Sigma J Eng & Nat Sci 37 (4), 2019, 1107-1128

Sigma Journal of Engineering and Natural Sciences
Sigma Mühendislik ve Fen Bilimleri Dergisi

Research Article

DATA CLUSTERING BASED ON FUZZY C-MEANS AND CHAOTIC WHALE OPTIMIZATION ALGORITHMS

Hatice ARSLAN*[1], Metin TOZ[2]

[1]*Computer Engineering Department of Institute of Science, Yildiz Technical University, ISTANBUL;* ORCID: 0000-0002-6166-8106
[2]*Computer Engineering Dept. of Technology Faculty, Duzce University;* ORCID: 0000-0001-9752-2718

Received: 27.07.2019 Accepted: 30.09.2019

ABSTRACT

Clustering is the process of sub-grouping data according to certain distance and similarity criteria. One of the most commonly used clustering algorithms in the literature is the Fuzzy C-Means (FCM) algorithm based on the fuzzy clustering principle. Although FCM is an efficient algorithm, random selection of initial cluster centers is a disadvantage since it easier trap the algorithm into local optimum. This problem can be solved by approaching the clustering problem as an optimization problem. In this article, Whale Optimization Algorithm (WOA), a global optimization algorithm developed by inspiration from hunting behaviors of humpback whales, has been improved with chaos maps using an adaptive normalization method and chaotic WOA algorithms are proposed. They are then hybridized with FCM algorithm. The performances of the proposed chaotic optimization algorithms are tested with thirteen different benchmark functions. Results are evaluated with means and standard deviations of the objective function values and with the Wilcoxon Sign Rank Test at 0.05 significance level. The clustering performances of the proposed hybrid algorithms measured according to the objective function, the Rand Index and the Adjusted Rand Index values and compared with the K-Means, FCM and some of the other hybrid algorithms for six different data sets selected from the UCI Repository database. In addition, the new hybrid clustering algorithms are improved by using Chebyshev distance function instead of the classical Euclidean distance for the FCM algorithm in order to increase their data clustering performances. As a result, it has been seen that the used chaos functions improve the optimization performance of WOA algorithm, integrating chaotic WOA algorithms with FCM algorithm enhances the disadvantages of FCM algorithm and changing the distance function increases clustering performance of the proposed algorithms.
Keywords: Data clustering, WOA, FCM, optimization, chaos.

1. INTRODUCTION

Population based meta-heuristic algorithms have been developed with inspiration from natural phenomena. These algorithms are often preferred since they do not require gradient information, exceeds to local optimum, are easy to implement, and can be used in many interdisciplinary fields [1]. These algorithms converge to an optimal solution rather than an exact solution. According to the NFL theorem [2], there is no algorithm that best solves all optimization problems. In other

* Corresponding Author: e-mail: haticearslan8154@gmail.com

words, a meta-heuristic algorithm may perform well for some problems while doing poorly for others. These algorithms are generally grouped in three different ways, physics-based, evolution-based, and swarm-based. Physics-based algorithms are developed based on natural physics rules. Simulated Annealing (SA) [3] which mimics the physical annealing process of the solids, Gravitational Search Algorithm (GSA) [4] using Newton's gravity and motion laws, Big Bang Big Crunch Algorithm (BB-BC) [5] inspired by the big bang theory, Gravity Local Search (GLS) [6], Black Hole Optimization (BH) [7] and Beam Algorithm (BA) [8] are examples of physics-based algorithms. The source of inspiration for evolutionary algorithms is Darwin's theory of evolution. The process that begins with the creation of a random population continues with the survival and proliferation of the best and most compatible individual. Genetic Algorithm (GA) [9], Genetic Programming (GP) [10], Evolution Strategy (ES) [11], Probability Based Incremental Learning (PBIL) [12] and Biology Based Optimization (BTO) [13] can be given as examples of evolution-based algorithms. Swarm-based algorithms have been enhanced with inspiration from behaviors of solving the problems encountered by living creatures acting collectively and behaviors of benefit from each other's experiences to solve a probable problem. Particle Swarm Optimization (PSO) [14] which models bird behaviors in order of food searches, Ant Colony Algorithm (ACA) [15] developed by mathematical modeling of ant colony behaviors, Artificial Bee Colony Algorithm (ABC) [16] inspired by the behaviors of honey bees' food search, Grey Wolf Optimizer (GWO) [17], Firefly Algorithm (FA) [18], The Ant Lion Optimizer (ALO) [19] and Sine Cosine Algorithm (SCA) [20] can be given as examples to these algorithms.

