International Journal of Soft Computing 13 (3): 108-122, 2018
ISSN: 1816-9503

Chaotic Sine-Cosine Optimization Algorithms

Dunia S. Tahir and Ramzy S. Ali
Department of Electrical Engineering, University of Basrah, Basrah, Iraq

Abstract: A Sine-Cosine Algorithm (SCA) is a new metaheuristic optimization algorithm. Sine-Cosine Algorithm (SCA) is inspired from the sine and cosine mathematical functions. The standard Sine-Cosine Algorithm (SCA) has some problems, like any of the other techniques such as slow convergence and falling into local solutions. To overcome these problems, this study suggested four different chaotic Sine-Cosine Algorithms (CSCAs) methods. The random parameters in the standard Sine-Cosine Algorithm (SCA) are replaced with the chaotic sequences to improve the performance of the standard algorithm. Five one dimensional various chaotic maps are implemented.The proposed chaotic Sine-Cosine Algorithms (CSCAs) methods are benchmarked on ten test benchmark functions. The statistical results showed that all chaotic Sine-Cosine Algorithms (CSCAs) methods can be outperformed the standard Sine-Cosine Algorithm (SCA) for these benchmark functions and the intermittency and circle maps are the best maps for boosting the performance of the first and fourth chaotic CSCAs. While the Gauss map is the most suitable variant for the second and third chaotic CSCAs methods, respectively. Additionally, the results proved that the fourth proposed algorithm with the circle map significantly overtook on the other proposed algorithms. The effectiveness of all chaotic Sine-Cosine Algorithms (CSCAs) methods are proved by comparing their results with the well-known metaheuristic methods such as the standard Sine-Cosine Algorithm (SCA), Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Differential Evolution (DE).

Key words: Chaos, sine-cosine optimization algorithm, chaotic sine-cosine optimization algorithms, differential, benchmarked, evolutional

INTRODUCTION

Through the amazing laws of nature, scholars have been inspired various metaheuristic optimization algorithms to solve complex problems in different scientific fields and technological such as mathematics, engineering, medicine and economics which were difficult to solve by the classical optimization methods. The reason behind the discovery of dozens of metaheuristic optimization algorithms is “No-Free Lunch Theorem”. (Wolpert and Macready, 1997; Yang, 2014). This theorem confirms that there is no absolute metaheuristic algorithm for solving all optimization problems. This prompted scholars to discover and develop different algorithms such as Genetic Algorithm (GA) (Holland, 1975), Ant Colony Algorithm (ACO) (Dorigo *et al.* , 1996), Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995), Differential Evolution (DE) (Storn and Price, 1997), Harmony Search (HS) (Greem *et al*., 2001), Biogeography-Based optimization (BBO) (Simon, 2008), Firefly Algorithm (FA) (Yang, 2014), Gravitation Search Algorithm (GSA) (Rashedi *et al*., 2009), Cuckoo Search (CS) (Yang and Deb, 2009), Bat Algorithm (BA) (Yang, 2010), Teaching Learning-Based Optimization (TLBO) (Rao *et al*., 2011), Krill Herd (KH) (Gandomi and Alavi, 2012), Water Cycle Algorithm (WCA) (Eskandar *et al*., 2012), Flower Pollination Algorithm (FPA) (Ang, 2012), Grey Wolf Optimizer (GWO) (Mirjalili *et al*., 2014), Social Spider Algorithm (SSA) (Zheng, 2015), Dragonfly Algorithm (DA) (Mirjalili, 2015), Water Wave Optimization (WWO) (Zheng, 2015), Multi-Verse Optimizer (MOV) (Mirjalili *et al*., 2016), Sine-Cosine Algorithm (SCA) (Mirjalili, 2016), Whale Optimization Algorithm (WOA) (Mirjalili and Lewis, 2016), Crow Search Algorithm (CSA) (Askarzadeh, 2016), Jaya Algorithm (JA) (Rao, 2016), Human Mental Search (HMS) (Seyed and Hossein, 2017), Grasshopper Optimization Algorithm (GOA) (Saremi *et al*., 2017), Kidney-Inspired Algorithm (KIA) (Jaaddi and Abdullah, 2017), Most Valuable Player Algorithm (MVPA) (Bouchekara, 2017) and Salp Swarm Algorithm (SSA) (Mirjalili *et al*., 2017).

A Sine-Cosine Algorithm (SCA) is a new robust and simple metaheuristic strategy proposed by Mirjalili (2016). The main idea of the SCA algorithm is inspired from the mathematical form of the sine and cosine functions. Hitherto, SCA algorithm has been used in various kinds of optimization problems to resolve diverse optimization problems like feature selections (Sindhu *et al*., 2017;

Corresponding Author: Dunia S. Tahir, Department of Electrical Engineering, University of Basrah, Basrah, Iraq

Hafez *et al*., 2016), image processing (Elattah *et al*., 2016), power system economics, shell and tube evaporator design problem (Turgut, 2017), energy applications (Kumar *et al*., 2017) and power system security (Mahdad and Srairi, 2018).

In recent times, chaos theory has produced a new branch of the metaheuristic optimization algorithms is called Chaotic Optimization Algorithm (COA). COA represents a promising method for solving complex engineering problems. The chaotic algorithms have advantages of short implementation time, easy to perform and high reliability of absconding from the local optima. Chaotic optimization algorithm employs chaotic variables rather than random variables. Chaos has three important dynamic properties: ergodicity, regularity and Semi-stochastic property (Mousavirad and Ebrahimpour-Komleh, 2017). Chaotic maps are able to effectively improve the performance of metaheuristic algorithms by fine-tuning parameters of these algorithms.

Saremi *et al*. (2014) employed a chaos-based technique for Biogeography-based optimization (Simon, 2008) to resolve unconstrained problems. Several chaotic algorithms are developed depending on replacing the selection, emigration and mutation probabilities with the chaotic maps. The simulations offered that the employment of Gauss map instead of the selection, emigration and/or mutation probabilities are enhanced the adjustment of the BBO algorithm.

Bingol and Alatas (2016) suggested six various Chaotic League Championship Algorithms (CLCAs). The results showed that the Chaotic League Championship Algorithms (CLCAs) increased the convergence speed and prevented of falling into the local solution.

Heidari *et al*. (2017) used different chaotic variants to boost the performance of Water cycle Algorithm (WCA) (Eskander *et al*., 2012). Three different Chaotic Water Cycle Algorithms (CWCAs) are suggested CWCA 1, CWCA 2 and 3 to solve unconstrained and constrained problems. The comparison of the success rate results showed that the CWCA 1 with Iterative map, CWCA 2 and CWCA 3 with sinusoidal map outperformed the basic Water Cycle Algorithm (WCA).

Mirjalili and Gandomi (2017) embedded ten chaotic maps into the Gravitational Search Algorithm (GSA). They proved that the Chaotic Gravitational Search Algorithm (CGSA) is able to outperform the basic version of Gravitational Search Algorithm (GSA) and other well-known algorithms.

Seyed *et al*. (2017) and Ewees *et al*. (2017) proposed feature selection techniques based on the Chaotic Crow Search Algorithm (CCSA) and Chaotic Multi-Verse Optimizer (CMVO), respectively. They revealed from the experimental results that the Chaotic Crow Search Algorithm (CCSA) and Chaotic Multi-Verse Optimizer (CMVO) are able to enhance the crow search algorithm and multi-verse optimizer in the terms of classification performance, number of selected features, convergence speed and stability quality. Tharwat and Hassanien (2017) proposed an efficient classification strategy (CALO-SVM) based on a Chaotic Ant Lion Algorithm (CALO). The simulations proved that (CALO-SVM) algorithm is outperformed PSO, GA and SEOA.

Up to now, chaotic variants have been embedded with different metaheuristic algorithms such as chaos-genetic algorithm (Liao and Tsao, 2006), chaos enhanced accelerated particle swarm algorithm (Gandomi *et al*., 2013a, b), chaotic harmony search algorithm (Yi *et al*., 2016), chaotic Cuckoo algorithm (Wang and Zhong, 2015) and (Feng *et al*., 2017), chaotic Bean Optimization Algorithm (CBOA) (Zhang and Feng, 2018), chaotic imperialist competitive algorithm (Talathari *et al*., 2012), chaotic fire algorithm (Gandomi *et al*., 2013a, b), chaotic bat optimization (Gandomi and Yang, 2014), chaotic bat swarm optimization (Jordehi, 2015a, b), chaotic Krill Herd algorithm (Wang *et al*., 2014), chaotic fruit fly algorithm (Mitic *et al*., 2015), chaotic seeker algorithm (Jordehi, 2015), chaotic teaching-learning based optimization (He *et al*., 2016), chaotic symbiotic organisms algorithm (Saha and Mukhherjee, 2018) and chaotic butterfly algorithm (Arora and Singh, 2017).

