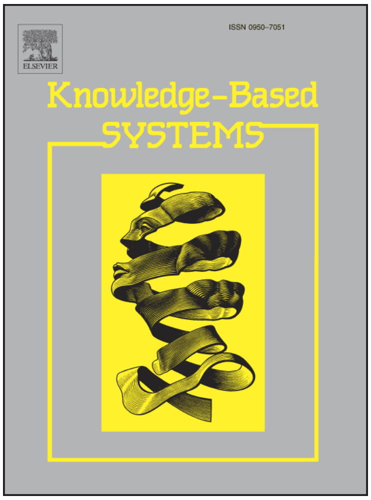


Chaotic fruit fly optimization algorithm

Marko Mitić, Najdan Vuković, Milica Petrović, Zoran Miljković

PII: S0950-7051(15)00314-7
DOI: http://dx.doi.org/10.1016/j.knosys.2015.08.010
Reference: KNOSYS 3251

To appear in: *Knowledge-Based Systems*

Received Date: 12 September 2014
Revised Date: 15 August 2015
Accepted Date: 17 August 2015

Please cite this article as: M. Mitić, N. Vuković, M. Petrović, Z. Miljković, Chaotic fruit fly optimization algorithm, *Knowledge-Based Systems* (2015), doi: http://dx.doi.org/10.1016/j.knosys.2015.08.010

Chaotic fruit fly optimization algorithm

Dr. Marko Mitić, Corresponding Author
Assistant Research Professor
University of Belgrade - Faculty of Mechanical Engineering
Production Engineering Department
Kraljice Marije 16; 11120 Belgrade 35; Serbia
Phone: +381 642835 287
e-mail: mmitic@mas.bg.ac.rs or miticm@gmail.com

Dr. Najdan Vuković
Assistant Research Professor
University of Belgrade - Faculty of Mechanical Engineering
Innovation Center
Kraljice Marije 16; 11120 Belgrade 35; Serbia
Phone: +381 63363 858
e-mail: nvukovic@mas.bg.ac.rs or najdanvuk@gmail.com

Milica Petrović, MSc
Teaching and Research Assistant
University of Belgrade - Faculty of Mechanical Engineering
Production Engineering Department
Kraljice Marije 16; 11120 Belgrade 35; Serbia
e-mail: mmpetrovic@mas.bg.ac.rs

Dr. Zoran Miljković
Full Professor
University of Belgrade - Faculty of Mechanical Engineering
Production Engineering Department
Kraljice Marije 16; 11120 Belgrade 35; Serbia
Phone: +381 11 3302 468
e-mail: zmiljkovic@mas.bg.ac.rs
url: http://cent.mas.bg.ac.rs/english/staff/zmiljkovic.htm

Abstract

Fruit fly optimization algorithm (FOA) is recently presented metaheuristic technique that is inspired by the behavior of fruit flies. This paper improves the standard FOA by introducing the novel parameter integrated with chaos. The performance of developed chaotic fruit fly algorithm (CFOA) is investigated in details on ten well known benchmark problems using ten different chaotic maps. Moreover, we performed comparison studies with basic FOA, FOA with Levy flight distribution, and other recently published chaotic algorithms. Statistical results on every optimization task indicate that the chaotic fruit fly algorithm (CFOA) has a very fast convergence rate. In addition, CFOA is compared with recently developed chaos enhanced algorithms such as chaotic bat algorithm, chaotic accelerated particle swarm optimization, chaotic firefly algorithm, chaotic artificial bee colony algorithm, and chaotic cuckoo search. Overall research findings show that FOA with Chebyshev map show superiority in terms of reliability of global optimality and algorithm success rate.

Keywords: Fruit fly optimization algorithm, Chaos, Metaheuristic technique, Optimization

1. Introduction

Recently, a large number of complex nonlinear optimization problems are solved using mathematical tools inspired by phenomena found in nature. In such cases, traditional algorithms often may not produce desired outcomes and therefore the alternate methods must be employed. Metaheuristic techniques are a well known global optimization approach that has been widely used to solve many different optimization issues [1, 2]. These methods mimic the social behaviour of species or natural mechanisms in order to find the best possible result for the given problem. In majority of these techniques, an algorithm starts by creating a random population of units which is then manipulated by using iterations and stochastic processes.

Many metaheuristic algorithms have been developed over the last few years. Most popular techniques in the field include harmony search [3], firefly algorithm [4], cuckoo search [5], bat algorithm [6], and krill heard [7], which are successfully applied for solving various optimization and real world problems [8]. One of the latest population-based techniques is the fruit fly optimization algorithm (FOA) [9]. So far, FOA has been used for: tuning of PID controller [10], semiconductor testing scheduling [11], power load forecasting [12], solving multidimensional knapsack problem [13] and continuous function optimization problem [14]. The algorithm proved to be very efficient, even in comparison with other state-of-the art techniques, such as mentioned harmony search [14]. However, similarly to other metaheuristic approaches, determination of algorithm-dependent parameters is still one of the key issues that influence FOA performance.

Latest trend in developing more effective metaheuristic techniques lies in their integration with chaos. Chaos theory is related to the study of chaotic dynamical systems that are sensitive to initial conditions [15]. Due to the ergodicity and mixing properties of chaos, algorithms can potentially carry out iterative step search at higher speeds then standard stochastic search with standard probability distributions [16, 17]. Recent research in the field refers to employment of chaos in: genetic algorithms [18], particle swarm optimization [19], harmony search [20], ant colony optimization [21], bee colony optimization [22], simulated annealing [23], firefly algorithm [24], bat algorithm [17], krill heard technique [25], and biogeography-based optimization [15]. Empirical studies in all of these approaches show that methods with chaos have a high-level mixing capability, which results in solutions with higher diversity and mobility.

Initial result of implementation of chaos in FOA is reported in [26]. In this study, FOA is integrated with logistic chaos map and such algorithm is tested in optimization of one function. However, it is known that different maps may lead to different behavior of algorithms, so the most of the aforementioned works in this domain investigate influence of various chaotic maps. Also, in order to obtain a full insight of the capabilities of the developed algorithm, it is necessary to test its performance on several different unimodal and multimodal nonlinear functions. Likewise, it is essential to compare the chaotic algorithm with the standard version, and also with other state-of-the-art techniques with chaos so as to obtain full insight in the capabilities of chaos enhanced algorithm.

This paper presents novel FOA method based on chaotic mapping with the following main contributions: (i) FOA algorithm is integrated with ten different chaos maps in order to find the most appropriate one for the problem in hand; (ii) algorithm performance is tested on ten different nonlinear functions; (iii) the developed method is compared with standard FOA and FOA with Levy distribution, as well as with other state-of-the-art chaos based metaheuristic techniques.

The rest of the paper is organized as follows. In Section 2 basic FOA is described. Section 3 presents developed chaotic levy FOA, with the detailed mathematical description of each of the employed map. Implementation details are stated in Section 4. Experimental results obtained for different unimodal and multimodal functions, comparison with other chaos optimization algorithms, as well as discussion of results are given in Section 5. Finally, Section 6 delivers the conclusion of this study.

2. Basic fruit fly optimization algorithm

Fruit fly optimization algorithm is inspired by the behaviour of the fruit flies found in nature [9]. Fruit fly (lat. *Drosophila*) is superior in comparison with other similar species, especially in terms of food foraging using osphresis and vision characteristics. The smell foraging phase enables an individual to search and locate food sources around the fruit fly swarm. For each of the food sources the smell concentration that corresponds to the fitness value is evaluated next. In the vision foraging phase maximum smell concentration value is allocated, and then the swarm is directed towards it. This foraging behaviour of fruit flies is presented in Fig. 1.

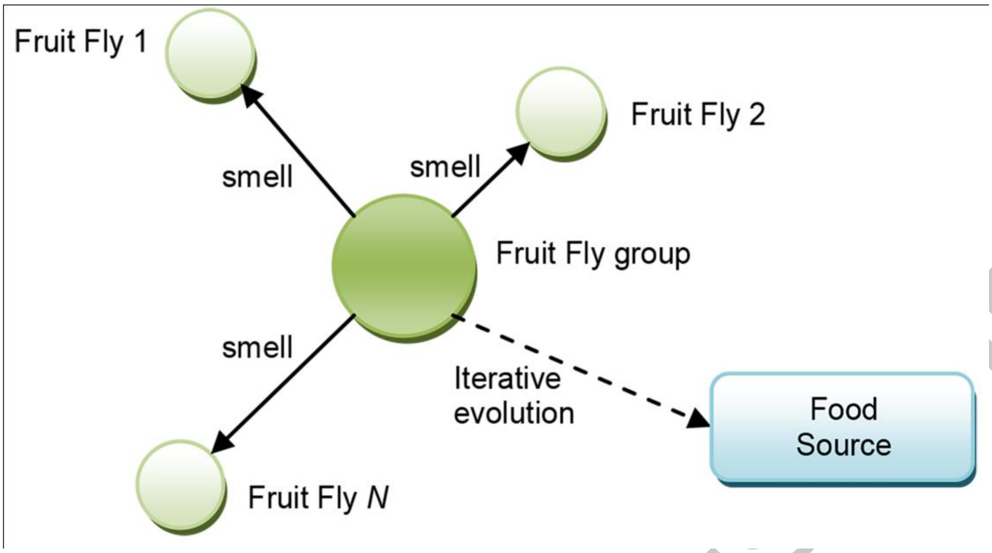


Fig.1. Iterative foraging process of fruit flies.

FOA can be summarized thorough 6 independent steps which are defined as follows [11]: (i) set the maximum number of iteration and population size; (ii) randomly initialize fruit fly swarm location using Eq. (1);

$$\delta_j = lower\_bound + (upper\_bound\text{-}lower\_bound) \times \texttt{rand()} \tag{1}$$

(iii) randomly generate a number of fruit flies around the fruit fly swarm so as to form a population (osphresis search process) using Eq.2;

$$x_{i,j} = \delta_j \pm \texttt{rand()},\ \ j = 1,\ldots,n \tag{2}$$

(iv) evaluate the entire fruit fly population in order to obtain the smell concentration value of each fruit fly in swarm; (v) determine the fruit fly with the maximum smell concentration and then direct the swarm towards the location of this best individual (vision search process), and finally (vi) finish the algorithm if the maximum number of generation is reached, or go to step (iii) otherwise.

The parameters $x_{i,j}$ and δ_j influence both search phases (osphresis and vision), and are responsible for generation of food sources. It is obvious that the way they are calculated in a significant manner determines the final algorithm solution. Numerous studies showed that random-based optimization algorithms perform better when using non-standard distributions (i.e. Gauss or uniform distribution) [16, 17]. Additionally, the properties of non-repetition and ergodicity of chaos can force an algorithm to carry out overall searches at higher speeds [16]. These are the main reasons for the development of improved FOA described next.

3. Chaotic fruit fly optimization algorithm

This section presents novel FOA by introducing a new parameter enhanced by chaos. We start the explanation of the chaotic fruit fly optimization algorithm (CFOA) as follows.

3.1 Algorithm initialization

Initial swarm location may have major influence in faster convergence and final outcome. As a first improvement, CFOA determines the initial fruit fly swarm location by choosing the best one among PS randomly generated solutions. Similarly to what is found in [14], this calculation of initial swarm position results in a faster convergence and a better algorithm solution at the end of experimental run.

3.2. Chaos FOA

In FOA, crucial influence on algorithm performance refers to the calculation of food sources. Basic implementation of this metaheuristic technique assumes randomization of $x_{i,j}$ variables using uniform distribution. This is often not a good choice, especially when dealing with complex nonlinear and multimodal problems. In order to enhance convergence and overall speed of FOA we introduce a new parameter, $alpha$, which is used for generation of food sources. Particularly, we modified the Eq. (2) so that it involves a chaotic variable as follows:

$$x_{i,j} = x_{i,j} + alpha(x_{i,j} - x_j^*),\ i = 1,...,PS,\ \ j = 1,...,n \tag{3}$$

where *X** is the currently best solution. In this manner, we force the individuals to move towards the *best so far* optimal solution in the chaotic manner. This proved to be a huge advantage in comparison with basic FOA and FOA with Levy distribution. The complete procedure of a novel CFOA is presented in Fig. 2.

