fig. 2 (compare with Fig. 3).The original mapping function is changed to a step-wise mapping function with low accuracy. This is the origin of many of the periodic orbits that we later used in our studies.

4.1. Determinism or Randomness

Instead of using random numbers, we employed time series generated by chaos generators (e.g. Fig. 1 and Eq. 1). Due to the fact, that numerical precision has clearly an impact on the occurrence of periodicity in chaos, logistic equation, Eq. 1 has been selected. Data series generated by this equation (numerical precision from the interval [1, 13]) with setting $A = 4$, see Tab. 1 which shows minimal and maximal period observed for current setting. The repeated generation of those series were started on random initial conditions. Algorithms selected for our experiments are reported in section 5.2. The setting of all versions of used algorithms is referred to in Tab. 2. It is simple to compute how many times the data series of pseudo-chaotic numbers (PCHNs) generated by DCHS have been used in EAs using this configuration and algorithm architecture.

All PRNG numbers were replaced by PCHNs (with different numerical precision and thus period length). As for comparison, the same algorithms in its canonical versions has been used, with classical PRNGs, just to compare performance.

In our experiments with chaos use, the PRNG has been used only once in each experiment - at the beginning of the chaotic series as the seed for the start of chaotic system iteration. That means if a chaotic series was generated $X\times$, then PRNG was used to uniformly generate X random numbers in the interval [0,1] that has been used as the initial starting value for chaotic generator. It is fact, that importance of the seeding, i.e. of the initialization is the most

Table 1: Periodicity dependence of Eq. 1 on various numerical precision (up to precision 13).

Numerical Precision	Minimal Period	Maximal Period
1	4	4
2	2	10
3	10	29
4	15	36
5	67	170
6	143	481
7	421	758
8	1030	4514
9	2277	11227
10	2948	35200
11	9668	57639
12	65837	489154
13	518694	518694

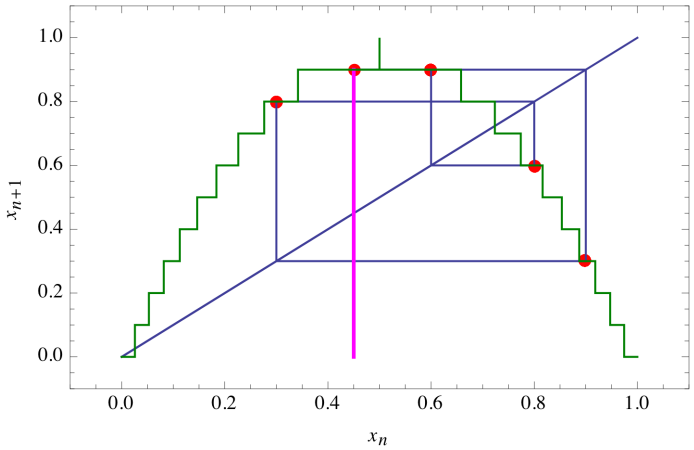

Figure 2: Cobweb diagram: a low numerical precision = 1 (one decimal behind zero) impact on the mapping function shape. Four periodic orbit are observable (blue line).

important because certain starting conditions can converge in a finite number of iterations into fixed points, or short period solutions, which are definitely not suited for the scope. They could be referred to as "pathological traps". However, to exclude such initial seeds would be out of the scope of our experiments and real-life use, we decided to not exclude this problem from our experimentation to demonstrate that even so, replacement PRNG and use chaos instead of, is still very robust and such pathological traps have likely rare and negligible impact.

5. Computational Setup

5.1. Test functions

To rigorously and thoroughly evaluate the impact of a deterministic chaos system on evolutionary algorithm performance, three well-known test suites of

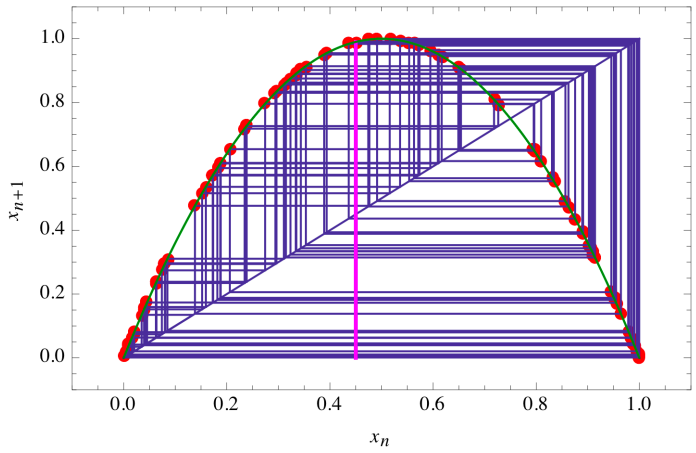

Figure 3: Cobweb diagram: full chaotic dynamics with of logistic equation. Compare with Fig. 2, the same initial conditions are used.

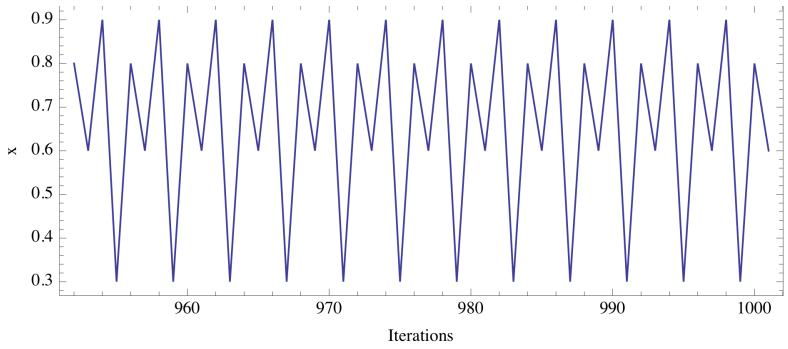

Figure 4: Periodicity of logistic equation under precision 1 and $x_{start} = 0.45$.

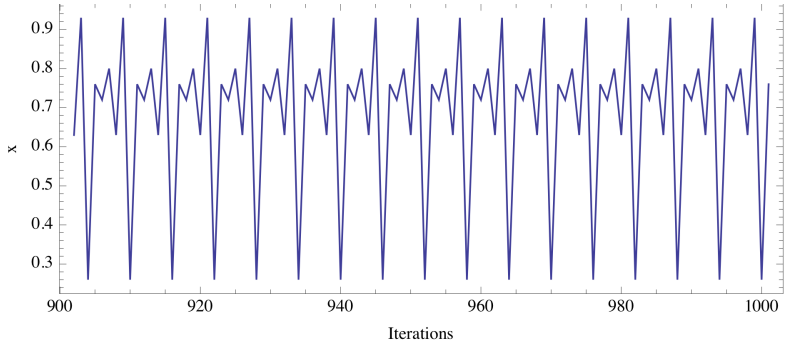

Figure 5: Periodicity of logistic equation under precision 2 and $x_{start} = 0.8$.

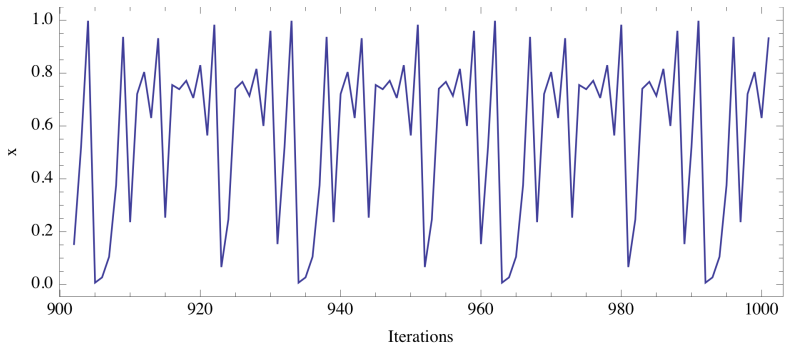

Figure 6: Periodicity of logistic equation under precision 3 and $x_{start} = 0.6$.

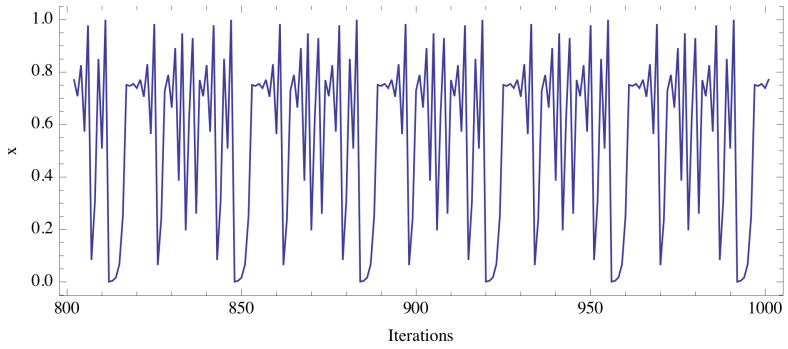

Figure 7: Periodicity of logistic equation under precision 4 and $x_{start} = 0.02$.

the IEEE Congress on Evolutionary Computation (IEEE CEC) are used and listed below.

- The first benchmark set is the IEEE CEC 2015 Competition on Learning-based Real Parameter Single Objective Optimization (CEC 2015, [29]);
- The second is the IEEE CEC 2017 Special Session and Competition on Single Objective Real Parameter Numerical Optimization (CEC 2017, [30]);
- And the last one is the IEEE CEC 2020 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization (CEC 2020, [31]).

The CEC 2015, 2017 and 2020 contains 15, 30 and 10 fitness functions, respectively. **A total of 55 functions** were used, including 6 Unimodal, 10 Simple Multimodal, 3 Basic, 16 Hybrid, and 20 Composition functions, challenging enough to rate the performance of any evolutionary algorithm. All test functions are shifted global optimum and are scalable, see [29], [30], [31] for more information.

5.2. Test algorithms

Ten popular algorithms, including the evolutionary computation and swarm intelligence algorithms, were selected for the experiment and listed below.

- Differential Evolution - DE/best/1/bin [32] (DE);
- Self-Organizing Migrating Algorithm AllToOne version [33, 34] (SOMA-ATO);
- Particle Swarm Optimization [35] (PSO);
- Artificial Bee Colony Algorithm [36] (ABC);
- Ant Colony Optimization for Continuous Domains [37] (ACO);
- Firefly Algorithms [38] (FA);
- Cultural Algorithm [39] (CA);
- Grey Wolf Optimizer [40] (GWO);
- The Whale Optimization Algorithm [41] (WOA);
- A Hybrid Firefly and Particle Swarm Optimization Algorithm [42] (HF-PSO).

These selected algorithms use many types of PRNGs, including uniformly distributed random numbers, continuous uniform random numbers, normally distributed random numbers, uniformly distributed pseudorandom integers, and so on. The performance of the algorithm using PRNGs is compared with

its performance when replacing PRNGs with a deterministic chaos system. Comparing performance between different algorithms is unnecessary and beyond the scope of this paper. The primary setting of all algorithms used in our experiments is reported in Table 2. Their meanings and additional parameters can be found in the original publications were cited.

Table 2: The detailed control parameters of the algorithms.

Algorithms	The control parameter values
DE	$pop = 100; F_{min} = 0.4; F_{max} = 1; Cr = 0.5$
SOMA	$pop = 50; Step = 0.11; PRT = 0.1; PathLength = 3.0$
PSO	$pop = 100; w_i = 1; w_{damp} = 0.99; c_1 = 1.5; c_2 = 2.0; v_{max} = 0.1 \times [-100, 100]^D; v_{min} = -v_{max}$
ABC	$pop = 50; n_{onlooker} = pop; l = 0.6D.pop; a = 1$
ACO	$pop = 50; n_{sample} = 40; q = 0.5; \zeta = 1$
FA	$pop = 25; \alpha = 0.2; \beta_0 = 1; \gamma = 1; \alpha_{damp} = 0.98; \delta = 0.05.[-100, 100]^D$
CA	$pop = 50; \alpha = 0.3; \beta = 0.5; p_{accept} = 0.35; n_{accept} = p_{accept}.pop$
GWO	$pop = 30; \vec{A} = 2\vec{a}.\vec{r_1} - \vec{a}; \vec{C} = 2.\vec{r_2}$
WOA	$pop = 30; b = 1; l = (a_2 - 1)rnd + 1$
HFPSO	$pop = 30; \alpha = 0.2; \beta_0 = 2; \gamma = 1; c_1 = c_2 = 1.49445; v_{max} = 0.1 \times [-100, 100]^D; v_{min} = -v_{max}; w_i = 0.9; w_f = 0.5$

5.3. Parameter settings

Tests on $10D$ is carried out for the total of 55 problems, with the search range of $[-100, 100]^D$. The maximum number of function evaluations is 10 000$\times D$ ($MaxFEs$ =100 000). Error value smaller than 10^{-8} will be taken as zero, as experimental settings requested in [29], [30], [31].