In this study, SCA has been embedded with different chaos maps. These chaotic maps have been employed instead of random parameters to increase the performance of the standard SCA in terms of convergence rate and quality solutions. Four chaotic sine cosine algorithm (CSCAs) have been proposed in this study. The performance of these algorithms is tested on several unimodal and multimodal benchmark functions. The initial value 0.7 has been used as the default value. Finally, all proposed chaotic (CSCAs) methods have been compared with the standard SCA and other well-known optimization algorithms.

MATERIALS AND METHODS

Sine-cosine algorithm: Sine-Cosine Algorithm (SCA) which was first proposed by Mirjalili (2016a, b) is inspired from the sine and cosine mathematical functions. Like other metaheuristic optimization algorithms, the optimization process of the SCA contains two mechanisms: exploitation versus exploration, intensification (local search) examines about the existent

preferable solutions and takes the preferable one while diversification (global search) permits for the metaheuristic algorithms to investigate more efficiently in the search space (Yang, 2014). To accentuate these two phases, two position updating equations have been modeled. Equation 1 and 2 are shown as:

$$Y_k^{l+1} = Y_k^l + A*\sin(B)*\left|C*Pos_k^l - Y_k^l\right| \quad (1)$$

$$Y_k^{l+1} = Y_k^l + A*\cos(B)*\left|C*Pos_k^l - Y_k^l\right|$$

Where:

Y_k^l = The position of the current solution in Kith dimension and lth iteration
A, B and C = The random numbers
Pos = The position of the target point in kth dimension

These two equations can be rearranged as follows:

$$Y_k^{l+1} = \begin{cases} Y_k^l + A*\sin(B)* \\ \left|C*Pos_k^l - Y_k^l\right| & D < 0.5 \\ Y_k^l + A*\cos(B)* \\ \left|C*Pos_k^l - Y_k^l\right| & D \geq 0.5 \end{cases}$$

where, D is a random number implemented by the standard uniform distribution. Sine-Cosine Algorithm (SCA) contains four parameters.

A is decided the dimension of the movement of the next solution which might be either in the space between the solution and target or outside it. The range of sine and cosine functions is changing dynamically to achieve balance between the diversification and intensification properties of the Sine Cosine Optimization (SCA) by the following Eq. 4:

$$A = z\text{-itera} * \left(\frac{z}{Max\_itera}\right) \quad (4)$$

Where:

itera = The current iteration
Max-itera = The maximum number of iterations
z = A constant term

- B is decided the amplitude of the movement in the search space
- C is decided the effective of the target point on the course of iterations which represents a random weight
- D is used to control the balance between diversification and intensification phases by switching between sine and cosine functions

To exploit the solution space, the repeated pattern of the sine and cosine functions make a solution to turn around another solution. A more detailed can be found by Mirjalili (2016a, b). The flowchart of a representation Sine-Cosine Algorithm (SCA) is shown in Fig. 1.

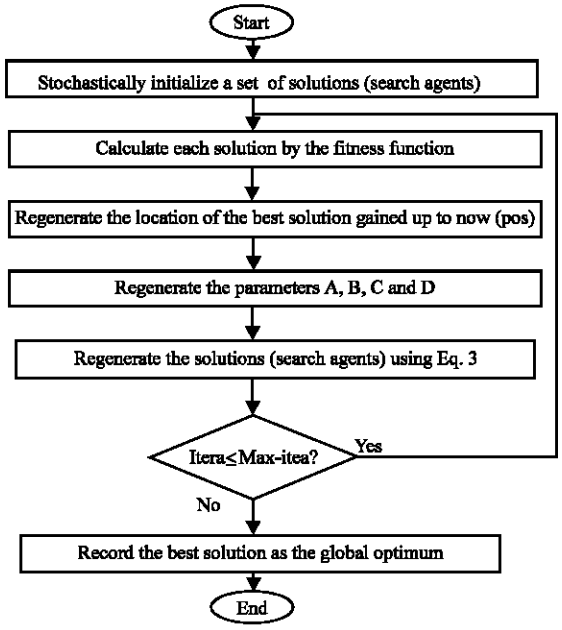

Fig. 1: Flowchart of Sine Cosine Algorithm (SCA)

Chaotic sine cosine algorithms: In this section, different chaotic maps have been presented and four various chaotic Sine Cosine Algorithm (CSCAs) methods have been proposed. Chaotic maps have been used instead of random parameters to boost the performance of the standard Sine Cosine Algorithm (SCA).

Chaotic maps: Chaos is a theory of expecting the unexpected things. It deals with the non-linear systems that are impossible to control or predict its behavior such as the states of brain, weather, turbulence and so on. The powerful mechanisms of chaos are ergodicity, randomness, regularity and sensitivity to the initial conditions. These mechanisms are transformed to different equations which are named chaotic maps. The optimization algorithms that are used these chaotic sequences instead of the random variables which are called Chaotic Optimization Algorithm (COA). The main concessions for chaotic optimization algorithms are high convergence rate and their ability for avoiding the local optima. These concessions have been made it suitable for improving the performance of algorithms (Seyed *et al*., 2017). In this study, 5 different one dimensional, non-invertible sequences are used to constitute four chaotic sine cosine algorithms. The features of chaotic sequences are described in Table 1.

Table 1: Chaotic sequences

Symbol	Name	Chaotic sequence	Range
CM1	Chebyshev (Saleh and Haeri, 2007)	$w_{j+1}=\cos\left(j\cos^{-1}\left(w_j\right)\right)$	(-1,1)
CM2	Circle (Hilborn *et al*., 2009)	$w_{j+1}=\mathrm{mod}\left(w_j+a2-\left(\frac{a1}{2\pi}\right)\sin\left(2\pi w_j\right),1\right)$	a1 = 0.5 and a2 = 0.2(0,1)
CM3	Gauss (He *et al*., 2001)	$w_{j+1}=\begin{cases}1 & w_j=0\\ \frac{1}{\mathrm{mod}\left(w_j,1\right)} & \text{otherwise}\end{cases}$	(0,1)
CM4	Intermittency (Forg *et al*., 2017)	$w_{j+1}=\begin{cases}\varepsilon+w_j+dw_j^n & 0\le w_j\le c\\ \frac{1}{\mathrm{mod}\left(w_j,1\right)} & c<w_j<1\end{cases}$	n = 2 and ε = 0.49(0,1) $d=\frac{1-\varepsilon-c}{c^n}$
CM5	Iterative (Zhenyu *et al*.,2006; Ott, 2002)	$w_{j+1}=\sin\left(\frac{a\pi}{w_j}\right)$	a = 0.7(-1,1)

Chaotic map utilization in SCA: The original SCA has three random parameters: B-D. This subsection is used the chaotic maps in four various methods to produce different variants of the chaotic Sine Cosine Algorithms (CSCAs) methods. Chaotic (CSCAs) methods with chaotic variants can be categorized and clarified briefly as follows:

- CSCA-1: B parameter of Eq. 1-3 is exchanged with the chaotic map
- CSCA-2: C parameter of Eq. 1-3 is exchanged with the chaotic map
- CSCA-3: D parameter of Eq. 3 is exchanged with the chaotic map
- CSCA-4: B, C and D parameters of Eq. 1-3 are exchanged with the chaotic maps

The remarks below shown how the chaotic maps are theoretically efficient, either individual manner or collectively:

- The chaotic maps for B or C random parameter help the chaotic (CSCAs) methods to select the direction of movement chaotically which improves exploration phase
- The chaotic maps for D random parameter makes the chaotic (CSCAs) methods to exploit the search space better than the standard SCA
- The chaotic maps for B-D random parameters provide various intensification and diversification patterns for the chaotic (CSCAs) methods during the optimization problems
- Because of the messy motions that are provided by the mess patterns, the proposed chaotic (CSCAs) methods can confirm either intensification or diversification phases
- Different chaotic operators assist the chaotic (CSCAs) methods to avoid the local optima
- In status of diversifying a desired area (s) of solution space, the chaotic (CSCAs) methods are improved the exploitation phase in order to diversify the better solutions

RESULTS AND DISCUSSION

In this study, ten benchmark functions are used to appraise the performance of the proposed chaotic (CSCAs) strategies (Jamil and Yang, 2013). Ten standard benchmark functions are used at 30 dimensions. These functions are classified into two categories: the first five functions are unimodal while the other five functions are multimodal. The first category is used for benchmarking intensification while the second kind is used to evaluate exploration.

Table 2 tabulates the benchmark functions are used in this study where LB and UB show upper and lower bounds of these functions. The global minimum value is zero for all functions. For all methods used in this study, the results are found over 30 independent runs. The population size and maximum iteration are assigned to 50 and 1000, respectively.

Firstly, various statistical criteria are used such as success rate and average rank to evaluate the performance of the proposed algorithms. The initial value 0.7 has been used as the default value for the chaotic CSCAs methods.