Population based algorithms consist of two parts; exploration of the search field (exploration phase) and use of the best result found (exploitation phase). During the exploration phase, the selected parameters must be as random as possible for better scanning of the search field [1]. The collapsed search region in the exploration phase is tested with the exploitation phase. That is, the optimum point in the exploration phase is used during the exploitation operation and is approached to optimum throughout the iteration. Thus, provide a good balance between exploration and exploitation phases is important for the performance of the algorithm [21]. However, due to the probabilistic behavior of population-based algorithms this balance is not easy to achieve [22]. When the literature is examined, it is seen that the integration of population-based algorithms with chaos theory increases the performance of both the exploration and exploitation phase. Zhang et al. in 2009, applied two chaotic maps to the PSO and the performance of the algorithm was improved [23]. In 2009, Wang and Yao proposed a Hybrid Genetic Algorithm based on chaos and PSO to improve the convergence and inadequate run-time performance of the genetic algorithm [24]. Atalas and his colleagues have improved the performance of the PSO by applying 12 chaotic maps to the PSO in 2009 [25]. They have also shown that the performance of the ABC [26] and Harmony Search (HS) algorithms [27] can be improved by chaos. In the work performed by Yan H. et al. in 2014, chaos has been used to improve the exploitation phase of the genetic algorithm and to increase the accuracy [28]. In addition, meta-heuristic algorithms also used in conjunction with chaos are GA [29], FA [30], SA [31], Differential Evolution (DE) [32] and Krill Herd Algorithm [33]. The examples given support the increase in performance when meta-heuristic optimization algorithms are used together with chaos. In this paper, Whale Optimization Algorithm (WOA) [1], developed by Mirjalili and Lewis based on the hunting behaviors of whales, is used together with chaotic maps to improve the performance of the algorithm. There are studies in the literature where the WOA algorithm was used with chaos functions. In the study done by Tanyıldızı and Cigal, the Logistic map was added to the WOA algorithm and WOA algorithms based on chaos were proposed [34]. Sun and Wang tried to solve the problem of trapped to local optimum by using the WOA algorithm to optimize the Elman neural network. Besides, a chaotic WOA algorithm was proposed to improve the diversity and eccentricity of search agents [35]. Oliva and colleagues applied four different chaos maps to the WOA algorithm for parameter estimation of solar batteries [36]. In this article, unlike other studies, 10 different chaotic maps are applied to the WOA after being passed through the normalization process proposed in [22]. The efficiency of the proposed algorithms is tested using 13 benchmark functions. Scientific significance of the results is measured by the Wilcoxon Sign Rank Test. In addition, a new approach has been improved to the solution of the data clustering problem by means of the proposed chaotic algorithms.

Clustering is the process of dividing a data set into different subsets where similar data are found in the same cluster. It is indicative of a good clustering being intra-cluster similarity is maximum and inter-cluster similarity is minimum [37]. Clustering is used in scientific and engineering applications such as image recognition, data mining, machine learning, signal processing and biology [38]. In the literature there are many clustering algorithms proposed for solving clustering problems. One of these algorithms is the fuzzy clustering based Fuzzy C-Means (FCM) algorithm proposed by Dunn [39] and developed by Bezdek [40]. In this algorithm, the data belongs to a cluster with certain membership grades. Therefore, one element in database can belong to more than one cluster at the same time. Although the FCM is an efficient algorithm, random selection of the initial cluster centers creates a disadvantage by making it easier to trap the algorithm to the local optimum. Clustering problem can also be considered as a kind of optimization problem. In recent years, meta-heuristic algorithms have begun to be widely used to solve such clustering problems [41]. Such algorithms look for an optimal solution for clustering problems and reduce the risk of trapping to the local optimum [38]. For this reason, the FCM algorithm is also combined with many meta-heuristic algorithms. According to the literature, the FCM is integrated with the meta-heuristic algorithms such as GA [42], DE [43], Ant Colony Optimization [44], PSO [45], Artificial Fish Swarm Optimization [46], fuzzy PSO [47], Support Vector Machines [48]. In this paper, FCM and chaotic WOA algorithms are combined and new hybrid clustering algorithms are developed. The proposed algorithms are based on optimizing the cluster centers with the chaotic WOA algorithms. For each cluster center, the FCM-CWOA algorithms updates the cluster centers while trying to minimize the objective function of the FCM algorithm. In addition, to improve performance of the proposed clustering algorithms, Euclid distance function of the FCM algorithm is replaced by the Chebyshev distance function. FCM-CWOA algorithms, the classical FCM algorithms and other optimization based hybrid algorithms are tested with six datasets selected from the UCI database [49]. The effect of changing the distance function of the FCM algorithm and of the normalization of chaos maps on the data clustering are evaluated by proposed algorithms (FCWOA-c and FCMWOA* algorithms). The obtained results are compared with the Rand Index and Adjusted Rand Index values and according to these indexes it is seen that the proposed clustering algorithms gives better results than the compared algorithm. As a result, it is observed that using WOA algorithm with normalized chaos maps increases the performance of the algorithm, integrating chaotic WOA algorithms with FCM algorithm improves disadvantages of FCM algorithm and changing distance function increases clustering performance of algorithms.

In the second part, WOA algorithm is explained in details; in the third part chaos maps and application methods are given. In the fourth part, the problem of data clustering is identified. Finally, in the fifth section, the study is briefly summarized and evaluated.

2. WHALE OPTIMIZATION ALGORITHM (WOA)

The whale optimization algorithm (WOA) is a global optimization algorithm developed by Mirjalili and Lewis [1], inspired by the hunting strategies of humpback whales. Humpback whales have a unique hunting behavior. They dive about 12 meters down in the water and form spiral-shaped bubbles around their prey, trapping their prey in air bubbles. Then, they swim to the surface to swallow their prey. These unique hunting behaviors of humpback whales are illustrated in figure 1.