The Wilcoxon rank-sum test (WRT) was applied at the 5% significance level to evaluate whether the differences between the results are significant [43, 44].

For PRGNs system: 10 algorithms were tested. For each algorithm, 55 functions were used. For each test function, 51 independent runs were performed. And for each independent run, 100 000 function evaluations were called (see Fig. 8).

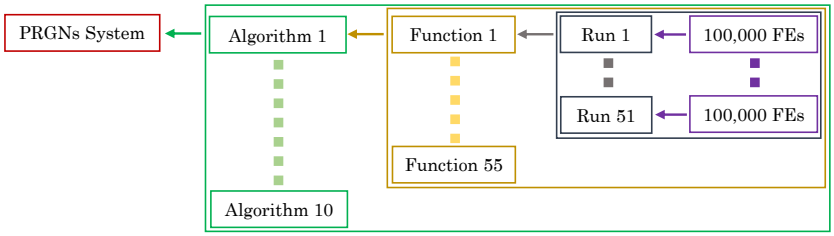

Figure 8: The PRGNs system layout for the test.

For deterministic chaos system: 551 chaotics series were generated. For each chaotics series, 10 algorithms were tested. For each algorithm, 55 functions were used. For each test function, 51 independent runs were performed. And for each independent run, 100,000 function evaluations were called (see Fig. 9).

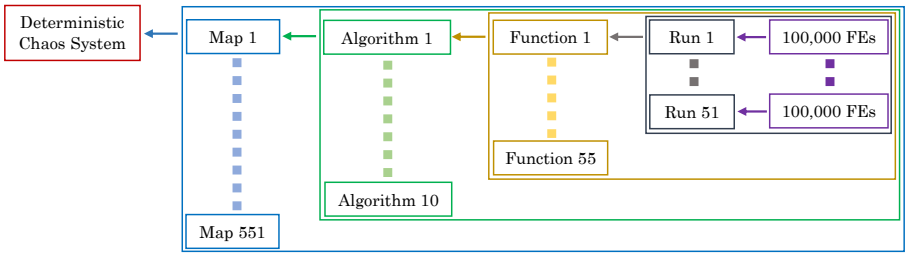

Figure 9: The deterministic chaos system layout for the test.

In 551 chaotics series: One chaotics series with full precision and 550 chaotics maps with limited precision. 550 chaotics maps were divided into 50 series and 11 levels of precisions. The difference between series of chaos maps is the first number x_0 created by the setting of *rng default* of Matlab (repeatable).

5.4. Hardware infrastructure

All tests were run on the Salomon supercomputer (https://www.it4i.cz/en), under CentOS 7.x operating system, Matlab R2015b version (8.6.0.267246) 64-bit (glnxa64), using Parallel Computing Toolbox, within 20 compute nodes, 24 cores of two twelve-core Intel Haswell processors for each node (480 workers), and 2560 GB RAM in total.

The simulation consumed about 150,000 core hours for the computation, equivalent to around 17.36 continuously working years of the single CPU with 2.5 GHz.

6. Results

The results obtained from the description of the experiments in the previous section were visualized as done in the following figures. Because it was a large amount of data that needed to be displayed in some readable but also compressed way, we decided to display, which can be seen in the following pictures. They show a graph for each algorithm, consisting of three colors. The x-axis shows the number of test functions used, the y-axis the accuracy with which the respective chaotic series was generated. We remind that lower accuracy equals shorter chaotic series, which is repeated earlier. Colors have the following meanings.

The green colour represents those cases where the algorithm-driven by the classic random number generator won. The yellow colour represents those cases where, from the point of view of statistical evaluation, both algorithms had the same performance, and the blue colour represents those cases in which the algorithm, which was powered by a chaotic number generator, won. Thus,

each figure represents a comparison of the selected algorithm with itself, with one version being driven by chance and the other version being driven by the chaos of different lengths of the chaotic series period, depending on the already mentioned calculation accuracy. It is clear from the figure that the random number generator clearly wins in the lower part of the calculation accuracy for chaotic series. More specifically, if the accuracy of the chaotic series calculation is low, or chaos has a short period, then the algorithm that uses the random generator gives better performance. However, it can be seen that from about precision 6, i.e. calculating to 6 decimal places, for many algorithms chaos begins to prevail and gives better results than the original algorithm with a random number generator. In this case, a draw marked in yellow in the middle can be added to the good of chaos, because its presence did not impair the performance of the algorithm. However, in the evaluation better - worse we do not take this into account. In some algorithms, the benefit of chaos is quite significant, as can be seen, for example, with the ACO algorithm or the cultural algorithm. The SOMA algorithm also shows strong signs of improvement over its classic canonical version. We note that this algorithm has many modern variants that have not been tested here. In the tables below, the exact numerical values and performance evaluations of the algorithm are published and which are shown in the figure10-11 and its numerical form in the tables 3 - 12.

Evolutionary algorithms also work with random integer numbers. During our experimentation, we had met some obstacles when chaotic series was used. Some algorithms use the *randi* (uniformly distributed pseudorandom integers) to randomly select individuals in populations or subpopulations (a small number of individuals). For these algorithms, replacing *randi* with DCHS leads to the inefficient working algorithm (for example DE, see Fig. 11).

The reason is visualized in Fig. 12. The logistic equation return just values between 0-1, Fig. 12, on the left. For population members selection, we need integers in the interval [1,N]. So firstly, we just expanded the series into the interval [1,N] like [1,6] (on the right). However, for bigger populations is shows itself as inefficient. We have found that acceptable correction is if the chaotic series is expanded into interval where $N >>$ population size, like [1,200] (in the middle) and then integer needed for the individual selection from the population is given by formula

$$Selected\_Individual = N \; modulo \; population\_size. \tag{2}$$

This has been verified as the efficient source of *chaotic integers*.

Lets shortly summarize findings about chaotically generated integers:

- If the algorithm only uses *rand*: replace rand by deterministic chaos generator ⇒ better performance (than original)
- If the algorithm uses many types of random (for example: *rand*, *unifrnd*, *randn*):
  - just replace one type ⇒ worse performance

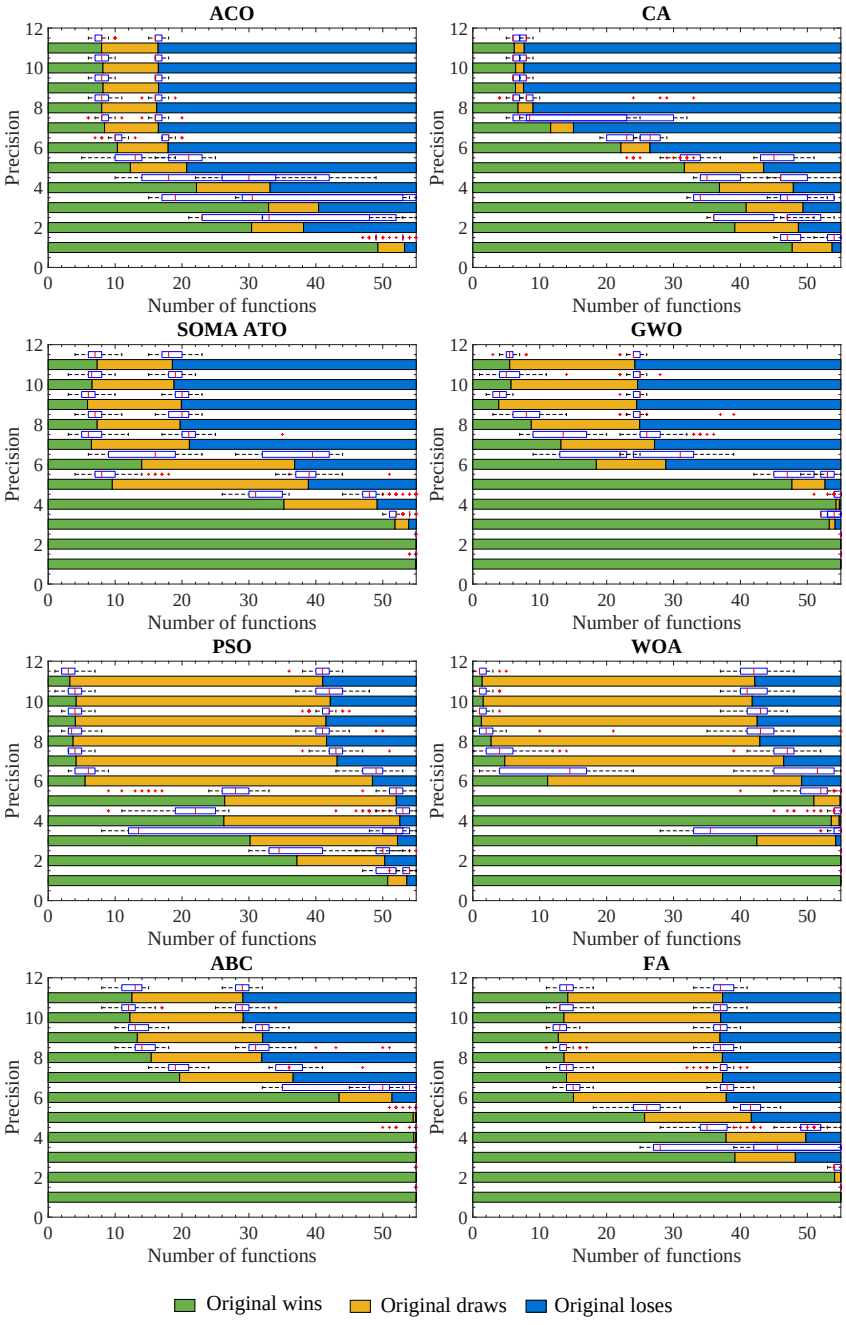

Figure 10: The graphically visualized results on the IEEE CEC 2015, 2017, and 2020.

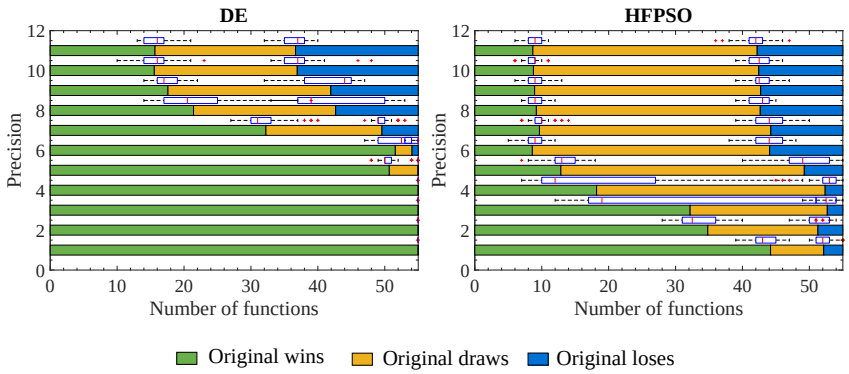

Figure 11: The graphically visualized results on the IEEE CEC 2015, 2017, and 2020.

– replace all types ⇒ better performance

- Replace integers - with small range [lower upper] ⇒ no change or worse performance (because it is no longer a Chaotic map).
- Replace integers - with huge range [lower upper] and use formula (2) ⇒ no change or better performance.

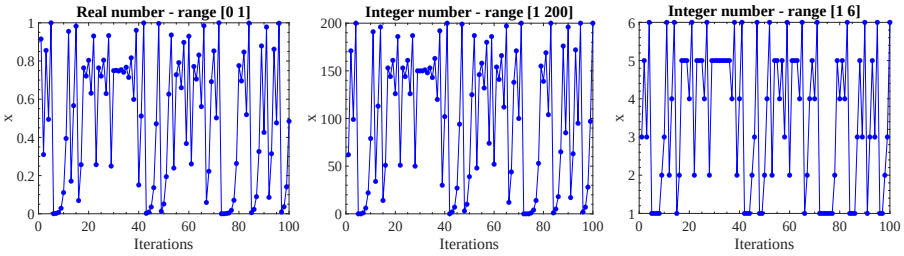

Figure 12: The problem of integer number sequences.