The best, worst, mean, standard deviation and average time of execution are used to verify the performance of the proposed chaotic CSCAs methods. Best represents the minimum result obtained of the total runs. While the worst shows the maximum result obtained of the total runs. Moreover, the proposed chaotic CSCAs are compared with other well-known algorithms. The Wilcoxon's rank sum test at 5% significance level is employed to show significance between two algorithms. Finally, the convergence graphs which are represented the convergence curves of each algorithm for the best result of the benchmark functions within the maximum number of iterations.

Evaluating the performance of the chaotic CSCAs methods according to the criteria of success rate and average rank: According to Mitic *et al*. (2015), success

Table 2: Benchmark functions

FN No.	Name	LB	UP	Equation
1	Brown	-1	4	$F(x)=\sum_{i=1}^{n-1}\left[\left(x_i^2\right)x_{i+1}^2+1+\left(x_{i+1}^2\right)x_i^2+1\right]$
2	Powell	-4	5	$F(x)=\sum_{i=1}^{d/4}\left[\left(x_{4i-3}+10x_{4i-2}\right)^2+5\left(x_{4i-1}-x_{4i}\right)^2+\left(x_{4i-2}-2x_{4i-1}\right)+10\left(x_{4i-3}-x_{4i}\right)^4\right]$
3	Schwefell 2.22	-10	10	$F(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$
4	Sphere	-5.12	5.12	$F(x)=\sum_{i=1}^{n}x_i^2$
5	Quartic without noise	-1.28	1.28	
6	Ackley	-32	32	$F(x)=-20e^{-0.2}\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}-e\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)+20+e$
7	Griewank	-600	600	$F(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{1}}\right)+1$
8	Rastrigin	-5.12	5.12	$F(x)=10n\sum_{i=1}^{n}\left[x_i^2-10\cos\left(2\pi x_i\right)\right]$
9	Wavy	-π	π	$F(x)=1-\frac{1}{n}\sum_{i=1}^{n}\cos\left(kx_i\right)e^{\frac{x_i^2}{2}}$ where k = 10. The number of local minima is kn and (k+1)n for odd and even k, respectively
10	Csendes	-1	1	$F(x)=\sum_{i=1}^{n}x_i^6\left(2+\sin\frac{1}{x_i}\right)$

rate criterion can apply for evaluating the performance of algorithms. The success rate variant SR is defined as (Gandomi *et al*., 2013a, b):

$$SR=\left(\frac{Se}{Te}\right)\times 100 \quad (4)$$

Where:

(Se) = The number of executions that can discover the optimal solution

(Te) = The total number of executions

The criteria for a successful run can be calculated as follows:

$$\left|X^{globalb}-X^{*}\right|\le\left(Up_b-Lo_b\right)\times\beta \quad (5)$$

Where:

($X^{global\ b}$) = The global best value by the new algorithms

(X^*) = The best solution

(Upb) and (Lob) = The upper and lower bounds, respectively

β = The value 10-14

The success rates of the standard SCA and all proposed algorithms on ten benchmark functions are introduced in Table 3-6. For accurate comparisons, the total success rates of the chaotic (CSCAs) methods are expressed in Table 7 which is represented the summation of total success rates at β = 10-14. From Table 7, some important points need to be clarified as follows:

The performance of CSCA-4 is significantly more advanced to the other chaotic techniques. According to the results of the success rates; Circle, Intermittency and Iterative maps are appropriate maps for the CSCA-4 method. For the CSCA-2, Chebyshev, Circle, Gauss and Iterative maps are the most suitable map to boost the efficiency of the standard SCA.

Based on the success rate results, there are more than one chaotic maps are the most efficacious maps for CSCA-1 and CSCA-3. For CSCA-1 the most appropriate maps are Circle, Intermittency and Iterative maps, respectively. While CSCA-3 the best maps are chebyshev, circle, gauss, intermittency and iterative, maps, respectively.

Some of chaotic maps have failed for improving the performance of the chaotic (CSCAs) methods such as Gauss and Intermittency maps at the CSCA-1 and Gauss map for the CSCA-4, respectively.

According to the success rate tests, CSCA-4 method represented the best algorithm by owning it the highest success rate but there are more than one of the chaotic maps that have the same value. To override this problem, average ranks test is used as a second test to distinguish the best chaotic map for each proposed chaotic (CSCAs) methods.

Table 3: Success rate of CSCA-1 with different chaotic maps for benchmark functions with $\beta = 10^{-14}$

Chaotic map name	FN1	FN2	FN3	FN4	FN5	FN6	FN7	FN8	FN9	FN10	Total
Chebyshev	100	100	100	100	100	86	100	100	100	100	986
Circle	100	100	100	100	100	100	100	100	100	100	1000
Gauss/mouse	0	0	0	0	0	0	0	0	0	0	0
Intermittency	100	100	100	100	100	100	100	100	100	100	1000
Iterative	100	100	100	100	100	100	100	100	100	100	1000
Standard SCA	0	0	0	0	0	0	0	0	0	16	16

Table 4: Success rate of CSCA-2 with different chaotic maps for benchmark functions with $\beta = 10^{-14}$

Chaotic map name	FN1	FN2	FN3	FN4	FN5	FN6	FN7	FN8	FN9	FN10	Total
Chebyshev	100	100	100	100	100	100	100	100	100	100	1000
Circle	100	100	100	100	100	100	100	100	100	100	1000
Gauss/mouse	100	100	100	100	100	100	100	100	100	100	1000
Intermittency	100	100	100	100	100	100	66	100	100	100	966
Iterative	100	100	100	100	100	100	100	100	100	100	1000
Standard SCA	0	0	0	0	0	0	0	0	0	16	16

Table 5: Success rate of CSCA-III with different chaotic maps for benchmark functions with $\beta = 10^{-14}$

Chaotic map name	FN1	FN2	FN3	FN4	FN5	FN6	FN7	FN8	FN9	FN10	Total
Chebyshev	100	100	100	100	100	100	100	100	100	100	1000
Circle	100	100	100	100	100	100	100	100	100	100	1000
Gauss/mouse	100	100	100	100	100	100	100	100	100	100	1000
Intermittency	100	100	100	100	100	100	100	100	100	100	1000
Iterative	100	100	100	100	100	100	100	100	100	100	1000
Standard SCA	0	0	0	0	0	0	0	0	0	16	16

Table 6: Success rate of CSCA-IV with different chaotic maps for benchmark functions with $\beta = 10^{-14}$

Chaotic map name	FN1	FN2	FN3	FN4	FN5	FN6	FN7	FN8	FN9	FN10	Total
Chebyshev	100	100	100	100	100	96	100	93	80	100	869
Circle	100	100	100	100	100	100	100	100	100	100	1000
Gauss/mouse	0	0	0	0	0	0	0	0	0	0	0
Intermittency	100	100	100	100	100	100	100	100	100	100	1000
Iterative	100	100	100	100	100	100	100	100	100	100	1000
Standard SCA	0	0	0	0	0	0	0	0	0	16	16

Table 7: Comparison of the total success rates for all benchmark functions

Chaotic map name	CSCA-I	CSCA-II	CSCA-III	CSCA-IV
Chebyshev	986	1000^{x}	1000^{x}	869
Circle	1000^{x}	1000^{x}	1000^{x}	1000^{x}
Gauss/mouse	0	1000^{x}	1000^{x}	0
Intermittency	1000^{x}	966	1000^{x}	1000^{x}
Iterative	1000^{x}	1000^{x}	1000^{x}	1000^{x}

The best maps (total success rate of the original algorithm is 16)

Table 8-11 show the average ranks of the chaotic CSCAs strategies for ten benchmark functions. The best (min) results are indicated in the bold type. From these tables, the following notes have been concluded:

- Intermittency map represents the best chaotic map for the CSCA-1 algorithm
- For the CSCA-2 and A-3, the most efficient map is the Gauss chaotic map
- Finally, Circle map appears as the most efficacious map for the CSCA-4 algorithm

According to assessments of the success rates and average ranks, circle map is selected as the best chaotic variant and the CSCA-4 algorithm is the best chaotic SCA algorithm. Chaotic SCA-4 with circle map can detect the high-quality solutions and avoid falling into local solutions.

Generally, the experiment results of the chaotic variants on ten benchmark functions show that the chaotic maps can be arranged for each of the four proposed algorithms from the best (min) to the worst (max) as follows:

- For CSCA-1: Intermittency<Circle<Iterative <Chebyshev<Gauss
- For CSCA-2: Gauss<Circle<Iterative<Chebyshev< Intermittency
- For CSCA-3: Gauss<Intermittency<Circle< Chebyshev<Iterative
- For CSCA-4: Circle<Intermittency<Iterative< Chebyshev<Gauss

It can deduce from the above results that the Intermittency map represents the best (min) results whereas the Gauss map provides the worst (max) results for the CSCA-I method.

