CHAOTIC CUCKOO OPTIMIZATION ALGORITHM FOR SOLVING GLOBAL OPTIMIZATION PROBLEMS

[1]SALWA EL- SAYED, [2]ABD EL-NASSER H. ZAYED, [3] MAHMOUD ISMAIL

[1] Teaching Assistant - Decision Support Department - Faculty of Computers & Informatics - Zagazig University

[2] Professor of Information Systems, Vice Dean for Student's affairs-Faculty of Computer Science, Misr International University (MIU)

[3]lecturer-Decision Support Department- Faculty of Computers & Informatics, Zagazig University

Email: [1]salwaelsayed_93@yahoo.com, [2]nasserhr@yahoo.com, [3]mahsabe@yahoo.com

ABSTRACT

In this paper, three chaotic algorithms based on Cuckoo Optimization Algorithm (COA) are introduced. For avoiding local optima and getting high convergence speed, chaotic theory is used. The first Chaotic Cuckoo Optimization Algorithm (CCOA1) uses chaotic maps to estimate Egg Laying Radius (ELR) coefficient. The second Chaotic Cuckoo Optimization Algorithm (CCOA2) uses chaotic maps to estimate the immigration coefficient (F). In the third Chaotic Cuckoo Optimization Algorithm (CCOA3), chaotic maps are incorporated in the immigration process to the goal point. Ten chaotic maps are applied to determine precisely which map can give the best results for each chaotic algorithm. To verify the efficiency of the proposed algorithms, a set of different types of benchmark problems is selected and tested. Also, Wilcoxon rank-sum test is performed to approve that the results are statistically significant. The results show how the three proposed algorithms can improve COA and also show the ability of CCOA3 to get better results than other compared algorithms. Besides, CCOA3 can achieve high convergence speed than COA and other proposed algorithms. Besides, three chaotic algorithms are tested on two engineering problems and compared with different algorithms from the literature in a fair comparison and results show the ability of the proposed algorithms in getting superior results.

Keywords: *Cuckoo Optimization Algorithm, Chaotic Maps, Metaheuristics, Optimization. Evolutionary Algorithms*

I. INTRODUCTION

Optimization is the process of finding the highest or least objective function value for a set of constraints. In other words, optimization means finding the best solution that can minimize or maximize objective function according to the given problem [1]. The optimization problem can be defined as how to get the values of the variables that can achieve the best value of the objective function. Mathematical optimization is essential because it works better than traditional methods (guess and check), and it also not time-consuming for solving a particular problem. Mathematical optimization is useful in many fields like scheduling, finance, inventory control, economics, transportation, etc.

According to [2], there are two methods of optimization. The first is exact methods, which means that a set of instructions is constructed to solve the problem. If these instructions are followed correctly, the optimal solution will obtain not just the good solution such as linear programming, integer programming, nonlinear programming, branch and bound, etc... While the second is approximate methods which have two types of algorithms, approximate algorithms that allow getting provable solution and heuristic algorithms which allow obtaining a good solution with acceptable performance. Heuristics derived from a Greek word, which means “to discover”, and it also means “rule of thumb.” The heuristic is a method that comes from experience and helps you to think through things such as the process of trial and error. Heuristic techniques help you to solve optimization problems faster than you will solve the problem by traditional methods. Heuristic techniques don’t guarantee an optimal solution to be got but only satisfy immediate goals, and they include CDS heuristics, NEH heuristics, Gupta’s heuristics, etc. Heuristic algorithms have two types, metaheuristics, and problem-specific heuristics. Problem-specific heuristics were constructed to solve a particular problem, and the algorithm can’t solve any other problem. While metaheuristics is a higher-level procedure designed to find the optimal solution (or near-optimal solution) for the optimization problem. Meta-heuristic algorithms can handle a wide range of problems, unlike traditional methods. There are many metaheuristic algorithms like Genetic Algorithm (GA) [3], Particle Swarm Optimization (PSO) [4], Ant Colony Optimization (ACO) [5], Artificial Bee Colony (ABC) [6], Cuckoo Search (CS) [7], Bat Algorithm (BA) [8] and etc.

All metaheuristic algorithms handle the problem in the same way because it has two essential processes that must be done. The first process called “exploration”, which means that the optimization begins with a collection of random solutions that will be combined and changed rapidly and randomly. The main goal in this process is to determine the desired area that can have a solution. The other process called “exploitation”, and the goal of this process is to determine the accuracy of the solutions which had got by exploration process by using fitness function and getting the global optimal solution. Metaheuristic algorithms have more merits, including flexibility in its behavior, because it can deal with different types of problems and also consider the problem as an anonymous box and control only inputs, outputs, and constraints of the given problem. It also has simplicity because it uses straightforward rules in bundles or swarms. Besides, metaheuristic algorithms can avoid local solutions and get the desired global optimal solution in a few numbers of iterations. The convergence speed of these algorithms is faster than traditional methods for solving the problem. These algorithms can also handle a large number of variables in contrast to traditional methods. According to [9], metaheuristic algorithms were divided into five major categories: swarm-based algorithms, nature-inspired algorithms, biogeographic-simulated algorithms, physics-based algorithms, and evolutionary algorithms. The inspiration of swarm-based algorithms based on the collective behavior of social insects such as bees, ants, and also birds. These algorithms depend on the interaction between swarm members and their environments like Ant Colony Optimization (ACO) [5], Artificial Bee Colony (ABC) [6], and Particle Swarm Optimization (PSO) [4]. The inspiration of nature-inspired algorithms based on Nature. Most of the current algorithms are nature-inspired such as Bat Algorithm (BA); [8] and Cuckoo Search (CS); [7]. The inspiration of biogeographic-simulated algorithms from biological organisms and it has two basic concepts, migration and mutation such as Spotted Hyena Optimizer (SHO) [10] and Grey Wolf Optimizer (GWO) [11]. Physics-based algorithms based on rules of physics. Scientists employ the laws of physics and chemistry to enhance optimization methods. By using these rules, it can transfer to discover the search space of the given problem and obtaining an optimal solution. The most popular physics-based algorithm is Simulated Annealing [12].

Evolutionary algorithms (EAs) were introduced by C. Darwin in 1859 when he showed evolution theory in his book [13]. EAs are random metaheuristics that can be stratified to a wide range of complicated problems. The success that was achieved by these algorithms leading to take attention from more researchers for studying these algorithms and making more developments for it. Evolutionary Algorithms such as Differential Evolution (DE) [14, 15], Genetic Algorithm (GA) [3], Evolution Strategy (ES) [16, 17].

It was the English Zoologist William Elford Leach who introduced the family Cuculidae (Cuckoos) in a guide to the contents of the British Museum that published in 1820 [18], but the evolutionary history remains unclear, and there is a minimal fossil record for them. Cuckoos are medium-sized birds that its size range from little bronze cuckoo at 17 g and 15 cm in length to Channel-billed cuckoo with 630 g and 63 cm in length [19]. These birds have slightly curved and somewhat longbows. They differ from most birds in that two of their fingers are pointing forward, and the other two are facing back. Both male and female cuckoo has white breasts with dark

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

stripes on it but, the head of the male and their appearance are grey while the head of the female and backs are grey or brown. Cuckoo was found in the ancient world throughout Europe, most of Asia and in Subsaharan Africa. Now, there are some species of cuckoos live in forests and woodland. The other some can live in open environments such as deserts, and there are migratory species that live in a wide range of habitats to achieve the most significant benefit from hosts. Cuckoo birds feed on insects and caterpillars (which is a butterfly larva and delicate father), for example, silkworms and cotton leaf worms including poisonous bristles caterpillars which other birds do not eat it.

The common cuckoo bird does not give care for its young like many other types of cuckoos in the ancient world but, it lays its eggs in the nest of other birds and leaves it until it hatches and then the other bird gives it the care as shown in figure 1. In other words, there are many species of cuckoo birds, which are brood parasites that never build their nests. Around 56 species and 3 in the new world are brood parasites [20]. Female cuckoo has the skill of deception to stealth and removes one egg laid by the host bird and lays its egg in the same place. Then it flies with the host egg in her bill. This process takes ten seconds at maximum. Matching eggs does not happen randomly, but cuckoo tries to imitate host eggs by attracting the color and pattern of their eggs in a way hoping not to be recognized by host birds. We can be marvel at this sophistication and intelligence, which this bird has to obligate host birds to host it. The question now, how host birds can recognize cuckoo egg? The answer to this question returns to the fact that many birds learn how to distinguish the strange egg and throw out of the nest. Host birds and cuckoos, each of them try to survive from the other [21]. More details about cuckoo lifestyle and its behavior in laying eggs will be discussed in the next section.

Figure 1: Brood parasitism cuckoo

2. CUCKOO OPTIMIZATION ALGORITHM (COA)

In 2011, Rajabioun introduced a new evolutionary algorithm called Cuckoo Optimization Algorithm [21]. The motivation of this algorithm based on the lifestyle of the cuckoo bird and its behavior in breeding and laying eggs. Like any other evolutionary algorithm, COA begins with an initial population of cuckoo birds. Each cuckoo has some eggs to lay in other nests (host nests). The eggs that are more similar to host eggs, only have the chance to grow and be a mature cuckoo. The others which are less likely to host eggs are recognized by host birds and throw out of the nest. So, it is more critical for cuckoos to determine the more suitable area (habitat) to lay eggs and guarantee survival for their youngsters. The more eggs that survive in a district, the more suitable habitat for laying eggs. After the growth of eggs and becoming mature, they compose societies, and each society has its habitat. The more eggs survive in one habitat, the more suitable habitat for all cuckoos to immigrate. The immigration process continues until most of the cuckoos are collected around the same position. The goal of COA is to optimize this position.

2.1 Generating Initial Cuckoo Habitat

For solving the optimization problems, habitat is represented with an array based on values of the variables of the problem. N_{var}, is the number of variables of the problem, and habitat is an array of $1\times N_{\text{var}}$, which means the current position of the cuckoo. This array can be represented by the following:

$$\text{habitat} = [\, x_1 , x_2 , x_3 , \ldots . x_{N_{\text{var}}} \,] \tag{1}$$

To determine the profit of this habitat, it is calculated by profit function F as the following:

$$\text{Profit}= f_p\,(\text{habitat})= f_p\,(x_1,x_2,x_3,\ldots .x_{N_{\text{var}}}) \tag{2}$$

To use COA in minimization problems, we can calculate cost function by the following:

$$\text{Cost=-profit=-}\ f_p\,(x_1,x_2,x_3,\ldots .x_{N_{\text{var}}}) \tag{3}$$

Then the initial habitat matrix can be created with size $N_{pop}\times N_{\text{var}}$, where N_{pop}, is the number of

Journal of Theoretical and Applied Information Technology
30th September 2020. Vol.98. No 18

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

population, and N_{var}, is the number of variables for the problem.

2.2 Calculating Egg Laying Radius (ELR)

Each cuckoo has a maximum and a minimum number of eggs to be laid. Each cuckoo can lay its eggs within a maximum distance from its position called ELR. Based on the number of eggs, ELR can be calculated for each cuckoo by the following equation:

$$\text{ELR} = \alpha \times \frac{\text{Number of current cuckoo's eggs}}{\text{Total number of eggs}} \times (\text{var}_{hi} - \text{var}_{low}) \quad (4)$$

Where α is an integer to control the maximum value of ELR, var_{hi} and var_{low}, are the upper limit and the lower limit of variables in the optimization problem, respectively.

2.3 Laying Eggs

Each cuckoo begins laying eggs in any host nest within its ELR. Figure 2 shows this process. As mentioned before, when all cuckoos finish laying eggs, some of these eggs are recognized and killed by host birds with probability *p*% (usually 10%) of all eggs. The other eggs still grow up until hatch and become a mature cuckoo (or a chick). When the cuckoo egg becomes a mature cuckoo, it tries to get rid of all other eggs and small chicks. Even if host egg hatches first, the cuckoo egg can also get rid of all others because the cuckoo chick is three times bigger than the host chick. Besides, the cuckoo chick will eat more than the host chick, and the host egg will die from hunger after a small period.