7. Conclusion

At the end of this paper, the following can be stated. A well-known logistic equation was used as a chaos generator, which, as has been verified and proven many times, generates deterministic chaos, and the behaviour of this equation was generated with varying degrees of calculation accuracy/precision. This accuracy had an effect on whether the chaotic series was theoretically infinite and therefore unrepeatable or began to repeat after N iterations. The resulting lengths of various lengths, i.e. infinitely long chaotic series but also a series of repeated partially chaotic series (so it was a kind of pseudo-chaotic periodic processes), were then used instead of random numbers in all used evolutionary

Table 3: Comparison of DE algorithms (original vs chaos version with different PopSize) on the CEC 2017 benchmark suite (10 dimensions, 51 runs, full precision, WRT at 5%).

CEC 2017	PopSize=50		PopSize=200		PopSize=500	
Function	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)
F_1	$1.24e+03(1.63e+03)$	$1.37e+03(1.69e+03)\approx$	$1.81e+03(1.80e+03)$	$1.54e+03(1.69e+03)\approx$	$6.44e+03(5.94e+03)$	$2.82e+03(2.94e+03)+$
F_2	$0.00e+00(0.00e+00)$	$0.00e+00(0.00e+00)\approx$	$0.00e+00(0.00e+00)$	$0.00e+00(0.00e+00)\approx$	$1.61e+02(4.19e+02)$	$4.54e+01(6.92e+01)+$
F_3	$0.00e+00(0.00e+00)$	$0.00e+00(0.00e+00)\approx$	$1.09e+02(5.82e+01)$	$2.98e+01(2.70e+01)+$	$1.51e+03(4.54e+02)$	$1.12e+03(3.70e+02)+$
F_4	$1.20e-01(5.66e-02)$	$2.19e-01(3.45e-01)-$	$2.14e+00(5.69e-01)$	$2.03e+00(6.91e-01)\approx$	$3.29e+00(8.51e-01)$	$3.51e+00(6.54e-01)\approx$
F_5	$1.00e+01(4.67e+00)$	$1.01e+01(3.96e+00)\approx$	$1.85e+01(2.98e+00)$	$6.34e+00(2.88e+00)+$	$2.14e+01(3.99e+00)$	$1.23e+01(3.66e+00)+$
F_6	$1.17e-06(6.95e-06)$	$8.46e-05(4.03e-04)-$	$0.00e+00(0.00e+00)$	$0.00e+00(0.00e+00)\approx$	$1.18e-04(4.20e-05)$	$1.24e-05(4.68e-06)+$
F_7	$2.52e+01(3.27e+00)$	$2.01e+01(4.29e+00)+$	$3.05e+01(4.42e+00)$	$1.55e+01(3.35e+00)+$	$3.46e+01(4.28e+00)$	$2.88e+01(4.67e+00)+$
F_8	$1.02e+01(5.50e+00)$	$9.13e+00(3.09e+00)\approx$	$1.90e+01(3.58e+00)$	$6.24e+00(2.31e+00)+$	$2.34e+01(3.54e+00)$	$1.43e+01(4.58e+00)+$
F_9	$0.00e+00(0.00e+00)$	$5.52e-02(1.48e-01)-$	$0.00e+00(0.00e+00)$	$0.00e+00(0.00e+00)\approx$	$1.57e-07(1.04e-07)$	$5.21e-09(1.10e-08)+$
F_{10}	$4.88e+02(2.74e+02)$	$4.10e+02(2.15e+02)\approx$	$8.53e+02(1.74e+02)$	$1.83e+02(1.14e+02)+$	$1.02e+03(1.42e+02)$	$6.14e+02(2.55e+02)+$
F_{11}	$1.17e+00(1.27e+00)$	$2.78e+00(2.16e+00)-$	$1.31e+00(1.36e+00)$	$1.85e+00(1.62e+00)\approx$	$5.78e+00(1.15e+00)$	$3.29e+00(1.82e+00)+$
F_{12}	$1.37e+04(1.33e+04)$	$1.37e+04(1.41e+04)\approx$	$2.41e+04(1.84e+04)$	$1.69e+04(1.40e+04)+$	$1.50e+05(1.11e+05)$	$6.02e+04(6.56e+04)+$
F_{13}	$8.38e+02(8.80e+02)$	$1.85e+03(4.99e+03)\approx$	$4.34e+02(4.71e+02)$	$1.06e+03(9.15e+02)-$	$3.15e+02(2.46e+02)$	$5.49e+02(5.19e+02)\approx$
F_{14}	$1.15e+01(1.81e+01)$	$1.61e+02(4.77e+02)-$	$1.20e+01(4.19e+00)$	$5.45e+00(7.06e+00)+$	$2.22e+01(3.95e+00)$	$1.08e+01(6.25e+00)+$
F_{15}	$1.03e+00(1.20e+00)$	$3.54e+02(2.24e+03)-$	$3.77e+00(2.18e+00)$	$4.43e+00(8.66e+00)-$	$9.46e+00(2.70e+00)$	$5.09e+00(3.43e+00)+$
F_{16}	$7.90e+00(2.92e+01)$	$7.71e+01(1.03e+02)-$	$7.75e-01(1.56e+00)$	$8.05e+00(2.51e+01)\approx$	$1.43e+00(7.48e-01)$	$2.13e+00(3.83e+00)-$
F_{17}	$4.56e+00(9.72e+00)$	$1.67e+01(2.41e+01)-$	$1.71e+00(3.96e+00)$	$3.97e+00(7.07e+00)-$	$1.29e+01(4.08e+00)$	$2.96e+00(5.11e+00)+$
F_{18}	$1.14e+02(1.93e+02)$	$2.06e+03(4.37e+03)\approx$	$1.05e+02(1.33e+02)$	$7.95e+01(1.72e+02)+$	$8.76e+02(1.47e+03)$	$3.34e+02(6.32e+02)+$
F_{19}	$4.23e-01(5.04e-01)$	$6.34e+02(1.89e+03)-$	$1.23e+00(1.36e+00)$	$1.01e+00(2.99e+00)+$	$4.89e+00(9.85e-01)$	$2.42e+00(2.20e+00)+$
F_{20}	$2.00e+00(4.30e+00)$	$7.86e+00(1.80e+01)-$	$4.28e-01(4.45e-01)$	$9.32e-01(1.92e+00)-$	$2.89e-01(2.83e-01)$	$7.10e-01(6.61e-01)\approx$
F_{21}	$1.91e+02(4.84e+01)$	$1.65e+02(5.68e+01)+$	$1.40e+02(5.72e+01)$	$1.56e+02(5.50e+01)\approx$	$1.33e+02(5.33e+01)$	$1.22e+02(4.33e+01)+$
F_{22}	$9.36e+01(2.55e+01)$	$1.01e+02(7.77e-01)-$	$8.69e+01(3.67e+01)$	$9.42e+01(2.47e+01)-$	$8.98e+01(3.49e+01)$	$8.88e+01(3.24e+01)+$
F_{23}	$3.07e+02(3.61e+00)$	$3.11e+02(5.75e+00)-$	$3.17e+02(4.13e+00)$	$3.08e+02(2.97e+00)+$	$3.21e+02(4.43e+00)$	$3.12e+02(4.43e+00)+$
F_{24}	$3.13e+02(7.88e+01)$	$3.22e+02(6.57e+01)\approx$	$3.16e+02(8.72e+01)$	$2.95e+02(9.73e+01)+$	$2.74e+02(1.20e+02)$	$2.60e+02(1.19e+02)+$
F_{25}	$4.20e+02(2.32e+01)$	$4.28e+02(2.30e+01)-$	$4.20e+02(2.32e+01)$	$4.19e+02(2.33e+01)\approx$	$4.21e+02(2.33e+01)$	$4.18e+02(2.34e+01)\approx$
F_{26}	$3.00e+02(1.70e+01)$	$3.10e+02(2.20e+01)-$	$2.94e+02(4.20e+01)$	$2.96e+02(4.33e+01)\approx$	$3.00e+02(6.42e-06)$	$3.00e+02(7.77e-06)+$
F_{27}	$3.93e+02(2.70e+00)$	$3.94e+02(3.16e+00)-$	$3.91e+02(2.90e+00)$	$3.92e+02(2.75e+00)\approx$	$3.91e+02(2.38e+00)$	$3.91e+02(2.88e+00)\approx$
F_{28}	$4.07e+02(1.39e+02)$	$4.14e+02(1.53e+02)\approx$	$3.21e+02(7.53e+01)$	$3.26e+02(8.20e+01)\approx$	$3.01e+02(1.11e+00)$	$3.09e+02(2.91e+01)\approx$
F_{29}	$2.64e+02(8.08e+00)$	$2.57e+02(2.10e+01)+$	$2.74e+02(1.01e+01)$	$2.60e+02(1.28e+01)+$	$2.84e+02(1.25e+01)$	$2.78e+02(1.52e+01)+$
F_{30}	$1.34e+05(2.98e+05)$	$1.49e+05(3.18e+05)-$	$3.46e+04(1.22e+05)$	$6.86e+04(2.21e+05)-$	$5.49e+04(4.26e+04)$	$4.26e+04(1.56e+05)+$
Chaos wins	+	3	+	12	+	23
Original wins	−	16	−	6	−	1
Similar	≈	11	≈	12	≈	6

Table 4: Comparison of algorithms (SOMA ATO, HFPSO, and ABC original vs chaos version) on the CEC 2017 benchmark suite (10 dimensions, 51 runs, full precision, WRT at 5%).