Table 8: Average rank of CSCA-I and SCA algorithms on the test functions

	Benchmark functions										
Algorithms	FN1	FN2	FN3	FN4	FN5	FN6	FN7	FN8	FN9	FN10	Total average rank
SCA	5	5	5	5	5	5	5.0	5.0	5.0	5	5.00
CSCA-I-CM1	4	4	4	4	4	4	2.5	2.5	2.5	4	3.55
CSCA-I-CM2	2	1	2	1	2	3	2.5	2.5	2.5	1	1.95
CSCA-I-CM3	6	6	6	6	6	6	6.0	6.0	6.0	6	6.00
CSCA-I-CM4	1	2	1	2	1	1	2.5	2.5	2.5	2	1.75
CSCA-I-CM5	3	3	3	3	3	2	2.5	2.5	2.5	3	2.75

Table 9: Average rank of CSCA-II and SCA algorithms on the test functions

	Benchmark functions										
Algorithms	FN1	FN2	FN3	FN4	FN5	FN6	FN7	FN8	FN9	FN10	Total average rank
SCA	6	6	6	6	6	6	6	6	6	6	6.0
CSCA-II-CM1	4	4	4	4	4	4	3	3	3	4	3.7
CSCA-II-CM2	2	2	2	2	2	2	3	3	3	2	2.3
CSCA-II-CM3	1	1	1	1	1	1	3	3	3	1	1.6
CSCA-II-CM4	5	5	5	5	5	5	3	3	3	5	4.4
CSCA-II-CM5	3	3	3	3	3	3	3	3	3	3	3.0

Table 10: Average rank of CSCA-III and SCA algorithms on the test functions

	Benchmark functions										
Algorithms	FN1	FN2	FN3	FN4	FN5	FN6	FN7	FN8	FN9	FN10	Total average rank
SCA	6	6	6	6	6	6.0	6	6	6	6	6.00
CSCA-III-CM1	4	2	4	2	5	1.5	3	3	3	5	3.25
CSCA-III-CM2	2	1	3	5	4	4.0	3	3	3	2	3.00
CSCA-III-CM3	5	5	1	1	3	1.5	3	3	3	1	2.65
CSCA-III-CM4	1	4	2	3	1	4.0	3	3	3	3	2.70
CSCA-III-CM5	3	3	5	4	2	4.0	3	3	3	4	3.40

Table 11: Average rank of CSCA-IV and SCA algorithms on the test functions

	Benchmark functions										
Algorithms	FN1	FN2	FN3	FN4	FN5	FN6	FN7	FN8	FN9	FN10	Total average rank
SCA	5	5	5	5	5	5	5.0	5	5	5.0	5.000
CSCA-IV-CM1	4	4	4	4	4	4	2.5	4	4	2.5	3.700
CSCA-IV-CM2	1	1	1	1	1	2	2.5	2	2	2.5	1.600
CSCA-IV-CM3	6	6	6	6	6	6	6.0	6	6	6.0	6.000
CSCA-IV-CM4	2	3	2	2	3	2	2.5	2	2	2.5	2.300
CSCA-IV-CM5	3	2	3	3	2	2	2.5	2	2	2.5	2.413

For the CSCA-2 and A-3 methods, the best results can be calculated by the Gauss map while the worst results are provided using the Intermittency and Iterative maps, respectively.

The Circle map can enhance the performance of the CSCA-4 method by finding the minimum solutions whereas the Gauss map lead to obtain the worst solutions.

Evaluating the performance of chaotic CSCAs strategies versus other algorithms: In this subsection, the performance of chaos-based CSCAs methods are compared with the conventional SCA and three well-known metaheuristic optimization algorithms at D = 30. These algorithms are Genetic Algorithm (GA), Particle Swarm Optimization (PSO) algorithm and Differential Evolution (DE) algorithm. Table 12 shows the values of the essential parameters of these algorithms. All algorithms are run with 30 independent runs, population size 50 and maximum number of iterations 1000.

Table 12: Parameters of selecting algorithms

Algorithms	Parameters
GA	Selection: Roulette-wheel
	Mutation: Uniform (p = 0.3)
	Crossover: Single point (p = 0.7)
	Type: Real coded
PSO	Cognitive constant (c1): 2
	Social constant (c2): 2
	Inertial weight (w): Linearly decreasing (0.9-0.3)
DE	Crossover constant (CR): 0.9
	Weight Factor (F): 0.8

In Table 13, the statistical results obtained by the proposed chaotic (CSCAs) methods, the conventional SCA, GA, PSO and DE are submitted. Table 13 proves that, on all statistical results and on ten benchmark functions, the CSCA-4 performs better than the other algorithms when searching for optima solutions. The CSCA-2, CSCSA-1 and CSCA-3 are the second, third and fourth most reliability algorithms, respectively. PSO, DE,

Table 13: Comparison results of CSCA-1, CSCA-2, CSCA-3, CSCA-4, SCA, GA, PSO and DE on ten selected benchmark functions at D = 30

FN No. Results	CSCA-1	CSCA-2	CSCA-3	CSCA-4	SCA	GA	PSO	DE
FN1								
Best	2.5902938e-61	1.5559960e-83	2.3671047e-39	1.1157635e-241	2.2707575e-12	7.0680616e-28	2.2935733e-13	3.1815605e-15
Worst	1.5581116e-53	2.6848115e-68	8.0932456e-32	7.9634879e-241	4.9400774e-05	1.0927036e+01	3.0931716e-10	6.3248615e-13
Mean	1.4042941e-54	1.4125046e-69	4.6561048e-33	5.0033067e-241	2.5846085e-06	5.1905899e-01	2.1127842e-11	5.3346718e-14
Std.	3.9637939e-54	5.3491452e-69	1.5095228e-32	0	9.2651029e-06	2.0088372e+00	5.6385824e-11	1.1256724e-13
T(s)	1.27	1.17	1.47	1.23	3.03	6.1	7.7	6.4
$SR10^{-14}$	100	100	100	100	0	33	0	20
Rank	3	2	4	1	7	8	6	5
FN2								
Best	1.5313952e-57	6.7085600e-76	7.8505977e-36	3.1171914e-245	1.0804986e-05	2.0393697e-02	6.2649122e-04	5.1982650e-04
Worst	2.7685052e-44	3.2233276e-66	4.4102110e-26	1.0031262e-237	1.9624383e+00	1.0139682e+03	5.4225157e-03	1.8715089e-03
Mean	2.0408477e-45	3.1245433e-67	1.5981567e-27	6.9716508e-239	8.0586496e-02	3.8265654e+01	2.9428188e-03	1.1520618e-03
Std.	6.5262890e-45	8.3876767e-67	8.0504404e-27	0	3.5671987e-01	1.8485443e+02	1.0576308e-03	3.3387212e-04
T(s)	1.27	1.47	1.2	1.17	1.57	4.03	9.8	5.53
$SR10^{-14}$	100	100	100	100	0	0	0	0
Rank	3	2	4	1	7	8	6	5
FN3								
Best	1.3880638e-33	1.2314875e-41	3.6313750e-22	1.3391716e-121	1.7914365e-08	5.1615615e-19	1.2657242e-07	6.6683604e-07
Worst	1.0419598e-29	5.6134617e-35	8.3066188e-18	3.1760650e-119	8.1554783e-05	9.5737642e-02	7.5729690e-06	7.0572790e-06
Mean	9.2529464e-31	2.0966919e-36	5.3815235e-19	9.1324938e-120	9.3058969e-06	4.2154391e-03	1.9943767e-06	2.4489242e-06
Std.	2.0933106e-30	1.0254148e-35	1.5308495e-18	8.4274771e-120	1.7976779e-05	1.8172208e-02	2.0573994e-06	1.6108759e-06
T(s)	1.33	1.37	1.03	1.1	2.1	4.6	6.9	6.1
$SR10^{-14}$	100	100	100	100	0	83	0	0
Rank	3	2	4	1	7	8	5	6
FN4								
Best	1.1463670e-59	5.1339684e-86	5.5588002e-39	1.8751241e-246	1.1650222e-09	1.0948814e-27	7.4548629e-13	1.0907517e-14
Worst	5.0657515e-52	6.4160424e-67	4.7824206e-32	2.3160655e-240	2.4377576e-05	3.6067250e-02	9.3344414e-11	4.7585369e-13
Mean	3.4131352e-53	3.3806838e-68	5.5406137e-33	2.1214015e-241	2.6283098e-06	1.2102956e-03	1.9335790e-11	7.5992596e-14
Std.	1.0734473e-52	1.2726310e-67	1.2466693e-32	0	5.1339913e-06	6.5835661e-03	2.5945881e-11	1.0760913e-13
T(s)	1.03	1.43	1.23	0.93	1.77	5.37	8.53	8.33
$SR10^{-14}$	100	100	100	100	0	56	0	0
Rank	3	2	4	1	7	8	6	5
Best	5.8602328e-114	8.6750265e-150	2.8291468e-70	0	1.3621276e-13	1.1141136e-15	3.5586286e-19	1.2057814e-2411
FN5								
Worst	1.1383977e-98	3.9740056e-129	6.5569495e-57	0	3.4914192e-06	5.9596469e-01	3.0351699e-16	8.5237312e-20
Mean	3.9297586e-100	2.3773101e-130	6.4682774e-58	0	2.8637451e-07	4.4650268e-02	4.3273148e-17	4.7465667e-21
Std.	2.0767836e-99	9.1103372e-130	1.5859222e-57	0	6.6117212e-07	1.1777735e-01	7.0359905e-17	1.6365150e-20
T(s)	1.5	2.07	1.7	1.73	3.2	4.77	7.67	7.4
$SR10^{-14}$	100	100	100	100	0	10	100	100
Rank 3	2	4	1	7	8	6	5	
FN6								
Best	8.8817842e-16	8.8817842e-16	8.8817842e-16	8.8817842e-16	1.3621276e-13	5.3693005e-01	4.5587695e-01	1.0885262e+00
Worst	4.4408921e-15	8.8817842e-16	4.4408921e-15	8.8817842e-16	3.4914192e-06	1.3788327e+00	9.9316086e-01	1.6453036e+00
Mean	2.4276877e-15	8.8817842e-16	4.3224683e-15	8.8817842e-16	2.8637451e-07	9.2085553e-01	6.3325208e-01	1.3897882e+00
Std.	1.7905923e-15	0	6.4863381e-16	0	6.6117212e-07	2.2535584e-01	1.2843678e-01	1.2835836e-01
T(s)	1.87	2.27	1.77	2.03	3.2	6.03	8.53	6.47
$SR10^{-14}$	100	100	100	100	0	0	0	0
Rank	3	1	4	1	5	7	6	8
FN7								
Best	0	0	0	0	1.0706630e-02	0	7.6143181e-10	6.4122041e-12
Worst	0	0	0	0	2.6324016e+00	4.5465133e+00	6.3860472e-02	1.7241032e-02
Mean	0	0	0	0	9.9875281e-01	2.3771373e-01	1.3369344e-02	2.1349666e-03
Std.	0	0	0	0	4.7398873e-01	8.5534608e-01	1.4764550e-02	5.0412958e-03
T(s)	1.23	1.93	1.63	1.53	3.83	5.3	7.93	7
$SR10^{-14}$	100	100	100	100	0	33	0	0
Rank	1	1	1	1	8	7	6	5
FN8								
Best	0	0	0	0	3.8138069e-08	1.9619374e+01	2.2884028e+01	1.1705960e+02
Worst	0	0	0	0	9.7205937e+01	7.4216656e+01	6.9647023e+01	2.0812405e+02
Mean	0	0	0	0	1.2950248e+01	4.3457430e+01	4.0959093e+01	1.6703773e+02
Std.	0	0	0	0	2.1886133e+01	1.5430151e+01	9.9395539e+00	2.3666387e+01
T(s)	1.23	1.93	2.1	1.4	1.93	5.23	8.03	7.93
$SR10^{-14}$	100	100	100	100	0	0	0	0
Rank	1	1	1	1	5	7	6	8
FN9								
Best	0	0	0	0	8.6605167e-12	1.4906787e-01	1.3708617e-01	5.3940303e-01