Algorithm 1. Chaotic FOA pseudocode

```
// Algorithm initialization
Set the population size PS and maximum number of iterations $I_{max}$
// Initialize fruit fly swarm location in the search space n
For i = 1,..,PS
    $x_{i,j} = lower\_bound + (upper\_bound\text{-}lower\_bound)$ x rand(), $j = 1,...,n$
EndFor
$\Delta \leftarrow \arg\left(\min_{i=1,2,...,PS} f(X_i)\right)$ // Set swarm location
// Set optimal solution and iteration counter:
$X^* = \Delta$
```

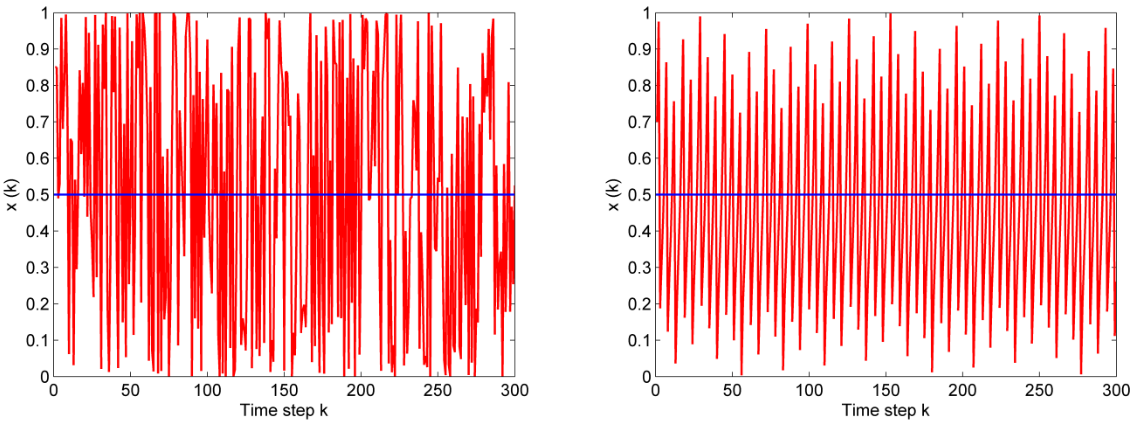
```
$Iter = 0$
Repeat
// Smell-based (osphresis) foraging phase
  For j =1,…,PS
     // Generate food source $X_i = \left(x_{i,1}, x_{i,2}, ..., x_{i,n}\right)$
     $alpha = \texttt{chaos}()$ // Determine chaotic parameter
     $x_{i,j} = x_{i,j} + alpha(x_{i,j} - x_j^*),\ i = 1,...,PS,\ \ j = 1,...,n$
     // Limit the result
     If $x_{i,j} > upper\_bound$ then
         $x_{i,j} = upper\_bound$
     EndIf
     If $x_{i,j} > upper\_bound$ then
         $x_{i,j} = upper\_bound$
     EndIf
  EndFor

// Vision-based foraging phase
  $X_{best} = \arg\min_{i=1,..,PS}\left(f(X_i)\right)$
// Find global best solution
   If $f\left(X_{best}\right) < f\left(\Delta\right)$ then
       $\Delta = X_{best}$
   EndIf
   If $f\left(\Delta\right) < f\left(X*\right)$ then
       $X* = \Delta$
   EndIf
Until the maximum number of iteration is reached: $Iter == Iter_{\max}$
```

Fig. 2 Procedure of Chaotic FOA

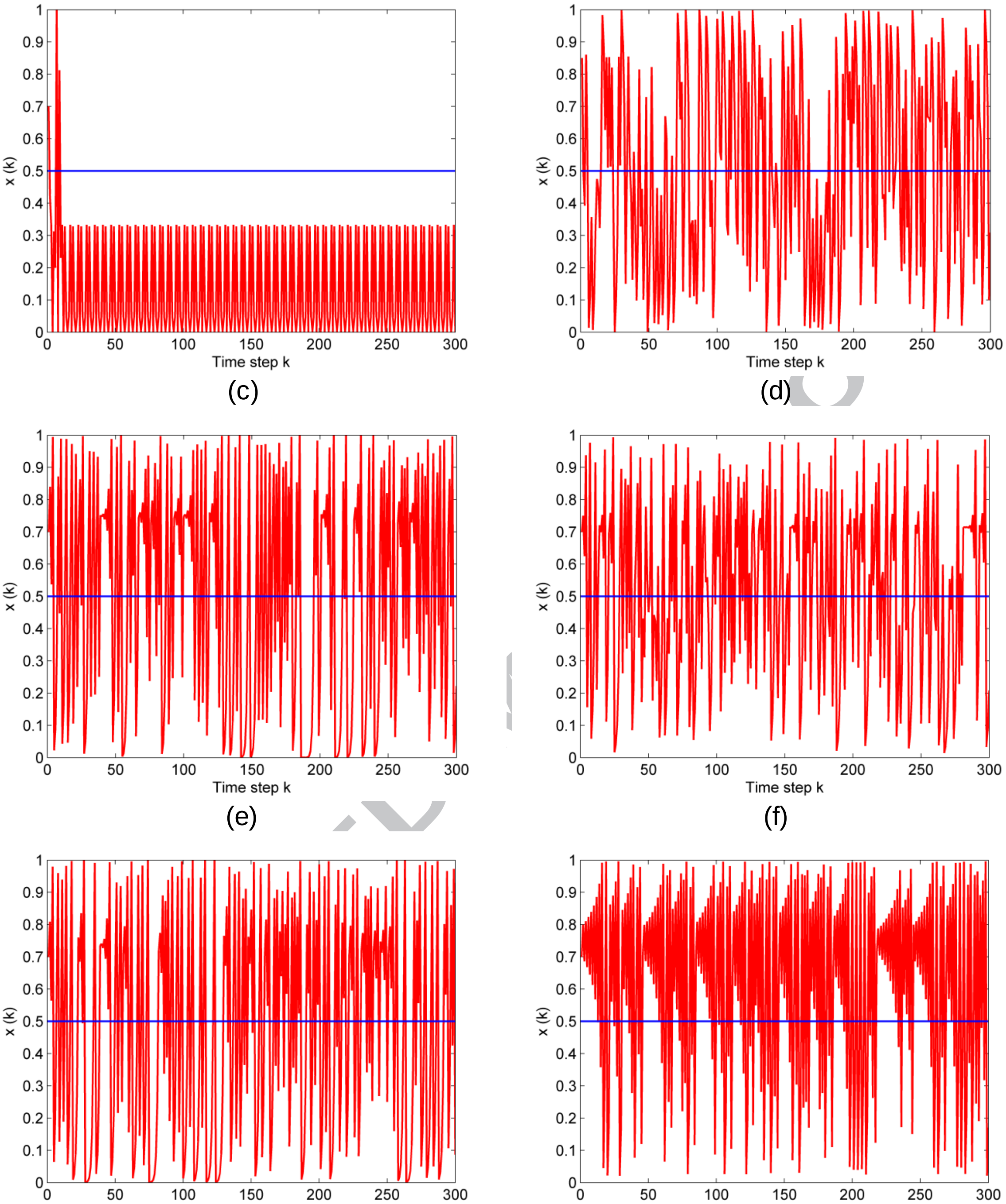
In this paper, we investigate the influence of ten different one-dimensional non-invertible chaotic maps, similarly to other recent studies [15]. A mathematical description and graphical presentation of these maps for 300 iterations are given in Table 1 and Fig. 3, respectively. It is important to note that the chaotic behavior is evident in Fig. 3 despite the lack of random component in equations given in Table 1. Each chaotic map presented in Fig. 4 has the starting point of 0.7. The maps that do not produce values in range of [0, 1] are normalized to fit into this scale.

Table 1. Chaotic maps used in this study

No.	Map name	Equation
1	Chebyshev	$x_{i+1} = \cos\left(i\cos^{-1}\left(x_i\right)\right)$
2	Circle	$x_{i+1} = \mathrm{mod}\left(x_i + b - \left(a/2\pi\right)\sin(2\pi x_i),1\right);\ a = 0.5, b = 0.2$
3	Gauss/Mouse	$x_{i+1} = \begin{cases} 1, & xi = 0 \\ \dfrac{1}{\mathrm{mod}\left(x_i,1\right)}, & otherwise \end{cases}$
4	Iterative	$x_{i+1} = \sin\left(\dfrac{a\pi}{x_i}\right),\ a = 0.7$
5	Logistic	$x_{i+1} = ax_i\left(1 - x_i\right),\ a = 4$
6	Piecewise	$x_{i+1} = \begin{cases} x_i/P, & 0 \le x_i < P \\ \dfrac{x_i - P}{0.5 - P}, & P \le x_i \le 0.5 \\ \dfrac{1 - P - x_i}{0.5 - P}, & 0.5 \le x_i < 1 - P \\ \dfrac{1 - x_i}{P}, & 1 - P \le x_i < 1 \end{cases},\ P = 0.4$
7	Sine	$x_{i+1} = \dfrac{a}{4}\sin\left(\pi x_i\right),\ a = 4$
8	Singer	$x_{i+1} = \mu\left(7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.301875x_i^4\right),\ \mu = 1.07$
9	Sinusoidal	$x_{i+1} = ax_i\sin\left(\pi x_i\right),\ a = 2.3$
10	Tent	$x_{i+1} = \begin{cases} x_i/0.7, & x_i < 0.7 \\ \dfrac{10}{3}\left(1 - x_i\right), & x_i \ge 0.7 \end{cases}$



(a) (b)



(c)

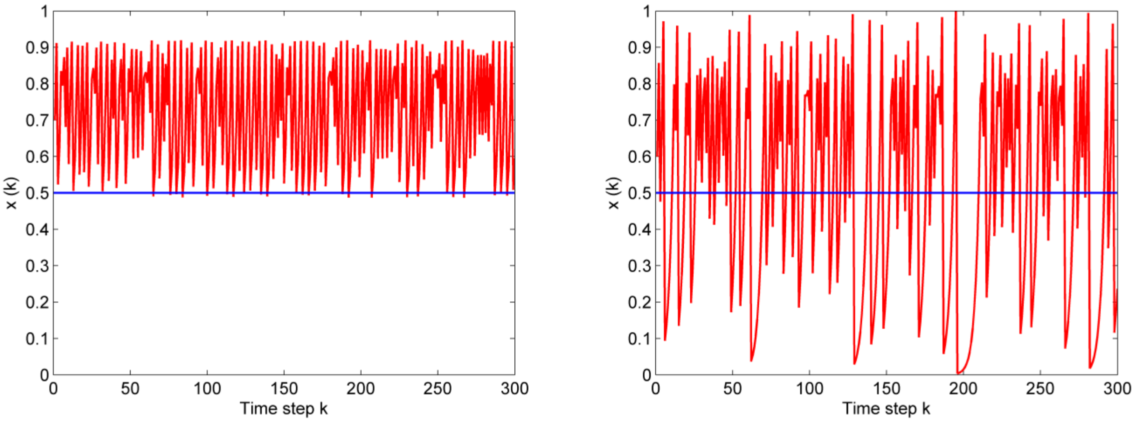
(d)

(e)

(f)

(g)

(h)



(i) (j)

Fig. 3. Visualization of implemented chaotic maps: (a) Chebyshev map, (b) Circle map, (c) Gauss/Mouse map, (d) Iterative map, (e) Logistic map, (f) Piecewise map, (g) Sine map, (h) Singer map, (i) Sinusoidal map, (j) Tent map

4. Implementation details

4.1 Test functions

Chaotic FOA presented in this study is benchmarked using 14 different well-known functions [14, 15, 27]. Total optimization set contains seven unimodal and seven multimodal examples. The mathematical description and graphical presentation of these functions are given in Table 2 and Fig. 4, respectively.

For all test problems global optimum is equal to $f(X^*)=0$. Boundaries for functions are equal to their known initial ranges. After the every algorithm iteration, boundary constraint for each $x_{i,j}$ parameter is applied.