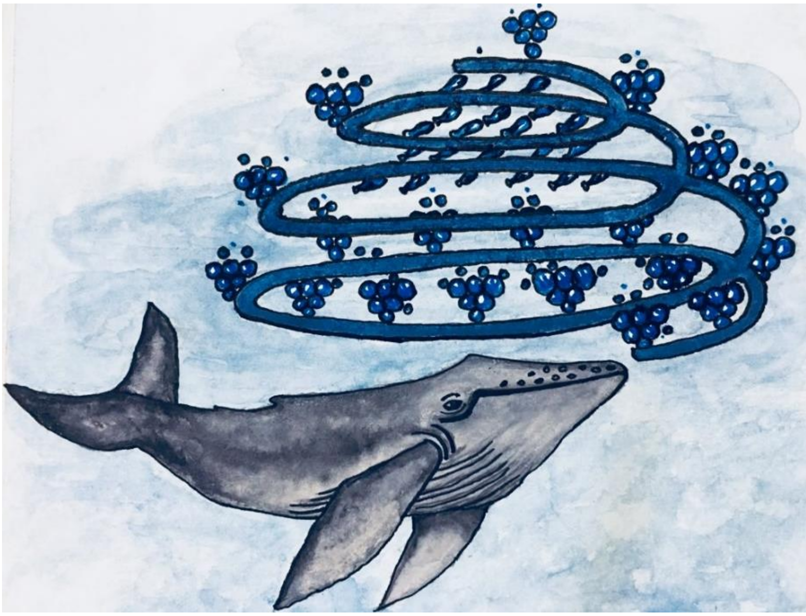


Figure 1. Hunting behavior of humpback whale

The mathematical model of the WOA algorithm consists of three basic steps; spinning, air bubble attack, and hunting. The algorithm assumes that the target hunt is the closest candidate solution to the optimal hunt model. Each humpback whale is considered a search agent. After the best search agent is identified according to the target prey, other search agents update their location accordingly. The mathematical model of this behavior is defined as follows [1].

$$\vec{D} = \left|\vec{C}.\overrightarrow{X^*}(t) - \vec{X}(t)\right| \tag{1}$$

$$\vec{X}(t+1) = \overrightarrow{X^*}(t) - \vec{A}.\vec{D} \tag{2}$$

where, t is current iteration number, $\vec{A}$ and $\vec{C}$ are two coefficient vectors, $\overrightarrow{X^*}$ is the best solution of position vector obtained so far, $\vec{X}$ is the position vector and $|\ |$ and $\cdot$ means absolute value and elementary multiplication, respectively. If there is a better solution $\overrightarrow{X^*}$ should be updated in every iteration. The vectors $\vec{A}$ and $\vec{C}$ are calculated as follows [1]:

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \tag{3}$$

$$\vec{C} = 2\vec{r} \tag{4}$$

where, $\vec{a}$ is a linearly decreasing number from 2 to 0 throughout the iterations (both during the exploration and exploitation phases) and $\vec{r}$ is a random vector in the range [0,1].

The search agent (X, Y) can update the location according to the best available (X^*, Y^*) location and by chancing the values of $\vec{A}$ and $\vec{C}$, it can reach to different places near the best search agent. By randomly defining the vector $\vec{r}$, it is possible to reach any position in the search space located between the lock points. Equation 2 allows the search agent to update its position in the neighborhood of the best solution available and to model encircling the prey. The modeling of the bubble-net attacking method of humpback whales involves two approaches [1].

Shrinking encircling mechanism represents the reduction of the circle around the prey by updating the value of $\vec{a}$ in the equation 3 with the following equation [1].

$$a = 2 - t\frac{2}{Maxiter} \tag{5}$$

Thus, $\vec{A}$ takes a random value in the interval [-a, a] with decreasing a from 2 to 0 during the iteration. The new position of a search agent can be defined anywhere between the starting position of the agent and the position of the best available agent, if we assign random values in the range [-1,1] for A [1].

Spiral updating computes the distance between the whale located at (X, Y) and the prey located at (X^*, Y^*). To model the helical movements of humpback whales, the following equation is established between the whale and the prey positions [1].

$$\vec{X}(t+1) = \overrightarrow{D'}.e^{bl}\cos(2\pi l) + \overrightarrow{X^*}(t) \tag{6}$$

where $\overrightarrow{D'} = \left|\overrightarrow{X^*}(t) - \vec{X}(t)\right|$ and shows the distance of the *i*th whale to prey. b is a constant that defines the logarithmic spiral shape; *l* is a random number in the range [-1,1], and · is elementary product.

The humpback whales swim creating shrinking spirals around their prey, simultaneously. To simulate this synchronous behavior, it is assumed that during optimization, the location of the whales has been updated with a probability of 50% among shrinking encircling mechanism and the spiral updating. It is mathematically expressed by the following equation [1].

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A}.\vec{D} & if\ p < 0.5 \\ \overrightarrow{D'}.e^{bl}\cos(2\pi l) + \overrightarrow{X^*}(t) & if\ p \geq 0.5 \end{cases} \tag{7}$$

where p is a random number between [-1,1].

Search for prey (exploration phase) imitates humpback whales doing random research according to each other's position. To make the search more comprehensive and to keep the whales away from each other, $\vec{A}$ is selected randomly as greater than 1 and less than -1. In contrast to the exploitation phase, during the exploration phase, the position of a search agent is updated with a randomly selected search agent. Selecting $|\vec{A}| > 1$ allows the WOA algorithm to conduct a global search. The mathematical model is as follows [1]:

$$\vec{D} = \left|\vec{C}.\overrightarrow{X_{rand}} - \vec{X}\right| \tag{8}$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A}.\vec{D} \tag{9}$$

where $\overrightarrow{X_{rand}}$ is a random position vector selected from the current population.

To summarize, the WOA algorithm begins with a series of random solutions. In each iteration, search agents update their positions to 50% probability, either randomly selected search agents, or the best solution so far. Depending on the value of p, WOA can choose between spiral or circular motion. Finally, the WOA is terminated by provide of the stopping criterion.