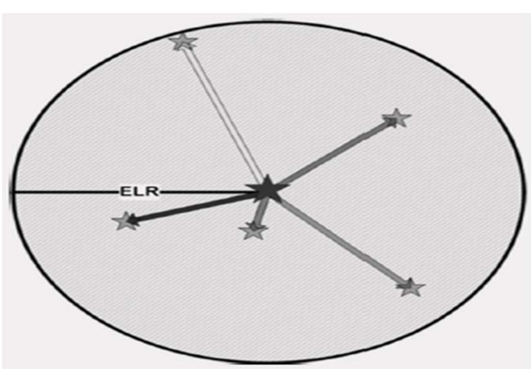


Figure 2: Laying eggs within ELR

2.4 Immigration of Cuckoos

When it is almost time to lay eggs for cuckoos, it searches for a more suitable habitat so; it can live in its area for some time, and then it will move to another habitat. As mentioned before, cuckoos compose groups in different areas in their environment. The best habitat is determined based on the more survival of eggs in it. This habitat is selected as a goal point for all cuckoos to immigrate. During immigration, it is more difficult to distinguish which cuckoo belongs to which group so, k-means clustering is used to solve this issue. Each cuckoo travels the only λ % of all distance to the goal point because it has a (θ) deviation from the goal point. λ ,θ , Can increase the chance of searching for more positions in its environment. λ , is a random number between (0, 1), and θ , is a random number between ($\frac{-\pi}{6}, \frac{\pi}{6}$). Figure 3 shows the immigration process.

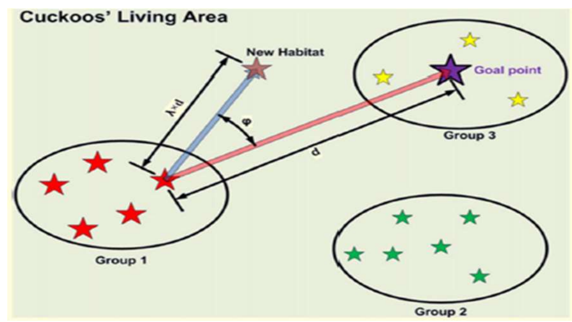


Figure 3: Immigration of cuckoo toward goal habitat [21].

2.5 Eliminating Cuckoos in Worst Habitats

Because of the limitations of food and failure in finding a suitable nest for laying eggs, the balance for the population of cuckoos must be performed. N_{max}, Can control this where N_{max} is the maximum number of cuckoos which can live in the environment at the same time. Cuckoos with better profit values can only be kept whereas the others are demise. There are many articles for enhancing the performance of COA and getting better results such [22-26].

3. CHAOTIC CUCKOO OPTIMIZATION ALGORITHM

Three chaotic algorithms are proposed. The first one (CCOA1) for defining ELR coefficient (α). The second is (CCOA2) for defining the immigration coefficient or degree of deviation (F). The third is (CCOA3) for defining a new proposed

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

parameter which incorporated in the immigration process. The three chaotic algorithms will be discussed in detail.

3.1 Chaotic Maps

We realize the vital role of chaos in improving metaheuristic algorithms and getting superior results over the original version of the algorithm like [27-29]. Chaos has the property of ergodicity and non-repetition, so it can speed the search process than stochastic methods. Chaos also helps to escape from local optima that any algorithm can fall in it. In addition to, chaos is used as a powerful technique in expanding search space and provide more effective exploration for the search area. From this point and for these reasons, we proposed to use chaos to obtain a more effective results. Table 1 and figure 4 show characteristics of the selected chaotic maps.

3.2 CCOA1

As seen in equation (4), ELR coefficient (α) is used to control ELR maximum value. This coefficient has a more important role in determining the allowed area for each cuckoo to lay its eggs. α is constant in the original COA but changing the value of α will effect on calculating this area. By utilizing chaotic maps for estimating this coefficient, it will allow different ELR for cuckoos even if they have the same number of eggs and also will allow to find different nests for laying their eggs. So, in CCOA1, chaotic maps are used to estimate the value of α.

3.3 CCOA2

After laying eggs and demising cuckoos with less fitness, the remaining cuckoos composing the clusters by using K-means. Then the fitness of each cluster is calculated and the cluster with highest fitness is selected as the most suitable region to immigrate. The most fitted cuckoo in the cluster with highest fitness, will be selected as the global optimal cuckoo. The more interesting point that in the original paper for COA, Rajabioun didn't determine a clear relationship between the next position and the current position of the habitat in the immigration process, but according to [30], the next position of the habitat can be determined by the following equation:

$$x_{next} = x_{current} + F\,(x_{goal} - x_{current}) \qquad (5)$$

Where x_{next} and $x_{current}$ are the positions of the next habitat and the current habitat, respectively. x_{goal} is the position of the global optimal cuckoo. F is the degree of deviation for each cluster and it is constant for all clusters and for all iterations in the original COA. F has a substantial role in determining the next position of the habitat. The constant value for F makes a high density of cuckoos in one place during immigration because the degree of deviation is equal. It also don't give chance for discovering any other best cuckoos in the other regions which may be better than the current global best. So, changing the value of (F), will achieve discovering the other best solutions and it will reduce the density of cuckoos in one domain. Chaotic maps have advantage of avoiding local optima and high convergence speed so, in CCOA2, chaotic maps are utilized to define F.

3.4 CCOA3

For improving the immigration process and getting the global optima than local optima, we propose a new parameter that will be incorporated in the immigration process. This parameter will determine how the current position will effect on the next position of the habitat. So, in CCOA3, the following equation will be used to determine the next position in the immigration process:

$$x_{next} = \beta \times x_{current} + F\,(x_{goal} - x_{current}) \qquad (6)$$

Where β is the chaotic sequence produced by chaotic maps. Changing the position of habitat by this equation will help to escape from local optima and getting better results. This equation also provides high convergence speed than the original algorithm. In addition, the high values for this parameter can achieve the high exploration process for the search area and covering the global search which was one of the main drawbacks of COA. Besides, the low values of this parameter enable to search the area around the current position and achieving the high exploitation process. So, introducing this parameter will achieve balance between exploration and exploitation process that most of the metaheuristic algorithms may fail to achieve it.

4. EXPERIMENTAL RESULTS AND STATISTICAL ANALYSIS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

To show the results the proposed algorithms, the experiments were performed using many benchmark problems. Table 2 shows properties of the selected benchmarks where d indicates the dimension of the problem, Range is the boundary of the problem search space and $f_{\min}$, is the optimal value for the problem. Benchmark functions are divided into three types; the first is unimodal functions (F1, F2), which have a unique global best solution. These functions can evaluate the exploitation process. The second type is shift-rotated functions (from F3 to F13) to increase the complexity. More details about these functions and their shift position can be found in [31]. The third type is multimodal functions, which have many global best solutions. These functions can evaluate the exploration process and show its competitive advantage in getting the desired results.

4.1 Three Chaotic Algorithms with Different Chaotic Maps

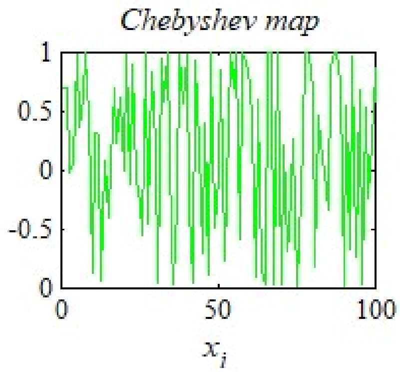
Ten chaotic maps were selected for the comparison as mentioned before. Tables (3-5) show the results of CCOA1, CCOA2 and CCOA3 respectively. Mean and standard deviation of 10 independent runs for different benchmark functions with ten chaotic maps with initial point 0.7 are listed in the tables. Moreover, the nonparametric statistical test called the Wilcoxon rank-sum test was performed at significance level 5% to determine whether the difference between chaotic maps is statistically significant. The statistical significance is calculated by using the *p*-values that are also reported in the tables. N/A indicates ''not applicable'', meaning that the corresponding algorithm could not be compared with itself. It is generally known that p-values <0.05 can be considered as sufficient evidence against the null hypothesis. Note that the problems that are not reported in the tables were tested and give the same result for all maps so, we didn't put it in the tables because it can't give a different result to distinguish the best map. The best results are highlighted in bold face. The results in table 3 show that Sine map (M7) can get better results for ELR coefficient and p-values verify that. So, Sine map is the more suitable map for calculating α and it will be used for getting the remaining results for CCOA1. The results in table 4 show that Singer map (M8) can get better results for the immigration coefficient so, Singer map will be used for calculating F in CCOA2 for getting the remaining results. The results in table 5 illustrate that the Gaussian map (M9) can give better results for six benchmark functions, while Iterative map and Tent map, each of them can provide better results for only one benchmark function. From these results, we can conclude that the more suitable map for calculating β is the Gaussian map. So, the results of CCOA3 is calculated using the Gaussian map (M9).

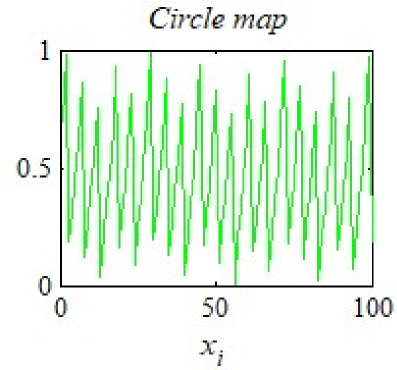
Table 1: Chaotic Maps

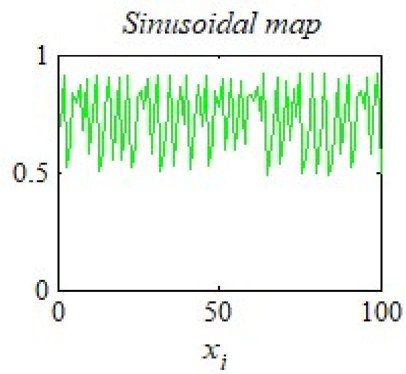
Name	Definition	Range	Map number
Chebyshev map	$x_{k+1} = \cos(k \cos^{-1}(x_k))$	(-1,1)	M1
Circle map	$x_{k+1} = \bmod(x_k + b - (a/2\pi)\sin(2\pi x_k),1)$, $a = 0.5 \ and \ b = 0.2$	(0,1)	M2
Sinusoidal map	$x_{k+1} = a x_k^2 \sin(\pi x_k), \quad a = 2.3$	(0,1)	M3
Iterative map	$x_{k+1} = \sin\left(\dfrac{a\pi}{x_k}\right), \quad a = 0.7$	(-1,1)	M4
Logistic map	$x_{k+1} = a x_k (1 - x_k), \ a = 4$	(0,1)	M5

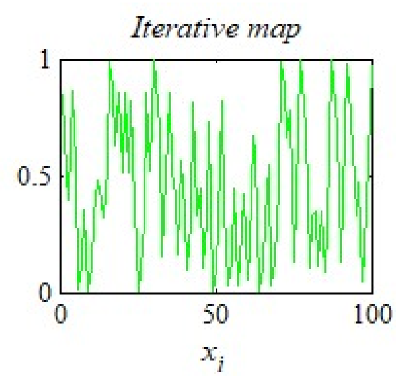
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

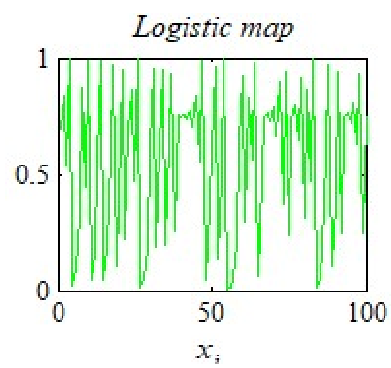
Piecewise map	$x_{k+1} = \begin{cases} \frac{x_k}{p} & 0 \le x_k < p \\ \frac{x_k - p}{0.5 - p} & p \le x_k < 0.5 \\ \frac{1 - p - x_k}{0.5 - p} & 0.5 \le x_k < 1 - p \\ \frac{1 - x_k}{p} & 1 - p \le x_k < 1 \end{cases}$, $P = 0.4$	(0,1)	M6
Sine map	$x_{k+1} = \frac{a}{4}\sin(\pi x_k), \quad a = 4$	(0,1)	M7
Singer map	$x_{k+1} = u(7.86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.302875x_k^4),$ u=1.07	(0,1)	M8
Gaussian map	$x_{k+1} = \begin{cases} 0 & x_k = 0 \\ \frac{1}{\mathrm{mod}(x_k, 1)} & otherwise \end{cases}$	(0,1)	M9
Tent map	$x_{k+1} = \begin{cases} \frac{x_k}{0.7} & x_k < 0.7 \\ \frac{10}{3}(1 - x_k) & x_k \ge 0.7 \end{cases}$	(0,1)	M10