Table 2. Benchmark functions

ID	Function name	Equation	Upper and Lower Bound	Dimension n	Type
F1	Sphere	$f(x)=\sum_{i=1}^{n} x_i^2$	Ub=100 Lb=-100	30	Unimodal
F2	Schwefel's problem 2.22	$f(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	Ub=10 Lb=-10	30	Unimodal
F3	Quartic	$f(x)=\sum_{i=1}^{n} ix_i^4+\mathrm{rand}$	Ub=1.28 Lb=-1.28	30	Unimodal
F4	Sum squares	$f(x)=\sum_{i=1}^{n} ix_i^2$	Ub=10 Lb=-10	30	Unimodal

F5	Sum of different power	$f\left(x\right)=\sum_{i=1}^{n}\lvert x_i\rvert^{i+1}$	Ub=1 Lb=-1	30	Unimodal
F6	Schwefel's problem 2.21	$f\left(x\right)=\max\left\{\lvert x_i\rvert,1\le i\le n\right\}$	Ub=100 Lb=-100	30	Unimodal
F7	Rosenbrock	$f\left(x\right)=\sum_{i=1}^{n-1}\left(100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right)$	Ub=30 Lb=-30	30	Unimodal
F8	Ackley	$f\left(x\right)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)$ $-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)\right)+20+e$	Ub=32 Lb=-32	30	Multimodal
F9	Griewank	$f\left(x\right)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	Ub=600 Lb=-600	30	Multimodal
F10	Alpine	$f\left(x\right)=\sum_{i=1}^{n}\lvert x_i\sin\left(x_i\right)+0.1x_i\rvert$	Ub=10 Lb=-10	30	Multimodal
F11	Powell	$f\left(x\right)=\sum_{i=1}^{n/4}\Big[\left(x_{4i-3}+10x_{4i-2}\right)^2+5\left(x_{4i-1}-x_{4i}\right)^2$ $+\left(x_{4i-2}-2x_{4i-1}\right)^4+10\left(x_{4i-3}-x_{4i}\right)^4\Big]$	Ub=5 Lb=-4	30	Multimodal
F12	Rastrigin	$f\left(x\right)=\sum_{i=1}^{n}\left(x_i^2-10\cos\left(2\pi x_i\right)+10\right)$	Ub=5.12 Lb=-5.12	30	Multimodal
F13	Generalized Penalized 1	$f\left(x\right)=\frac{\pi}{n}\Big\{10\sin^2\left(\pi y_1\right)+\sum_{i=1}^{n-1}\left(y_i-1\right)^2\left[1+10\sin^2\left(\pi y_{i+1}\right)\right]$ $+\left(y_n-1\right)^2\Big\}+\sum_{i=1}^{n}\mu\left(x_i,10,100,4\right)$ $y_i=1+\frac{1}{4}\left(x_i+1\right),\ \mu\left(x_i,a,k,m\right)=\begin{cases}k\left(x_i-a\right)^m, & x_i>a\\ 0, & -a\le x_i\le a\\ k\left(-x_i-a\right)^m, & x_i<a\end{cases}$	Ub=50 Lb=-50	30	Multimodal
F14	Solomon problem	$f\left(x\right)=1-\cos\left(2\pi\sqrt{\sum_{i=1}^{n}x_i^2}\right)+0.1\sqrt{\sum_{i=1}^{n}x_i^2}$	Ub=100 Lb=-100	30	Multimodal

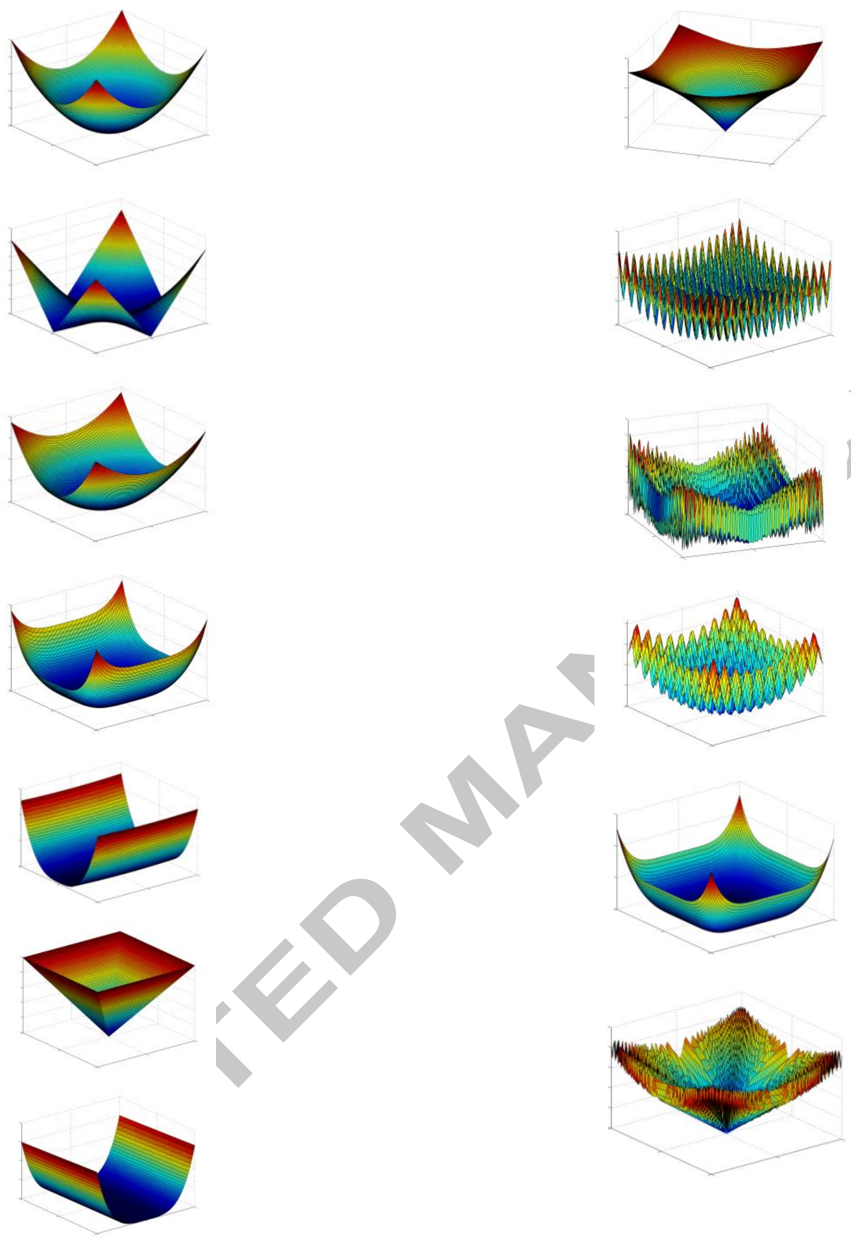


Fig.4 Visualization of benchmark functions for *n=2 (column wise): F1-F7 in the first column, F8, F9, F10, F12, F13, F14 in the second row*

4.2 Success criterion

Beside the usual measures for algorithm evaluations such as best, mean and median results, in this paper we additionally apply success rate criterion. The success rate parameter S_r is defined as [24]:

$$S_r = \frac{N_{success}}{N_{all}} \times 100 \quad (4)$$

where $N_{success}$ is the number of successful trials, and N_{all} is number of trials. Similarly to other studies [17,19], one experimental run is considered successful if the final algorithm solution is close to the searched optimum. The closeness criterion depends on the search space of a particular function, and is defined as [24]:

$$\left| X^{gbest} - X^* \right| \leq (upper\_bound - lower\_bound) \times 10^{-4} \quad (5)$$

where X^{gbest} is the obtained global best result of the developed algorithm.

4.3. Test studies and initialization

In this paper, we tested each function with 50 independent algorithm runs. The initial conditions of each test are completely different, so the algorithm outcome is practically independent of the starting position of the fruit fly swarm. To completely evaluate the CFOA performance, we used statistical measures such as median and mean objective values, as well as their standard deviations. This information is provided for each chaotic map and every tested function.

Additionally, extensive studies regarding parameter settings are carried out. From conducted experiments, we concluded that a population of 50 individuals and 700 iterations per experimental run is sufficient for all testing cases. Likewise, in all of the experiments, initial value of 0.95 for parameter $alpha$ proved to be a good choice.

Finally, it is also important to note that every reported result in this paper is obtained using Matlab software that runs on a desktop PC with 4GBs RAM and Windows 7 (64 bit) operating system.

5. Experimental studies

5.1.CFOA computational results on benchmark problems

Computational results for all functions using all maps are given in Table 3 – Table 16 (best result is given in bold font). Chaotic FOA is tested on 14 benchmark problems using Chebyshev, Circle, Gauss/Mouse, Iterative, Logistic, Piecewise, Sine, Singer, Sinusoidal, and Tent map. Beside the mentioned statistical measures, we added best and worst results obtained over 50 independent runs, as well as average time for one such test.

Overall, the experimental results prove the usefulness of the implementation of chaos in FOA. In each tested case, CFOA final output was very close to the desired function optimum. For ten out of the fourteen functions, the best minimum value is obtained before the maximum number of iteration is reached. Algorithm also converged in case of F3 and F8, while CFOA did not solve minimization problems in functions F7 and F13. We also provide additional information on the exact iteration in which the function optimum is reached.

Experiments show that Chebyshev, Iterative, Logistic, Sine and Singer map provide best results in all cases (Table 3 – Table 16). Particularly, these maps reached maximum algorithm performance before the end of one experimental cycle, except for F3 and F8 (for F7 and F13 algorithm did not converged). Among then, Chebyshev map yields the best results in terms of fastest algorithm convergence.

Statistical results given in Table 8 for F6 show that all chaotic maps except map No. 9 has the same influence on the algorithm performance. As the matter of fact, by observing Table 3 – Table 16 one should note that Sinusoidal map is the worst choice for generation of parameter $alpha$. This conclusion supports the results obtained for success rate showed in Table 17. Except for functions F6, F9 and F14, CFOA based on Sinusoidal map did not manage to successfully converge. All of the other types of CFOA show impressive performance in terms of algorithm performance and convergence. This is especially evident in comparison with other chaotic state-of-the-art algorithms, which did not succeed to achieve such a consistent, high level result.

Table 3. Results of CFOA for Sphere function (F1) after 50 independent runs

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]
1	Chebyshev map	**<1E-309 (Iter. 540)***	<1E-309	<1E-309	<1E-309	N/A	0.5244
2	Circle map	4.4913E-234	3.6272E-230	2.8973E-229	4.2116E-228	N/A	0.4009
3	Gauss/Mouse map	1.0276E-295	3.4043E-295	5.1591E-295	6.270E-294	N/A	0.3926
4	Iterative map	<1E-309 (Iter. 659)*	<1E-309	<1E-309	<1E-309	N/A	0.4323
5	Logistic map	<1E-309 (Iter. 630)*	<1E-309	<1E-309	<1E-309	N/A	0.4478
6	Piecewise map	1.742E-278	5.4917E-278	1.4802E-277	1.3074E-276	N/A	0.3091
7	Sine map	<1E-309 (Iter. 644)*	<1E-309	<1E-309	<1E-309	N/A	0.4426
8	Singer map	<1E-309 (Iter. 606)*	<1E-309	<1E-309	<1E-309	N/A	0.4805
9	Sinusoidal map	1.0889	1.9278	1.8828	2.4395	3.2092 E-1	0.3946
10	Tent map	1.0274E-268	1.0484E-267	1.3309E-267	5.2944E-267	N/A	0.3903

* Iteration in which the best result is achieved.

Table 4. Results of CFOA for Schwefel’s problem 2.22 (F2) after 50 independent runs

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]
1	Chebyshev map	**<1E-155 (Iter. 540)***	<1E-155	<1E-155	<1E-155	N/A	0. 5753

2	Circle map	7.3752E-117	3.2026E-115	1.1038E-114	1.0059E-113	1.9044 E-114	0. 4546
3	Gauss/Mouse map	5.7087E-148	1.0723E-147	1.1916E-147	2.8293E-147	5.2981 E-148	0. 4452
4	Iterative map	<1E-155 (Iter. 655)*	<1E-155	<1E-155	<1E-155	N/A	0. 4877
5	Logistic map	<1E-155 (Iter. 628)*	<1E-155	<1E-155	<1E-155	N/A	0. 5025
6	Piecewise map	1.5945E-139	5.6953E-139	5.7616E-139	1.0727E-138	1.8388 E-139	0. 4441
7	Sine map	<1E-155 (Iter. 643)*	<1E-155	<1E-155	<1E-155	N/A	0. 4951
8	Singer map	<1E-155 (Iter. 604)*	<1E-155	<1E-155	<1E-155	N/A	0. 5352
9	Sinusoidal map	2.9735	3.6929	3.6606	3.9835	2.2755 E-001	0. 4432
10	Tent map	3.0217E-134	6.0680E-134	6.7105E-134	1.6321E-133	3.315E-134	0. 4456

* Iteration in which the best result is achieved.