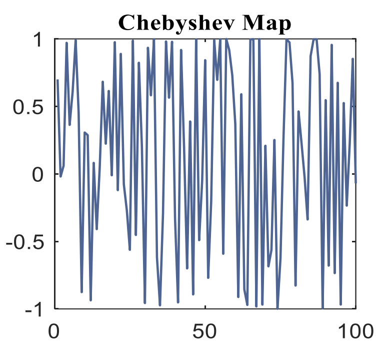
3. CHAOTIC MAPS

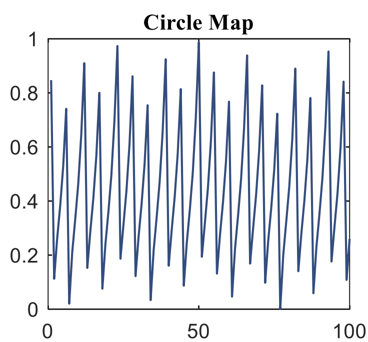
Chaos is defined as the randomness produced by mathematically simple deterministic systems [25]. It can also be expressed as an arrangement within the irregularity that focuses on the behavior of dynamic systems that are highly sensitive to their initial values. That is, small changes in the initial conditions can result in large differences (sensitivity). Chaos has similar scatter performance for a random value (randomness). It also consists of values that do not repeat within a certain interval (ergodicity) [26], [50]. Therefore, using chaotic variables instead of random variables in optimization algorithms reduces the likelihood of repeating randomly selected numbers and accumulating at a certain interval. Thus, the problem of trapping to the local optimum of the optimization problems can be solved [50]. The chaotic maps used in this study are shown in Table 1 [22], [50].

Table 1. Equations of and range of chaotic maps

Chaotic Maps	Equation	Interval
Chebyshev	$x_{i+1} = \cos(i \cos^{-1}(x_i))$	(-1,1)
Circle	$x_{i+1} = mod\left(x_i + b - \left(\frac{a}{2\pi}\right)\sin 2\pi x_i, 1\right), a = 0.5\ b = 0.2$	(0,1)
Gauss/mouse	$x_{i+1} \begin{cases} x_i = 0 \\ \frac{1}{mod(x_i, 1)} & diğer \end{cases}$	(0,1)
Iterative	$x_{i+1} = \sin\frac{a\pi}{x_i} \quad a = 0.7$	(-1,1)
Logistic	$x_{i+1} = ax_i(1 - x_i) \quad a = 4$	(0,1)
Piecewise	$x_{i+1} = \begin{cases} \frac{x_i}{d} & 0 \leq x_i < d \\ \frac{x_i - d}{0.5 - d} & d \leq x_i < \frac{1}{2} \\ \frac{1 - d - x_i}{0.5 - d} & \frac{1}{2} \leq x_i < 1 - d \\ \frac{1 - x_i}{d} & 1 - d \leq x_i < 1 \end{cases}$	(0,1)
Sine	$x_{i+1} = \frac{a}{4}\sin(\pi x_i) \quad a = 4$	(0,1)
Singer	$x_{i+1} = \mu(7.86x_i - 23.31x_i^2 + 28.75x_i^3 - 13.302875x_i^4), \ \mu = 2.3$	(0,1)
Sinusoidal	$x_{i+1} = ax_i^2 \sin(\pi x_i), \quad a = 2.3$	(0,1)
Tent	$x_{i+1} = \begin{cases} \frac{x_i}{0.7} x_i < 0.7 \\ \frac{10}{3}(1 - x_i)x_i \geq 0.7 \end{cases}$	(0,1)

The graphs of the chaotic maps are given in figure 2.