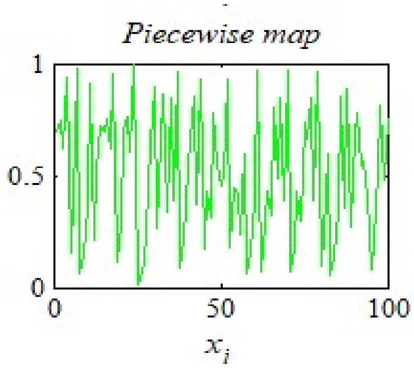

Chebyshev map
1
0.5
0
-0.5
0
0
50
100
x_i


Circle map
1
0.5
0
0
50
100
x_i


Sinusoidal map
1
0.5
0
0
50
100
x_i


Iterative map
1
0.5
0
0
50
100
x_i


Logistic map
1
0.5
0
0
50
100
x_i


Piecewise map
1
0.5
0
0
50
100
x_i

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

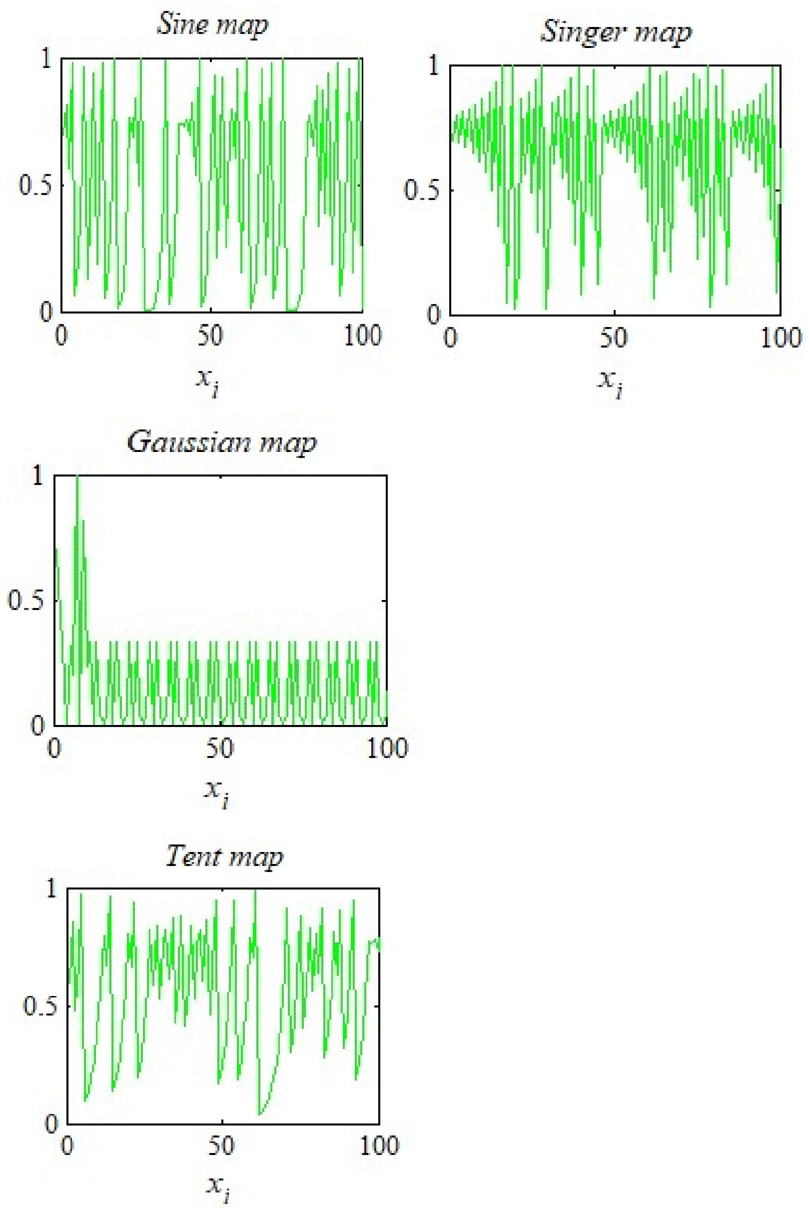


Figure 4: Chaotic maps (M1-M10)

Table 2: Benchmark Functions

Test problems	Equation	d	Range	$f_{\min}$	type
1-Zakharov	$f(x)=\sum_{i=1}^{d} x_i^2+\left(\sum_{i=1}^{d} 0.5ix_i\right)^2+\left(\sum_{i=1}^{d} 0.5ix_i\right)^4$	30	[-5,10]	0	unimodal
2-Easom	$f(x)=-\cos(x_1)\cos(x_2)e^{(-(x_1-\pi)^2-(x_2-\pi)^2)}$	2	[-100,100]	-1	unimodal
3-Schwefel (2.21)	$f(x)=\max\{\lvert x_i\rvert,\ 1\le i\le d\}$	30	[-100,100]	0	unimodal

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4-Rosenbrock	$f(x)=\sum_{i=1}^{d-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	[-30,30]	0	unimodal
5-Step	$f(x)=\sum_{i=1}^{d}(\lvert x_i+0.5\rvert)^2$	30	[-100,100]	0	unimodal
6-Quartic	$f(x)=\sum_{i=1}^{d} ix_i^4+random(0,1)$	30	[-1.28,1.28]	0	unimodal
7-Schwefel 2.22	$f(x)=\sum_{i=1}^{d}\lvert x_i\rvert+\prod_{i=1}^{d}\lvert x_i\rvert$	30	[-10,10]	0	unimodal
8-Schwefel (2.26)	$f(x)=\sum_{i=1}^{d}-x_i\sin(\sqrt{x_i})$	30	[-500,500]	-418.9829*d	multimodal
9-Rastrigin	$f(x)=\sum_{i=1}^{d}[x_i^2-10\cos(2\pi x_i)+10]$	30	[-5.12,5.12]	0	multimodal
10-Ackley	$f(x)=-20\exp(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2})-\exp(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i))+20+e$	30	[-32,32]	0	multimodal
11-Griewank	$f(x)=\frac{1}{4000}\sum_{i=1}^{d}x_i^2-\prod_{i=1}^{d}\cos(\frac{x_i}{\sqrt{i}})+1$	30	[-600,600]	0	multimodal
12-Penalty 1	$f(x)=\frac{\pi}{d}\times\{10\sin^2(\pi y_1)+\sum_{i=1}^{d-1}(y_i-1)^2[1+10\sin^2(\pi y_{i+1})]+(y_d-1)^2\}+\sum_{i=1}^{d}u(x_i,a,k,m)$ where $y_i=1+\frac{x_i+1}{4}$ $u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & if\ \ x_i>a\\ 0 & if\ \ -a<x_i<a\\ k(-x_i-a)^m & if\ \ x_i<-a\end{cases}$ $, a=10,\ \ k=100,\ \ m=4$	30	[-50,50]	0	multimodal
13-Penalty 2	$f(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^{d}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+(x_d-1)^2[1+\sin^2(2\pi x_d)]\}+\sum_{i=1}^{d}u(x_i,a,k,m)$ $,a=5, k=100,\ m=4$	30	[-50,50]	0	multimodal
14-Beale	$f(x)=(1.5-x_1+x_1x_2)^2+(2.25-x_1+x_1x_2^2)^2+(2.625-x_1+x_1x_2^2)^2$	2	[-4.5,4.5]	0	multimodal
15-Three-Hump Camel Function	$f(x)=2x_1^2-1.05x_1^4+\frac{x_1^6}{6}+x_1x_2+x_2^2$	2	[-5,5]	0	multimodal

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

16-Matyas	$f(x)=0.26(x_1^2+x_2^2)-0.48x_1x_2$	2	[-10,10]	0	multimodal
17-Schaffer Function N.2	$f(x)=0.5+\frac{\sin^2(x_1^2-x_2^2)-0.5}{(1+0.001(x_1^2+x_2^2))^2}$	2	[-100,100]	0	multimodal

Table 3: Results of CCOA1 for Chaotic Maps

Function/ Map	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
F1 Mean	8.93E-18	**2.88E-23**	9.34E-19	7.38E-22	2.05E-18	9.70E-17	9.82E-15	5.97E-16	1.21E-11	2.19E-20
SD	2.82E-17	**9.12E-23**	2.70E-18	2.33E-21	6.35E-18	3.07E-16	3.10E-14	1.52E-15	3.80E-11	6.47E-20
p-value	9.68E-01	**N/A**	9.68E-01	9.68E-01	9.68E-01	9.68E-01	9.68E-01	9.68E-01	9.68E-01	9.68E-01
F3 Mean	4.67E+01	5.03E+01	5.03E+01	3.90E+01	4.67E+01	4.46E+01	4.51E+01	**3.88E+01**	5.80E+01	4.95E+01
SD	1.43E+01	2.27E+01	1.50E+01	1.11E+01	1.13E+01	1.81E+01	1.59E+01	**8.02E+00**	2.85E+01	1.46E+01
p-value	1.62E-01	3.47E-01	1.03E-01	9.36E-01	1.42E-01	6.74E-01	7.95E-01	**N/A**	3.75E-02	6.43E-02
F 4 Mean	2.10E+00	2.35E+00	6.91E+00	1.65E+00	1.64E+00	2.32E+00	**7.00E-01**	1.90E+00	1.73E+01	3.58E+00
SD	1.43E+00	7.99E-01	5.57E+00	1.34E+00	1.35E+00	1.82E+00	**6.53E-01**	2.30E+00	1.47E+01	1.04E+00
p-value	1.39E-02	1.32E-03	4.40E-04	7.51E-02	5.36E-02	1.14E-02	**N/A**	4.72E-01	3.08E-01	1.80E-04
F5 Mean	2.54E+04	2.14E+04	1.52E+04	1.60E+04	2.44E+04	2.28E+04	**1.34E+04**	2.03E+04	2.56E+04	2.81E+04
SD	1.46E+04	1.51E+04	8.21E+03	9.39E+03	1.53E+04	1.28E+04	**7.30E+03**	6.89E+03	2.78E+04	2.28E+04
p-value	3.75E-02	1.87E-01	5.49E-01	5.69E-01	1.50E-01	6.43E-02	**N/A**	5.36E-02	7.95E-01	8.91E-02
F7 Mean	2.16E+07	8.82E+05	3.15E+05	1.10E+06	9.28E+04	6.53E+07	**7.88E+03**	1.32E+05	6.81E+05	2.04E+06
SD	6.77E+07	2.56E+06	6.51E+05	3.44E+06	2.23E+05	2.05E+08	**2.33E+04**	4.16E+05	2.15E+06	4.87E+06
p-value	1.21E-01	8.18E-01	4.97E-01	9.12E-01	5.69E-01	6.74E-01	**N/A**	9.68E-01	3.47E-01	5.22E-01
F10 Mean	1.66E+01	1.70E+01	1.70E+01	1.61E+01	1.69E+01	1.65E+01	1.61E+01	1.75E+01	**1.16E+01**	1.69E+01
SD	1.98E+00	2.29E+00	2.93E+00	2.59E+00	9.03E-01	2.38E+00	1.80E+00	1.54E+00	**4.82E+00**	2.44E+00
p-value	3.75E-02	2.57E-02	1.73E-02	3.75E-02	3.16E-02	2.57E-02	5.36E-02	1.73E-02	**N/A**	2.57E-02
F12 Mean	**1.93E+02**	1.66E+03	2.47E+04	2.31E+05	3.92E+02	9.97E+02	4.63E+02	7.75E+04	3.60E+06	8.73E+03
`SD	**5.13E+02**	4.33E+03	7.48E+04	5.73E+05	1.19E+03	3.09E+03	1.42E+03	1.58E+05	1.14E+07	1.82E+04
p-value	**N/A**	9.12E-01	1.87E-01	2.26E-01	2.89E-01	3.84E-01	7.28E-01	1.21E-01	4.07E-01	9.68E-01
F14 Mean	1.52E-01	3.05E-01	3.05E-01	1.52E-01	1.52E-01	1.52E-01	3.82E-01	2.29E-01	2.35E-01	**7.62E-02**
SD	3.21E-01	3.94E-01	3.94E-01	3.21E-01	3.21E-01	3.21E-01	4.03E-01	3.68E-01	3.79E-01	**2.41E-01**
p-value										**N/A**