CEC 2017	SOMA ATO		HFPSO		ABC	
Function	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)
F_1	$1.66e+03(2.12e+03)$	$1.14e+01(1.61e+01)+$	$2.57e+03(2.94e+03)$	$3.62e+03(4.10e+03)\approx$	$5.91e+02(6.18e+02)$	$1.29e+04(2.30e+04)-$
F_2	$1.43e+00(7.09e+00)$	$0.00e+00(0.00e+00)+$	$7.37e+00(3.46e+01)$	$6.28e+02(4.44e+03)\approx$	$2.49e+06(3.09e+06)$	$1.78e+06(7.67e+06)+$
F_3	$1.67e+03(9.72e+02)$	$7.57e-02(1.21e-01)+$	$0.00e+00(0.00e+00)$	$0.00e+00(0.00e+00)\approx$	$1.39e+04(3.89e+03)$	$1.21e+03(2.90e+03)+$
F_4	$4.02e+00(1.73e+00)$	$3.96e+00(1.28e+00)\approx$	$1.31e+00(7.76e+00)$	$1.47e+00(5.45e-01)-$	$3.89e+00(1.40e-01)$	$7.04e+00(1.38e+01)-$
F_5	$7.15e+00(1.60e+00)$	$4.55e+00(1.68e+00)+$	$1.64e+01(6.01e+00)$	$1.09e+01(5.18e+00)+$	$2.73e+01(3.74e+00)$	$1.92e+01(8.85e+00)+$
F_6	$2.13e-05(6.60e-06)$	$2.54e-07(1.62e-06)+$	$9.78e-02(4.08e-01)$	$1.07e-02(7.61e-02)\approx$	$3.12e-06(6.35e-06)$	$5.76e-02(2.13e-01)-$
F_7	$1.94e+01(2.39e+00)$	$1.59e+01(3.56e+00)+$	$1.79e+01(3.87e+00)$	$1.90e+01(3.18e+00)\approx$	$3.82e+01(4.43e+00)$	$2.78e+01(8.51e+00)+$
F_8	$7.54e+00(2.01e+00)$	$4.19e+00(1.49e+00)+$	$1.27e+01(6.14e+00)$	$8.74e+00(2.87e+00)+$	$2.64e+01(3.85e+00)$	$1.77e+01(7.48e+00)+$
F_9	$7.37e-07(1.10e-06)$	$0.00e+00(0.00e+00)+$	$0.00e+00(0.00e+00)$	$0.00e+00(0.00e+00)\approx$	$0.00e+00(0.00e+00)$	$2.35e-01(6.46e-01)-$
F_{10}	$3.23e+02(1.00e+02)$	$1.48e+02(1.17e+02)+$	$5.42e+02(2.10e+02)$	$4.07e+02(2.36e+02)+$	$1.45e+03(1.18e+02)$	$8.40e+02(3.78e+02)+$
F_{11}	$3.29e+00(1.26e+00)$	$2.56e+00(1.81e+00)+$	$9.48e+00(9.19e+00)$	$5.91e+00(3.78e+00)+$	$6.87e+00(1.11e+00)$	$9.19e+00(5.24e+00)\approx$
F_{12}	$2.88e+04(4.39e+04)$	$1.35e+04(1.33e+04)+$	$1.56e+04(1.36e+04)$	$1.43e+04(1.28e+04)\approx$	$2.52e+06(1.29e+06)$	$2.58e+04(3.06e+04)+$
F_{13}	$2.41e+02(3.52e+02)$	$8.37e+00(3.51e+00)+$	$5.84e+03(6.44e+03)$	$5.94e+03(6.08e+03)\approx$	$1.12e+04(3.03e+03)$	$1.07e+04(1.62e+04)+$
F_{14}	$1.63e+00(9.68e-01)$	$1.61e+00(1.18e+00)\approx$	$6.43e+01(3.50e+01)$	$5.14e+01(2.14e+01)\approx$	$7.21e+02(4.77e+02)$	$4.58e+01(2.66e+01)+$
F_{15}	$1.14e+00(6.25e-01)$	$7.76e-01(7.50e-01)+$	$4.87e+01(5.87e+01)$	$3.49e+01(3.12e+01)\approx$	$3.62e+03(1.90e+03)$	$6.33e+02(9.63e+02)+$
F_{16}	$2.02e+00(3.13e+00)$	$2.08e+00(3.07e+00)\approx$	$2.44e+02(1.28e+02)$	$2.32e+02(1.44e+02)\approx$	$3.97e+01(1.83e+01)$	$7.09e+01(6.97e+01)\approx$
F_{17}	$2.01e+00(3.28e+00)$	$1.43e+00(2.51e+00)+$	$5.24e+01(3.43e+01)$	$4.24e+01(3.27e+01)\approx$	$4.61e+01(6.56e+00)$	$4.56e+01(2.00e+01)+$
F_{18}	$4.34e+00(3.03e+00)$	$5.46e+00(1.04e+01)-$	$1.11e+04(1.06e+04)$	$1.06e+04(1.04e+04)\approx$	$6.69e+04(4.69e+04)$	$6.70e+03(1.54e+04)+$
F_{19}	$2.97e-01(3.63e-01)$	$1.13e+01(1.55e+01)-$	$1.16e+02(2.53e+02)$	$3.55e+02(1.82e+03)-$	$9.93e+02(5.90e+02)$	$6.61e+02(1.22e+03)+$
F_{20}	$9.94e-02(2.20e-01)$	$2.09e-01(3.85e-01)-$	$7.02e+01(5.92e+01)$	$8.26e+01(6.57e+01)\approx$	$2.41e+01(2.07e+00)$	$2.60e+01(7.60e+00)\approx$
F_{21}	$1.07e+02(8.82e+00)$	$9.90e+01(2.50e+01)+$	$1.97e+02(4.53e+01)$	$1.85e+02(5.00e+01)+$	$1.65e+02(2.11e+01)$	$2.07e+02(4.15e+01)-$
F_{22}	$6.56e+01(3.49e+01)$	$7.24e+01(4.16e+01)\approx$	$1.02e+02(1.87e+01)$	$1.15e+02(9.84e+01)\approx$	$1.03e+02(1.45e+00)$	$2.03e+02(3.10e+02)\approx$
F_{23}	$3.11e+02(2.92e+00)$	$3.02e+02(4.30e+01)+$	$3.26e+02(1.31e+01)$	$3.18e+02(7.45e+00)+$	$3.30e+02(4.13e+00)$	$3.25e+02(1.09e+01)+$
F_{24}	$1.20e+02(4.35e+01)$	$1.26e+02(7.37e+01)-$	$3.03e+02(1.02e+02)$	$3.21e+02(8.19e+01)\approx$	$3.53e+02(2.08e+01)$	$3.57e+02(9.11e+00)\approx$
F_{25}	$4.01e+02(8.14e+00)$	$4.19e+02(2.30e+01)-$	$4.22e+02(5.28e+01)$	$4.05e+02(6.63e+01)\approx$	$4.35e+02(1.18e+01)$	$4.28e+02(2.50e+01)\approx$
F_{26}	$2.28e+02(1.17e+02)$	$2.75e+02(8.49e+01)-$	$4.60e+02(3.60e+02)$	$3.22e+02(2.39e+02)+$	$4.89e+02(8.49e+01)$	$5.40e+02(1.51e+02)\approx$
F_{27}	$3.92e+02(2.37e+00)$	$3.93e+02(2.77e+00)\approx$	$4.10e+02(2.63e+01)$	$4.02e+02(2.50e+01)+$	$4.57e+02(3.24e+01)$	$4.77e+02(3.40e+01)-$
F_{28}	$2.66e+02(9.05e+01)$	$2.62e+02(9.80e+01)+$	$4.80e+02(1.43e+02)$	$4.64e+02(1.44e+02)\approx$	$4.92e+02(6.59e+00)$	$4.87e+02(1.27e+01)\approx$
F_{29}	$2.63e+02(8.87e+00)$	$2.51e+02(7.32e+00)+$	$3.02e+02(5.65e+01)$	$2.84e+02(3.54e+01)\approx$	$3.42e+02(2.20e+01)$	$3.32e+02(5.85e+01)\approx$
F_{30}	$3.57e+04(4.78e+04)$	$3.83e+03(4.29e+03)+$	$5.06e+05(8.52e+05)$	$2.81e+05(5.13e+05)\approx$	$7.15e+02(3.50e+02)$	$1.95e+03(2.50e+03)-$
Chaos wins	+	19	+	8	+	14
Original wins	−	6	−	2	−	7
Similar	≈	5	≈	20	≈	9

Table 5: Comparison of algorithms (GWO, WOA, and ACOR original vs chaos version) on the CEC 2017 benchmark suite (10 dimensions, 51 runs, full precision, WRT at 5%).

CEC 2017	GWO		WOA		ACOR	
Function	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)
F_1	$3.08e+08(3.98e+08)$	$9.42e+07(1.68e+08)+$	$3.93e+05(8.14e+05)$	$1.88e+05(2.73e+05) \approx$	$1.90e+03(2.31e+03)$	$7.85e+02(1.18e+03)+$
F_2	$1.52e+09(3.01e+09)$	$3.75e+07(2.31e+08)+$	$2.10e+04(3.19e+04)$	$1.37e+04(3.25e+04) \approx$	$7.96e+05(5.03e+06)$	$0.00e+00(0.00e+00)+$
F_3	$1.78e+04(9.02e+03)$	$1.14e+03(1.87e+03)+$	$5.14e+02(7.52e+02)$	$2.67e+02(3.37e+02) \approx$	$8.02e+02(1.34e+03)$	$0.00e+00(0.00e+00)+$
F_4	$3.67e+01(2.39e+01)$	$1.47e+01(1.79e+01)+$	$1.78e+01(2.80e+01)$	$2.08e+01(3.35e+01) \approx$	$2.16e+00(1.38e-01)$	$2.88e+00(3.95e-01)-$
F_5	$3.63e+01(1.45e+01)$	$1.65e+01(8.30e+00)+$	$5.34e+01(1.78e+01)$	$4.52e+01(1.67e+01)+$	$2.86e+01(4.64e+00)$	$2.01e+01(6.40e+00)+$
F_6	$7.76e+00(7.55e+00)$	$1.14e+00(1.25e+00)+$	$3.05e+01(1.30e+01)$	$2.93e+01(1.28e+01) \approx$	$0.00e+00(0.00e+00)$	$2.22e-07(1.59e-06) \approx$
F_7	$5.68e+01(1.62e+01)$	$3.31e+01(1.02e+01)+$	$7.78e+01(2.21e+01)$	$8.02e+01(2.47e+01) \approx$	$3.81e+01(4.78e+00)$	$3.10e+01(3.31e+00)+$
F_8	$3.14e+01(1.56e+01)$	$1.40e+01(6.49e+00)+$	$3.98e+01(1.24e+01)$	$3.93e+01(1.60e+01) \approx$	$2.88e+01(4.62e+00)$	$2.02e+01(4.74e+00)+$
F_9	$1.18e+02(2.36e+02)$	$1.62e+01(3.90e+01)+$	$4.13e+02(2.81e+02)$	$4.71e+02(3.46e+02) \approx$	$0.00e+00(0.00e+00)$	$8.91e-03(6.36e-02) \approx$
F_{10}	$1.46e+03(5.83e+02)$	$4.97e+02(2.82e+02)+$	$1.02e+03(3.31e+02)$	$9.35e+02(3.55e+02) \approx$	$1.70e+03(1.18e+02)$	$1.29e+03(1.70e+02)+$
F_{11}	$3.90e+02(1.05e+03)$	$3.08e+01(2.11e+01)+$	$1.09e+02(9.81e+01)$	$9.69e+01(7.19e+01) \approx$	$6.55e+00(1.51e+00)$	$7.61e-01(7.11e-01)+$
F_{12}	$3.05e+06(3.51e+06)$	$4.21e+05(6.65e+05)+$	$3.83e+06(5.03e+06)$	$3.88e+06(5.00e+06) \approx$	$2.79e+04(8.38e+04)$	$7.38e+03(3.64e+03)+$
F_{13}	$2.01e+04(1.42e+04)$	$1.14e+04(8.40e+03)+$	$1.43e+04(1.26e+04)$	$1.44e+04(1.17e+04) \approx$	$8.92e+03(7.91e+03)$	$1.10e+04(5.52e+03)-$
F_{14}	$6.26e+03(6.96e+03)$	$5.27e+02(1.11e+03)+$	$4.55e+02(9.68e+02)$	$2.25e+02(4.04e+02)+$	$8.92e+02(8.66e+02)$	$3.26e+03(2.67e+03)-$
F_{15}	$1.91e+04(1.74e+04)$	$1.30e+03(1.43e+03)+$	$2.39e+03(2.08e+03)$	$2.59e+03(2.86e+03) \approx$	$6.19e+03(3.33e+03)$	$1.54e+03(1.78e+03)+$
F_{16}	$2.47e+02(1.39e+02)$	$1.41e+02(1.29e+02)+$	$2.65e+02(1.36e+02)$	$2.18e+02(1.28e+02) \approx$	$3.93e+01(2.33e+01)$	$3.57e+00(8.41e+00)+$
F_{17}	$1.34e+02(6.42e+01)$	$4.84e+01(1.58e+01)+$	$8.46e+01(3.69e+01)$	$9.54e+01(4.75e+01) \approx$	$4.24e+01(1.07e+01)$	$3.89e+01(1.14e+01)+$
F_{18}	$7.03e+04(4.89e+04)$	$2.97e+04(1.42e+04)+$	$1.69e+04(1.21e+04)$	$1.27e+04(1.23e+04)+$	$1.03e+05(7.94e+04)$	$6.94e+03(6.31e+03)+$
F_{19}	$6.53e+04(1.33e+05)$	$3.01e+03(4.75e+03)+$	$1.75e+04(3.92e+04)$	$1.60e+04(2.31e+04) \approx$	$2.60e+03(2.64e+03)$	$4.17e+03(3.27e+03)-$
F_{20}	$1.87e+02(9.43e+01)$	$6.59e+01(4.30e+01)+$	$1.55e+02(7.46e+01)$	$1.43e+02(6.95e+01) \approx$	$2.04e+01(2.11e-01)$	$2.29e+01(5.39e+00) \approx$
F_{21}	$2.35e+02(2.72e+01)$	$2.08e+02(3.17e+01)+$	$2.24e+02(5.63e+01)$	$2.06e+02(6.51e+01) \approx$	$2.29e+02(8.25e+00)$	$2.23e+02(5.58e+00)+$
F_{22}	$1.53e+02(2.13e+02)$	$1.17e+02(2.54e+01)+$	$1.31e+02(1.24e+02)$	$1.33e+02(1.34e+02) \approx$	$2.06e+02(3.70e+02)$	$1.00e+02(2.52e-01)+$
F_{23}	$3.49e+02(1.81e+01)$	$3.20e+02(9.43e+00)+$	$3.49e+02(2.00e+01)$	$3.38e+02(1.91e+01)+$	$3.33e+02(4.79e+00)$	$3.17e+02(8.42e+00)+$
F_{24}	$3.71e+02(3.72e+01)$	$3.45e+02(1.27e+01)+$	$3.54e+02(8.24e+01)$	$3.64e+02(4.05e+01) \approx$	$3.60e+02(4.26e+00)$	$3.50e+02(6.18e+00)+$
F_{25}	$4.51e+02(1.81e+01)$	$4.29e+02(1.46e+01)+$	$4.36e+02(5.26e+01)$	$4.41e+02(2.96e+01) \approx$	$4.42e+02(1.31e+01)$	$4.46e+02(7.01e+00)-$
F_{26}	$7.88e+02(4.17e+02)$	$4.81e+02(3.43e+02)+$	$8.87e+02(5.71e+02)$	$6.88e+02(5.28e+02)+$	$6.92e+02(6.70e+01)$	$5.03e+02(1.46e+02)+$
F_{27}	$4.20e+02(2.81e+01)$	$3.94e+02(1.06e+01)+$	$4.34e+02(3.74e+01)$	$4.10e+02(2.32e+01)+$	$4.81e+02(2.15e+01)$	$4.09e+02(4.83e+01)+$
F_{28}	$6.40e+02(1.23e+02)$	$6.00e+02(4.58e+01)+$	$6.36e+02(1.68e+02)$	$5.93e+02(6.37e+01)+$	$4.96e+02(8.08e+00)$	$4.73e+02(3.75e+00)+$
F_{29}	$3.75e+02(6.51e+01)$	$2.88e+02(3.91e+01)+$	$4.27e+02(7.80e+01)$	$3.97e+02(8.64e+01) \approx$	$3.45e+02(4.14e+01)$	$2.73e+02(9.72e+00)+$
F_{30}	$4.07e+06(5.31e+06)$	$7.01e+05(7.75e+05)+$	$5.34e+05(9.98e+05)$	$2.54e+05(4.20e+05) \approx$	$1.96e+03(1.57e+03)$	$2.47e+02(2.29e+01)+$
Chaos wins	+	30	+	7	+	22
Original wins	−	0	−	0	−	5
Similar	≈	0	≈	23	≈	3