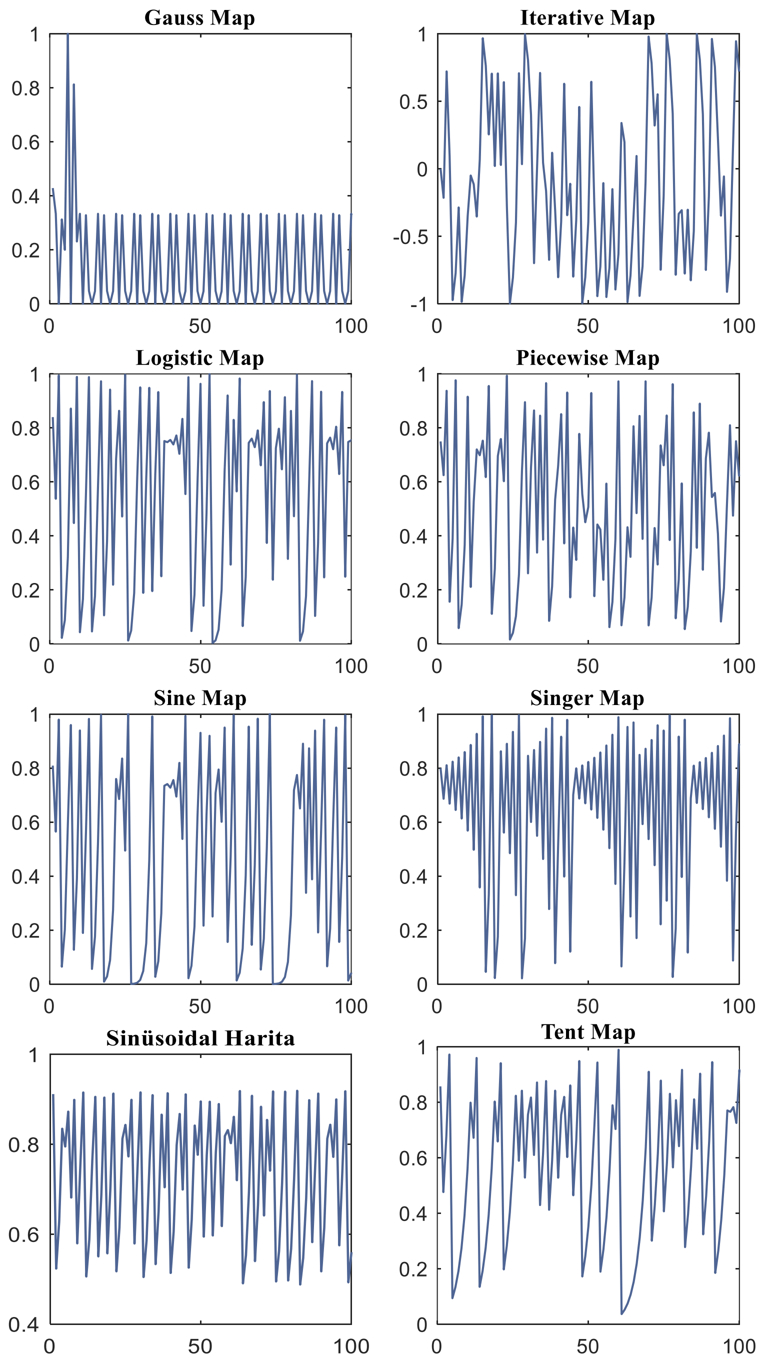


Figure 2. Graphs of Chaotic Maps

3.1. Application of Chaotic Maps to the WOA Algorithm

In the WOA algorithm, hunting strategy of humpback whales is modeled. The whales create air bubbles around their prey and surround them and prevent them from escaping. This behavior is modeled by the Equation 2. By updating the value a in the Equation 3, it is possible to narrow the circle around the prey. In the WOA algorithm, a value is chosen from randomly generated numbers. In this work, a value has been replaced by chaotic maps to provide both decreasing linearly in the range [0,2] and randomness obtained by chaos maps. In the WOA algorithm, a value decreases linearly in the range [0,2]. Accordingly, each of the chaotic maps has been normalized to a predefined range $[k, l]$, this range represents the added chaos effect to the a value. Normalized chaos maps have been added to the current a value. The normalization process is performed using the following formulas as done in [22].

$$x(t)_n = \frac{(x(t)-m)\times(l-k)}{(m-n)} + c \tag{10}$$

Where $x(t)$ and $x(t)_n$ are non-normalized and normalized values of the chaotic map at t'th iteration, $[m, n]$ is the interval of the chaos map given in Table 1, and $[k, l] = [0\ 0.05]$ is the normalization interval used in this study. d value is reduced throughout iterations by the following formula [22].

$$d(t) = d - \frac{t}{T}(l - k) \tag{11}$$

In this Equation, t represents the current iteration and T represents the maximum number of iterations. Chaotic maps are normalized and then combined with a value. This process is shown in Figure 3-4 for Chebyshev Map.

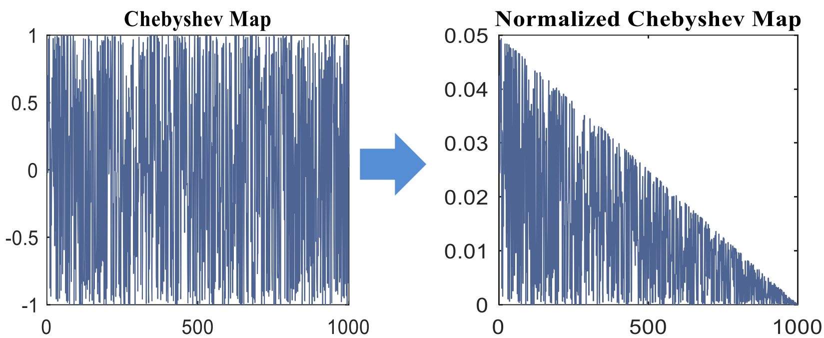


Figure 3. Normalization graph of Chebyshev Map

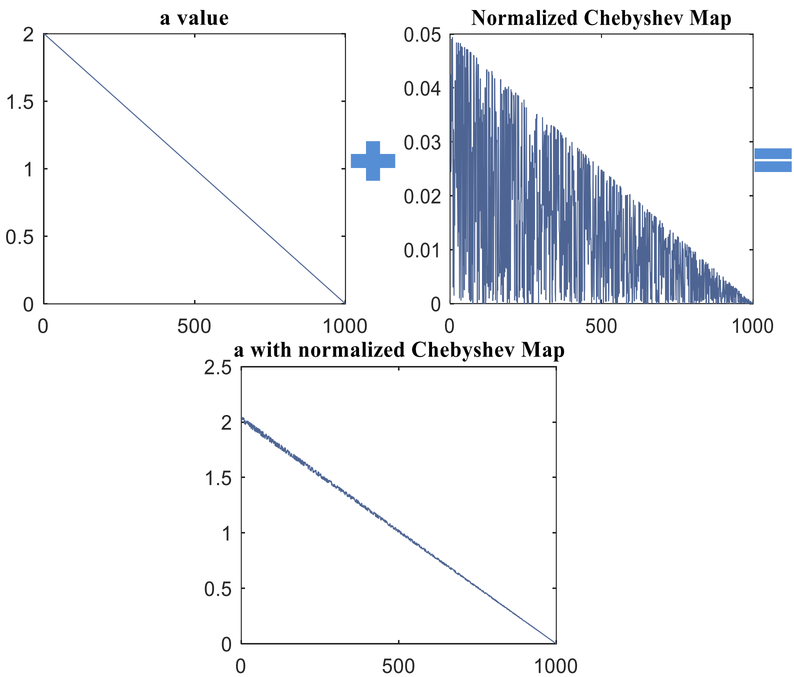
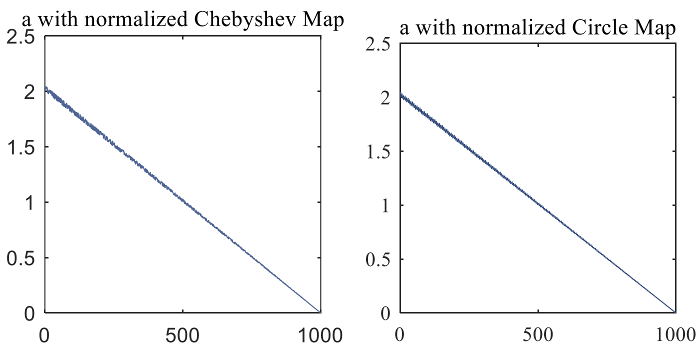