Table 5. Results of CFOA for Quartic function (F3) after 50 independent runs

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]
1	Chebyshev map	8.7628E-006	2.2289E-004	2.4749E-003	3.8616 E-002	6.1081 E-003	0. 6125
2	Circle map	4.4841E-007	2.1225E-005	2.8433E-005	1.2379 E-004	2.8216 E-005	0. 6384
3	Gauss/Mouse map	4.7370E-007	2.6933E-005	3.3066E-005	1.2131 E-004	3.0446 E-005	0. 6311
4	Iterative map	1.8656E-007	1.5941E-005	2.3617E-005	8.6441 E-005	2.2242 E-005	0. 64
5	Logistic map	4.6066E-007	1.9228E-005	2.4341E-005	1.114E-004	2.3347 E-005	0. 6301
6	Piecewise map	6.0483E-007	1.7433E-005	2.7048E-005	1.4334 E-004	2.8233 E-005	0. 6276
7	Sine map	**7.2192E-008**	1.5060E-005	2.0866E-005	1.0440 E-004	1.9906 E-005	0. 6352
8	Singer map	2.4731E-007	2.2611E-005	3.1783E-005	1.1282 E-004	2.8293 E-005	0. 6478

9	Sinusoidal map	3.2799E+002	6.9723E+002	7.3836E+002	1.2452E+003	2.3508E+002	0. 6049
10	Tent map	7.0523E-007	1.8834E-005	2.4080E-005	8.0337E-005	2.1499E-005	0. 628

Table 6. Results of CFOA for Sum squares function (F4) after 50 independent runs

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]
1	Chebyshev map	**<1E-308 (Iter. 539)***	<1E-308	<1E-308	<1E-308	N/A	0.5009
2	Circle map	9.5256E-232	9.8355E-230	2.5643E-228	6.8626E-227	N/A	0. 3779
3	Gauss/Mouse map	1.5323E-295	8.8688E-295	1.8718E-294	1.5569E-293	N/A	0. 369
4	Iterative map	<1E-308 (Iter. 655)*	<1E-308	<1E-308	<1E-308	N/A	0. 4124
5	Logistic map	<1E-308 (Iter. 628)*	<1E-308	<1E-308	<1E-308	N/A	0. 4255
6	Piecewise map	6.3267E-278	2.4457E-277	5.6883E-277	1.0103E-275	N/A	0. 3683
7	Sine map	<1E-308 (Iter. 642)*	<1E-308	<1E-308	<1E-308	N/A	0. 4202
8	Singer map	<1E-308 (Iter. 605)*	<1E-308	<1E-308	<1E-308	N/A	0. 4587
9	Sinusoidal map	4.2542	7.364	7.2234	9.4625	1.1496	0. 3717
10	Tent map	6.4293E-268	2.9082E-267	3.9658E-267	1.6777E-266	N/A	0. 369

* Iteration in which the best result is achieved.

Table 7. Results of CFOA for Sum of different power function (F5) after 50 independent runs

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]
1	Chebyshev map	**<1E-309 (Iter. 541)***	<1E-309	<1E-309	<1E-309	N/A	0. 6196
2	Circle map	1.3305E-231	4.3830E-230	2.6402E-229	4.7710E-228	N/A	0. 5956

3	Gauss/Mouse map	6.3003E-297	4.2919E-296	6.7760E-296	2.4588E-295	N/A	0. 5895
4	Iterative map	<1E-309 (Iter. 656)*	<1E-309	<1E-309	<1E-309	N/A	0. 6175
5	Logistic map	<1E-309 (Iter. 630)*	<1E-309	<1E-309	<1E-309	N/A	0. 6235
6	Piecewise map	2.5570E-279	8.5835E-279	1.3152E-278	9.1050E-278	N/A	0. 5885
7	Sine map	<1E-309 (Iter. 644)*	<1E-309	<1E-309	<1E-309	N/A	0. 6232
8	Singer map	<1E-309 (Iter. 606)*	<1E-309	<1E-309	<1E-309	N/A	0. 6497
9	Sinusoidal map	6.7858E+003	7.2111E+006	1.0157E+008	2.1105E+009	3.320896E+008	0. 5368
10	Tent map	1.0233E-269	1.0813E-268	1.4729E-268	7.1666E-268	N/A	0. 5876

* Iteration in which the best result is achieved.

Table 8. Results of CFOA for Schwefel's problem 2.21 (F6) after 50 independent runs

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]
1	Chebyshev map	**<1E-155 (Iter. 535)***	<1E-155	<1E-155	<1E-155	N/A	0. 3465
2	Circle map	2.5806E-118	3.504E-117	2.3144E-117	1.1603E-116	3.3689E-117	0. 3545
3	Gauss/Mouse map	2.6879E-150	1.9504E-149	1.6698E-149	6.4141E-149	1.3876E-149	0. 3387
4	Iterative map	<1E-155 (Iter. 650)*	<1E-155	<1E-155	<1E-155	N/A	0. 3447
5	Logistic map	<1E-155 (Iter. 624)*	<1E-155	<1E-155	<1E-155	N/A	0. 3671
6	Piecewise map	1.7012E-141	7.1163E-141	5.5271E-141	2.5251E-140	5.1416E-141	0. 3718
7	Sine map	<1E-155 (Iter. 641)*	<1E-155	<1E-155	<1E-155	N/A	0. 3965
8	Singer map	<1E-155 (Iter. 600)*	<1E-155	<1E-155	<1E-155	N/A	0. 3757
9	Sinusoidal map	1.8929E-002	2.454E-002	2.4369E-002	3.1503E-002	3.2026E-003	0.3375

10	Tent map	1.8361E-136	5.4338E-136	5.3438E-136	1.0073E-135	2.3030 E-135	0. 3385

* Iteration in which the best result is achieved.

Table 9. Results of CFOA for Rosenbrock function (F7) after 50 independent runs

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]
1	Chebyshev map	2.8321 E+001	2.8611 E+001	2.8637 E+001	2.8746 E+001	1.1409 E-001	0.4007
2	Circle map	2.8484 E+001	2.8692 E+001	2.8704 E+001	2.8754 E+001	5.7014 E-002	0.4028
3	Gauss/Mouse map	2.8524 E+001	2.8697 E+001	2.8713 E+001	2.8752 E+001	5.2389 E-002	0.409
4	Iterative map	**2.8464 E+001**	2.8679 E+001	2.8711 E+001	2.8758 E+001	7.8703 E-002	0.3889
5	Logistic map	2.8593 E+001	2.8702 E+001	2.8718 E+001	2.8763 E+001	5.0520 E-002	0.3889
6	Piecewise map	2.8531 E+001	2.8682 E+001	2.8704 E+001	2.8763 E+001	6.0911 E-002	0.9436
7	Sine map	2.8554 E+001	2.8710 E+001	2.8727 E+001	2.8769 E+001	5.1559 E-002	0.3955
8	Singer map	2.8625 E+001	2.8708 E+001	2.8719 E+001	2.8751 E+001	3.4434 E-002	0.4885
9	Sinusoidal map	3.0186 E+001	3.1651 E+001	3.1585 E+001	3.2725 E+001	6.4110 E-001	0.4744
10	Tent map	2.8566 E+001	2.869 E+001	2.8707 E+001	2.8761 E+001	5.8085 E-002	0.3984

Table 10. Results of CFOA for Ackley function (F8) after 50 independent runs

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]
1	Chebyshev map	8.8818E-016	8.8818E-016	8.8818E-016	8.8818E-016	N/A	1.5635
2	Circle map	8.8818E-016	8.8818E-016	8.8818E-016	8.8818E-016	N/A	1.5627
3	Gauss/Mouse map	8.8818E-016	8.8818E-016	8.8818E-016	8.8818E-016	N/A	1.5561

4	Iterative map	8.8818E-016	8.8818E-016	8.8818E-016	8.8818E-016	N/A	1.5663
5	Logistic map	8.8818E-016	8.8818E-016	8.8818E-016	8.8818E-016	N/A	1.5547
6	Piecewise map	8.8818E-016	8.8818E-016	8.8818E-016	8.8818E-016	N/A	1.5548
7	Sine map	8.8818E-016	8.8818E-016	8.8818E-016	8.8818E-016	N/A	1.5600
8	Singer map	8.8818E-016	8.8818E-016	8.8818E-016	8.8818E-016	N/A	1.5686
9	Sinusoidal map	6.4717E-002	1.3612E-001	1.0353E-001	1.0228E-001	1.5995 E-002	1.5853
10	Tent map	8.8818E-016	8.8818E-016	8.8818E-016	8.8818E-016	N/A	1.5534

Table 11. Results of CFOA for Griewank function (F9) after 50 independent runs

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]
1	Chebyshev map	**<1E-16 (Iter. 19)***	<1E-16	<1E-16	<1E-16	N/A	0. 4288
2	Circle map	<1E-16 (Iter. 29)*	<1E-16	<1E-16	<1E-16	N/A	0. 4269
3	Gauss/Mouse map	<1E-16 (Iter. 23)*	<1E-16	<1E-16	<1E-16	N/A	0. 42
4	Iterative map	<1E-16 (Iter. 22)*	<1E-16	<1E-16	<1E-16	N/A	0. 4256
5	Logistic map	<1E-16 (Iter. 21)*	<1E-16	<1E-16	<1E-16	N/A	0. 418
6	Piecewise map	<1E-16 (Iter. 28)*	<1E-16	<1E-16	<1E-16	N/A	0. 4193
7	Sine map	<1E-16 (Iter. 21)*	<1E-16	<1E-16	<1E-16	N/A	0. 4249
8	Singer map	<1E-16 (Iter. 21)*	<1E-16	<1E-16	<1E-16	N/A	0. 4302
9	Sinusoidal map	5.3456E-006	8.331E-006	8.082E-006	1.0221E-005	1.1899 E-006	0. 4356
10	Tent map	<1E-16 (Iter. 27)*	<1E-16	<1E-16	<1E-16	N/A	0. 4199

* Iteration in which the best result is achieved.

Table 12. Results of CFOA for Alpine function (F10) after 50 independent runs

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]
1	Chebyshev map	**<1E-156 (Iter. 539)***	<1E-156	<1E-156	<1E-156	N/A	0. 5068
2	Circle map	1.4005E-117	2.6463E-116	7.2709E-116	6.3057E-115	1.1476E-115	0. 383
3	Gauss/Mouse map	6.8852E-149	1.1367E-148	1.3004E-148	3.1424E-148	5.8945E-149	0. 3781
4	Iterative map	<1E-156 (Iter. 655)*	<1E-156	<1E-156	<1E-156	N/A	0. 4178
5	Logistic map	<1E-156 (Iter. 628)*	<1E-156	<1E-156	<1E-156	N/A	0. 4347
6	Piecewise map	2.1937E-140	5.5474E-140	6.5224E-140	2.1841E-139	3.6248E-140	0. 3776
7	Sine map	<1E-156 (Iter. 643)*	<1E-156	<1E-156	<1E-156	N/A	0. 4257
8	Singer map	<1E-156 (Iter. 605)*	<1E-156	<1E-156	<1E-156	N/A	0. 4675
9	Sinusoidal map	3.2171E-001	5.0950E-001	5.147E-001	6.8731E-001	7.5072E-002	0. 3844
10	Tent map	2.2373E-135	5.7899E-135	6.5307E-135	1.3816E-134	2.5618E-135	0. 377

* Iteration in which the best result is achieved.

Table 13. Results of CFOA for Powell function (F11) after 50 independent runs

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]
1	Chebyshev map	**<1E-309 (Iter. 541)***	<1E-309	<1E-309	<1E-309	N/A	0. 6053
2	Circle map	1.6305E-231	6.1197E-229	1.4662E-227	2.8049E-226	N/A	0. 4998
3	Gauss/Mouse map	1.4981E-294	6.3239E-294	1.5752E-293	1.5329E-292	N/A	0. 4930
4	Iterative map	<1E-309 (Iter. 656)*	<1E-309	<1E-309	<1E-309	N/A	0. 53
5	Logistic map	<1E-309 (Iter. 630)*	<1E-309	<1E-309	<1E-309	N/A	0. 5413

6	Piecewise map	3.0790E-277	1.0412E-276	1.8439E-276	2.113E-275	N/A	0. 4943
7	Sine map	<1E-309 (Iter. 645)*	<1E-309	<1E-309	<1E-309	N/A	0. 5387
8	Singer map	<1E-309 (Iter. 606)*	<1E-309	<1E-309	<1E-309	N/A	0. 5692
9	Sinusoidal map	3.2176E+001	5.4547E+001	5.5258E+001	8.2022E+001	8.8919	0. 4778
10	Tent map	2.2341E-267	1.8193E-266	2.4639E-266	1.1808E-265	N/A	0. 4919

* Iteration in which the best result is achieved.