Table 6: Comparison of algorithms (FA, PSO, and CA original vs chaos version) on the CEC 2017 benchmark suite (10 dimensions, 51 runs, full precision, WRT at 5%).

CEC 2017	FA		PSO		CA	
Function	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)
F_1	$3.85e+03(4.27e+03)$	$4.19e+03(3.63e+03) \approx$	$1.79e+03(1.66e+03)$	$2.41e+03(2.59e+03) \approx$	$4.36e+03(6.32e+03)$	$1.45e+07(2.48e+07)-$
F_2	$3.92e-02(1.96e-01)$	$0.00e+00(0.00e+00) \approx$	$0.00e+00(0.00e+00)$	$0.00e+00(0.00e+00) \approx$	$9.71e+83(8.71e+68)$	$4.13e+63(2.95e+64)+$
F_3	$1.24e-05(3.27e-06)$	$9.16e-04(2.57e-04)-$	$0.00e+00(0.00e+00)$	$0.00e+00(0.00e+00) \approx$	$4.85e+04(2.38e+04)$	$3.29e+04(1.03e+04)+$
F_4	$2.78e+00(1.12e+00)$	$1.92e+00(3.35e-01)+$	$2.64e+00(9.39e+00)$	$2.84e+00(7.52e-01)-$	$5.93e+00(2.06e-01)$	$8.53e+00(8.19e-01)-$
F_5	$1.74e+01(8.18e+00)$	$8.99e+00(3.90e+00)+$	$1.69e+01(7.95e+00)$	$1.49e+01(7.50e+00) \approx$	$4.23e+01(5.02e+00)$	$1.27e+01(6.22e+00)+$
F_6	$1.58e-03(2.38e-04)$	$1.40e-02(1.89e-03)-$	$3.83e-01(8.88e-01)$	$1.83e-01(4.12e-01)+$	$6.86e+01(8.81e+00)$	$1.74e-06(7.56e-06)+$
F_7	$1.73e+01(3.01e+00)$	$1.71e+01(2.96e+00) \approx$	$1.91e+01(4.29e+00)$	$1.72e+01(4.05e+00)+$	$5.21e+01(5.23e+00)$	$2.41e+01(6.61e+00)+$
F_8	$1.32e+01(5.13e+00)$	$7.24e+00(3.01e+00)+$	$1.29e+01(5.18e+00)$	$1.03e+01(3.77e+00)+$	$4.32e+01(6.31e+00)$	$1.16e+01(6.01e+00)+$
F_9	$2.67e-06(7.27e-07)$	$2.07e-04(5.38e-05)-$	$8.91e-03(6.36e-02)$	$0.00e+00(0.00e+00) \approx$	$9.31e+00(1.93e+01)$	$1.92e-02(4.83e-02)+$
F_{10}	$6.07e+02(3.30e+02)$	$2.36e+02(1.67e+02)+$	$6.55e+02(2.53e+02)$	$5.48e+02(2.30e+02)+$	$2.09e+03(1.93e+02)$	$6.73e+02(3.63e+02)+$
F_{11}	$5.95e+00(3.28e+00)$	$4.61e+00(2.77e+00)+$	$1.76e+01(1.09e+01)$	$1.09e+01(6.78e+00)+$	$3.33e+03(2.33e+03)$	$7.96e+02(1.09e+03)+$
F_{12}	$1.34e+04(1.18e+04)$	$1.65e+04(1.62e+04) \approx$	$9.25e+03(7.31e+03)$	$1.40e+04(1.01e+04)-$	$3.23e+08(1.67e+08)$	$5.72e+05(9.68e+05)+$
F_{13}	$4.20e+03(5.32e+03)$	$4.06e+03(5.03e+03) \approx$	$6.56e+03(5.15e+03)$	$5.55e+03(4.62e+03) \approx$	$1.28e+06(1.38e+06)$	$1.59e+04(5.38e+03)+$
F_{14}	$4.95e+01(8.45e+01)$	$2.52e+01(1.15e+01)+$	$5.48e+01(2.17e+01)$	$5.55e+01(2.27e+01) \approx$	$8.22e+03(8.10e+03)$	$2.43e+03(1.50e+03)+$
F_{15}	$8.59e+01(3.31e+02)$	$2.76e+01(2.37e+01) \approx$	$6.18e+01(4.84e+01)$	$4.06e+01(2.43e+01)+$	$5.27e+04(4.49e+04)$	$3.75e+03(3.19e+03)+$
F_{16}	$7.22e+01(7.74e+01)$	$1.37e+01(1.81e+01)+$	$2.09e+02(1.30e+02)$	$1.69e+02(1.40e+02)+$	$2.14e+02(5.76e+01)$	$1.36e+01(1.72e+01)+$
F_{17}	$2.71e+01(1.88e+01)$	$2.61e+01(1.15e+01) \approx$	$5.27e+01(3.07e+01)$	$4.73e+01(3.25e+01) \approx$	$9.29e+01(2.01e+01)$	$4.38e+01(1.47e+01)+$
F_{18}	$7.66e+03(7.19e+03)$	$1.07e+04(9.04e+03) \approx$	$4.18e+03(6.40e+03)$	$4.01e+03(6.06e+03) \approx$	$8.54e+06(1.06e+07)$	$1.33e+04(1.40e+04)+$
F_{19}	$2.45e+02(7.24e+02)$	$1.65e+01(1.75e+01)+$	$2.13e+02(5.35e+02)$	$1.01e+02(1.90e+02) \approx$	$5.76e+04(9.30e+04)$	$4.91e+03(3.56e+03)+$
F_{20}	$1.62e+01(1.57e+01)$	$1.11e+01(2.14e+01)+$	$6.79e+01(5.54e+01)$	$6.28e+01(5.62e+01) \approx$	$1.88e+02(3.46e+01)$	$5.89e+01(1.27e+01)+$
F_{21}	$1.72e+02(5.55e+01)$	$1.65e+02(5.85e+01) \approx$	$1.77e+02(5.51e+01)$	$1.70e+02(5.65e+01) \approx$	$2.38e+02(1.99e+01)$	$2.09e+02(1.79e+01)+$
F_{22}	$9.79e+01(1.59e+01)$	$9.00e+01(3.08e+01) \approx$	$1.02e+02(8.75e-01)$	$1.12e+02(9.49e+01) \approx$	$1.41e+03(4.13e+02)$	$1.14e+02(1.02e+02)+$
F_{23}	$3.18e+02(6.82e+00)$	$3.10e+02(3.26e+00)+$	$3.20e+02(8.10e+00)$	$3.17e+02(9.27e+00)+$	$3.64e+02(9.14e+00)$	$3.16e+02(5.58e+00)+$
F_{24}	$2.98e+02(9.91e+01)$	$3.40e+02(3.78e+00)-$	$2.94e+02(9.88e+01)$	$2.84e+02(1.03e+02) \approx$	$3.79e+02(4.82e+00)$	$3.49e+02(7.08e+00)+$
F_{25}	$4.22e+02(2.32e+01)$	$4.27e+02(2.33e+01) \approx$	$4.21e+02(2.33e+01)$	$4.20e+02(2.36e+01) \approx$	$4.58e+02(4.97e+00)$	$4.46e+02(6.92e+00)+$
F_{26}	$3.30e+02(6.13e+01)$	$3.00e+02(3.83e-03) \approx$	$2.87e+02(3.95e+01)$	$3.26e+02(2.07e+02) \approx$	$9.11e+02(6.73e+01)$	$5.28e+02(1.04e+02)+$
F_{27}	$3.96e+02(3.87e+00)$	$3.90e+02(7.33e-01)+$	$4.07e+02(2.78e+01)$	$4.01e+02(1.88e+01) \approx$	$4.74e+02(2.32e+01)$	$4.45e+02(2.14e+01)+$
F_{28}	$4.14e+02(1.38e+02)$	$4.26e+02(1.52e+02)-$	$4.53e+02(1.47e+02)$	$4.71e+02(1.48e+02) \approx$	$4.98e+02(2.65e+00)$	$4.98e+02(1.66e+00) \approx$
F_{29}	$2.77e+02(3.22e+01)$	$2.42e+02(8.48e+00)+$	$3.05e+02(3.84e+01)$	$2.87e+02(3.84e+01)+$	$6.81e+02(1.28e+02)$	$4.65e+02(1.51e+02)+$
F_{30}	$1.88e+05(3.60e+05)$	$2.67e+05(3.99e+05) \approx$	$8.43e+04(2.44e+05)$	$1.19e+05(3.19e+05) \approx$	$7.97e+05(8.59e+05)$	$2.95e+03(2.68e+03)+$
Chaos wins	+	12	+	9	+	27
Original wins	−	5	−	2	−	2
Similar	≈	13	≈	19	≈	1

Table 7: Comparison of algorithms (SOMA ATO, HFPSO, and ABC original vs chaos version) on the CEC 2020 benchmark suite (10 dimensions, 51 runs, full precision, WRT at 5%).