Figure 4. Sum of normalized Chebyshev map with a value

In Figure 5 all the normalized chaos maps with a value is shown.



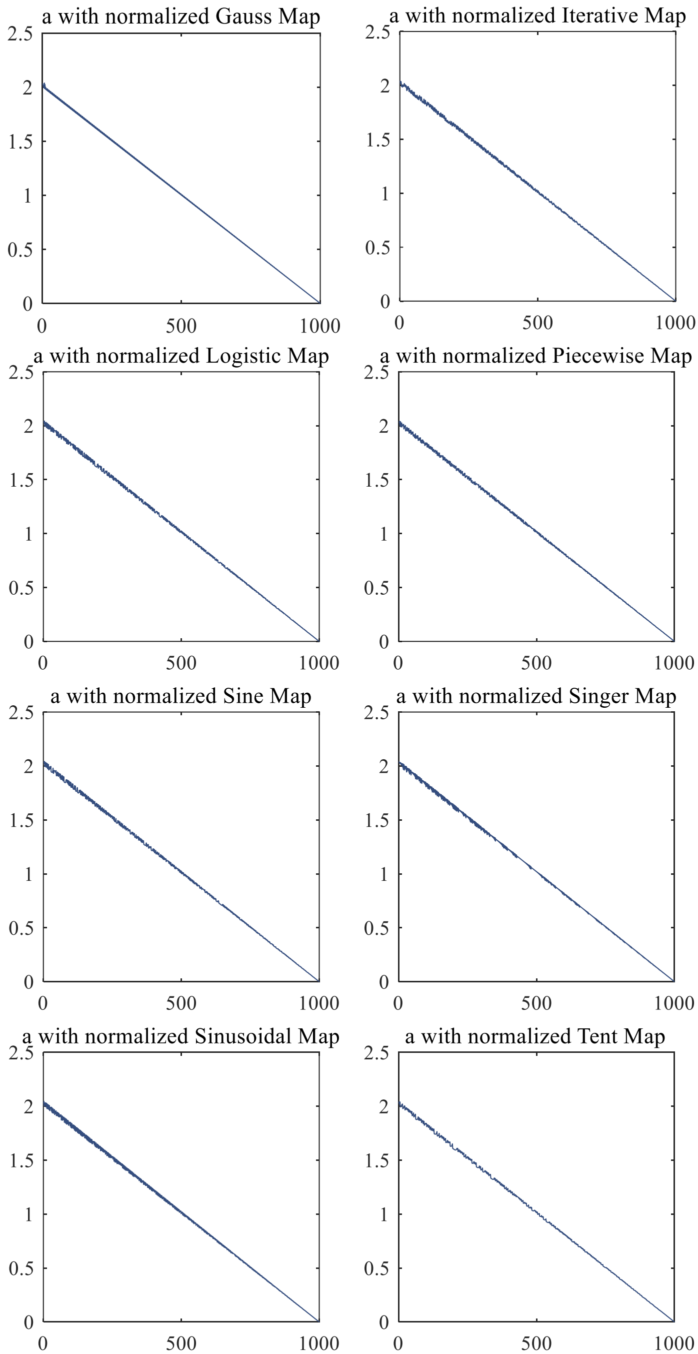


Figure 5. Normalized chaos maps with a values

4. EXPERIMENTAL STUDIES

The performance of chaos-based WOA algorithms developed in this paper has been tested with 13 different benchmark functions that are frequently used in optimization problems [1]. These functions are composed from both single-mode functions (F1-F7) and multi-mode (F8-F13) functions [1]. The equations of these functions are given in Table 2.

Table 2. Equations of benchmark functions

benchmark functions	Dimension	range	f_{min}
$F_1(x)=\sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$F_2(x)=\sum_{i=1}^{n} \lvert x_i \rvert+\prod_{i=1}^{n} \lvert x_i \rvert$	30	[-10,10]	0
$F_3(x)=\sum_{i=1}^{n}\left(\sum_{j-1}^{i} x_j\right)^2$	30	[-100,100]	0
$F_4(x)=max_i\{\lvert x_i \rvert, 1\leq i\leq n\}$	30	[-100,100]	0
$F_5(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	[-30,30]	0
$F_6(x)=\sum_{i=1}^{n}(\lvert x_i+0.5 \rvert)^2$	30	[-100,100]	0
$F_7(x)=\sum_{i=1}^{n} ix_i^4+random(0,1)$	30	[-1.28,1.28]	0
$F_8(x)=\sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	30	[-500,500]	-418.982x5
$F_9(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i)+10]$	30	[-5.12,5.12]	0
$F_{10}(x)=-20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)-exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)+20+e$	30	[-32,32]	0
$F_{11}(x)=\frac{1}{4000}\sum_{i=1}^{n} x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[-600,600]	0
$F_{12}(x)=\frac{\pi}{n}\{10\sin(\pi y_1)+\sum_{i=1}^{n}(y_i-1)^2[1+10sin^2(\pi y_{i+1})]+(y_n-1)^2\}+\sum_{i=1}^{n} u(x_i,10,100,4)$	30	[-50,50]	0