Table 13: Continue

FN No. Results	CSCA-1	CSCA-2	CSCA-3	CSCA-4	SCA	GA	PSO	DE
Worst	0	0	0	0	1.3276471e-01	3.2859637e-01	4.6430734e-01	6.9537607e-01
Mean	0	0	0	0	1.1991135e-02	2.4854778e-01	2.8086648e-01	6.5108134e-01
Std.	0	0	0	0	2.8836541e-02	3.7059400e-02	6.3532021e-02	3.5955894e-02
T(s)	1.47	1.87	1.67	1.57	2.63	5.97	9.13	8.37
SR10^{-14}	100	100	100	100	0	0	0	0
Rank	1	1	1	1	5	6	7	8
FN10								
Best	1.1490905e-178	6.7837347e-235	3.8714686e-103	0	3.4337643e-21	3.5330725e-08	6.7525375e-23	6.7487857e-30
Worst	8.2376512e-145	6.7837347e-235	1.8764117e-86	0	8.6321310e-04	1.0328578e-02	2.1795243e-17	1.1410678e-07
Mean	4.7754237e-146	6.7837347e-235	8.1709735e-88	0	3.3554301e-05	5.6662283e-04	1.2982219e-18	4.0873754e-09
Std.	1.8386651e-145	6.7837347e-235	3.4491859e-87	0	1.5885517e-04	2.0582404e-03	4.3481726e-18	2.0809814e-08
T(s)	2.4	3.3	2.67	2.9	2.5	6.63	9.73	9.07
SR10^{-14}	100	100	100	100	16	0	100	86
Rank	3	2	4	1	7	8	5	6
Average	Rank	2.4	1.6	3.1	1	6.5	7.5	5.9	6.1
Total	Rank	3	2	4	1	7	8	5	6

Table 14: p-values of wilcoxon rank sum test for CSCA-1, SCA, GA, PSO and DE on ten selected benchmark functions at D = 30 and 30 runs

FN No.	CSCA-I vs. SCA	CSCA-I vs. GA	CSCA-I vs. PSO	CSCA-I vs. DE
1	3.019e-11	3.019e-11	3.019e-11	3.019e-11
2	3.019e-11	3.019e-11	3.019e-11	3.019e-11
3	3.019e-11	3.019e-11	3.019e-11	3.019e-11
4	3.019e-11	3.019e-11	3.019e-1	3.019e-11
5	3.019e-11	3.019e-11	3.019e-11	3.019e-11
6	1.405e-11	1.405e-11	1.405e-11	1.405e-11
7	1.211e-12	1.701e-08	1.211e-12	1.211e-12
8	1.211e-12	1.211e-12	1.211e-12	1.211e-12
9	1.211e-12	1.211e-12	1.211e-12	1.211e-12
10	3.019e-11	3.019e-11	3.019e-11	3.018e-11

Table 15: p-values of wilcoxon rank sum test for CSCA-II, SCA, GA, PSO and DE on ten selected benchmark functions at D = 30 and 30 runs

FN No.	CSCA-II vs. SCA	CSCA-II vs. GA	CSCA-II vs. PSO	CSCA-II vs. DE
1	3.019e-11	3.019e-11	3.019e-11	3.019e-11
2	3.019e-11	3.019e-11	3.019e-11	3.019e-11
3	3.019e-11	3.019e-11	3.019e-11	3.019e-11
4	3.019e-11	3.019e-11	3.019e-11	3.019e-11
5	3.019e-11	3.019e-11	3.019e-11	3.019e-11
6	1.211e-12	1.211e-12	1.211e-12	1.211e-12
7	1.211e-12	1.701e-08	1.211e-12	1.211e-12
8	1.211e-12	1.211e-12	1.211e-12	1.211e-12
9	1.211e-12	1.211e-12	1.211e-12	1.211e-12
10	3.019e-11	3.019e-11	3.019e-11	3.018e-11

Table 16: p-values of wilcoxon rank sum test for CSCA- 3, SCA, GA, PSO and DE on ten selected benchmark functions at D = 30 and 30 runs

FN No.	CSCA-3 vs. SCA	CSCA-3 vs. GA	CSCA-3 vs. PSO	CSCA-3 vs. DE
1	3.019e-11	3.019e-11	3.019e-11	3.019e-11
2	3.019e-11	3.019e-11	3.019e-11	3.019e-11
3	3.019e-11	2.669e-09	3.019e-11	3.019e-11
4	3.019e-11	3.019e-11	3.019e-11	3.019e-11
5	3.019e-11	3.019e-11	3.019e-11	3.019e-11
6	1.720e-12	1.720e-12	1.720e-12	1.720e-12
7	1.211e-12	1.701e-08	1.211e-12	1.211e-12
8	1.211e-12	1.211e-12	1.211e-12	1.211e-12
9	1.211e-12	1.211e-12	1.211e-12	1.211e-12
10	3.019e-11	3.019e-11	3.019e-11	3.018e-11

Table 17: p-values of wilcoxon rank sum test for CSCA- IV, SCA, GA, PSO and DE on ten selected benchmark functions at D =30 and 30 runs

FN No.	CSCA-IV vs. SCA	CSCA-IV vs. GA	CSCA-IV vs. PSO	CSCA-IV vs. DE
1	3.019e-11	3.019e-11	3.019e-11	3.019e-11
2	3.019e-11	3.019e-11	3.019e-11	3.019e-11
3	3.019e-11	3.019e-11	3.019e-11	3.019e-11
4	3.019e-11	3.019e-11	3.019e-11	3.019e-11
5	1.211e-12	1.211e-12	1.211e-12	1.211e-12
6	1.211e-12	1.211e-12	1.211e-12	1.211e-12
7	1.211e-12	1.701e-08	1.211e-12	1.211e-12
8	1.211e-12	1.211e-12	1.211e-12	1.211e-12
9	1.211e-12	1.211e-12	1.211e-12	1.211e-12
10	1.211e-12	1.211e-12	1.211e-12	1.210e-12

Conventional SCA and GA are ranked as the fifth, sixth, seventh and eighth most efficacious algorithms, respectively.