CEC 2020	SOMA ATO		HFPSO		ABC	
Function	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)
F_1	$1.31e+03(1.83e+03)$	$2.76e+01(4.58e+01)+$	$1.01e+03(1.12e+03)$	$2.49e+03(3.51e+03)\approx$	$9.33e+02(1.33e+03)$	$1.06e+04(1.71e+04)-$
F_2	$1.77e+02(8.09e+01)$	$6.93e+01(6.87e+01)+$	$3.06e+02(2.68e+02)$	$2.18e+02(1.24e+02)\approx$	$1.37e+03(1.59e+02)$	$7.29e+02(2.96e+02)+$
F_3	$1.97e+01(2.53e+00)$	$1.67e+01(2.94e+00)+$	$1.46e+01(6.65e-01)$	$1.99e+01(4.60e+00)-$	$3.71e+01(3.68e+00)$	$2.76e+01(7.80e+00)+$
F_4	$1.50e+00(3.16e-01)$	$9.70e-01(3.27e-01)+$	$6.79e-01(1.35e-01)$	$1.56e+00(3.67e+00)-$	$1.83e+00(3.58e-01)$	$1.52e+00(6.22e-01)+$
F_5	$1.90e+04(2.40e+04)$	$5.87e+02(1.01e+03)+$	$3.39e+03(2.44e+03)$	$2.46e+03(2.64e+03)\approx$	$5.13e+04(2.65e+04)$	$1.20e+04(1.63e+04)+$
F_6	$4.14e+00(1.68e+01)$	$1.72e+00(2.98e+00)\approx$	$1.24e+02(1.60e+02)$	$1.65e+02(8.01e+01)-$	$7.50e+01(2.90e+01)$	$1.12e+02(8.06e+01)-$
F_7	$7.59e+02(1.36e+03)$	$5.48e+00(1.10e+01)+$	$2.22e+02(2.52e+02)$	$1.39e+02(1.48e+02)\approx$	$1.45e+04(7.35e+03)$	$6.00e+02(6.49e+02)+$
F_8	$7.69e+01(3.42e+01)$	$7.70e+01(3.73e+01)\approx$	$1.01e+02(1.53e-01)$	$1.23e+02(1.27e+02)-$	$1.03e+02(1.19e+00)$	$1.74e+02(1.80e+02)\approx$
F_9	$1.24e+02(3.74e+01)$	$1.31e+02(7.19e+01)-$	$3.54e+02(9.20e+00)$	$3.29e+02(6.83e+01)+$	$3.45e+02(2.85e+01)$	$3.58e+02(1.48e+01)\approx$
F_{10}	$4.01e+02(1.10e+01)$	$4.14e+02(2.19e+01)\approx$	$4.20e+02(2.33e+01)$	$4.26e+02(2.36e+01)-$	$4.30e+02(1.37e+01)$	$4.21e+02(3.94e+01)\approx$
Chaos wins	+	6	+	1	+	5
Original wins	−	1	−	5	−	2
Similar	≈	3	≈	4	≈	3

Table 8: Comparison of algorithms (GWO, WOA, and ACOR original vs chaos version) on the CEC 2020 benchmark suite (10 dimensions, 51 runs, full precision, WRT at 5%).

CEC 2020	GWO		WOA		ACOR	
Function	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)
F_1	$2.12e+07(9.72e+07)$	$2.70e+07(9.83e+07)-$	$3.34e+05(7.26e+05)$	$2.14e+05(3.40e+05)\approx$	$8.27e+02(9.97e+02)$	$1.16e+03(1.91e+03)\approx$
F_2	$3.82e+02(2.63e+02)$	$4.48e+02(2.68e+02)\approx$	$9.65e+02(2.77e+02)$	$9.69e+02(2.56e+02)\approx$	$1.60e+03(1.86e+02)$	$1.26e+03(2.01e+02)+$
F_3	$2.83e+01(9.54e+00)$	$2.91e+01(8.09e+00)\approx$	$8.24e+01(2.49e+01)$	$7.28e+01(2.21e+01)\approx$	$4.09e+01(2.05e+00)$	$2.97e+01(3.78e+00)+$
F_4	$3.29e+00(1.07e+01)$	$1.76e+00(9.01e-01)\approx$	$6.23e+00(4.34e+00)$	$5.17e+00(2.95e+00)\approx$	$2.62e+00(2.10e-01)$	$1.74e+00(3.52e-01)+$
F_5	$4.50e+04(1.12e+05)$	$5.59e+03(4.72e+03)\approx$	$1.39e+05(2.82e+05)$	$8.35e+04(1.33e+05)\approx$	$2.27e+04(1.48e+04)$	$6.60e+04(5.87e+04)-$
F_6	$1.32e+02(9.58e+01)$	$1.38e+02(7.97e+01)\approx$	$2.00e+02(1.01e+02)$	$1.74e+02(8.80e+01)\approx$	$5.97e+01(3.59e+01)$	$7.33e+00(1.79e+01)+$
F_7	$5.49e+03(4.61e+03)$	$4.30e+03(3.85e+03)\approx$	$1.80e+04(1.39e+04)$	$1.70e+04(1.43e+04)\approx$	$3.58e+03(1.19e+03)$	$1.15e+04(8.56e+03)-$
F_8	$1.21e+02(5.99e+01)$	$1.26e+02(7.78e+01)-$	$1.74e+02(2.35e+02)$	$1.78e+02(2.65e+02)\approx$	$1.04e+02(6.33e-01)$	$1.00e+02(2.83e-01)+$
F_9	$3.45e+02(1.03e+01)$	$3.47e+02(1.19e+01)\approx$	$3.72e+02(4.53e+01)$	$3.66e+02(4.19e+01)\approx$	$3.64e+02(5.84e+00)$	$3.51e+02(5.62e+00)+$
F_{10}	$4.30e+02(1.83e+01)$	$4.35e+02(1.85e+01)\approx$	$4.26e+02(6.44e+01)$	$4.33e+02(6.40e+01)\approx$	$4.24e+02(2.42e+01)$	$4.44e+02(1.17e+01)-$
Chaos wins	+	0	+	0	+	6
Original wins	−	2	−	0	−	3
Similar	≈	8	≈	10	≈	1

Table 9: Comparison of algorithms FA, PSO, and CA original vs chaos version) on the CEC 2020 benchmark suite (10 dimensions, 51 runs, full precision, WRT at 5%).

CEC 2020	FA		PSO		CA	
Function	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)
F_1	$3.58e+03(3.91e+03)$	$6.80e+03(3.93e+03)-$	$2.49e+03(3.16e+03)$	$2.35e+03(2.87e+03)\approx$	$3.53e+03(2.46e+03)$	$4.40e+07(1.47e+08)-$
F_2	$5.44e+02(2.90e+02)$	$2.40e+02(1.45e+02)+$	$4.22e+02(2.61e+02)$	$3.94e+02(2.05e+02)\approx$	$2.12e+03(1.19e+02)$	$6.01e+02(2.67e+02)+$
F_3	$1.79e+01(4.43e+00)$	$1.66e+01(3.70e+00)+$	$1.97e+01(4.86e+00)$	$1.79e+01(3.53e+00)+$	$5.46e+01(1.96e+00)$	$2.35e+01(5.83e+00)+$
F_4	$8.36e-01(3.31e-01)$	$8.38e-01(2.58e-01)\approx$	$9.47e-01(3.22e-01)$	$9.37e-01(3.28e-01)\approx$	$3.15e+00(5.54e-01)$	$9.37e-01(5.22e-01)+$
F_5	$2.40e+03(2.76e+03)$	$4.31e+03(4.26e+03)\approx$	$2.30e+03(1.98e+03)$	$2.26e+03(2.14e+03)\approx$	$2.91e+06(9.08e+05)$	$4.63e+05(3.95e+05)+$
F_6	$1.09e+02(1.01e+02)$	$4.98e+01(6.21e+01)+$	$1.48e+02(9.08e+01)$	$1.57e+02(8.34e+01)\approx$	$7.32e+02(9.80e+01)$	$1.96e+01(9.12e+00)+$
F_7	$1.71e+02(1.12e+02)$	$1.60e+02(1.39e+02)\approx$	$2.25e+02(1.62e+02)$	$1.50e+02(1.20e+02)+$	$3.46e+05(2.32e+05)$	$8.16e+04(7.08e+04)+$
F_8	$9.77e+01(1.62e+01)$	$9.35e+01(2.67e+01)\approx$	$9.86e+01(1.55e+01)$	$9.30e+01(2.79e+01)+$	$1.35e+03(4.62e+02)$	$1.08e+02(5.44e+01)+$
F_9	$3.23e+02(7.45e+01)$	$3.08e+02(8.41e+01)+$	$3.06e+02(9.02e+01)$	$2.88e+02(1.02e+02)\approx$	$3.81e+02(2.63e+00)$	$3.50e+02(4.30e+00)+$
F_{10}	$4.29e+02(2.29e+01)$	$4.25e+02(2.32e+01)\approx$	$4.14e+02(2.20e+01)$	$4.29e+02(2.22e+01)-$	$4.59e+02(4.25e+00)$	$4.47e+02(6.81e+00)+$
Chaos wins	+	4	+	3	+	9
Original wins	−	1	−	1	−	1
Similar	≈	5	≈	6	≈	0

Table 10: Comparison of algorithms (SOMA ATO, HFPSO, and ABC original vs chaos version) on the CEC 2015 benchmark suite (10 dimensions, 51 runs, full precision, WRT at 5%).

CEC 2015	SOMA ATO		HFPSO		ABC	
Function	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)
F_1	$1.90e+05(1.65e+05)$	$5.97e+04(5.64e+04)+$	$1.13e+04(8.36e+03)$	$4.83e+04(3.95e+04)-$	$7.03e+06(2.25e+06)$	$3.26e+06(5.16e+06)+$
F_2	$6.85e+03(7.10e+03)$	$4.52e+02(8.12e+02)+$	$6.25e+03(7.88e+03)$	$6.40e+03(7.74e+03)\approx$	$4.89e+03(4.05e+03)$	$1.85e+04(2.28e+04)-$
F_3	$2.01e+01(3.55e-02)$	$2.01e+01(2.75e-02)-$	$2.00e+01(2.05e-03)$	$2.00e+01(1.04e-02)-$	$2.03e+01(6.62e-02)$	$2.03e+01(8.17e-02)\approx$
F_4	$6.65e+00(1.85e+00)$	$5.32e+00(1.82e+00)+$	$1.45e+01(5.35e+00)$	$8.63e+00(3.67e+00)+$	$2.77e+01(3.79e+00)$	$1.77e+01(7.87e+00)+$
F_5	$2.34e+02(1.08e+02)$	$2.01e+02(1.19e+02)\approx$	$3.68e+02(1.93e+02)$	$2.82e+02(1.47e+02)\approx$	$1.41e+03(1.46e+02)$	$8.90e+02(3.75e+02)+$
F_6	$1.68e+03(1.73e+03)$	$1.95e+02(2.21e+02)+$	$2.83e+03(2.58e+03)$	$2.39e+03(2.81e+03)\approx$	$2.67e+04(1.48e+04)$	$6.33e+03(9.32e+03)+$
F_7	$4.43e-01(2.65e-01)$	$4.18e-01(3.68e-01)+$	$1.65e+00(1.00e+00)$	$1.46e+00(9.74e-01)\approx$	$2.52e+00(1.88e-01)$	$2.61e+00(6.50e-01)\approx$
F_8	$2.17e+03(3.50e+03)$	$6.93e+01(7.73e+01)+$	$1.40e+03(1.46e+03)$	$8.89e+02(1.02e+03)\approx$	$9.31e+03(5.80e+03)$	$7.12e+03(7.79e+03)+$
F_9	$1.00e+02(3.75e-02)$	$1.00e+02(3.94e-02)+$	$1.03e+02(1.67e+01)$	$1.00e+02(3.65e-01)\approx$	$1.00e+02(4.34e-02)$	$1.00e+02(6.12e-02)\approx$
F_{10}	$1.26e+03(1.40e+03)$	$3.68e+02(3.31e+02)+$	$9.87e+02(8.14e+02)$	$7.50e+02(6.28e+02)+$	$1.51e+04(6.93e+03)$	$6.42e+03(8.93e+03)+$
F_{11}	$6.47e+01(1.15e+02)$	$1.32e+02(1.48e+02)\approx$	$2.87e+02(7.52e+01)$	$2.78e+02(1.12e+02)\approx$	$3.31e+02(2.91e+01)$	$3.93e+02(9.57e+01)-$
F_{12}	$1.03e+02(4.90e-01)$	$1.02e+02(4.85e-01)+$	$1.02e+02(8.11e-01)$	$1.02e+02(8.46e-01)\approx$	$1.04e+02(6.61e-01)$	$1.03e+02(1.08e+00)+$
F_{13}	$2.95e+01(1.56e+00)$	$2.72e+01(2.16e+00)+$	$3.98e+01(4.43e+00)$	$3.77e+01(5.66e+00)\approx$	$3.90e+01(1.38e+00)$	$3.58e+01(3.64e+00)+$
F_{14}	$2.40e+03(1.15e+03)$	$2.63e+03(9.34e+02)\approx$	$4.29e+03(3.73e+03)$	$4.42e+03(3.51e+03)\approx$	$3.71e+02(1.65e+01)$	$3.54e+02(5.64e+01)+$
F_{15}	$1.00e+02(9.37e-06)$	$1.00e+02(0.00e+00)+$	$1.00e+02(0.00e+00)$	$1.00e+02(0.00e+00)\approx$	$1.00e+02(0.00e+00)$	$1.00e+02(0.00e+00)\approx$
Chaos wins	+	11	+	2	+	9
Original wins	−	1	−	2	−	2
Similar	≈	3	≈	11	≈	4

Table 11: Comparison of algorithms (GWO, WOA, and ACOR original vs chaos version) on the CEC 2015 benchmark suite (10 dimensions, 51 runs, full precision, WRT at 5%).