$y_i = 1 + \frac{x_i+1}{4} u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$			
$F_{13}(x) = 0.1\left\{sin^2(\beta\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[-50,50]	0

The WOA algorithm and the proposed CWOA algorithms have been run 30 times in succession. Besides, maximum number of iteration was defined as 1000 and population size was 50. In order to evaluate the mutual performances of the algorithms, mean and standard deviation values were calculated for each run. The best results are indicated in bold type. In addition, nonparametric Wilcoxon Sign Rank Test [51] was calculated at the significance level of 0.05 in order to show the significant differences between the performance of the algorithms. According to Wilcoxon sign rank test used for statistical evaluation of the results, p-values that are less than 0.05 can be considered as strong evidences against the null hypothesis. The p value of less than 0.05 is underlined. All work was done by using MATLAB R2017b program on a computer with Intel Core i7-7700HQ CPU 2.80GHz processor and 16GB Ram in the same conditions. The obtained results of chaos-based WOA algorithms are given in Table 3.

The F1-F7 functions are single-mode functions since they have a single local optimum. These functions allow to evaluate the exploitation phase performance of meta-heuristic algorithms [1]. When Table 3 is examined, it is seen that chaos-based WOA algorithms for F1, F2, F3, F5, and F6 functions all yield better results than the WOA algorithm in terms of mean of the objective function values and their standard deviations. At the same time, when p values are examined, it is observed that these results are significantly different. Although the CWOA1, CWOA3, CWOA4, CWOA7, CWOA9 and CWOA10 for the F4 function give better results on average, the results are not significant when p values are considered. All chaos-based algorithms for F7 function are better in terms of average and standard deviation values, but there is a significant difference only for the CWOA4 algorithm. As a result, chaos-based WOA algorithms perform better for 5 out of the 7 functions, so the performance of the exploitation phase seems to be increased. The graphs of benchmark functions for two dimensions are illustrated in figure 6.

The functions F8-F13 are multimodal functions with more than one local optimum. For this reason, multimodal functions also allow us to evaluate the performance of the exploration phase [1]. When the results are examined, for F8 function in CWOA1 and CWOA8 algorithms, for F10 function in all algorithms except CWOA1, CWAO5 and CWOA7, for F9 function in all algorithms except CWOA1 and CWOA7, for F11 function in all algorithms except CWOA2 and CWOA6, in all algorithms for F12 function and for F13 function in CWOA1, CWOA5, CWOA6 and CWOA8 algorithms, no significant difference is found although better results are obtained than WOA in terms of standard deviation and mean values.