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

	5.49E-01	6.24E-01	2.11E-01	5.49E-01	7.64E-01	4.30E-01	3.47E-01	2.71E-01	9.68E-01	

Table 4: Results of CCOA2 for Chaotic Maps

Function/ Map	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
F3 Mean SD p-value	4.25E+01 1.26E+01 3.84E-01	5.47E+01 1.23E+01 1.39E-02	4.57E+01 1.56E+01 2.71E-01	5.48E+01 1.21E+01 1.14E-02	5.34E+01 1.43E+01 4.14E-02	4.45E+01 1.98E+01 4.72E-01	4.66E+01 1.65E+01 1.62E-01	**3.73E+01** **1.28E+01** **N/A**	6.42E+01 1.21E+01 1.00E-03	4.90E+01 9.99E+00 3.16E-02
F6 Mean SD p-value	1.19E-02 1.02E-02 7.95E-01	1.06E-02 5.62E-03 1.62E-01	9.74E-03 6.33E-03 7.04E-01	1.35E-02 8.37E-03 8.19E-02	**8.40E-03** **1.64E-03** **N/A**	1.19E-02 4.56E-03 3.75E-02	1.70E-02 7.74E-03 7.36E-03	1.37E-02 7.20E-03 1.03E-01	1.42E-02 9.84E-03 3.84E-01	8.95E-03 5.01E-03 9.68E-01
F 7 Mean SD p-value	1.14E+04 3.57E+04 3.16E-02	1.30E+08 4.10E+08 9.06E-03	**5.21E+01** **6.71E+01** **N/A**	6.08E+04 1.92E+05 2.09E-02	1.44E+07 4.52E+07 1.14E-02	1.13E+07 3.15E+07 8.91E-02	1.66E+05 4.31E+05 1.73E-02	7.17E+01 6.74E+01 3.08E-01	4.80E+10 1.52E+11 4.40E-04	9.13E+01 1.76E+02 8.18E-01
F9 Mean SD p-value	3.00E+02 4.77E+01 1.21E-01	3.10E+02 3.94E+01 1.55E-02	2.76E+02 3.90E+01 5.69E-01	2.91E+02 3.08E+01 1.42E-01	3.02E+02 4.33E+01 1.03E-01	2.92E+02 5.17E+01 2.26E-01	2.90E+02 3.39E+01 2.42E-01	2.92E+02 5.06E+01 2.71E-01	3.52E+02 4.50E+01 1.80E-04	**2.71E+02** **2.18E+01** **N/A**
F13 Mean SD p-value	4.25E+04 8.82E+04 1.42E-01	1.46E+04 2.81E+04 1.73E-02	7.64E+03 1.68E+04 8.49E-01	2.96E+05 9.18E+05 5.36E-02	3.77E+04 9.84E+04 2.71E-01	7.62E+05 1.73E+06 5.36E-02	3.59E+05 8.82E+05 1.39E-02	**2.27E+02** **4.69E+02** **N/A**	1.75E+08 1.32E+08 1.80E-04	3.96E+05 1.17E+06 7.51E-02
F15 Mean SD p-value	9.54E-77 2.99E-76 9.68E-01	3.61E-76 9.72E-76 9.68E-01	1.07E-78 3.34E-78 9.68E-01	2.82E-76 8.91E-76 9.68E-01	1.41E-72 4.45E-72 9.68E-01	1.87E-71 5.92E-71 9.68E-01	**9.63E-80** **2.95E-79** **N/A**	2.48E-76 7.80E-76 9.68E-01	3.38E-78 9.75E-78 9.68E-01	1.60E-70 5.06E-70 9.68E-01
F16 Mean `SD p-value	4.61E-59 1.46E-58 9.68E-01	**4.90E-80** **1.55E-79** **N/A**	2.04E-65 6.44E-65 9.68E-01	2.86E-73 9.02E-73 9.68E-01	4.81E-72 1.51E-71 9.68E-01	2.31E-72 7.09E-72 9.68E-01	4.79E-71 1.51E-70 9.68E-01	4.10E-46 1.30E-45 9.68E-01	1.24E-74 3.81E-74 9.68E-01	5.87E-64 1.86E-63 9.68E-01

Table 5: Results of CCOA3 for Chaotic Maps

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Function/ Map	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
F3 Mean	3.28E-01	4.25E-01	1.32E+00	2.59E-01	4.60E-01	3.09E-01	1.75E-01	4.74E-01	**1.33E-01**	3.72E-01
SD	2.62E-01	1.03E-01	3.13E-01	9.57E-02	3.14E-01	1.84E-01	1.83E-01	3.61E-01	**1.84E-01**	2.21E-01
p-value	4.55E-02	3.62E-03	1.80E-04	4.55E-02	1.14E-02	4.55E-02	2.11E-01	9.06E-03	**N/A**	1.39E-02
F5 Mean	1.43E+01	1.16E+01	4.68E+01	1.42E+01	1.09E+01	1.58E+01	1.29E+01	2.15E+01	**6.69E+00**	1.39E+01
SD	6.13E+00	6.37E+00	2.09E+01	7.00E+00	4.98E+00	6.13E+00	4.52E+00	9.73E+00	**2.93E+00**	4.26E+00
p-value	7.36E-03	7.51E-02	1.80E-04	3.16E-02	6.43E-02	1.68E-03	1.14E-02	2.78E-03	**N/A**	1.52E-03
F7 Mean	1.46E+00	9.03E-01	5.11E+00	8.81E-01	1.04E+00	1.39E+00	5.41E-01	2.50E+00	**3.47E-01**	1.06E+00
SD	7.50E-01	5.33E-01	1.21E+00	9.52E-01	4.77E-01	5.78E-01	4.71E-01	6.00E-01	**4.01E-01**	5.32E-01
p-value	2.22E-03	2.09E-02	1.80E-04	8.91E-02	7.36E-03	1.16E-03	2.26E-01	1.80E-04	**N/A**	9.06E-03
F8 Mean	-2.60E+03	-2.50E+03	-2.44E+03	**-3.70E+03**	-2.86E+03	-2.99E+03	-2.68E+03	-2.50E+03	-2.56E+03	-2.47E+03
SD	7.93E+02	8.21E+02	8.89E+02	**1.49E+03**	8.85E+02	1.50E+03	6.99E+02	4.19E+02	8.90E+02	1.17E+03
p-value	1.03E-01	6.43E-02	5.36E-02	**N/A**	1.87E-01	2.71E-01	8.91E-02	3.75E-02	7.51E-02	8.91E-02
F9 Mean	3.74E+00	1.33E+00	2.96E+01	1.47E+00	2.61E+00	2.28E+00	9.96E-01	9.14E+00	**2.08E-01**	3.25E+00
SD	3.01E+00	1.09E+00	7.30E+00	2.29E+00	1.55E+00	1.24E+00	8.90E-01	6.88E+00	**3.05E-01**	3.20E+00
p-value	1.80E-04	4.66E-03	1.80E-04	2.09E-02	4.40E-04	7.80E-04	9.06E-03	4.40E-04	**N/A**	1.68E-03
F10 Mean	3.97E-01	1.45E-01	1.63E+00	2.83E-01	2.60E-01	2.37E-01	2.27E-01	5.14E-01	**9.94E-02**	3.32E-01
SD	2.63E-01	1.54E-01	4.58E-01	2.18E-01	1.62E-01	1.55E-01	2.08E-01	3.90E-01	**1.37E-01**	1.64E-01
p-value	2.78E-03	1.87E-01	1.80E-04	2.09E-02	4.55E-02	3.75E-02	3.16E-02	9.06E-03	**N/A**	5.78E-03
F12 Mean	1.43E-01	1.31E-01	1.35E-01	8.26E-02	1.03E-01	6.35E-02	8.16E-02	7.23E-02	**5.15E-02**	6.65E-02
SD	1.01E-01	1.71E-01	7.09E-02	6.05E-02	1.82E-01	4.20E-02	6.52E-02	5.79E-02	**4.44E-02**	4.03E-02
p-value	1.73E-02	2.11E-01	9.06E-03	1.87E-01	7.95E-01	5.69E-01	3.84E-01	4.30E-01	**N/A**	4.30E-01
F13 Mean	2.48E+00	2.10E+00	4.15E+00	2.66E+00	1.91E+00	1.64E+00	2.75E+00	2.53E+00	1.84E+00	**1.47E+00**
SD	1.69E+00	1.07E+00	1.74E+00	1.89E+00	1.19E+00	1.02E+00	1.27E+00	1.20E+00	9.75E-01	**9.45E-01**
p-value	1.87E-01	1.87E-01	1.68E-03	1.42E-01	4.72E-01	6.74E-01	2.09E-02	7.51E-02	4.30E-01	**N/A**

4.2 Performance of the Three Chaotic Algorithms against Other Compared Algorithms

For evaluating the performance of the proposed algorithms, the comparison is performed with other metaheuristic algorithms. The proposed algorithms

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

were compared with standard Cuckoo Optimization Algorithm (COA), Cuckoo Search(CS), Differential Evolution (DE), Covariance Matrix Adaptation-Evolution Strategy (CMA-ES), Particle Swarm Optimization (PSO), Gravitational Search Algorithm (GSA) and chaotic cuckoo search (CCS) [32] Table 6 shows parameter settings for the . compared algorithms except CMA-ES and GSA . these two algorithms were executed with 30 and 100 for population size and maximum iteration respectively. the other parameters forthe two algorithms were set as default. The parameters for CCOA1, CCOA2 and CCOA3 are the same as COA except the specified parameters, which calculated .by chaotic maps

Table 6: Parameter Settings Of Compared Algorithms

Algorithm	Parameters	Value
COA	Initial no. cuckoos	5
	Minimum no. of eggs for each cuckoo	2
	Maximum no. of eggs for each cuckoo	10
	No. of clusters	1
	ELR coefficient (α)	1
	Migration coefficient (F)	0.3
	Population variance that cuts the optimization	1.00E−08
	Population size ($N_{\max}$)	30
	Maximum no. of iterations	100
CS	Discovery rate (p_a)	0.25
	Population size	30
	Maximum no. of iterations	100
DE	scale factor (F)	[0.2, 0.8]
	Crossover Probability (CR)	0.2
	Population size	30
	Maximum no. of iterations	100
PSO	Acceleration coefficients C1, C2	2
	Inertia Weight Factor	[0.2, 0.9]
	Population size	30
	Maximum no. of iterations	100
CCS	Discovery rate (p_a)	0.25
	Elitism parameter	2
	Population size	30
	Maximum no. of iterations	100

The results of the proposed algorithms and the selected algorithms for comparisons are reported in tables (7-10). Table 7 shows mean optimization results and standard deviation for 20 independent runs for solving the selected benchmark functions. As observed from the table, CCOA3 can get the best solution for most of the problems. CCOA3 has the ability to give the optimal solution for F1, F2, F11, F15, F16 and F17 which verifies its ability in solving both unimodal and multimodal problems. In addition, CCOA3 can get better solution than other compared algorithms for F6, F9, F10, F12 and F13 while COA, CCOA1 and CCOA2 can give the optimal solution for F2, F11 and F17. CMA-ES gives the optimal solution for two problems F14 and F17 while DE and PSO give the optimal solution for only one problem F2. CCS gives better solution than other compared algorithms for F3, F5 and F7. CS and GSA fail to get any optimal solution or better results the other compared algorithms.

For deep analysis for the performance of the proposed algorithms against other compared algorithms, a nonparametric statistical test, Wilcoxon's rank-sum test is executed at 5 % significance level. The p values calculated in the Wilcoxon's rank-sum are given in table 8. The p-values >0.05 are underlined and the best results are highlighted in bold face. The results in table 8 show how CCOA3 can achieve significant difference between the other algorithms. Best optimization results and worst optimization results are reported in tables 9 and 10 which, verify the ability of CCOA3 in getting the desired results. Figures (5-7) show the convergence curves of CCOA1, CCOA2 and CCOA3 respectively against COA. The three figures illustrate how the proposed algorithms can improve the standard COA specially, CCOA3 which approve its superiority in getting better results for most of the problems over COA.