CEC 2015	GWO		WOA		ACOR	
Function	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)
F_1	$3.41e+06(3.15e+06)$	$2.58e+06(2.78e+06)\approx$	$4.43e+06(2.47e+06)$	$4.05e+06(2.78e+06)\approx$	$5.73e+06(3.29e+06)$	$2.96e+05(9.64e+04)+$
F_2	$4.61e+06(5.40e+06)$	$6.01e+06(5.87e+06)\approx$	$4.86e+05(8.11e+05)$	$9.45e+04(7.30e+04)+$	$5.21e+03(5.60e+03)$	$6.51e+03(4.47e+03)-$
F_3	$2.04e+01(7.64e-02)$	$2.04e+01(7.07e-02)\approx$	$2.01e+01(1.07e-01)$	$2.01e+01(9.44e-02)\approx$	$2.04e+01(6.36e-02)$	$2.04e+01(8.54e-02)\approx$
F_4	$1.47e+01(6.48e+00)$	$1.40e+01(5.09e+00)\approx$	$3.98e+01(1.65e+01)$	$3.89e+01(1.43e+01)\approx$	$2.85e+01(3.64e+00)$	$1.84e+01(7.94e+00)+$
F_5	$4.70e+02(2.42e+02)$	$4.72e+02(2.91e+02)\approx$	$9.47e+02(2.89e+02)$	$9.04e+02(3.05e+02)\approx$	$1.64e+03(1.68e+02)$	$1.30e+03(1.57e+02)+$
F_6	$1.70e+04(2.02e+04)$	$1.36e+04(1.34e+04)+$	$2.66e+05(3.27e+05)$	$2.82e+05(3.38e+05)\approx$	$9.30e+03(1.18e+04)$	$5.84e+02(8.71e+02)+$
F_7	$2.43e+00(2.18e+00)$	$2.44e+00(1.09e+00)\approx$	$5.61e+00(1.38e+00)$	$5.12e+00(1.14e+00)\approx$	$3.19e+00(2.81e-01)$	$2.57e+00(3.10e-01)+$
F_8	$3.47e+04(2.33e+05)$	$2.02e+03(1.19e+03)\approx$	$5.63e+03(4.74e+03)$	$6.42e+03(5.31e+03)\approx$	$5.49e+03(3.93e+03)$	$6.27e+02(8.36e+02)+$
F_9	$1.00e+02(1.50e-01)$	$1.00e+02(1.29e-01)\approx$	$1.01e+02(2.35e-01)$	$1.01e+02(2.62e-01)\approx$	$1.00e+02(4.40e-02)$	$1.00e+02(4.43e-02)+$
F_{10}	$3.31e+03(4.49e+03)$	$3.03e+03(3.30e+03)\approx$	$8.15e+03(1.07e+04)$	$9.21e+03(8.12e+03)\approx$	$2.21e+04(1.21e+04)$	$1.45e+03(1.31e+03)+$
F_{11}	$3.00e+02(8.68e+01)$	$2.83e+02(1.34e+02)\approx$	$3.25e+02(1.05e+02)$	$3.03e+02(3.91e+01)\approx$	$5.69e+02(8.49e+01)$	$3.89e+02(1.19e+02)+$
F_{12}	$1.02e+02(7.92e-01)$	$1.02e+02(8.04e-01)\approx$	$1.08e+02(2.60e+00)$	$1.08e+02(3.29e+00)\approx$	$1.04e+02(8.60e+00)$	$1.02e+02(3.17e-01)+$
F_{13}	$3.15e+01(4.43e+00)$	$3.09e+01(3.93e+00)\approx$	$4.14e+01(3.84e+00)$	$4.00e+01(3.22e+00)+$	$4.15e+01(3.19e+00)$	$3.21e+01(3.51e+00)+$
F_{14}	$5.91e+03(2.48e+03)$	$5.81e+03(2.70e+03)\approx$	$6.46e+03(1.28e+03)$	$6.34e+03(1.47e+03)+$	$3.50e+02(1.08e+01)$	$7.27e+02(1.47e+03)-$
F_{15}	$1.07e+02(4.99e+00)$	$1.09e+02(5.45e+00)-$	$1.01e+02(5.91e-01)$	$1.00e+02(1.06e-01)+$	$1.00e+02(0.00e+00)$	$1.00e+02(0.00e+00)\approx$
Chaos wins	+	1	+	4	+	11
Original wins	−	1	−	0	−	2
Similar	≈	13	≈	11	≈	2

Table 12: Comparison of algorithms (FA, PSO, and CA original vs chaos version) on the CEC 2015 benchmark suite (10 dimensions, 51 runs, full precision, WRT at 5%).

CEC 2015	FA		PSO		CA	
Function	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)	Original Mean (Std Dev)	Chaos Mean (Std Dev)
F_1	$9.95e+04(1.16e+05)$	$1.61e+05(1.97e+05)\approx$	$1.06e+04(9.46e+03)$	$7.25e+04(4.59e+04)-$	$1.35e+08(7.17e+07)$	$5.87e+07(2.32e+07)+$
F_2	$6.09e+03(8.00e+03)$	$1.28e+04(1.39e+04)\approx$	$6.67e+03(7.16e+03)$	$6.75e+03(7.44e+03)\approx$	$1.01e+04(1.29e+04)$	$2.70e+07(9.68e+07)-$
F_3	$1.91e+01(4.83e+00)$	$1.97e+01(2.82e+00)-$	$2.00e+01(2.64e-02)$	$2.02e+01(1.35e-01)-$	$2.03e+01(7.46e-02)$	$2.03e+01(7.27e-02)\approx$
F_4	$1.31e+01(4.94e+00)$	$8.25e+00(4.14e+00)+$	$1.58e+01(7.76e+00)$	$1.18e+01(5.16e+00)+$	$4.18e+01(5.13e+00)$	$1.05e+01(5.89e+00)+$
F_5	$5.87e+02(2.89e+02)$	$2.45e+02(1.85e+02)+$	$5.27e+02(2.02e+02)$	$3.89e+02(2.12e+02)+$	$2.10e+03(1.46e+02)$	$6.71e+02(3.39e+02)+$
F_6	$2.59e+03(2.34e+03)$	$4.32e+03(3.16e+03)-$	$1.73e+03(2.06e+03)$	$1.20e+03(9.40e+02)\approx$	$1.74e+06(1.44e+06)$	$7.15e+04(9.78e+04)+$
F_7	$1.01e+00(4.68e-01)$	$1.21e+00(4.91e-01)-$	$2.07e+00(9.91e-01)$	$1.19e+00(8.60e-01)+$	$3.23e+00(2.44e-01)$	$2.34e+00(2.70e-01)+$
F_8	$1.64e+03(2.66e+03)$	$7.09e+02(8.50e+02)+$	$1.46e+03(1.73e+03)$	$1.15e+03(1.40e+03)\approx$	$3.30e+05(3.26e+05)$	$5.44e+04(8.06e+04)+$
F_9	$1.00e+02(4.98e-02)$	$1.00e+02(4.56e-02)\approx$	$1.00e+02(5.64e-02)$	$1.00e+02(3.96e-02)+$	$1.00e+02(5.28e-02)$	$1.00e+02(4.30e-02)-$
F_{10}	$1.21e+03(1.54e+03)$	$8.39e+02(1.15e+03)\approx$	$9.37e+02(7.30e+02)$	$7.52e+02(4.76e+02)\approx$	$4.51e+05(3.54e+05)$	$1.11e+05(1.42e+05)+$
F_{11}	$2.71e+02(8.93e+01)$	$2.83e+02(7.08e+01)\approx$	$2.77e+02(8.05e+01)$	$2.71e+02(8.92e+01)\approx$	$4.94e+02(9.63e+01)$	$4.09e+02(5.23e+01)+$
F_{12}	$1.02e+02(4.37e-01)$	$1.02e+02(3.67e-01)\approx$	$1.02e+02(7.13e-01)$	$1.02e+02(5.42e-01)\approx$	$1.40e+02(2.70e+01)$	$1.03e+02(5.54e+00)+$
F_{13}	$3.26e+01(4.32e+00)$	$2.95e+01(2.40e+00)+$	$3.50e+01(3.82e+00)$	$3.29e+01(4.44e+00)+$	$4.48e+01(7.75e-01)$	$3.83e+01(1.80e+00)+$
F_{14}	$5.12e+03(2.31e+03)$	$5.12e+03(2.10e+03)\approx$	$2.61e+03(2.69e+03)$	$2.99e+03(2.97e+03)\approx$	$2.87e+03(1.27e+03)$	$5.23e+02(6.05e+01)+$
F_{15}	$1.00e+02(2.88e-04)$	$1.00e+02(3.01e-03)-$	$1.00e+02(0.00e+00)$	$1.00e+02(0.00e+00)\approx$	$1.00e+02(0.00e+00)$	$1.01e+02(2.77e+00)-$
Chaos wins	+	4	+	5	+	11
Original wins	−	4	−	2	−	3
Similar	≈	7	≈	8	≈	1

algorithms and algorithms of swarm intelligence. In other words, whenever a random number was needed, a chaotic series number was used.

According to the results presented in the results section, it can be stated that deterministic chaos clearly improved the performance of the algorithms used, which was demonstrated by statistical testing. Thus, it can be said that the fact that purely random processes are probably not necessary within these algorithms has been demonstrated numerically. This finding is, after all, entirely in line with the fact that comes from the field of control theory, namely that random processes are an unwelcome destructive force in the case of dynamic system control, which is very rarely beneficial. In other words, a dynamic that is subject to a process and has feedback should not contain random processes unless it is directly the goal of control. In control theory itself, random processes are understood as white noise, and everything is done to erase/eliminate this component of behaviour from the behaviour of dynamic systems. Indeed, deliberate injections of (small) noise are normally used to trigger otherwise hidden dynamics, thus revealing the true nature of the system, or to prevent the control law from pathological failures induced by variables which stay at constant values for prolonged periods. This is typical of control techniques which need an accurate identification of the model (persistent excitability), and of systems which are prevented from operating when one or more variables are inside certain intervals (dead zone systems). However, when the classic control is in the process (autopilot, reactor control, traffic control, etc) then noise is no acceptable signal there. That is a point of view of this idea here. Thus, there is a discrepancy between what the evolutionary community still understands about the need for random processes in evolutionary algorithms themselves and how randomness is understood in the control of dynamic systems (we recall that both evolutionary and swarm algorithms can be understood as discrete dynamic systems).

So this raises a number of interesting questions that deserve further research. Questions such as how complex and how periodic a given numerical series must be for the algorithm to work if used instead of random series, whether there is a link between the chaotic mode and the nonlinearity of the test function on which the algorithm runs, what in this way, these algorithms process information (in terms of control theory - feedback systems that process their output information) about their previous activities N steps back and forth.

Based on the results obtained in these experiments, the repeatability and accuracy of which can not be doubted, it can be stated that this area, although previously visited by many researchers and performed basic experiments, still offers a wide range of interesting questions and areas for further research. He also points out that the evolutionary algorithm community should begin to understand these algorithms as dynamic systems, for which there is already a productive mathematical apparatus that deals with the control of such systems, their stability and many other aspects. We firmly believe that applying this pre-existing mathematical apparatus to evolutionary and swarm intelligence algorithms would help shed light on the many unanswered questions that exist in the algorithm community today and provide exciting and powerful mathematical tools for researching these algorithms.

Thus, our experiments have regularly shown that chaos as a generator of pseudorandom numbers, if we can call it that, certainly increases the performance of evolutionary algorithms. However, what is a striking fact, which contradicts the generally accepted dogma, is that chaos is not a random process and still increases the performance of the algorithm. We believe that it is an interesting research area that combines computer science in terms of algorithm study and theoretical cybernetics and control theory.

References

[1] I. Zelinka, S. Celikovský, H. Richter, G. Chen (Eds.), Evolutionary Algorithms and Chaotic Systems, Vol. 267 of Studies in Computational Intelligence, Springer, 2010.
URL http://dblp.uni-trier.de/db/series/sci/sci267.html

[2] R. Caponetto, L. Fortuna, S. Fazzino, M. Xibilia, Chaotic sequences to improve the performance of evolutionary algorithms, IEEE Transactions on Evolutionary Computation 7 (3) (2003) 289–304. doi:10.1109/TEVC.2003.810069.