Table 14. Results of CFOA for Rastrigin function (F12) after 50 independent runs

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]
1	Chebyshev map	**<1E-14 (Iter. 28)***	<1E-14	<1E-14	<1E-14	N/A	0.4356
2	Circle map	<1E-14 (Iter. 44)*	<1E-14	<1E-14	<1E-14	N/A	0.4288
3	Gauss/Mouse map	<1E-14 (Iter. 35)*	<1E-14	<1E-14	<1E-14	N/A	0. 4209
4	Iterative map	<1E-14 (Iter. 34)*	<1E-14	<1E-14	<1E-14	N/A	0. 4264
5	Logistic map	<1E-14 (Iter. 32)*	<1E-14	<1E-14	<1E-14	N/A	0. 4187
6	Piecewise map	<1E-14 (Iter. 41)*	<1E-14	<1E-14	<1E-14	N/A	0. 4216
7	Sine map	<1E-14 (Iter. 32)*	<1E-14	<1E-14	<1E-14	N/A	0.4239
8	Singer map	<1E-14 (Iter. 31)*	<1E-14	<1E-14	<1E-14	N/A	0. 4334
9	Sinusoidal map	1.7288E+002	2.2372E+002	2.2186E+002	2.7775E+002	2.3696E+001	0. 4409
10	Tent map	<1E-14 (Iter. 41)*	<1E-14	<1E-14	<1E-14	N/A	0. 4201

* Iteration in which the best result is achieved.

Table 15. Results of CFOA for Generalized Penalized 1 function (F13) after 50 independent runs

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]
1	Chebyshev map	1.1228	1.5302	1.5557	1.593	8.7364 E-002	1.2749
2	Circle map	1.2706	1.5181	1.5531	1.6084	8.8587 E-002	1.2476
3	Gauss/Mouse map	1.4398	1.5413	1.5461	1. 5976	4.1850 E-002	1.3467
4	Iterative map	1.3689	1.5397	1.5546	1.6112	6.4712 E-002	1.5209
5	Logistic map	1.1022	1.502	1.5473	1.6018	1.1446 E-001	1.1897
6	Piecewise map	1.1220	1.5191	1.5421	1.6027	9.7464 E-002	1.1677
7	Sine map	1.2849	1.5158	1.5431	1.5921	8.1600 E-002	1.1703
8	Singer map	1.1282	1.5022	1.5413	1.6061	1.0507 E-001	1.1879
9	Sinusoidal map	1.1335	1.516	1.5368	1.6114	9.1038 E-002	1.1865
10	Tent map	1.3973	1.5442	1.5574	1.5899	5.1043 E-002	1.1604

* Iteration in which the best result is achieved.

Table 16. Results of CFOA for Solomon problem function (F14) after 50 independent runs

No.	Chaotic map	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]
1	Chebyshev map	**<1E-155 (Iter. 532)***	<1E-155	<1E-155	<1E-155	N/A	0. 3866
2	Circle map	3.5685E-119	1.9739E-117	7.5868E-118	1.262E-116	3.1014 E-117	0. 3999
3	Gauss/Mouse map	1.1417E-150	2.904E-150	2.758E-150	5.6703E-150	1.1667 E-150	0. 3857
4	Iterative map	<1E-155 (Iter. 651)*	<1E-155	<1E-155	<1E-155	N/A	0. 3954
5	Logistic map	<1E-155 (Iter. 622)*	<1E-155	<1E-155	<1E-155	N/A	0. 4044

6	Piecewise map	6.0266E-142	1.1025E-141	1.0353E-141	2.52E-141	4.4473 E-142	0. 4699
7	Sine map	<1E-155 (Iter. 644)*	<1E-155	<1E-155	<1E-155	N/A	0. 4261
8	Singer map	<1E-155 (Iter. 598)*	<1E-155	<1E-155	<1E-155	N/A	0. 4306
9	Sinusoidal map	6.1731E-002	1.0683E-001	1.0454E-001	1.6998E-002	3.2026 E-003	0.3965
10	Tent map	5.4541E-137	1.4337E-136	1.0454E-136	6.7018E-137	2.3030 E-135	0. 4127

* Iteration in which the best result is achieved.

Table 17. Success rate of CFOA for different chaotic maps

No.	Chaotic map	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14
1	Chebyshev map	100	100	100	100	100	100	0	100	100	100	100	100	0	100
2	Circle map	100	100	100	100	100	100	0	100	100	100	100	100	0	100
3	Gauss/Mouse map	100	100	100	100	100	100	0	100	100	100	100	100	0	100
4	Iterative map	100	100	100	100	100	100	0	100	100	100	100	100	0	100
5	Logistic map	100	100	100	100	100	100	0	100	100	100	100	100	0	100
6	Piecewise map	100	100	100	100	100	100	0	100	100	100	100	100	0	100
7	Sine map	100	100	100	100	100	100	0	100	100	100	100	100	0	100
8	Singer map	100	100	100	100	100	100	0	100	100	100	100	100	0	100
9	Sinusoidal map	0	0	0	0	0	100	0	0	100	0	0	0	0	100
10	Tent map	100	100	100	100	100	100	0	100	100	100	100	100	0	100

Additionally, the algorithm is tested using random values instead of chaotic sequences. Particularly, alpha parameter in Eq. (3) is drawn from the uniform distribution (within range [0,1]), and results are presented in Table 18. It is evident that this algorithm with random alpha successfully converged in same benchmark problems. However, employment of chaotic variable proved the be the optimal choice since it can be observed that in all cases algorithm with chaos alpha converged faster.

Table 18. Results of CFOA for Solomon problem function (F14) after 50 independent runs

Function No.	Best	Worst	Median	Mean	Std. dev.	Ave. time [s]
F1	<1E-309 (Iter. 686)*	<1E-309	<1E-309	<1E-309	N/A	0.34453
F2	<1E-155 (Iter. 688)*	<1E-155	<1E-155	<1E-155	N/A	0.3762
F3	2.5522E-005	8.31173E-004	2.7058E-004	2.15436E-004	2.2342E-004	0.5472
F4	<1E-308 (Iter. 677)*	<1E-308	<1E-308	<1E-308	N/A	0.3736
F5	<1E-309 (Iter. 688)*	<1E-309	<1E-309	<1E-309	N/A	0.5144
F6	<1E-155 (Iter. 669)*	<1E-155	<1E-155	<1E-155	N/A	0.3457
F7	2.8544 E+001	2.8751 E+001	2.8692 E+001	2.8711 E+001	6.2931E-002	0.4075
F8	8.8818E-016	8.8818E-016	8.8818E-016	8.8818E-016	N/A	2.8363
F9	<1E-16 (Iter. 36)*	<1E-16	<1E-16	<1E-16	N/A	0.4037
F10	<1E-156 (Iter. 680)*	<1E-156	<1E-156	<1E-156	N/A	0.3862
F11	<1E-309 (Iter. 677)*	<1E-309	<1E-309	<1E-309	N/A	0.5064
F12	<1E-14 (Iter. 54)*	<1E-14	<1E-14	<1E-14	N/A	0.3644
F13	1.4117	1.6081	1.5587	1.565	3.8052E-002	1.1292
F14	<1E-155 (Iter. 674)*	<1E-155	<1E-155	<1E-155	N/A	0.3856

* Iteration in which the best result is achieved.

5.2. Comparison study

5.2.1 Comparison with FOA based methods

In order to further evaluate the developed algorithm, we compared CFOA with basic FOA and FOA with Levy distribution. We implemented Chebyshev map since the previous results indicate that this is the best choice in all test cases. Similarly to the experiment showed above, we tested these algorithms on ten selected benchmark problems with 50 independent runs. Results are presented in Table 19. Convergence curves for obtained mean values of these algorithms are showed in Fig. 5 – Fig. 14.

From Table 19 it is evident that CFOA gives best results in comparison with other two algorithms. For each optimization function CFOA proved to be the best choice in terms of obtained statistical measures. Likewise, success rate of CFOA in each tested case reached maximum of 100%, while the FOA and LFOA showed much worse results. Only in the case of optimization of function F5, all of three algorithms converged with maximum rate. These results indicate that integration of chaos in FOA significantly improves overall algorithm performance.

Table 19. Comparison results of CFOA, basic FOA, and Levy FOA on ten selected benchmark problems

ID	Algorithm	Best	Mean	Median	Worst	Std. dev.	Ave. time [s]	Success rate
F1	FOA	4.4457E-005	1.2112	1.6675	2.3415	9.2286E-001	0.2913	38
	LFOA	4.5391E-001	0.6892	7.4226E-001	1.1459	1.7101E-001	2.2122	0
	CFOA	<1E-309	<1E-309	<1E-309	<1E-309	N/A	0.5244	100
F2	FOA	3.6472E-002	1.0476	3.70934E-002	3.9644	1.7267	0.3364	0
	LFOA	1.1649	1.6231	1.6224	1.9694	1.7991E-001	2.2892	0
	CFOA	<1E-155	<1E-155	<1E-155	<1E-155	N/A	0. 5753	100
F3	FOA	1.1774E-003	8.6099E+001	3.046E-003	8.4905E+002	2.5261E+002	0.5436	36
	LFOA	7.4109 E+001	1.9794E+002	1.7291E+002	5.0049E+002	1.1246E+002	2.4189	0
	CFOA	<1E-309	<1E-309	<1E-309	<1E-309	N/A	0. 6053	100
F4	FOA	6.8805E-004	4.9840	6.4152E+000	9.1661E+000	3.6883E+000	0.3839	38
	LFOA	1.1915E+0	2.4598	2.3191E+00	3.393E+000	4.9255E-	2.2937	0

		00		0		001		
	CFOA	<1E-14	<1E-14	<1E-14	<1E-14	N/A	0.4356	100
F5	FOA	1.48849E-006	1.5270E+004	1.511703E-006	4.0388E+006	5.8043E+005	0.6464	80
	LFOA	1.3391E+001	2.3069E+004	6.1164E+003	1.1518E+003	1.6189E+005	2.4914	0
	CFOA	7.2192E-008	1.0440E-004	2.0866E-005	1.5060E-005	1.9906E-005	0. 6352	100
F8	FOA	5.3905E-004	7.7500E-002	9.3652E-002	1.2312E-001	4.7789E-002	3.052	30
	LFOA	2.7445E-002	4.1907E-002	4.0171E-002	5.1889E-002	5.6992E-003	4.8830	100
	CFOA	<1E-308	<1E-308	<1E-308	<1E-308	N/A	0.5009	100
F9	FOA	2.3209E-006	6.4021E-006	7.1121E-006	9.7443E-006	2.1991E-006	0.4552	100
	LFOA	1.2485E-006	2.4056E-006	2.3591E-006	3.4427E-006	4.979E-007	2.3726	100
	CFOA	<1E-309	<1E-309	<1E-309	<1E-309	N/A	0. 6196	100
F10	FOA	3.7878E-001	5.1800E-001	5.2334E-001	6.5158E-001	6.5343E-002	0.4397	0
	LFOA	3.6139E-001	4.9140E-001	4.9818E-001	6.3516E-001	6.5539E-002	2.4056	0
	CFOA	8.8818E-016	8.8818E-016	8.8818E-016	8.8818E-016	N/A	1.5534	100
F11	FOA	1.2561E-003	2.3872E+001	1.3007E+001	6.5019E+001	2.5369E+001	0.5163	54
	LFOA	5.1671E+000	1.0132E+001	1.0344E+001	2.0608E+001	3.6303	2.4252	0
	CFOA	<1E-16	<1E-16	<1E-16	<1E-16	N/A	0. 4288	100
F12	FOA	8.8347E-003	1.2585E+002	1.928E+002	2.7090E+002	1.0926E+002	0.339	34
	LFOA	6.088E+001	7.7409E+001	7.4376E+001	1.0089E+002	8.1507E+000	2.2066	0

CFOA	<1E-156	<1E-156	<1E-156	<1E-156	N/A	0. 5068	100

Similarly to what is found in Table 14, results presented in Fig. 5- Fig.14 show the superiority of CFOA (note that values on x-axis and y-axis depicts iterations and fitness, respectively). For every optimization task proposed algorithm successfully found the optimal solution in minimal number of iterations. In comparison with FOA and LFOA, CFOA converge much faster and produce significantly better final outcome. It is also important to note that the CFOA initialization (as described in Section 3) has a major influence on algorithm performance and convergence.