The results on all functions confirm that the CSCA-4 method with Circle chaotic map outperformed over the conventional SCA, GA, PSO and DE algorithms. Table 14-17 show the p-values between the CSCA-1 to 4 and other metaheuristic optimization algorithms. The p>5% indicate that the rank sum test rejects the null hypothesis and the differences between the proposed chaotic CSCAs methods and other compared algorithms are significant.

Generally, these results prove that the chaotic CSCAs strategies are able to outperform the standard SCA, GA, PSO and DE algorithms.

Figure 2 shows the convergence curves of the CSCA-1 to 4 with the conventional SCA, GA, PSO and DE algorithms for ten benchmark functions. As in this Fig. 2, the chaotic CSCAs methods have a faster convergence rate than other algorithms.

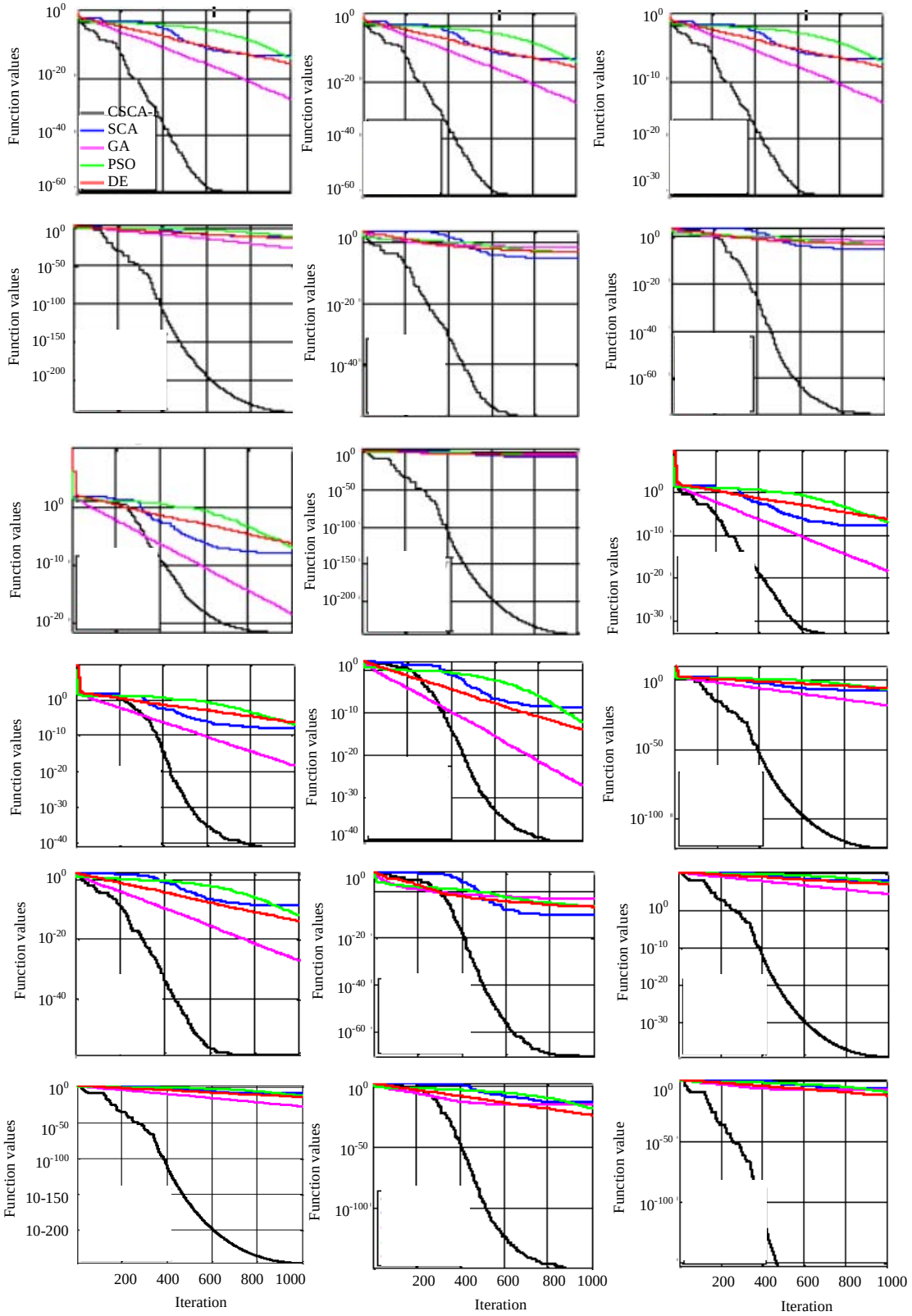

Fig. 2: Continue

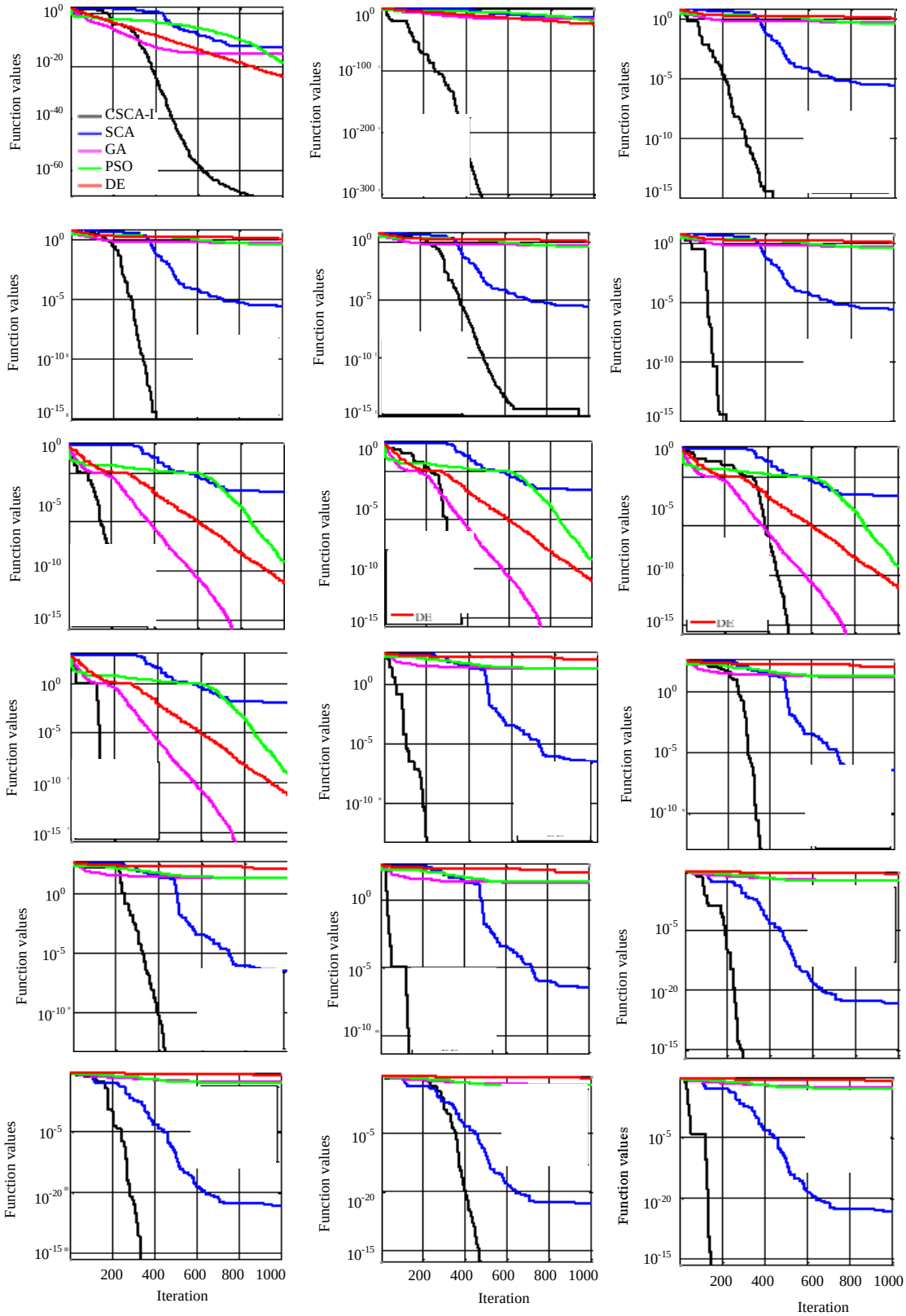

Fig. 2: Continue

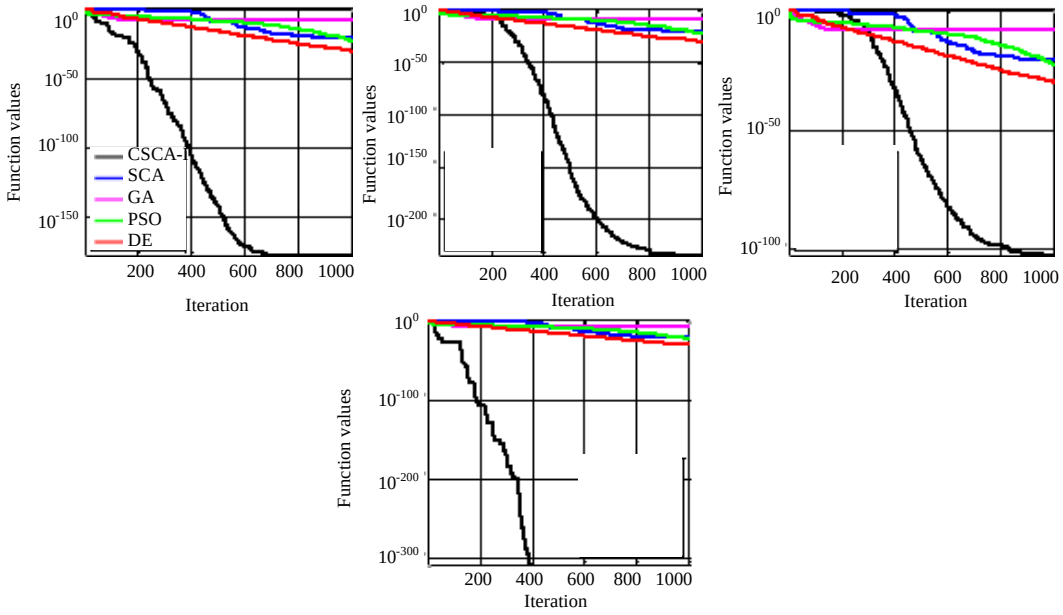

Fig. 2: Convergence curves for CSCA-1, CSCA-2, CSCA-3 and CSCA-4 with standard SCA, GA, PSO and DE alogorithms; FN1Brown function at D = 30 using CM4 for CSCA-1; FN1Brown function at D = 30 using CM4 for CSCA-2; FN1Brown function at D = 30 using CM4 for CSCA-3; FN1Brown function at D = 30 using CM4 for CSCA-4; Powell function at D = 30 using CM4 for CSCA-1; Powell function at D = 30 using CM4 for CSCA-2;Schwefel function at D = 30 using CM3 for CSCA-3; Powell function at D = 30 using CM4 for CSCA-4; Schwefel 2.22 function at D = 30 using CM4 for CSCA-1; Schwefel 2.22 function at D = 30 using CM4 for CSCA-2; Sphere function at D = 30 using CM3 for CSCA-3; Schwefel 2.22 function at D = 30 using CM4 for CSCA-4; Sphere function at D = 30 using CM4 for CSCA-1; Sphere function at D = 30 using CM3 for CSCA-2;Powell function at D = 30 using CM3 for CSCA-2; Sphere function at D = 30 using CM2 for CSCA-4; Quartic function at D = 30 using CM4 for CSCA-1; Quartic function at D = 30 using CM3 for CSCA-2; Quartic function at D = 30 using CM3 for CSCA-3; Quartic function at D = 30 using CM2 for CSCA-4; Ackley function at D = 30 using CM4 for CSCA-1; Ackley function at D = 30 using CM3 for CSCA-2; Ackley function at D = 30 using CM3 for CSCA-3; Ackley function at D = 30 using CM2 for CSCA-4; Griewank function at D = 30 using CM4 for CSCA-1; Griewank function at D = 30 using CM3 for CSCA-2; Griewank function at D = 30 using CM3 for CSCA-3; Griewank function at D = 30 using CM2 for CSCA-4; Rastrigin function at D = 30 using CM4 for CSCA-1; Rastrigin function at D = 30 using CM3 for CSCA-2; Rastrigin function at D = 30 using CM3 for CSCA-3; Rastrigin function at D = 30 using CM2 for CSCA-4; Wavy function at D = 30 using CM4 for CSCA-1; Wavy function at D = 30 using CM3 for CSCA-2; Wavy function at D = 30 using CM3 for CSCA-3; Wavy function at D = 30 using CM2 for CSCA-4; Csendes function at D = 30 using CM4 for CSCA-1; Csendes function at D = 30 using CM3 for CSCA-2; Csendes function at D = 30 using CM3 for CSCA-3; Csendes function at D = 30 using CM2 for CSCA-4

CONCLUSION

In this study, five chaotic variants were used to improve the performance of standard SCA. Two kinds of benchmark functions were employed: unimodal and multimodal. The random parameters of the conventional SCA were substituted with the chaotic variants to increase the convergence rate and abscond from the local solutions. Four chaotic CSCAs strategies were suggested. These algorithms were called CSCA-1 to 4, respectively.