Table 7: Mean Optimization Results And Standard Deviation For Benchmark Functions

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Function/algorithm	COA	CS	DE	CMA-ES	PSO	GSA	CCS	CCOA 1	CCOA 2	CCOA 3
F1 Mean	1.30E-18	3.92E+02	4.00E+02	6.04E+02	2.84E+02	1.10E+02	3.09E+02	2.05E-21	4.35E-13	**0.00E+00**
SD	3.39E-18	4.56E+01	6.89E+01	1.77E+02	7.79E+01	2.65E+01	8.59E+01	9.16E-21	1.92E-12	**0.00E+00**
F2 Mean	**-1.00E+00**	-6.59E-01	**-1.00E+00**	-9.29E-01	**-1.00E+00**	-9.50E-01	-9.99E-01	**-1.00E+00**	**-1.00E+00**	**-1.00E+00**
SD	4.10E-05	3.56E-01	1.83E-04	2.27E-01	8.16E-14	2.24E-01	7.23E-04	**0.00E+00**	**0.00E+00**	1.66E-05
F3 Mean	5.72E+01	4.15E+01	6.05E+01	3.39E+00	1.87E+01	1.91E+01	**1.21E-04**	4.03E+01	4.56E+01	1.14E-01
SD	1.22E+01	4.34E+00	3.39E+00	7.91E-01	2.91E+00	1.81E+00	**1.96E-04**	1.38E+01	1.31E+01	1.22E-01
F4 Mean	4.48E+00	3.61E+06	7.37E+05	2.95E+03	3.27E+04	1.39E+05	2.49E+05	**1.03E+00**	4.61E+00	2.81E+00
SD	4.23E+00	1.28E+06	3.22E+05	4.20E+03	3.71E+04	7.96E+04	1.63E+05	**9.12E-01**	2.84E+00	2.97E+00
F5 Mean	2.12E+04	5.22E+03	1.55E+03	1.18E+01	1.43E+02	2.77E+03	**7.72E-08**	1.77E+04	1.36E+04	2.24E+00
SD	1.39E+04	1.52E+03	3.54E+02	3.80E+00	7.46E+01	8.35E+02	**1.04E-07**	1.50E+04	1.23E+04	1.26E+00
F6 Mean	8.83E-03	1.90E+00	7.05E-01	1.91E-02	1.78E-01	3.32E-01	2.53E+01	6.75E-03	1.39E-02	**4.82E-03**
SD	4.53E-03	6.43E-01	2.09E-01	1.16E-02	5.62E-02	1.45E-01	1.05E+01	4.78E-03	9.69E-03	**5.64E-03**
F7 Mean	3.78E+11	1.06E+03	1.53E+01	3.23E+00	8.15E+00	1.13E+01	**8.53E-05**	2.72E+07	8.49E+09	8.18E-02
SD	1.69E+12	4.26E+03	1.96E+00	4.69E-01	5.90E+00	3.78E+00	**2.07E-04**	1.21E+08	3.74E+10	1.33E-01
F8 Mean	-2.27E+03	-6.20E+03	-6.29E+03	-6.41E+04	**-7.50E+03**	-2.50E+03	-6.78E+03	-1.96E+03	-2.10E+03	-1.09E+03
SD	7.24E+02	1.84E+02	2.87E+02	7.57E+04	**8.09E+02**	5.77E+02	5.55E+02	7.10E+02	8.42E+02	4.88E+02
F9 Mean	3.24E+02	2.21E+02	1.78E+02	1.96E+02	1.19E+02	7.67E+01	2.73E+02	3.27E+02	2.90E+02	**1.55E-01**
SD	3.94E+01	1.70E+01	1.39E+01	8.17E+00	3.19E+01	1.96E+01	3.42E+01	3.74E+01	4.17E+01	**2.27E-01**
F10 Mean	1.78E+01	1.68E+01	1.02E+01	2.11E+00	4.38E+00	7.92E+00	9.67E+00	1.60E+01	1.54E+01	**1.17E-01**
SD	1.80E+00	1.13E+00	8.21E-01	2.41E-01	6.37E-01	9.45E-01	9.07E-01	3.03E+00	2.54E+00	**1.45E-01**
F11 Mean	**0.00E+00**	4.75E+01	1.54E+01	1.12E+00	2.55E+00	4.08E+02	3.72E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**
SD	**0.00E+00**	8.89E+00	2.04E+00	3.46E-02	6.40E-01	5.49E+01	1.13E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**
F12 Mean	3.60E+03	4.86E+05	3.49E+04	2.35E-01	1.13E+01	1.06E+02	1.61E+02	1.73E+05	7.74E+03	**1.03E-01**
SD	1.05E+04	5.44E+05	6.10E+04	8.53E-02	5.29E+00	2.23E+02	3.84E+02	7.53E+05	2.33E+04	**8.93E-02**
F13 Mean	6.36E+05	4.26E+06	7.17E+05	2.86E+00	9.58E+01	1.00E+05	5.93E+04	2.13E+06	1.70E+05	**6.70E-01**
SD	1.36E+06	2.63E+06	4.60E+05	1.30E+00	9.13E+01	1.21E+05	1.74E+05	9.26E+06	5.15E+05	**4.67E-01**
F14 Mean	2.29E-01	1.95E-05	1.55E-07	**0.00E+00**	7.62E-02	4.61E-02	8.19E-04	3.21E-01	2.29E-01	2.31E-01
SD	3.58E-01	2.13E-05	2.37E-07	**0.00E+00**	2.35E-01	3.92E-02	8.80E-04	4.04E-01	3.58E-01	4.12E-01

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

F15 Mean	2.70E-73	4.01E-07	4.89E-18	6.76E-61	2.85E-16	2.09E-18	8.88E-05	8.40E-16	1.03E-70	**0.00E+00**
SD	1.07E-72	5.97E-07	8.70E-18	1.99E-60	5.41E-16	2.15E-18	1.27E-04	2.97E-15	4.54E-70	**0.00E+00**
F16 Mean	1.56E-70	1.48E-08	9.40E-09	7.05E-61	2.16E-13	3.78E-03	2.57E-05	4.40E-19	5.02E-56	**0.00E+00**
SD	6.79E-70	1.82E-08	1.26E-08	2.16E-60	3.91E-13	9.42E-03	2.90E-05	1.41E-18	2.25E-55	**0.00E+00**
F17 Mean	**0.00E+00**	4.80E-04	1.95E-05	**0.00E+00**	2.61E-14	4.78E-02	1.88E-05	**0.00E+00**	**0.00E+00**	**0.00E+00**
SD	**0.00E+00**	5.30E-04	3.94E-05	**0.00E+00**	6.74E-14	4.57E-02	4.77E-05	**0.00E+00**	**0.00E+00**	**0.00E+00**

Table 8: P-Values For The Compared Algorithms

Function/algorithm	COA	CS	DE	CMA-ES	PSO	GSA	CCS	CCOA 1	CCOA 2	CCOA 3
F1	9.92E-01	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	9.92E-01	9.92E-01	N/A
F2	2.85E-01	1.00E-05	2.85E-01	5.96E-01	9.92E-01	7.95E-01	1.00E-05	N/A	N/A	1.24E-03
F3	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	N/A	1.00E-05	1.00E-05	3.73E-01
F4	1.32E-03	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	N/A	1.40E-04	1.47E-01
F5	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	N/A	1.00E-05	1.00E-05	1.00E-05
F6	1.94E-03	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	6.58E-02	1.00E-05	N/A
F7	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	N/A	1.00E-05	1.00E-05	3.17E-01
F8	1.00E-05	1.00E-05	1.00E-05	1.00E-05	N/A	1.00E-05	4.10E-03	1.00E-05	1.00E-05	1.00E-05
F9	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	N/A
F10	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	N/A
F11	N/A	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	N/A	N/A	N/A
F12	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	N/A
F13	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	1.00E-05	N/A
F14	1.00E-05	1.00E-05	1.00E-05	N/A	5.96E-01	1.00E-05	1.00E-05	1.55E-02	1.55E-02	1.00E-05
F15	9.92E-01	1.00E-05	9.92E-01	9.92E-01	9.92E-01	9.92E-01	1.00E-05	9.92E-01	9.92E-01	N/A
F16	9.92E-01	6.01E-02	1.07E-01	9.92E-01	9.92E-01	1.00E-05	1.00E-05	9.92E-01	9.92E-01	N/A
F17	N/A	1.00E-05	1.00E-05	N/A	9.92E-01	1.00E-05	1.00E-05	N/A	N/A	N/A

Table 9: Best Optimization Results

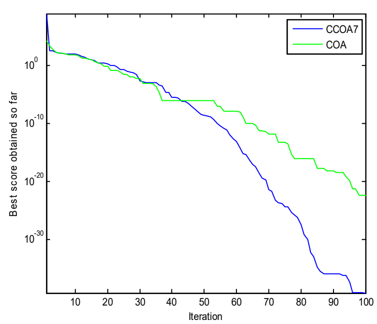
Function/algorithm	COA	CS	DE	CMA-ES	PSO	GSA	CCS	CCOA 1	CCOA 2	CCOA 3
F1	1.83E-56	2.98E+02	2.70E+02	3.41E+02	1.88E+02	6.67E+01	1.41E+02	6.73E-63	9.80E-46	0.00E+00
F2	-1.00E+00	-9.85E-01	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00	-1.00E+00
F3	3.53E+01	3.21E+01	5.53E+01	2.14E+00	1.18E+01	1.55E+01	0.00E+00	1.84E+01	2.69E+01	2.37E-14
F4	4.03E-01	1.79E+06	2.84E+05	2.28E+02	4.31E+03	4.87E+04	5.96E+04	2.80E-01	9.65E-02	5.13E-02
F5	4.54E+03	2.52E+03	7.78E+02	5.73E+00	5.88E+01	1.51E+03	4.96E-11	2.55E+03	1.82E+03	4.65E-01
F6	3.40E-03	5.47E-01	3.52E-01	1.70E-03	7.75E-02	1.11E-01	1.22E+01	8.27E-04	3.60E-03	6.40E-04
F7	4.20E+01	6.06E+01	1.14E+01	2.52E+00	2.15E+00	5.90E+00	0.00E+00	2.23E+01	1.65E+01	1.74E-14
F8	-4.11E+03	-6.48E+03	-6.79E+03	-3.74E+05	-8.98E+03	-3.73E+03	-8.24E+03	-3.40E+03	-4.12E+03	-2.53E+03
F9	2.66E+02	1.82E+02	1.50E+02	1.83E+02	7.60E+01	4.42E+01	2.28E+02	2.65E+02	2.25E+02	0.00E+00
F10	1.39E+01	1.30E+01	8.10E+00	1.63E+00	3.39E+00	6.30E+00	8.03E+00	1.04E+01	1.01E+01	4.44E-15
F11	0.00E+00	2.55E+01	1.11E+01	1.07E+00	1.57E+00	3.26E+02	2.37E+01	0.00E+00	0.00E+00	0.00E+00
F12	4.59E+04	2.28E+06	1.62E+01	1.09E-01	4.19E+00	1.01E+01	1.65E+03	5.17E+00	2.78E+00	3.55E-01
F13	5.53E+06	9.34E+06	8.51E+04	7.61E-01	1.99E+01	1.12E+02	7.87E+05	1.66E+01	1.03E+01	1.63E+00
F14	8.25E-09	1.17E-08	2.66E-10	0.00E+00	4.15E-15	4.90E-03	1.01E-05	8.23E-12	9.61E-10	4.32E-07
F15	0.00E+00	5.57E-09	1.10E-21	5.40E-66	9.27E-19	3.67E-20	0.00E+00	1.17E-29	0.00E+00	0.00E+00
F16	0.00E+00	6.23E-10	2.38E-11	4.14E-65	3.50E-17	4.35E-08	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F17	0.00E+00	5.27E-07	2.95E-09	0.00E+00	0.00E+00	3.01E-04	2.32E-09	0.00E+00	0.00E+00	0.00E+00

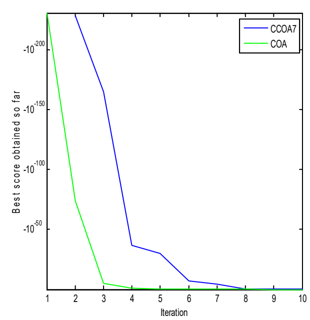
Table 10: Worst Optimization Results

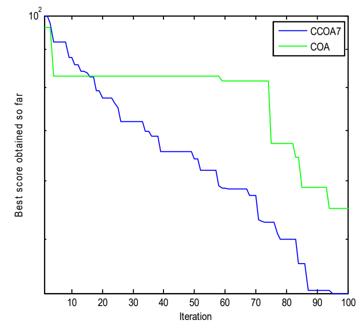
Function/algorithm	COA	CS	DE	CMA-ES	PSO	GSA	CCS	CCOA 1	CCOA 2	CCOA 3
F1	1.34E-17	4.52E+02	5.35E+02	9.23E+02	5.05E+02	1.48E+02	4.61E+02	4.10E-20	8.59E-12	0.00E+00
F2	-1.00E+00	-6.90E-05	-9.99E-01	-1.09E-01	-1.00E+00	-1.98E-04	-9.97E-01	-1.00E+00	-1.00E+00	-1.00E+00
F3	7.37E+01	4.97E+01	6.52E+01	5.21E+00	2.61E+01	2.14E+01	7.66E-04	7.21E+01	7.53E+01	3.46E-01
F4	1.33E+01	6.98E+06	1.41E+06	1.52E+04	1.18E+05	4.26E+05	7.45E+05	2.85E+00	1.10E+01	1.06E+01
F5	5.16E+04	1.02E+04	2.16E+03	1.85E+01	3.77E+02	5.25E+03	3.52E-07	5.45E+04	5.14E+04	5.35E+00
F6	2.02E-02	3.10E+00	1.08E+00	4.24E-02	2.77E-01	7.20E-01	5.30E+01	1.66E-02	3.86E-02	2.38E-02
F7	7.56E+12	1.92E+04	2.04E+01	4.37E+00	2.28E+01	1.94E+01	8.73E-04	5.43E+08	1.67E+11	4.43E-01