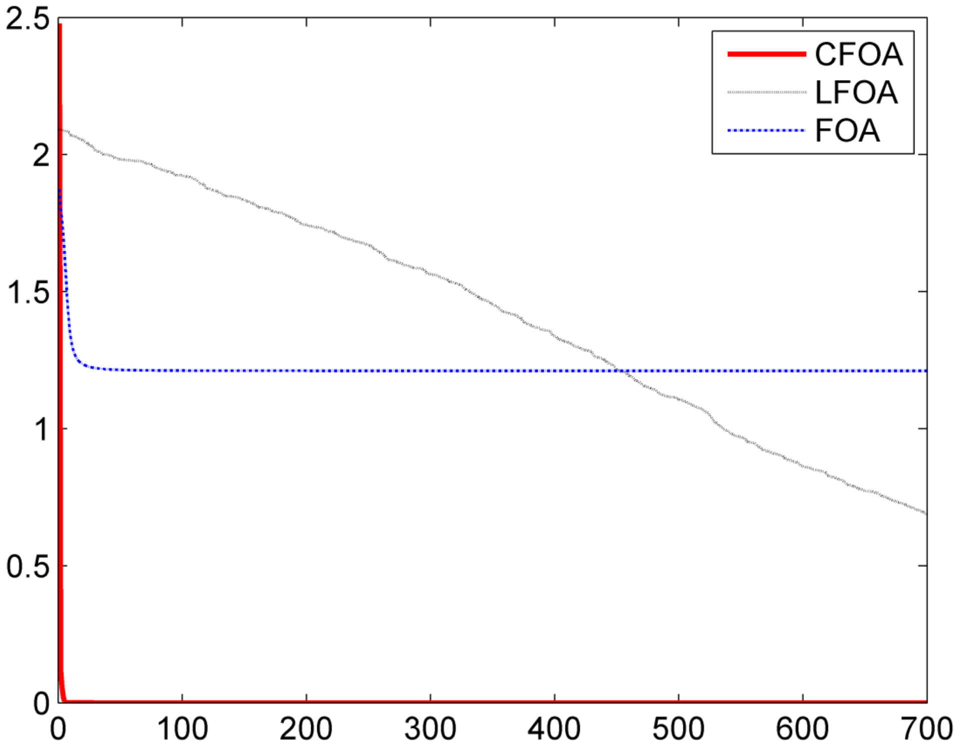


Fig.5 CFOA. LFOA, and FOA convergence curves for F1

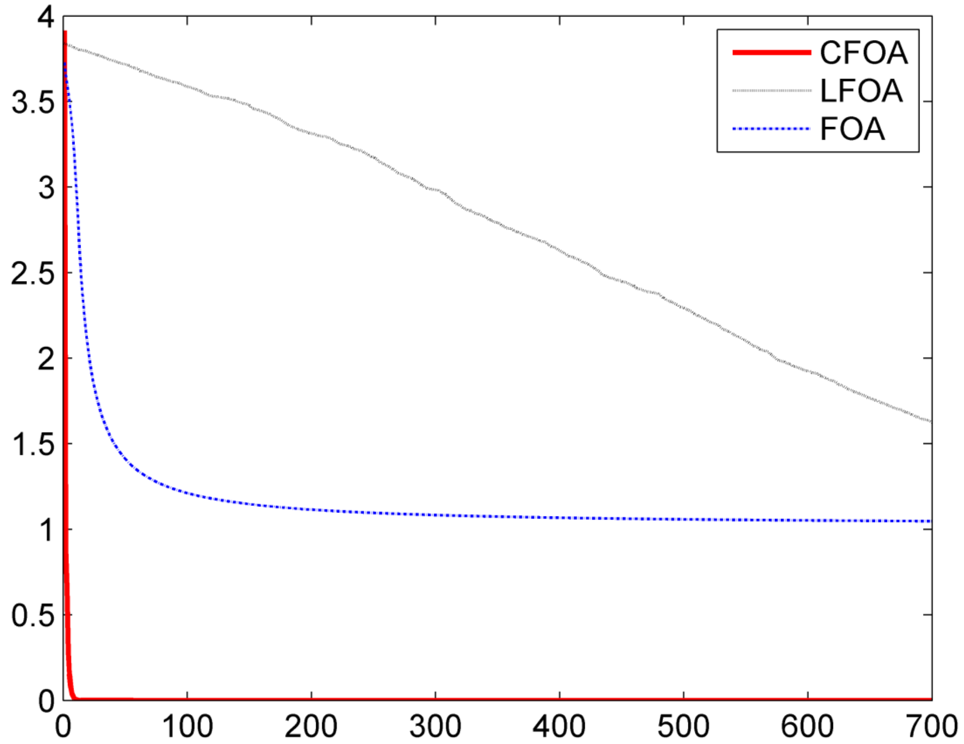


Fig.6 CFOA. LFOA, and FOA convergence curves for F2

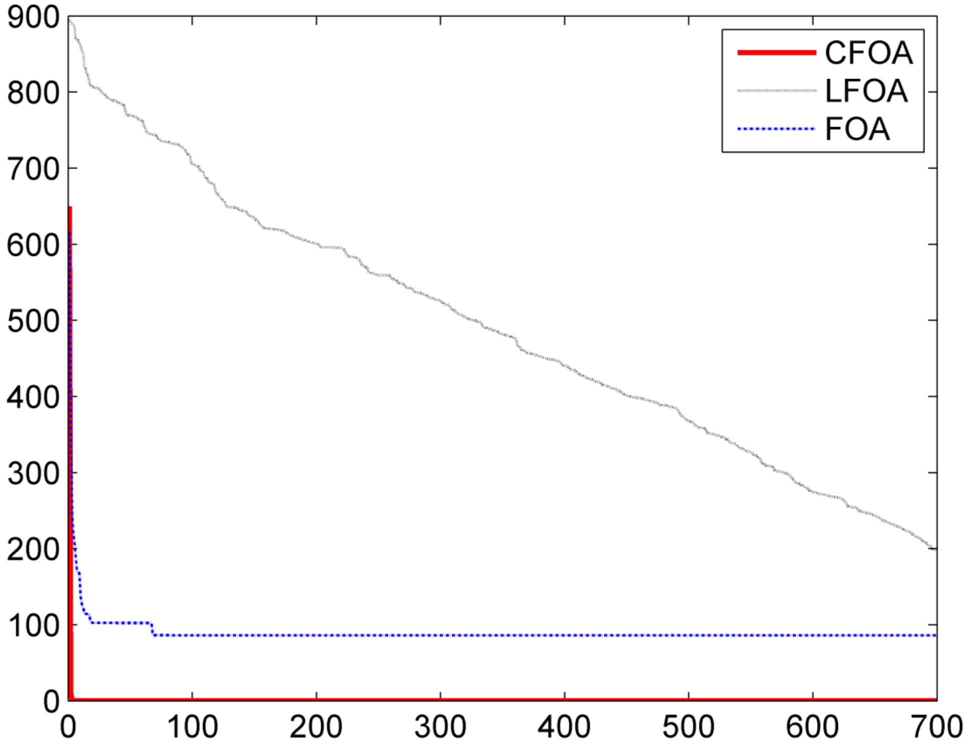


Fig.7 CFOA. LFOA, and FOA convergence curves for F3

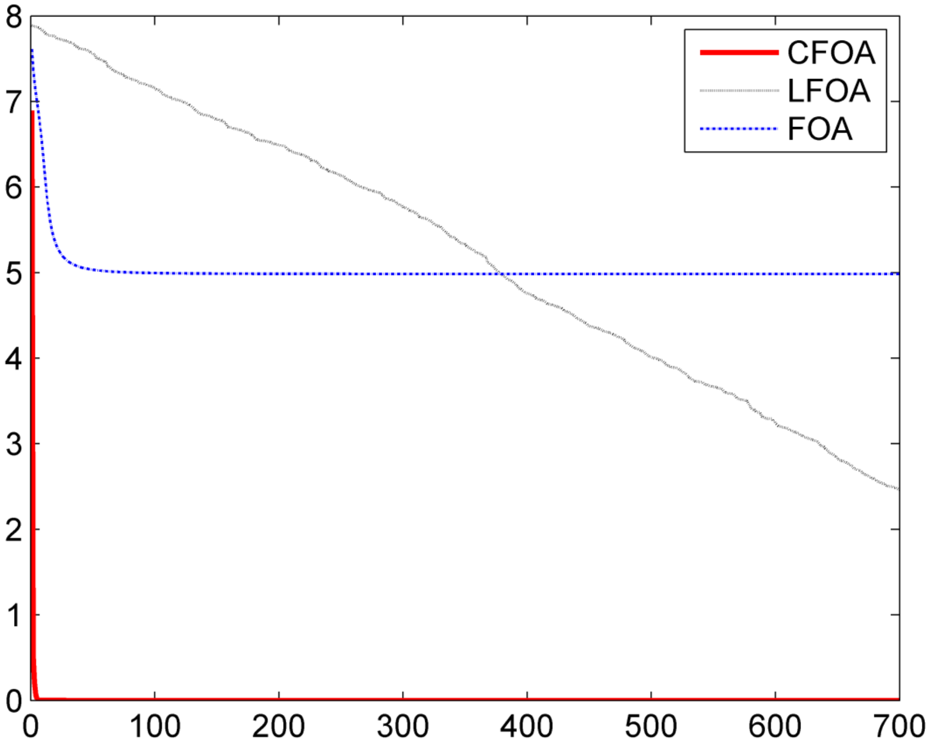


Fig.8 CFOA. LFOA, and FOA convergence curves for F4

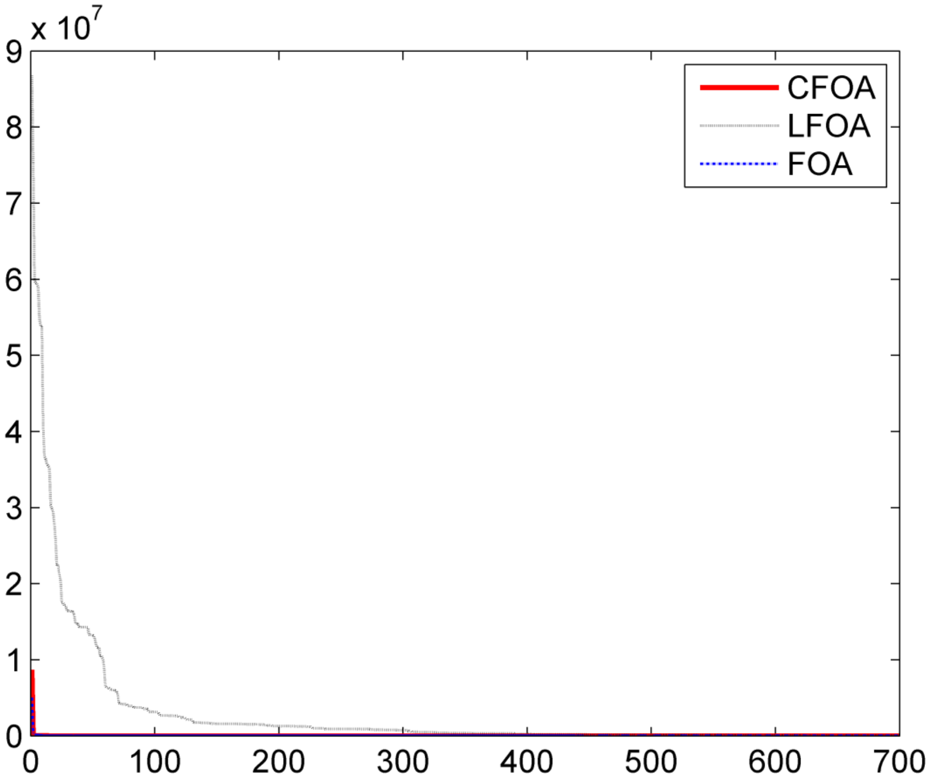


Fig.9 CFOA. LFOA, and FOA convergence curves for F5

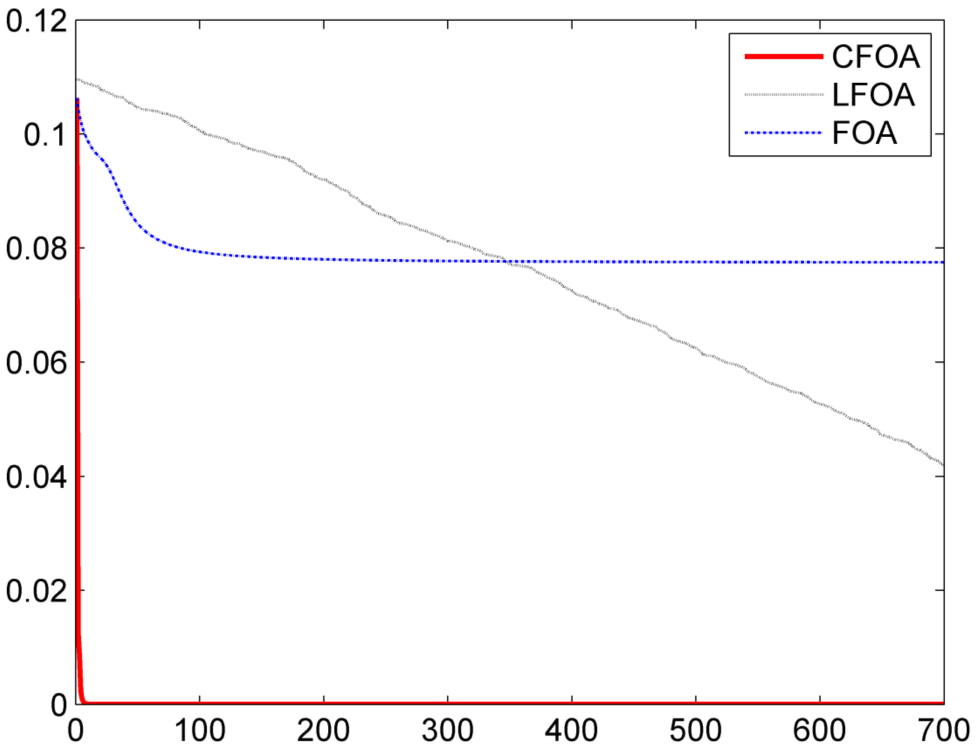


Fig.10 CFOA. LFOA, and FOA convergence curves for F8

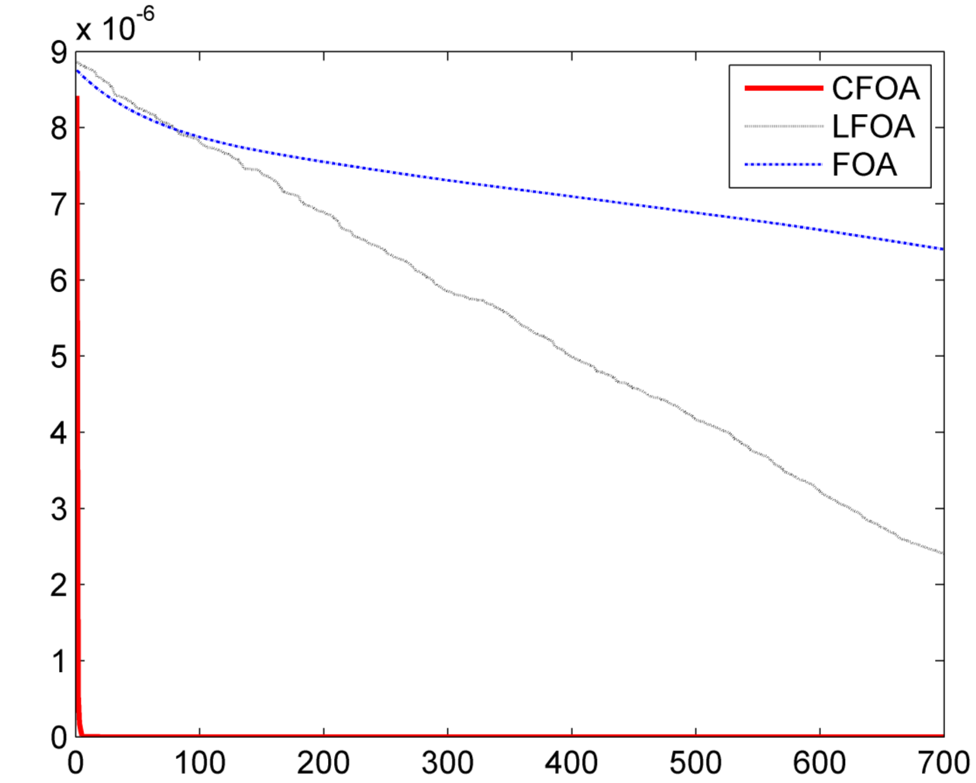


Fig.11 CFOA. LFOA, and FOA convergence curves for F9

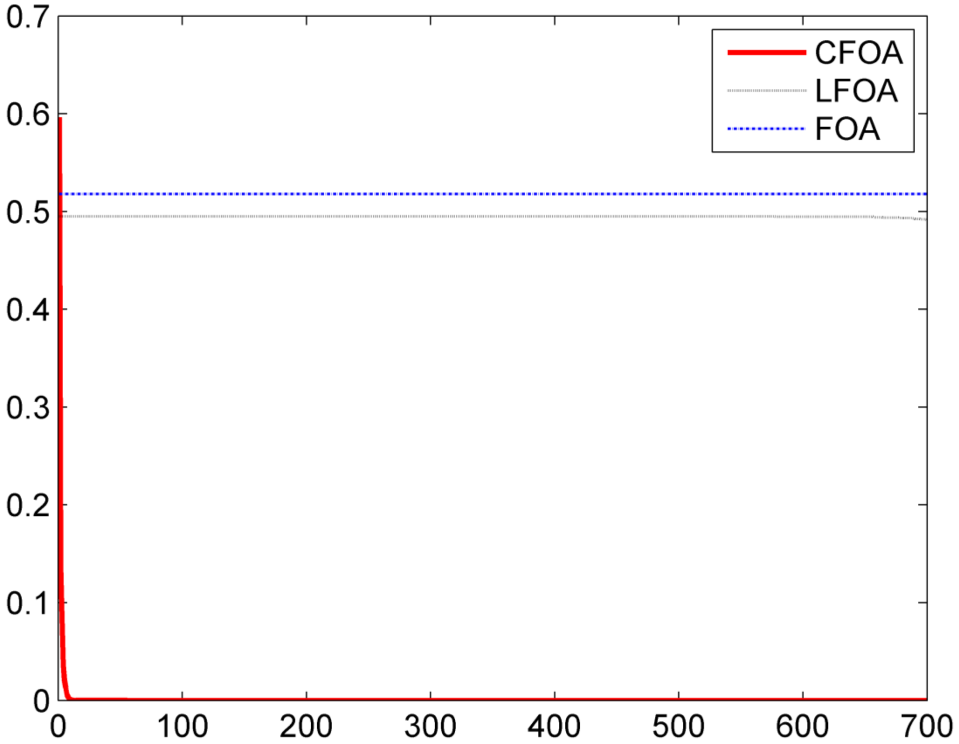


Fig.12 CFOA. LFOA, and FOA convergence curves for F10

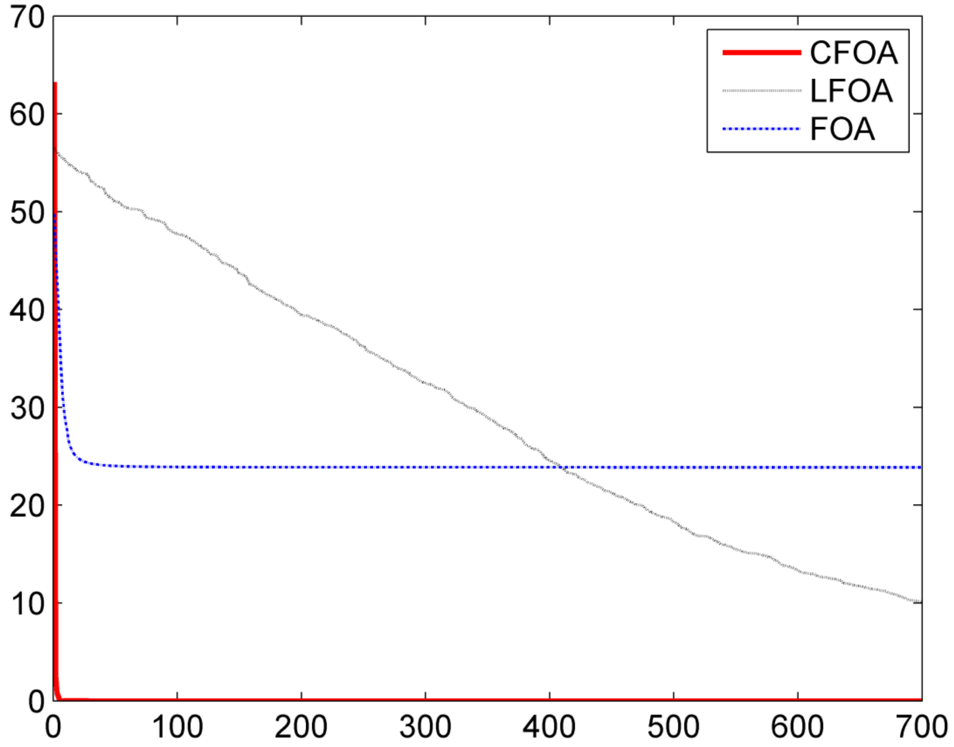


Fig.13 CFOA. LFOA, and FOA convergence curves for F11

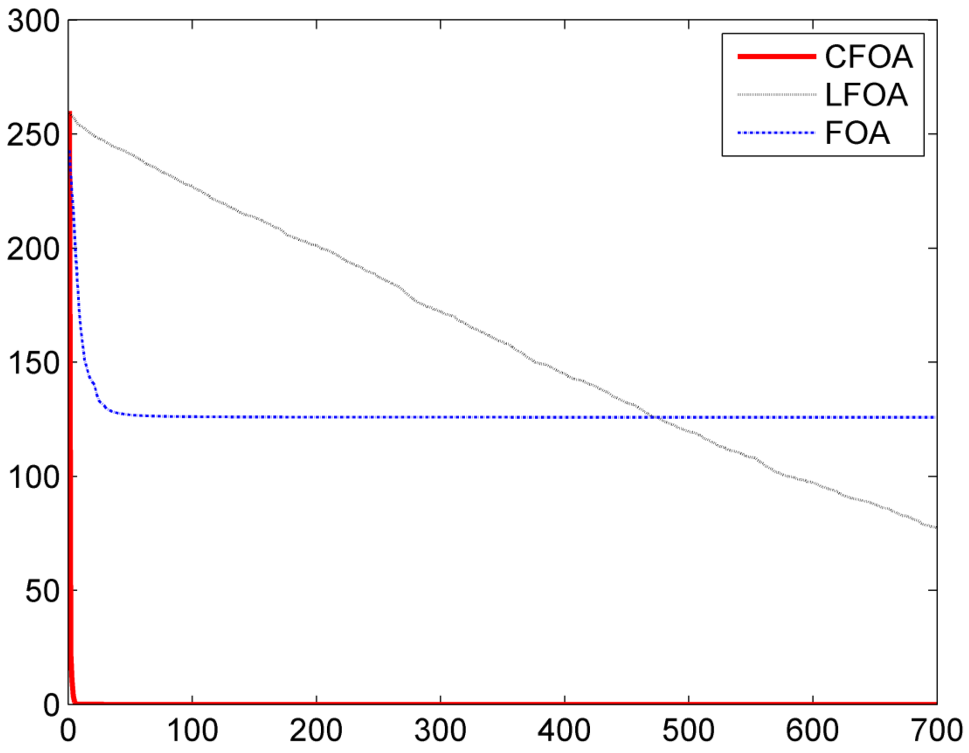


Fig.14 CFOA. LFOA, and FOA convergence curves for F12

Furthermore, we compare developed CFOA with other recently introduced FOA based methods. Similarly to previous case, Table 20 show superiority of CFOA in comparison with algorithms presented in literature [14, 28].

Table 20. Comparison of CFOA with other recently introduced FOA-based algorithms

Algorithm Function	iFOA [14]	mFOA [28]	CFOA
F1 (Sphere)	4.96E-013	3.1E-004	**<1E-309**
F7 (Rosenbrock)	/	**9.770**	28.321
F8 (Ackley)	4.85E-011	0.017	**8.8818E-016**
F9 (Griewank)	1.23E-002	0.015	**<1E-16**
F12 (Rastrigin)	4.78E-010	4.502	**<1E-14**

5.2.2 Comparison with other chaos enchanced metaheuristic methods

In addition to the above experiments, we compare the CFOA with recently developed state-of-the-art chaotic algorithms. Besides CFOA, we implemented chaotic bat algorithm (CBA), chaotic accelerated particle swarm optimization (CAPSO), chaotic firefly algorithm (CFA), chaotic artificial bee colony algorithm (CABC), and chaotic cuckoo search (CCS).