Several statistical criteria were used like success rate at different levels of stopping conditions and average rank of mean solution. The final results for all these criteria for various benchmark functions demonstrated that CSCA-4 method with Circle map has a superior performance compared to other CSCAs methods. Additionally, the most effective chaotic map for the CSCA-1 was the Intermittency map while the chaotic map could boost the performance of the CSCA-2 and 3 was the Gauss map. The initial value 0.7 has been used as the default value for the proposed chaotic algorithms.

The comparison test was provided for evaluating the performance of the proposed algorithms with other metaheuristic optimization algorithms such as the standard SCA, GA, PSO and DE. This test was confirmed the superiority of the chaotic proposed algorithms on the others not only in terms of quality solutions but also took execution time less than other algorithms.

RECOMMENDATIONS

There are many important trends that can be worked out in the future. First, the CSCAs methods would be assigned to resolve practical engineering problems. Second, the CSCAs strategies could be combined with other state-of-art algorithms to introduce new hybridization algorithms. Last but not least, using other chaotic variants and study the performance of the proposed algorithms.

REFERENCES

Ang, X.S., 2012. Flower Pollination Algorithm for Global Optimization. In: Unconventional Computation and Natural Computation, Durand-Lose, J. and N. Jonoska (Eds.). Springer, Berlin, Germany, ISBN:978-3-642-32893-0, pp: 240-249.

Arora, S. and S. Singh, 2017. An improved butterfly optimization algorithm with chaos. J. Intell. Fuzzy Syst., 32: 1079-1088.

Askarzadeh, A., 2016. A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm. Comput. Struct., 169: 1-12.

Bingol, H. and B. Alatas, 2016. Chaotic league championship algorithms. Arabian J. Sci. Eng., 41: 5123-5147.

Bouchekara, H.R.E.H., 2017. Most valuable player algorithm: A novel optimization algorithm inspired from sport. Oper. Res., 1: 1-57.

Das, S., A. Bhattacharya and A.K. Chakraborty, 2017. Solution of short-term hydrothermal scheduling using sine cosine algorithm. Soft Comput., 1: 1-19.

Dorigo, M., V. Maniezzo and A. Colorni, 1996. Ant system: Optimization by a colony of cooperating agents. IEEE Trans. Syst. Man Cybern. Part B: Cybern., 26: 29-41.

Elfattah, M.A., S. Abuelenin, A.E. Hassanien and J.S. Pan, 2016. Handwritten Arabic manuscript image Binarization using sine cosine optimization algorithm. Proceedings of the 10th International Conference on Genetic and Evolutionary Computing, November 7-9, 2016, Springer, Fuzhou, China, ISBN:978-3-319-48489-1, pp: 273-280.

Eskandar, H., A. Sadollah, A. Bahreininejad and M. Hamdi, 2012. Water cycle algorithm-A novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput. Struct., 110: 151-166.

Ewees, A.A., M.A.E. Aziz and A.E. Hassanien, 2017. Chaotic multi-verse optimizer-based feature selection. Neural Comput. Appl., 1: 1-16.

Feng, J., J. Zhang, X. Zhu and W. Lian, 2017. A novel chaos optimization algorithm. Multimedia Tools Appl., 76: 17405-17436.

Gandomi, A.H. and A.H. Alavi, 2012. Krill herd: A new bio-inspired optimization algorithm. Commun. Nonlinear Sci. Numer. Simul., 17: 4831-4845.

Gandomi, A.H. and X.S. Yang, 2014. Chaotic bat algorithm. J. Comput. Sci., 5: 224-232.