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

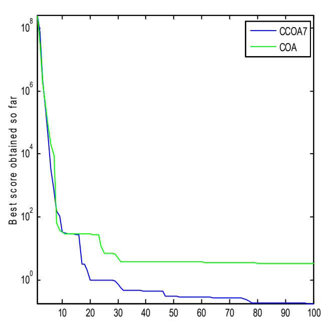
F8	-1.51E+03	-5.78E+03	-5.67E+03	-2.16E+04	-5.79E+03	-1.76E+03	-6.12E+03	-1.04E+03	-1.14E+03	-6.45E+02
F9	4.35E+02	2.53E+02	1.99E+02	2.12E+02	1.87E+02	1.15E+02	3.47E+02	3.84E+02	3.82E+02	8.18E-01
F10	2.00E+01	1.88E+01	1.21E+01	2.55E+00	5.40E+00	9.72E+00	1.10E+01	1.98E+01	2.01E+01	4.32E-01
F11	0.00E+00	6.25E+01	2.01E+01	1.19E+00	3.86E+00	5.35E+02	6.03E+01	0.00E+00	0.00E+00	0.00E+00
F12	4.59E+04	2.28E+06	2.77E+05	3.80E-01	2.41E+01	9.22E+02	1.65E+03	3.37E+06	8.82E+04	3.55E-01
F13	5.53E+06	9.34E+06	1.83E+06	5.54E+00	3.86E+02	4.82E+05	7.87E+05	4.15E+07	2.00E+06	1.63E+00
F14	7.62E-01	8.24E-05	1.02E-06	0.00E+00	7.62E-01	1.32E-01	2.36E-03	9.21E-01	7.62E-01	1.01E+00
F15	4.79E-72	2.06E-06	3.14E-17	6.60E-60	2.37E-15	8.45E-18	5.05E-04	1.29E-14	2.03E-69	0.00E+00
F16	3.04E-69	5.42E-08	4.08E-08	9.45E-60	1.58E-12	4.24E-02	9.41E-05	6.29E-18	1.00E-54	0.00E+00
F17	0.00E+00	1.80E-03	1.67E-04	0.00E+00	2.98E-13	1.39E-01	2.12E-04	0.00E+00	0.00E+00	0.00E+00

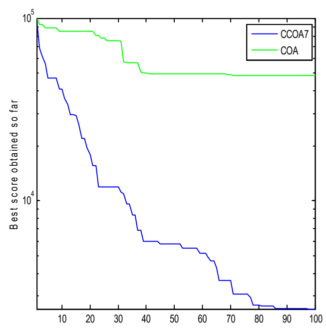


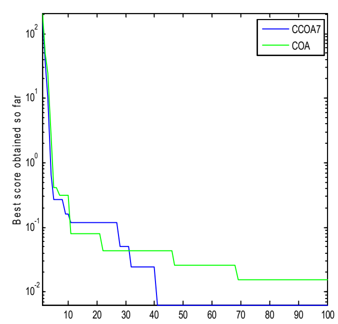




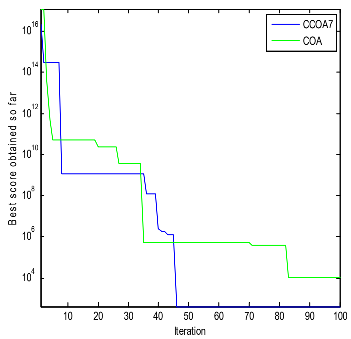
F1 F2 F3



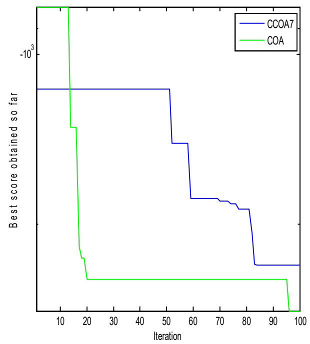




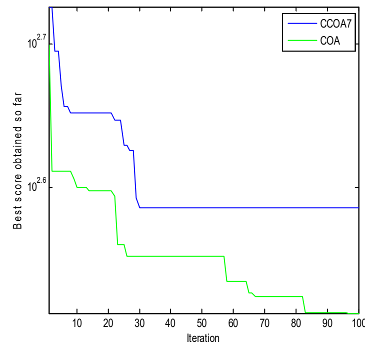
F4 F5 F6



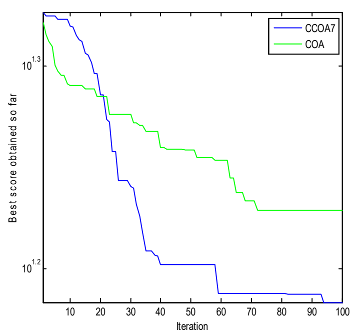
F7



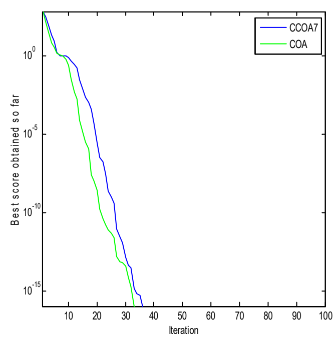
F8



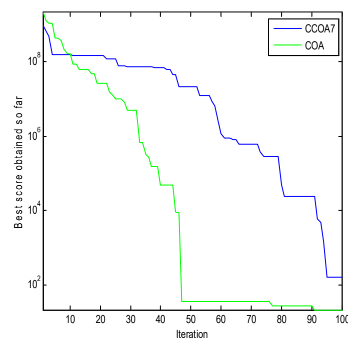
F9



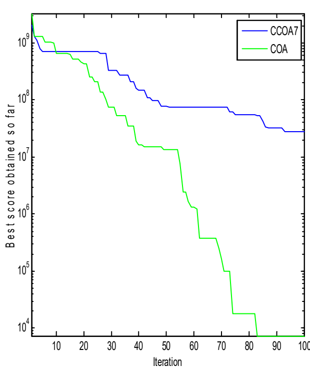
F10



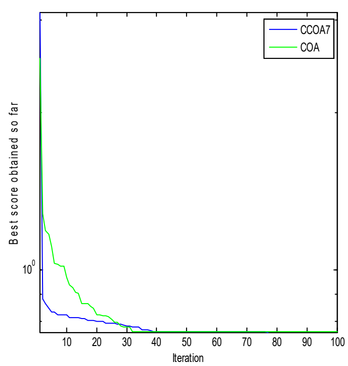
F11



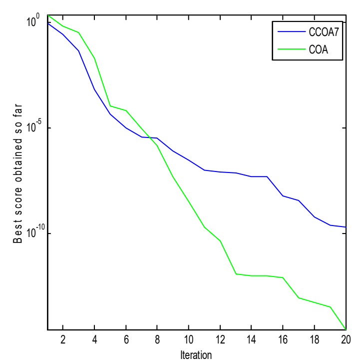
F12



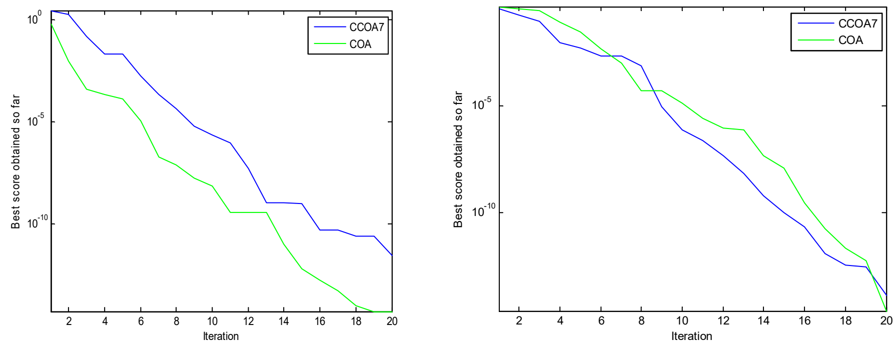
F13



F14

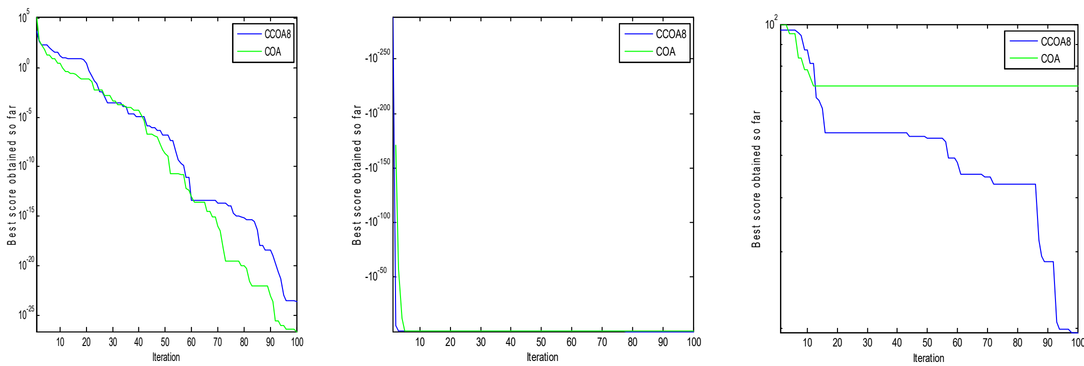


F15

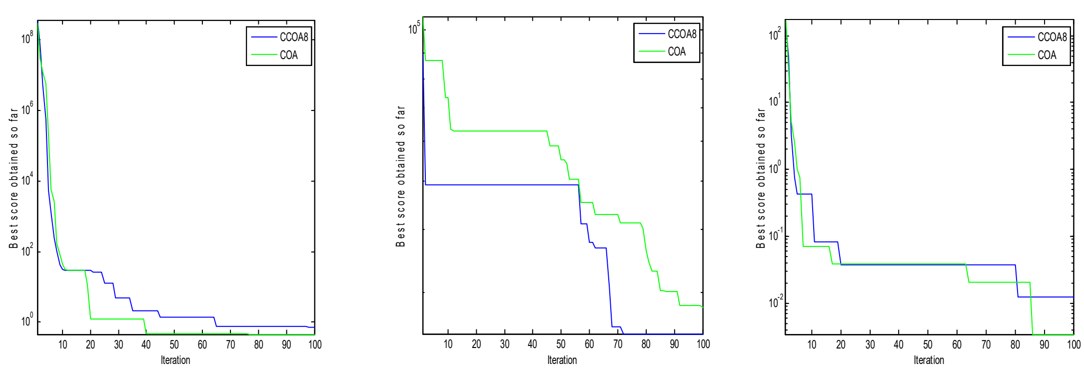


F16 F17

Figure 5: Convergence Curves Of CCOA1 And COA On Solving Benchmark Problems

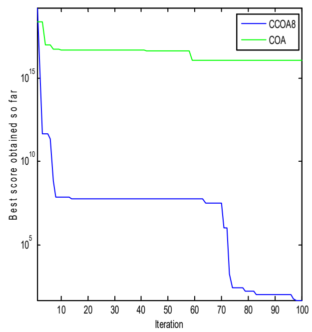


F1 F2 F3

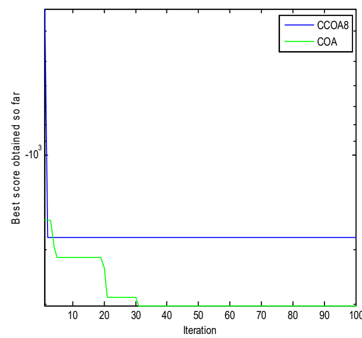


F4 F5 F6

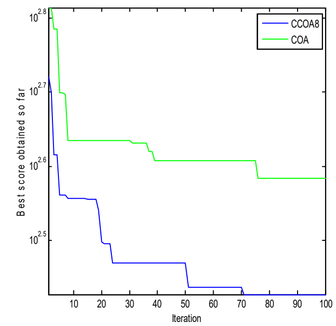
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