Population size, maximum number of iterations, and specific algorithm parameters are presented in Table 21, while comparison results are given in Table 22. All aforementioned algorithms are implemented and tested in Matlab software under the same initial conditions. The algorithms are tested on 3 unimodal and 3 multimodal functions with global optimum equal to zero. Best chaotic maps are employed for each individual algorithm: Sinusoidal map for CBA and CAPSO [17,19], Logistic map for CCS [29], and Gauss map for CABC and CFA [22, 24]. Results indicate that the CFOA outperforms all other algorithms in terms of mean final optimization result, except for the F8 and F9 in which the identical result is obtained as with employed CABC. This is an additional proof that the implementation of chaotic component remarkably improves the standard FOA. Finally, Friedman test also confirms the superiority of CFOA (Table 22).

Table 21. Setup for comparison experiments

Algorithm	Population size (PS)	Max number of iterations	Specific algorithm parameters
CBA	50	700	$A = 0.5$; $Q_{\min} = 0$; $Q_{\max} = 2$; r is chaotic variable
CAPSO	50	700	$\gamma = 0.7$; $\alpha = \gamma^i$, $i = iteration$; β is chaotic variable
CFA	50	700	$\alpha = 0.2$; $\gamma = 1$; $\delta = 0.97$; $\beta_0 = 1$; β is chaotic variable
CABC	50	700	$food\_numner = PS/2 = 25$; $limit = 100$; $c_{i,j}$ is chaotic variable
CCS	50	700	$p_a = 0.25$; c_s is chaotic variable

Table 22. Comparison of CFOA with other recent chaotic algorithms (best results are given in bold).

Algorithm Function	CBA	CAPSO	CFA	CABC	CCS	CFOA
F1 (Sphere)	5.73E-012	9.74E-221	4.65E-023	1.55E-024	4.76E-031	**<1E-309**
F2 (Schwefel's problem 2.22)	3.27E-006	2.38E-110	8.07E-012	1.02E-018	9.68E-017	**<1E-155**
F5 (Sum of different power)	1.98E-014	6.49E-018	2.43E-015	7.00E-022	3.24E-040	**<1E-309**
F8 (Ackley)	1.71E-005	4.44E-015	3.67E-012	**8.8818E-016**	6.13E-014	**8.8818E-016**
F9 (Griewank)	4.65E-012	7.40E-003	2.96E-002	**<1E-16**	3.31E-014	**<1E-16**
F10 (Alpine)	5.32E-007	1.73E-111	4.88E-013	2.22E-016	2.66E-010	**<1E-156**
Friedman test rank	5.6667	3.0	5.0	2.6667	3.5	**1.1667**

6. Conclusions

Implementation of chaos in metaheuristic algorithms has become increasingly popular in research community. Because of the ergodicity and mixing property of chaos, these types of algorithms show better behavior than original variants in terms of final outcome. In this paper we investigate the effectiveness of ten different chaotic maps in improving the performance of recently developed fruit fly optimization algorithm (FOA). In order to fairly compare these maps, we employ fourteen different well known unimodal and multimodal functions. By introducing new chaotic parameter and different initialization setting in basic FOA, novel algorithm is enhanced in terms of convergence speed and overall performance. Statistical results on every optimization task indicate that the chaotic fruit fly algorithm (CFOA) has a very fast convergence. Moreover, Chebyshev map proved to be the best map in terms of final algorithm solution. Likewise, we compare the CFOA with the standard FOA and Levy flight FOA as well as other FOA based approaches in terms of finding global optimum of a particular function. Additionally, the comparison study with other state-of-the-art chaos enhanced algorithms such as chaotic bat algorithm, chaotic accelerated particle swarm optimization, chaotic firefly algorithm, chaotic artificial bee colony algorithm, and chaotic cuckoo search is provided. CFOA proved to be better or at least as equal to the performance of aforementioned algorithm in all test cases. Finally, it can be concluded that overall findings confirm the superiority of CFOA in terms of statistical results, reliability of global optimality, and algorithm success rate.

In future research studies, CFOA could be applied for solving real world engineering problems. Specifically, the performance of the CFOA in structural model updating, gear train design, and pressure vessel design needs to be investigated. Likewise, it would be interesting to see the influence of other chaotic maps on the developed algorithm.

Acknowledgement

This research is supported by the Serbian Government – the Ministry of Education, Science and Technological Development under grant TR35004 (2011-2015).

References

1. Gendreau, M, Potvin, J. Y. Handbook of metaheuristics (Vol. 2). New York: Springer; 2010

2. Yang XS. Nature-inspired metaheuristic algorithms. Luniver Press; 2008

3. Lee KS, Geem ZW. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput Meth Appl Mech Eng 2005;194: 3902–33.

4. Yang XS. Engineering optimization: an introduction with metaheuristic applications. John Wiley & Sons; 2010

5. Yang XS, Deb S. Engineering optimisation by cuckoo search. Int J Math Model Num Opt 2010;1(4):330-43.

6. Yang XS. A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010) 2010, p.65-74.

7. Gandomi AH, Alavi AH. Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simulat 2012;17(12):4831-45.

8. Yang XS, Gandomi AH, Talatahari S., Alavi AH (Eds.). Metaheuristics in water, geotechnical and transport engineering. Elsevier; 2012

9. Pan WT. A new fruit fly optimization algorithm: taking the financial distress model. Knowl Based Syst 2012;26:69–74.

10. Li C, Xu S, Li W, Hu L. A novel modified fly optimization algorithm for designing the self-tuning proportional integral derivative controller. J Converg Inform Technol 2012;7:69–77.

11. Zheng XL, Wang L, Wang, SY. A novel fruit fly optimization algorithm for the semiconductor final testing scheduling problem. Knowl Based Syst 2014;57:95-103.

12. Li H, Guo S, Li C, Sun J. A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm. Knowl Based Syst 2013;37:378–87.

13. Wang L, Zheng XL, Wang SY. A novel binary fruit fly optimization algorithm for solving the multidimensional knapsack problem. Knowl Based Syst 2013;48:17–23.

14. Pan QK, Sang HY, Duan JH, Gao L. An improved fruit fly optimization algorithm for continuous function optimization problems. Knowl Based Syst 2014;62:69-83.

15. Saremi S, Mirjalili S, Lewis A. Biogeography-based optimisation with chaos. Neural Comput Appl 2014; doi: 10.1007/s00521-014-1597-x

16. Coelho L, Mariani VC. Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization. Expert Syst Appl 2008;34:1905–13.

17. Gandomi AH., Yang XS. Chaotic bat algorithm. J Comput Sci 2014;5(2):224-32.

18. Gharoonifard G, Moein-darbari F, Deldari H, Morvaridi A. Scheduling of scientific workflows using a chaos-genetic algorithm. Proc Comput Sci 2010;1:1445–54.

19. Gandomi AH, Yun GJ, Yang XS, Talatahari S. Chaos-enhanced accelerated particle swarm algorithm. Commun Nonlinear Sci Numer Simulat 2013;18(2):327–40.

20. Alatas B. Chaotic harmony search algorithms. Appl Math Comput 2010;216:2687–99

21. Gong W, Wang S. Chaos ant colony optimization and application. In: 4th International Conference on Internet Computing for Science and, Engineering, 2009, p.301–303.

22. Alatas B. Chaotic bee colony algorithms for global numerical optimization. Expert Syst Appl 2010;37:5682–7.

23. Mingjun J, Huanwen T. Application of chaos in simulated annealing. Chaos Soliton Fract 2004;21:933–41.

24. Gandomi AH, Yang XS, Talatahari S, Alavi AH. Firefly algorithm with chaos. Commun Nonlinear Sci Numer Simulat 2013;18(1):89–98.

25. Saremi S, Mirjalili SM, Mirjalili S. Chaotic krill herd optimization algorithm. Proc Techn 2014;12:180-5.

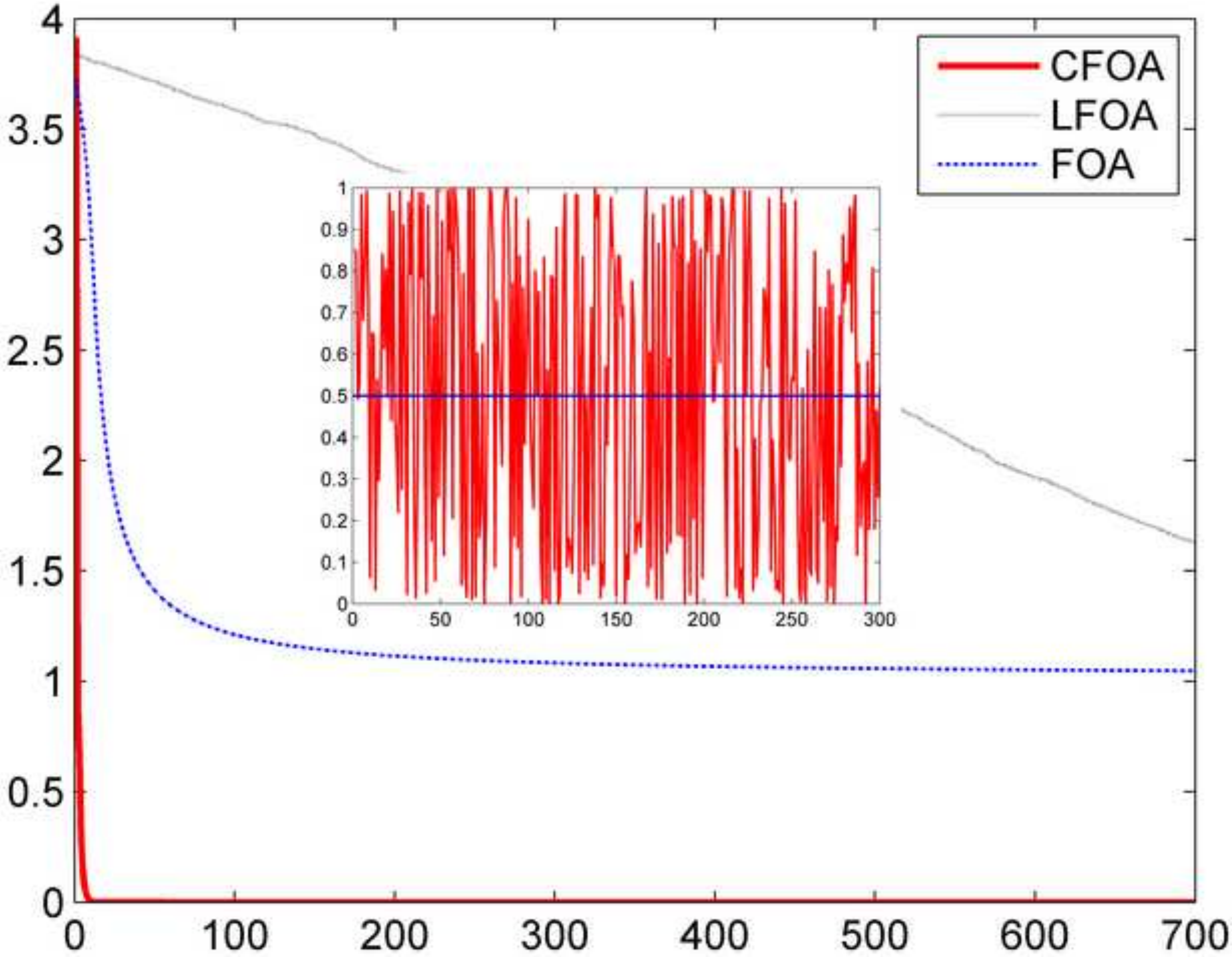
26. Cheng H, Liu CZ. Mixed Fruit Fly Optimization Algorithm Based on Chaotic Mapping. Comput Eng 2013;5:050.

27. Yao X, Liu Y, Lin G. Evolutionary programming madefaster. IEEE Trans Evol Comput 1999;3:82–102.

28. Pan WT. Using modified fruit fly optimisation algorithm to perform the function test and case studies. Connection Sci, 2013;25(2-3):151-160.

29. Lin JH, Lee HC. Emotional chaotic cuckoo search for the reconstruction of chaotic dynamics. Latest advances in systems science & computational intelligence, WSEAS Press, Athens, 2012, p.123-128.

Graphical Abstract



Highlights:

- Development of new method named chaotic fruit fly optimization algorithm (CFOA)
- Fruit fly algorithm (FOA) is integrated with ten different chaos maps
- Novel algorithm is tested on ten different well known benchmark problems
- CFOA is compared with FOA, FOA with Levy distribution, and similar chaotic methods
- Experiments show superiority of CFOA in terms of obtained statistical results