Gandomi, A.H., G.J. Yun, X.S. Yang and S. Talatahari, 2013b. Chaos-enhanced accelerated particle swarm optimization. Commun. Nonlinear Sci. Numer. Simul., 18: 327-340.

Gandomi, A.H., X.S. Yang, S. Talatahari and A.H. Alavi, 2013a. Firefly algorithm with chaos. Commun. Nonlinear Sci. Numer. Simul., 18: 89-98.

Geem, Z.W., J.H. Kim and G.V. Loganathan, 2001. A new heuristic optimization algorithm: Harmony search. Simulation, 76: 60-68.

Hafez, A.I., H.M. Zawbaa, E. Emary and A.E. Hassanien, 2016. Sine cosine optimization algorithm for feature selection. Proceedings of the 2016 International Symposium on Innovations in Intelligent Systems and Applications (INISTA'16), August 2-5, 2016, IEEE, Sinaia, Romania, ISBN:978-1-4673-9911-1, pp: 1-5.

He, D., C. He, L.G. Jiang, H.W. Zhu and G.R. Hu, 2001. Chaotic characteristics of a one-dimensional iterative map with infinite collapses. IEEE. Trans. Circuits Syst. I Fundam. Theor. Appl., 48: 900-906.

He, X., J. Huang, Y. Rao and L. Gao, 2016. Chaotic teaching-learning-based optimization with Levy flight for global numerical optimization. Comput. Intell. Neurosci., 2016: 43-43.

Heidari, A.A., R.A. Abbaspour and A.R. Jordehi, 2017. An efficient chaotic water cycle algorithm for optimization tasks. Neural Comput. Appl., 28: 57-85.

Hilborn, R.C., 2000. Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers. 2nd Edn., Oxford University Press, Oxford, England, UK., Pages: 651.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence. University of Michigan Press, Ann Arbor, Michigan, USA., ISBN:9780472084609, Pages: 183.

Jaddi, N.S., J. Alvankarian and S. Abdullah, 2017. Kidney-inspired algorithm for optimization problems. Commun. Nonlinear Sci. Numer. Simul., 42: 358-369.

Jamil, M. and X.S. Yang, 2013. A literature survey of benchmark functions for global optimisation problems. Intl. J. Math. Modell. Numer. Optimisation, 4: 150-194.

Jordehi, A.R., 2015a. Chaotic Bat Swarm Optimisation (CBSO). Appl. Soft Comput., 26: 523-530.

Jordehi, A.R., 2015b. Seeker optimisation (human group optimisation) algorithm with chaos. J. Exp. Theor. Artif. Intell., 27: 753-762.

Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. Proc. IEEE Int. Conf. Neural Networks, 4: 1942-1948.

Kumar, N., I. Hussain, B. Singh and B.K. Panigrahi, 2017. Single sensor-based MPPT of partially shaded PV system for battery charging by using Cauchy and Gaussian sine cosine optimization. IEEE. Trans. Energy Convers., 32: 983-992.

Liao, G.C. and T.P. Tsao, 2006. Application of a fuzzy neural network combined with a chaos genetic algorithm and simulated annealing to short-term load forecasting. IEEE Trans. Evolut. Comput., 10: 330-340.

Mahdad, B. and K. Srairi, 2018. A new interactive sine cosine algorithm for loading margin stability improvement under contingency. Electr. Eng., 100: 913-933.

Mirjalili, S. and A. Lewis, 2016. The whale optimization algorithm. Adv. Eng. Software, 95: 51-67.

Mirjalili, S. and A.H. Gandomi, 2017. Chaotic gravitational constants for the gravitational search algorithm. Appl. Soft Comput., 53: 407-419.

Mirjalili, S., 2016b. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete and multi-objective problems. Neural Comput. Appl., 27: 1053-1073.

Mirjalili, S., 2016a. SCA: A sine cosine algorithm for solving optimization problems. Knowl. Based Syst., 96: 120-133.

Mirjalili, S., A.H. Gandomi, S.Z. Mirjalili, S. Saremi and H. Faris *et al*., 2017. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. Adv. Eng. Software, 114: 163-191.

Mirjalili, S., S.M. Mirjalili and A. Hatamlou, 2016. Multi-verse optimizer: A nature-inspired algorithm for global optimization. Neural Comput. Appl., 27: 495-513.

Mirjalili, S., S.M. Mirjalili and A. Lewis, 2014. Grey wolf optimizer. Adv. Eng. Software, 69: 46-61.

Mitic, M., N. Vukovic, M. Petrovic and Z. Miljkovic, 2015. Chaotic fruit fly optimization algorithm. Knowl. Based Syst., 89: 446-458.

Mousavirad, S.J. and H. Ebrahimpour-Komleh, 2017. Human mental search: A new population-based metaheuristic optimization algorithm. Appl. Intell., 47: 850-887.

Ott, E., 2002. Chaos in Dynamical Systems. 2nd Edn., Cambridge University Press, Cambridge, England, UK., Pages: 218.

Rao, R., 2016. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Intl. J. Ind. Eng. Computations, 7: 19-34.

Rao, R.V., V.J. Savsani and D.P. Vakharia, 2011. Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. Comput. Aided Des., 43: 303-315.

Rashedi, E., H. Nezamabadi-Pour and S. Saryazdi, 2009. GSA: A gravitational search algorithm. Inform. Sci., 179: 2232-2248.

Saha, S. and V. Mukherjee, 2018. A novel chaos-integrated symbiotic organisms search algorithm for global optimization. Soft Comput., 22: 3797-3816.

Saremi, S., S. Mirjalili and A. Lewis, 2014. Biogeography-based optimisation with chaos. Neural Comput. Appl., 25: 1077-1097.

Saremi, S., S. Mirjalili and A. Lewis, 2017. Grasshopper optimisation algorithm: Theory and application. Adv. Eng. Software, 105: 30-47.

Sayed, G.I., A.E. Hassanien and A.T. Azar, 2017. Feature selection via a novel chaotic crow search algorithm. Neural Comput. Appl., 1: 1-18.

Simon, D., 2008. Biogeography-based optimization. IEEE Trans. Evol. Comput., 12: 702-713.

Sindhu, R., R. Ngadiran, Y.M. Yacob, N.A.H. Zahri and M. Hariharan, 2017. Sine-cosine algorithm for feature selection with elitism strategy and new updating mechanism. Neural Comput. Appl., 28: 2947-2958.

Storn, R. and K. Price, 1997. Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim., 11: 341-359.

Talatahari, S., B.F. Azar, R. Sheikholeslami and A.H. Gandomi, 2012. Imperialist competitive algorithm combined with chaos for global optimization. Commun. Nonlinear Sci. Numer. Simul., 17: 1312-1319.

Tavazoei, M.S. and M. Haeri, 2007. Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. Appl. Math. Comput., 187: 1076-1085.

Tharwat, A. and A.E. Hassanien, 2018. Chaotic antlion algorithm for parameter optimization of support vector machine. Appl. Intell., 48: 670-686.

Turgut, O.E., 2017. Thermal and economical optimization of a shell and tube evaporator using hybrid backtracking search-sine cosine algorithm. Arabian J. Sci. Eng., 42: 2105-2123.

Wang, G.G., L. Guo, A.H. Gandomi, G.S. Hao and H. Wang, 2014. Chaotic krill herd algorithm. Inf. Sci., 274: 17-34.

Wang, L. and Y. Zhong, 2015. Cuckoo search algorithm with chaotic maps. Math. Prob. Eng., 2015: 1-14.

Wolpert, D.H. and W.G. Macready, 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput., 1: 67-82.

Yang, X.S. and S. Deb, 2009. Cuckoo search via Levy flights. Proceedings of the World Congress on Nature and Biologically Inspired Computing (NaBIC), December 9-11, 2009, IEEE, Coimbatore, India, ISBN:978-1-4244-5053-4, pp: 210-214.

Yang, X.S., 2010. A New Metaheuristic Bat-Inspired Algorithm. In: Nature Inspired Cooperative Strategies for Optimization (NICSO 2010), Gonzalez, J.R., D.A. Pelta, C. Cruz, G. Terrazas and N. Krasnogor (Eds.). Springer, Berlin, Germany, ISBN:978-3-642-12537-9, pp: 65-74.

Yang, X.S., 2014. Nature-Inspired Optimization Algorithms. Elsevier, New York, USA., ISBN:9780124167452, Pages: 300.

Yi, J., X. Li, C.H. Chu and L. Gao, 2016. Parallel chaotic local search enhanced harmony search algorithm for engineering design optimization. J. Intell. Manuf., 2016: 1-24.

Zhang, X. and T. Feng, 2018. Chaotic bean optimization algorithm. Soft Comput., 22: 67-77.

Zheng, Y.J., 2015. Water wave optimization: A new nature-inspired metaheuristic. Comput. Oper. Res., 55: 1-11.