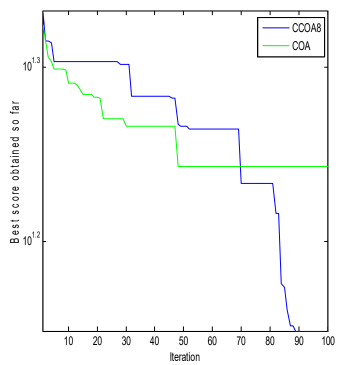
F7



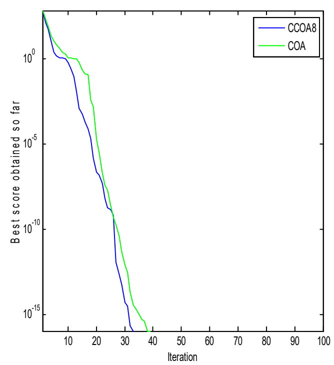
F8



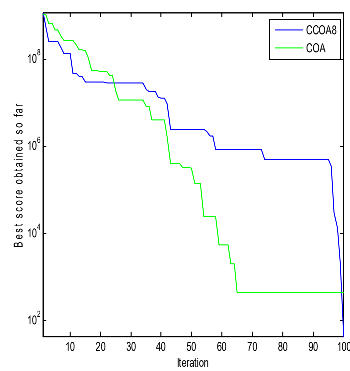
F9



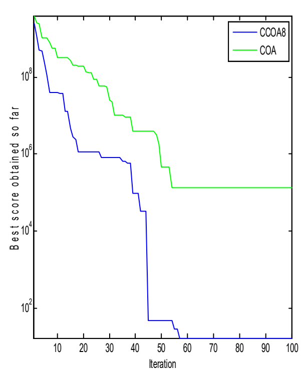
F10



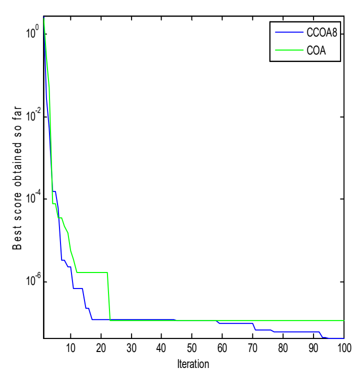
F11



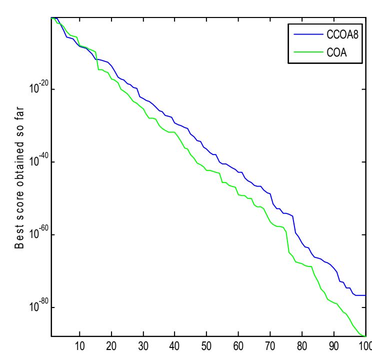
F12



F13

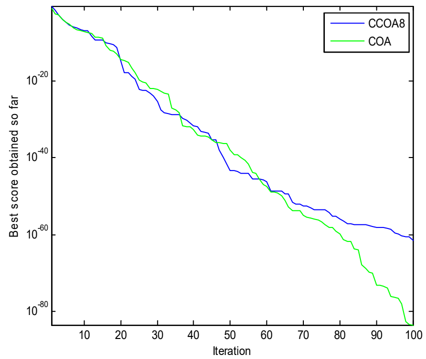


F14

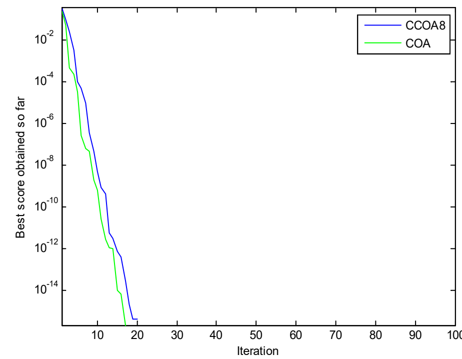


F15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

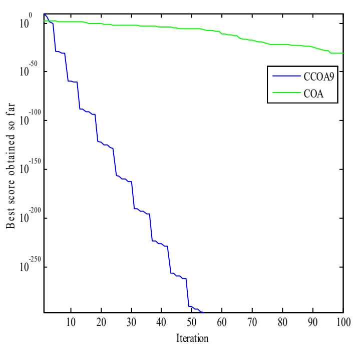


F16

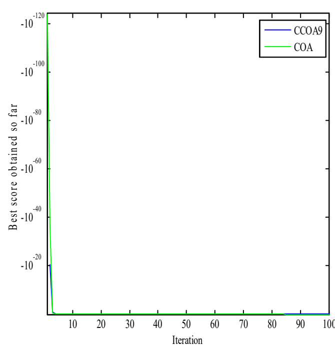


F17

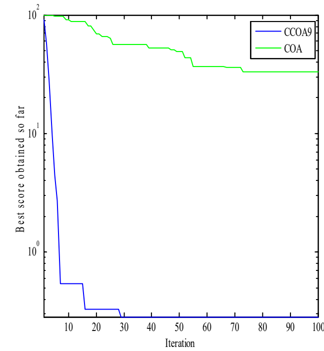
Figure 6: Convergence curves of CCOA2 and COA on solving benchmark problems



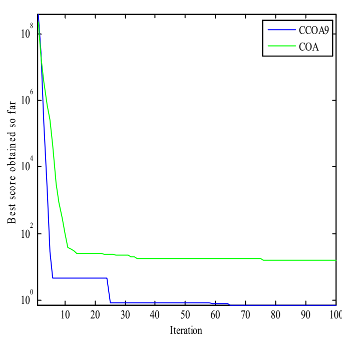
F1



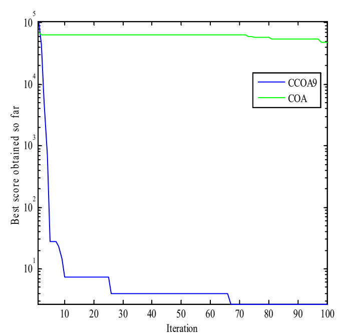
F2



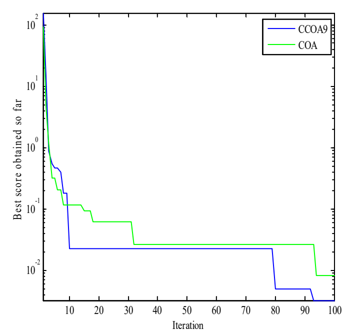
F3



F4

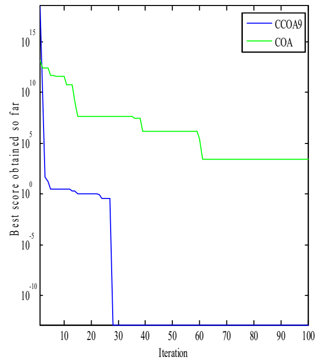


F5

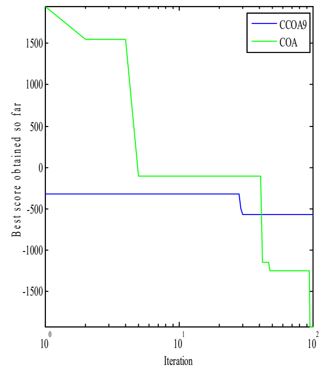


F6

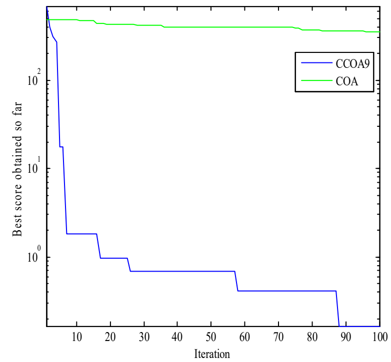
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



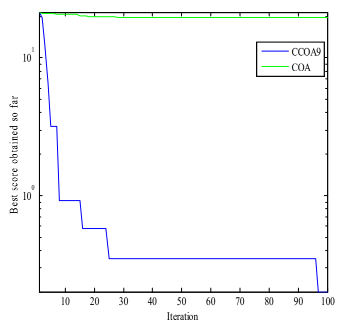
F7



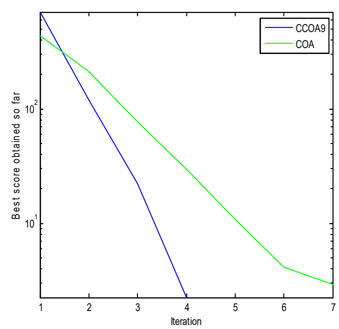
F8



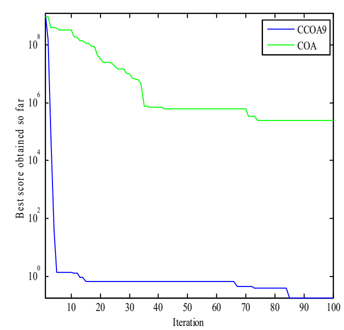
F9



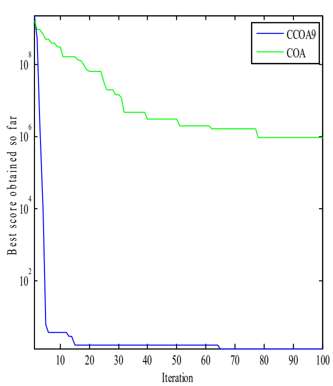
F10



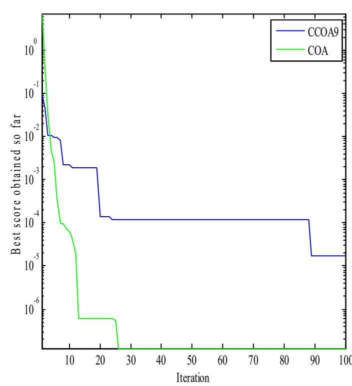
F11



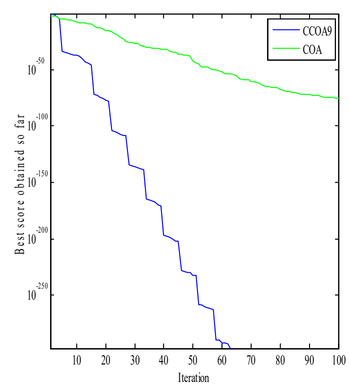
F12



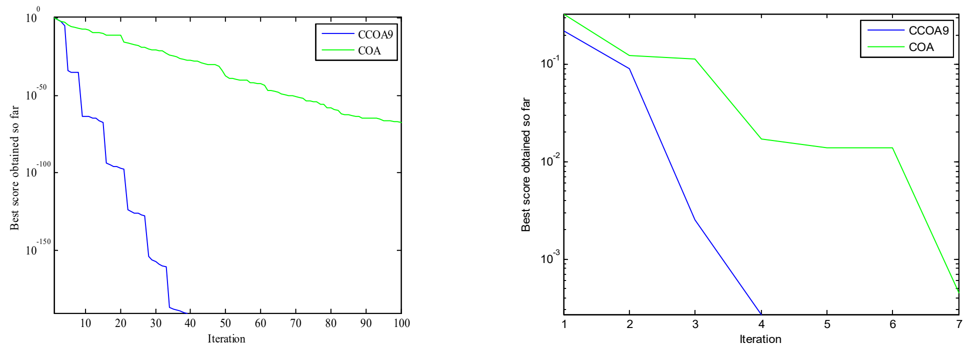
F13



F14



F15



F16 F17

Figure 7: Convergence curves of CCOA3 and COA on solving benchmark problems

4.3 Sensitivity Analysis for CCOA3

To verify consistency and ability of CCOA3, it is tested with different parameters. As showed in table 7, CCOA3 can give the optimal solution for F1, F6, F9, F10, F11, F12 and F13. Table 11 shows results of CCOA3 with population size 50 and dimension 50 while Table 12 shows results of CCOA3 with population size 50 and dimension 100.. We increased population size and dimension to examine high capability and consistency of CCOA3. As seen in the tables, CCOA3 can give also the best results for the test problems at different parameters and high dimension compared to other algorithms, which mentioned before in table 7.

Table 11: Results of CCOA3 with population size =50 and dim=50

Function	Mean	SD	Min	Max
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F6	3.82E-03	3.21E-03	4.07E-04	1.23E-02
F9	7.99E-02	1.10E-01	0.00E+00	3.43E-01
F10	4.75E-02	4.93E-02	4.44E-15	1.35E-01
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	6.89E-02	5.34E-02	1.50E-03	1.75E-01
F13	1.06E+00	6.67E-01	1.36E-01	2.51E+00

Table 12: Results of CCOA3 with population size =50 and dim=100

Function	Mean	SD	Min	Max
F1	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F6	4.57E-03	3.86E-03	1.02E-04	1.43E-02
F9	2.24E-01	2.96E-01	0.00E+00	9.76E-01
F10	9.98E-02	8.43E-02	2.22E-14	2.97E-01
F11	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	8.38E-02	1.17E-01	1.50E-03	4.91E-01
F13	2.59E+00	1.98E+00	2.01E-01	6.76E+00

5. THE PROPOSED ALGORITHMS FOR ENGINEERING DESIGN PROBLEMS

This section applies the three proposed algorithms for two engineering problems, Tension/Compression Spring design problem and Pressure vessel design problem. For a fair comparison with literature, the same penalty function is used to handle constraints.