[3] M. Pluhacek, R. Senkerik, D. Davendra, K. Zuzana, I. Zelinka, On the behavior and performance of chaos driven pso algorithm with inertia weight, Computers & Mathematics with Applications 66 (2) (2013) 122–134, nostradamus 2012. doi:https://doi.org/10.1016/j.camwa.2013.01.016.
URL https://www.sciencedirect.com/science/article/pii/S0898122113000333

[4] M. Pluháček, V. Budíková, R. Šenkeřík, Z. Oplatková, I. Zelinka, On the performance of enhanced pso algorithm with lozi chaotic map-an initial study, MENDEL 2012.

[5] R. Lozi, Emergence of randomness from chaos, International Journal of Bifurcation and Chaos 22 (02) (2012) 1250021. doi:10.1142/S0218127412500216.
URL https://doi.org/10.1142/S0218127412500216

[6] X.-Y. Wang, L. Yang, Design of pseudo-random bit generator based on chaotic maps, International Journal of Modern Physics B 26 (32) (2012) 1250208. doi:10.1142/S0217979212502086.
URL https://doi.org/10.1142/S0217979212502086

[7] Y. Sun, L. Zhang, X. Gu, A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems, Neurocomputing 98 (2012) 76–89, bio-inspired computing and applications (LSMS-ICSEE ' 2010). doi:https://doi.org/10.1016/j.neucom.2011.08.043.
URL https://www.sciencedirect.com/science/article/pii/S0925231212004298

[8] W.-C. Hong, Y. Dong, W. Y. Zhang, L.-Y. Chen, B. K. Panigrahi, Cyclic electric load forecasting by seasonal svr with chaotic genetic algorithm, International Journal of Electrical Power & Energy Systems 44 (1) (2013) 604–614. doi:https://doi.org/10.1016/j.ijepes.2012.08.010.
URL https://www.sciencedirect.com/science/article/pii/S0142061512004462

[9] R. Senkerik, D. Davendra, I. Zelinka, Z. Oplatkova, M. Pluhacek, Optimization of the batch reactor by means of chaos driven differential evolution, in: V. Snášel, A. Abraham, E. S. Corchado (Eds.), Soft Computing Models in Industrial and Environmental Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 93–102. doi:10.1007/978-3-642-32922-7_10.
URL https://link.springer.com/chapter/10.1007/978-3-642-32922-7_10

[10] D. Davendra, I. Zelinka, R. Senkerik, Chaos driven evolutionary algorithms for the task of pid control, Computers & Mathematics with Applications 60 (4) (2010) 1088–1104, pCO' 2010. doi:https://doi.org/10.1016/j.camwa.2010.03.066.
URL https://www.sciencedirect.com/science/article/pii/S0898122110002567

[11] K. Persohn, R. Povinelli, Analyzing logistic map pseudorandom number generators for periodicity induced by finite precision floating-point representation, Chaos, Solitons & Fractals 45 (3) (2012) 238–245. doi:https://doi.org/10.1016/j.chaos.2011.12.006.
URL https://www.sciencedirect.com/science/article/pii/S0960077911002384

[12] X.-y. Wang, X. Qin, A new pseudo-random number generator based on cml and chaotic iteration, Nonlinear Dynamics 70 (2) (2012) 1589–1592. doi:10.1007/s11071-012-0558-0.
URL https://link.springer.com/article/10.1007/s11071-012-0558-0

[13] N. K. Pareek, V. Patidar, K. K. Sud, A random bit generator using chaotic maps., IJ Network Security 10 (1) (2010) 32–38.

[14] M. Bucolo, R. Caponetto, L. Fortuna, M. Frasca, A. Rizzo, Does chaos work better than noise?, IEEE Circuits and Systems Magazine 2 (3) (2002) 4–19. doi:10.1109/MCAS.2002.1167624.

[15] H. Hu, L. Liu, N. Ding, Pseudorandom sequence generator based on the chen chaotic system, Computer Physics Communications 184 (3) (2013) 765–768. doi:https://doi.org/10.1016/j.cpc.2012.11.017.
URL https://www.sciencedirect.com/science/article/pii/S0010465512003931

[16] A. Pluchino, A. Rapisarda, C. Tsallis, Noise, synchrony, and correlations at the edge of chaos, Phys. Rev. E 87 (2013) 022910. doi:10.1103/PhysRevE.87.022910.
URL https://link.aps.org/doi/10.1103/PhysRevE.87.022910

[17] M. El-Shorbagy, A. Mousa, S. Nasr, A chaos-based evolutionary algorithm for general nonlinear programming problems, Chaos, Solitons & Fractals 85 (2016) 8–21. doi:https://doi.org/10.1016/j.chaos.2016.01.007.
URL https://www.sciencedirect.com/science/article/pii/S0960077916000163

[18] A. Viktorin, M. Pluhacek, R. Senkerik, Success-history based adaptive differential evolution algorithm with multi-chaotic framework for parent selection performance on cec2014 benchmark set, in: 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 4797–4803. doi:10.1109/CEC.2016.7744404.

[19] H. Chen, W. Li, X. Yang, A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems, Expert Systems with Applications 158 (2020) 113612. doi:https://doi.org/10.1016/j.eswa.2020.113612.
URL https://www.sciencedirect.com/science/article/pii/S095741742030436X

[20] S. Talatahari, M. Azizi, Optimization of constrained mathematical and engineering design problems using chaos game optimization, Computers & Industrial Engineering 145 (2020) 106560. doi:https://doi.org/10.1016/j.cie.2020.106560.
URL https://www.sciencedirect.com/science/article/pii/S0360835220302941

[21] M. Misaghi, M. Yaghoobi, Improved invasive weed optimization algorithm (IWO) based on chaos theory for optimal design of PID controller, Journal of Computational Design and Engineering 6 (3) (2019) 284–295. doi:10.1016/j.jcde.2019.01.001.
URL https://doi.org/10.1016/j.jcde.2019.01.001

[22] Z. Song, S. Gao, Y. Yu, J. Sun, Y. Todo, Multiple chaos embedded gravitational search algorithm, IEICE Transactions on Information and Systems 100 (4) (2017) 888–900. doi:10.1587/transinf.2016EDP7512.

[23] S. Talatahari, M. Azizi, Chaos game optimization: a novel metaheuristic algorithm, Artificial Intelligence Review 54 (2) (2021) 917–1004. doi:10.1007/s10462-020-09867-w.
URL https://link.springer.com/article/10.1007/s10462-020-09867-w

[24] A. Mozaffari, M. Emami, A. Fathi, A comprehensive investigation into the performance, robustness, scalability and convergence of chaos-enhanced

evolutionary algorithms with boundary constraints, Artificial Intelligence Review 52 (4) (2019) 2319–2380. doi:10.1007/s10462-018-9616-4.
URL https://link.springer.com/article/10.1007/s10462-018-9616-4

[25] S. Ahmed, M. Mafarja, H. Faris, I. Aljarah, Feature selection using salp swarm algorithm with chaos, in: Proceedings of the 2nd International Conference on Intelligent Systems, Metaheuristics & Swarm Intelligence, ISMSI '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 65–69. doi:10.1145/3206185.3206198.
URL https://doi.org/10.1145/3206185.3206198

[26] S. Hinojosa, D. Oliva, E. Cuevas, G. Pajares, O. Avalos, J. Gálvez, Improving multi-criterion optimization with chaos: a novel multi-objective chaotic crow search algorithm, Neural Computing and Applications 29 (8) (2018) 319–335. doi:10.1007/s00521-017-3251-x.
URL https://link.springer.com/article/10.1007/s00521-017-3251-x

[27] Q. Zhang, H. Chen, J. Luo, Y. Xu, C. Wu, C. Li, Chaos enhanced bacterial foraging optimization for global optimization, IEEE Access 6 (2018) 64905–64919. doi:10.1109/ACCESS.2018.2876996.

[28] A. M. Anter, M. Ali, Feature selection strategy based on hybrid crow search optimization algorithm integrated with chaos theory and fuzzy c-means algorithm for medical diagnosis problems, Soft Computing 24 (3) (2020) 1565–1584. doi:10.1007/s00500-019-03988-3.
URL https://link.springer.com/article/10.1007/s00500-019-03988-3

[29] J. Liang, B. Qu, P. Suganthan, Q. Chen, Problem definitions and evaluation criteria for the cec 2015 competition on learning-based real-parameter single objective optimization, Technical Report201411A, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore 29 (2014) 625–640.

[30] N. Awad, M. Ali, J. Liang, B. Qu, P. Suganthan, Problem definitions and evaluation criteria for the cec 2017 special session and competition on single objective bound constrained real-parameter numerical optimization, in: Technical Report, Nanyang Technological University Singapore, 2016.

[31] C. Yue, K. Price, P. Suganthan, J. Liang, M. Ali, B. Qu, N. Awad, P. Biswas, Problem definitions and evaluation criteria for the cec 2020 special session and competition on single objective bound constrained numerical optimization, Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China, Tech. Rep 201911.

[32] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, Journal of global optimization

11 (4) (1997) 341–359. doi:10.1023/A:1008202821328.
URL https://link.springer.com/article/10.1023/A:1008202821328

[33] I. Zelinka, SOMA — Self-Organizing Migrating Algorithm, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 167–217. doi:10.1007/978-3-540-39930-8_7.
URL https://doi.org/10.1007/978-3-540-39930-8_7

[34] I. Zelinka, SOMA—Self-organizing Migrating Algorithm, Springer International Publishing, Cham, 2016, pp. 3–49. doi:10.1007/978-3-319-28161-2_1.
URL https://doi.org/10.1007/978-3-319-28161-2_1

[35] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, Vol. 4, 1995, pp. 1942–1948 vol.4. doi:10.1109/ICNN.1995.488968.

[36] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, A comprehensive survey: artificial bee colony (abc) algorithm and applications, Artificial Intelligence Review 42 (1) (2014) 21–57. doi:10.1007/s10462-012-9328-0.
URL https://link.springer.com/article/10.1007/s10462-012-9328-0

[37] K. Socha, M. Dorigo, Ant colony optimization for continuous domains, European Journal of Operational Research 185 (3) (2008) 1155 – 1173. doi:https://doi.org/10.1016/j.ejor.2006.06.046.
URL http://www.sciencedirect.com/science/article/pii/S0377221706006333

[38] X.-S. Yang, Firefly algorithms for multimodal optimization, in: International symposium on stochastic algorithms, Springer, 2009, pp. 169–178. doi:10.1007/978-3-642-04944-6_14.
URL https://link.springer.com/chapter/10.1007/978-3-642-04944-6_14

[39] M. G. H. Omran, A novel cultural algorithm for real-parameter optimization, International Journal of Computer Mathematics 93 (9) (2016) 1541–1563. arXiv:https://doi.org/10.1080/00207160.2015.1067309, doi:10.1080/00207160.2015.1067309.
URL https://doi.org/10.1080/00207160.2015.1067309

[40] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, Advances in Engineering Software 69 (2014) 46 – 61. doi:https://doi.org/10.1016/j.advengsoft.2013.12.007.
URL http://www.sciencedirect.com/science/article/pii/S0965997813001853

[41] S. Mirjalili, A. Lewis, The whale optimization algorithm, Advances in Engineering Software 95 (2016) 51 – 67. doi:https:

//doi.org/10.1016/j.advengsoft.2016.01.008.
URL http://www.sciencedirect.com/science/article/pii/S0965997816300163

[42] İbrahim Berkan Aydilek, A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems, Applied Soft Computing 66 (2018) 232 – 249. doi:https://doi.org/10.1016/j.asoc.2018.02.025.
URL http://www.sciencedirect.com/science/article/pii/S156849461830084X

[43] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm and Evolutionary Computation 1 (1) (2011) 3 – 18. doi:https://doi.org/10.1016/j.swevo.2011.02.002.
URL http://www.sciencedirect.com/science/article/pii/S2210650211000034

[44] J. Carrasco, S. García, M. Rueda, S. Das, F. Herrera, Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review, Swarm and Evolutionary Computation 54 (2020) 100665. doi:https://doi.org/10.1016/j.swevo.2020.100665.
URL http://www.sciencedirect.com/science/article/pii/S2210650219302639