5.1 Tension/Compression Spring Design Problem

The main objective of this problem is to minimize the weight of a coil spring under compression subject to some constraints such as shear stress, surge frequency, and minimum deflection [33]. This problem has three variables: x_1 is the wire diameter, x_2 is the mean coil diameter, and x_3 is the number of active coils, as seen in figure 8. This problem can be formulated as the following:

$$\min \quad f(x) = (x_3+2)x_2x_1^2$$

$$s.t. \quad g_1(x) = 1 - \frac{x_2^3x_3}{71785x_1^4} \le 0$$

$$g_2(x) = \frac{x_2(4x_2 - x_1)}{x_1^3(12566x_2 - x_1)} + \frac{1}{5108x_1^2} - 1 \le 0$$

$$g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \le 0$$

$$g_4(x) = \frac{2(x_1+x_2)}{3} - 1 \le 0$$

$$0.05 \le x_1 \le 2, \quad 0.25 \le x_2 \le 1.3, \quad 2 \le x_3 \le 15$$

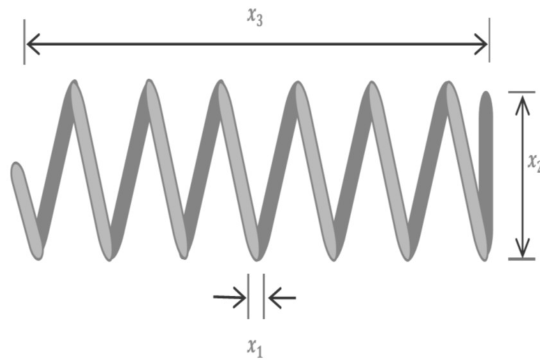


Figure 8: Tension/Compression Spring Design Problem

5.2 Pressure Vessel Design Problem

The main objective of this problem is to minimize the total cost of materials, forming, and welding [33]. This problem has four design variables including, $x_1 (T_s)$ is the thickness of the shell, $x_2 (T_h)$ is the thickness of the head, $x_3 (R)$ is the inner radius, and x_4 (L) is the length of the cylindrical section of the vessel, not including the head. Figure 9 shows the design of this problem. This problem can be formulated as the following:

$$\min \ f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

$$s.t$$

$$g_1(x) = -x_1 + 0.0193x_3 \le 0$$

$$g_2(x) = -x_3 + 0.00954x_3 \le 0$$

$$g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0$$

$$g_4(x) = x_4 - 240 \le 0$$

$$0 \le x_1, x_2 \le 99, \quad 10 \le x_3, x_4 \le 200$$

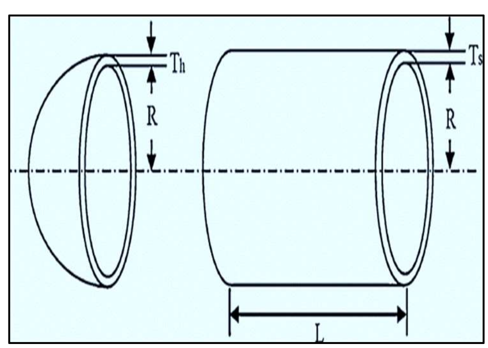


Figure 9: Pressure vessel design problem

Table 13: Results for Tension/Compression Spring Design Problem

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm	x_1	x_2	x_3	$f_{\min}(x)$
CCOA1	5.40E-02	4.14E-01	8.58E+00	1.28E-02
CCOA2	5.31E-02	3.91E-01	9.54E+00	1.2718E-02
CCOA3	**5.22E-02**	**3.685E-01**	**1.06579E+01**	**1.27E-02**
COA	5.24E-02	3.751E-01	1.03522E+01	1.2722E-02
GA [34]	5.8231E-02	5.2106E-01	5.8845E+00	1.3931E-02
TS[35]	N/A	N/A	N/A	1.2935E-02
PSO [36]	N/A	N/A	N/A	1.2857E-02
ACO[37]	N/A	N/A	N/A	1.3223E−02

Table 14: Results for Pressure vessel design problem

Algorithm	x_1	x_2	x_3	x_4	$f_{\min}(x)$
CCOA1	**0.9837**	**0.4877**	**50.8759**	**91.7856**	**6361.5**
CCOA2	1.0762	0.5279	54.9484	64.5892	6710.5
CCOA3	0.8083	0.5104	40.9811	191.9664	6410.2
COA	12.5136	46.7569	62.1525	142.1177	653510
SA[38]	1.125000	0.625000	58.290000	43.6930000	7197.7000
HS[38]	1.125000	0.625000	58.278900	43.75490000	7198.433
GSA [39]	1.1250	0.6250	55.9886598	84.4542025	8538.8359
Branch-bound [39]	1.1250	0.6250	47.700000	117.701000	8129.1036

5.3 Statistical Results For Engineering Design Problems

For spring problem, the proposed algorithms are compared with standard COA, Genetic Algorithm (GA) [34], Tabu Search (TS) [35], Particle Swarm Optimization (PSO) [36] and Ant Colony Optimization (ACO) [37]. Parameters for the compared algorithms, are the same as suggested by authors. As seen in table 13, results verifies the high capability of CCOA3 in getting better results.

For Pressure vessel design problem, the proposed algorithms are compared with standard COA, Simulated Annealing (SA) [38], Harmony Search (HS) [38], Gravitational Search Algorithm GSA [39] and Branch-bound [39]. All parameters are left the same as suggested by authors for compared algorithms. Table 14 shows results for this problem and illustrate how CCOA1 can get better results than the other compared algorithms.

6. CONCLUSION AND FUTURE WORKS

In this study, three chaotic Cuckoo Optimization Algorithm are introduced. In the first algorithm (CCOA1), ELR coefficient is calculated by using chaotic maps and the results showed that Sine map is the best chaotic map for this calculation. In the second algorithm (CCOA2), chaotic maps are used to calculate the immigration coefficient and the results showed that singer map is the most proper for this calculation. In the third algorithm (CCOA3), Chaos is combined with the immigration process to enhance the performance and the results showed that the most suitable map for getting the high performance is Gaussian map. The proposed algorithms are tested on several of unconstrained benchmark problems and the results showed that the proposed algorithms can improve the original algorithm which is the main purpose of this paper. In addition, the proposed algorithms can achieve better results than other compared algorithms. Besides, Wilcoxon test was performed to prove that the results are statistically significant, and the results verified that CCOA3 has a high capability in getting the desired results and also has a high convergence speed for solving unconstrained problems. As well

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

as, two engineering problems are examined and the results showed that CCOA3 can get the best results for Tension/Compression Spring design problem and CCOA1 can get the best results for Pressure Vessel Design Problem. As seen from the results, incorporating chaos in the immigration process helps in avoiding local optimum and increasing convergence speed. We can conclude that the proposed algorithms are more efficient for solving a large number of problems and getting superior results.

For future work, we suggest that the proposed algorithms can be performed for other benchmark functions and more real-world problems because it has the capability in giving better results. Also, chaos can be combined with different methods in the original algorithm for achieving the high performance. Moreover**,** CCOA3 can be combined with other metaheuristic algorithms and getting more desired results.

REFERENCES

[1] Rao, S.S., 2019. Engineering optimization: theory and practice. John Wiley & Sons.
[2] Talbi, E.G., 2009. Metaheuristics: from design to implementation (Vol. 74). John Wiley & Sons.
[3] Holland, J.H., 1975. Adaptation in natural and artificial systems, univ. of mich. press. Ann Arbor.
[4] Kennedy, J. and Eberhart, R., 1995, November. Particle swarm optimization. In Proceedings of ICNN'95-International Conference on Neural Networks (Vol. 4, pp. 1942-1948). IEEE.
[5] Dorigo, M. and Di Caro, G., 1999, July. Ant colony optimization: a new meta-heuristic. In Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406) (Vol. 2, pp. 1470-1477). IEEE.
[6] Tereshko, V. and Loengarov, A., 2005. Collective decision making in honey-bee foraging dynamics. Computing and Information Systems, 9(3), p.1.
[7] Yang, X.S. and Deb, S., 2009, December. Cuckoo search via Lévy flights. In 2009 World congress on nature & biologically inspired computing (NaBIC) (pp. 210-214). IEEE.
[8] Yang, X.S., 2010. A new metaheuristic bat-inspired algorithm. In Nature inspired cooperative strategies for optimization (NICSO 2010) (pp. 65-74). Springer, Berlin, Heidelberg.
[9] Kumar, A. and Bawa, S., 2019. Generalized ant colony optimizer: swarm-based meta-heuristic algorithm for cloud services execution. Computing, 101(11), pp.1609-1632.
[10] Dhiman, G. and Kumar, V., 2017. Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Advances in Engineering Software, 114, pp.48-70.
[11] Mirjalili, S., Mirjalili, S.M. and Lewis, A., 2014. Grey wolf optimizer. Advances in engineering software, 69, pp.46-61.
[12] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P., 1983. Optimization by simulated annealing. science, 220(4598), pp.671-680.
[13] Darwin, C., 1859. On the Origin of Species by Means of Natural Selection Or the Preservation of Favoured Races in the Struggle for Life. H. Milford; Oxford University Press.
[14] Storn, R., 1995. Differrential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, International Computer Science Institute, 11.
[15] Storn, R. and Price, K., 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4), pp.341-359.
[16]. Rechenberg, I., 1965. Cybernetic solution path of an experimental problem. Royal Aircraft Establishment Library Translation 1122.
[17] Vent, W., 1975. Rechenberg, Ingo, Evolutionsstrategie—Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. 170 S. mit 36 Abb. Frommann-Holzboog-Verlag. Stuttgart 1973. Broschiert. Feddes Repertorium, 86(5), pp.337-337.
[18] Leach, W.E., 1820. Eleventh room. Synopsis of the contents of the British Museum, pp.65-70.
[19] Payne, R.B., 1997. Family Cuculidae (cuckoos). Handbook of the birds of the world, 4, pp.508-607.
[20] Payne, R.B. and Sorensen, M.D., 2005. The cuckoos (Vol. 15). Oxford University Press.
[21] Rajabioun, R., 2011. Cuckoo optimization algorithm. Applied soft computing, 11(8), pp.5508-5518.
[22] Naseri, K.N., 2014. A hybrid cuckoo-gravitation algorithm for cost-optimized QFD decision-making problem. J Math Comput Sci, 9, pp.342-351.
[23] Mellal, M.A., Salhi, A. and Williams, E.J., 2020. Accelerated cuckoo optimization algorithm for the multi-objective welding process. Journal of the Brazilian Society of

Mechanical Sciences and Engineering, 42(1), p.6.
[24] Boveiri, H.R. and Elhoseny, M., 2018. A-COA: an adaptive cuckoo optimization algorithm for continuous and combinatorial optimization. Neural Computing and Applications, pp.1-25.
[25] Mousavirad, S.J. and Ebrahimpour-Komleh, H., 2015, November. Entropy based optimal multilevel thresholding using cuckoo optimization algorithm. In 2015 11th International Conference on Innovations in Information Technology (IIT) (pp. 302-307). IEEE.
[26] Mohammadrezapour, O., Yoosefdoost, I. and Ebrahimi, M., 2019. Cuckoo optimization algorithm in optimal water allocation and crop planning under various weather conditions (case study: Qazvin plain, Iran). Neural Computing and Applications, 31(6), pp.1879-1892.
[27] Kaur, G. and Arora, S., 2018. Chaotic whale optimization algorithm. Journal of Computational Design and Engineering, 5(3), pp.275-284.
[28] Sayed, G.I., Khoriba, G. and Haggag, M.H., 2018. A novel chaotic salp swarm algorithm for global optimization and feature selection. Applied Intelligence, 48(10), pp.3462-3481.
[29] Cai, J., Ma, X., Li, L. and Haipeng, P., 2007. Chaotic particle swarm optimization for economic dispatch considering the generator constraints. Energy conversion and management, 48(2), pp.645-653.
[30] Mahmoudi, S. and Lotfi, S., 2015. Modified cuckoo optimization algorithm (MCOA) to solve graph coloring problem. Applied soft computing, 33, pp.48-64.
[31] Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H. and Mirjalili, S.M., 2017. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software, 114, pp.163-191.
[32] Wang, G.G., Deb, S., Gandomi, A.H., Zhang, Z. and Alavi, A.H., 2016. Chaotic cuckoo search. Soft Computing, 20(9), pp.3349-3362.
[33] Rao, R.V. and Waghmare, G.G., 2017. A new optimization algorithm for solving complex constrained design optimization problems. Engineering Optimization, 49(1), pp.60-83.
[34] Du, T.S., Ke, X.T., Liao, J.G. and Shen, Y.J., 2018. DSLC-FOA: Improved fruit fly optimization algorithm for application to structural engineering design optimization problems. Applied Mathematical Modelling, 55, pp.314-339.
[35] Ali, M.Z. and Reynolds, R.G., 2014. Cultural algorithms: a Tabu search approach for the optimization of engineering design problems. Soft Computing, 18(8), pp.1631-1644.
[36] Cai, S.H., Long, W. and Jiao, J.J., 2015. Hybridizing artificial bee colony with biogeography-based optimization for constrained mechanical design problems. Journal of Central South University, 22(6), pp.2250-2259.
[37] Abdel-Basset, M. and Shawky, L.A., 2019. Flower pollination algorithm: a comprehensive review. Artificial Intelligence Review, 52(4), pp.2533-2557.
[38] KS, S.R. and Murugan, S., 2017. Memory based hybrid dragonfly algorithm for numerical optimization problems. Expert Systems with Applications, 83, pp.63-78.
[39] Mirjalili, S., Mirjalili, S.M. and Hatamlou, A., 2016. Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Computing and Applications, 27(2), pp.495-51.