Table 3. Statistical results of chaos-based WOA algorithms

F1	Woa	CWoa1	CWoa2	CWoa3	CWoa4	CWoa5	CWoa6	CWoa7	CWoa8	CWoa9	CWoa10
Mean	9,82E-150	**1,20E-166**	**1,20E-166**	**1,20E-166**	**1,20E-166**	**1,20E-166**	**1,20E-166**	**1,20E-166**	**1,20E-166**	**1,20E-166**	**1,20E-166**
Std	5,37E-149	0	0	0	0	0	0	0	0	0	0
p value	1,73E-06	1,73E-06	1,73E-06	1,73E-06	1,73E-06	1,73E-06	1,73E-06	1,73E-06	1,73E-06	1,73E-06	1,73E-06
F2	Woa	CWoa1	CWoa2	CWoa3	CWoa4	CWoa5	CWoa6	CWoa7	CWoa8	CWoa9	CWoa10
Mean	4,95E-103	**7,41E-108**	**2,29E-107**	**1,21E-108**	**2,62E-108**	**1,37E+105**	**1,84E-106**	**4,64E-107**	**4,95E-106**	**7,80E+109**	**7,05E-107**
Std	2,66E-102	2,32E-107	1,07E+106	6,36E+108	5,97E-108	7,52E-105	6,33E-106	2,47E-106	1,81E-105	2,27E+108	2,81E-106
p value	0	0,0007	0,0017	0,0001	0,0014	0,0034	0,0316	0,0001	0,0082	0	0,001
F3	Woa	CWoa1	CWoa2	CWoa3	CWoa4	CWoa5	CWoa6	CWoa7	CWoa8	CWoa9	CWoa10
Mean	20412,19	**10500**	**9600**	**10600**	**18100**	**10000**	**12400**	**10700**	**9420**	**11900**	**10600**
Std	11982,64	7006,34	6881,43	6748,98	5173,25	5821,39	8403,73	8589,64	5665	7441,81	5622,94
p value	0	0,0012	0,0004	0,0006	0,0047	0,0001	0,0093	0,0032	0,001	0,0021	0,0003
F4	Woa	CWoa1	CWoa2	CWoa3	CWoa4	CWoa5	CWoa6	CWoa7	CWoa8	CWoa9	CWoa10
Mean	30,7143	**20,899**	40,4574	**22,2949**	**28,0525**	30,7529	35,6355	**25,058**	35,439	**30,538**	**27,3173**
Std	28,5513	24,5102	28,2303	24,8089	28,085	29,7233	28,8572	24,3903	30,019	27,9364	25,6309
p value	0	0,1714	0,165	0,3389	0,6288	0,9099	0,3493	0,4048	0,544	0,8451	0,8612
F5	Woa	CWoa1	CWoa2	CWoa3	CWoa4	CWoa5	CWoa6	CWoa7	CWoa8	CWoa9	CWoa10
Mean	27,2006	**26,6553**	**25,8317**	36,6487	**26,672**	**26,6498**	**26,5722**	**26,7133**	**26,6659**	**26,651**	**26,606**
Std	0,4039	0,4125	4,7743	0,3165	0,2537	0,2566	0,3168	0,478	0,5211	0,2545	0,3113
p value	0	0,0001	0,0001	0	0	0	0	0,0001	0,0005	0	0
F6	Woa	CWoa1	CWoa2	CWoa3	CWoa4	CWoa5	CWoa6	CWoa7	CWoa8	CWoa9	CWoa10
Mean	0,0935	**0,0045**	**0,0038**	**0,0054**	**0,0034**	**0,0033**	**0,0073**	**0,0031**	**0,0033**	**0,004**	**0,0044**
Std	0,1123	0,007	0,0015	0,0047	0,0014	0,0016	0,0199	0,0013	0,0015	0,0026	0,0026
p value	0	0	0	0	0	0	0	0	0	0	0
F7	Woa	CWoa1	CWoa2	CWoa3	CWoa4	CWoa5	CWoa6	CWoa7	CWoa8	CWoa9	CWoa10
Mean	0,0014	**0,0008**	**0,001**	**0,001**	**0,0008**	**0,0013**	**0,001**	**0,0009**	**0,0009**	**0,0017**	**0,0009**
Std	0,0016	0,001	0,001	0,001	0,0011	0,0013	0,0013	0,001	0,0008	0,0026	0,0008
p value	0	0,0687	0,102	0,2712	0,0111	0,7655	0,0598	0,0719	0,1986	0,7036	0,0545
F8	Woa	CWoa1	CWoa2	CWoa3	CWoa4	CWoa5	CWoa6	CWoa7	CWoa8	CWoa9	CWoa10
Mean	-11893,89	**-11970,86**	-11541,29	-11884,6	-11681,34	-11860,94	-11652,51	-11606,59	**-12091,77**	-11496,99	-11839,47
Std	1174,24	832,63	1424,41	1107,18	1244,34	1184,17	1176,79	1398,24	985,96	1506,79	1231,29
p value	0	0,3709	0,3185	0,8451	0,4048	0,813	0,165	0,4405	0,3185	0,3709	0,8612
F9	Woa	CWoa1	CWoa2	CWoa3	CWoa4	CWoa5	CWoa6	CWoa7	CWoa8	CWoa9	CWoa10
Mean	1,89E-15	5,68E-15	**0,00E+00**	**0,00E+00**	**0,00E+00**	**0,00E+00**	**0,00E+00**	5,68E-15	**0,00E+00**	**0,00E+00**	**0,00E+00**
Std	1,04E-14	2,29E-14	0,00E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	2,29E-14	0,00E+00	0,00E+00	0,00E+00
p value	1	0,75	1	1	1	1	1	0,75	1	1	1
F10	Woa	CWoa1	CWoa2	CWoa3	CWoa4	CWoa5	CWoa6	CWoa7	CWoa8	CWoa9	CWoa10
Mean	4,32E-15	4,80E-15	**3,85E-15**	**3,73E-15**	**3,73E-15**	4,32E-15	**3,61E-15**	4,44E-15	**3,49E-15**	**3,97E-15**	**3,97E-15**
Std	2,72E-15	2,16E-15	2,10E-15	2,36E-15	2,54E-15	2,55E-15	2,22E-15	2,47E-15	2,79E-15	2,42E-15	2,03E-15
p value	0	0,4986	0,4283	0,3173	0,375	1	0,2435	0,8332	0,1938	0,6076	0,5586
F11	Woa	CWoa1	CWoa2	CWoa3	CWoa4	CWoa5	CWoa6	CWoa7	CWoa8	CWoa9	CWoa10
Mean	0,0013	0,0019	**0,001**	0,0028	0,0019	0,0068	**0**	0,0019	0,0033	0,0059	0,0062
Std	0,0071	0,0107	0,0054	0,0106	0,0072	0,0184	0	0,0106	0,0103	0,0206	0,0164
p value	1	1	1	0,75	1	0,125	1	1	0,625	0,375	0,125
F12	Woa	CWoa1	CWoa2	CWoa3	CWoa4	CWoa5	CWoa6	CWoa7	CWoa8	CWoa9	CWoa10
Mean	0,0077	**0,001**	**0,0019**	**0,0031**	**0,0015**	**0,0022**	**0,0026**	**0,0016**	**0,0008**	**0,0008**	**0,0027**
Std	0,0345	0,0017	0,0035	0,0092	0,0036	0,0043	0,0058	0,0026	0,001	0,0016	0,0102
p value	0	0,1915	0,8612	0,6733	0,1109	0,6884	0,7655	0,813	0,3933	0,1359	0,1714
F13	Woa	CWoa1	CWoa2	CWoa3	CWoa4	CWoa5	CWoa6	CWoa7	CWoa8	CWoa9	CWoa10
Mean	0,0294	**0,0251**	0,045	0,0614	0,0332	**0,023**	**0,0272**	0,0334	**0,0276**	0,0393	0,0366
Std	0,0304	0,029	0,0424	0,0715	0,0545	0,0222	0,0327	0,0372	0,0267	0,0477	0,0726
p value	0	0,829	0,1254	0,0545	0,9754	0,544	0,36	0,7189	0,5716	0,4653	0,6435

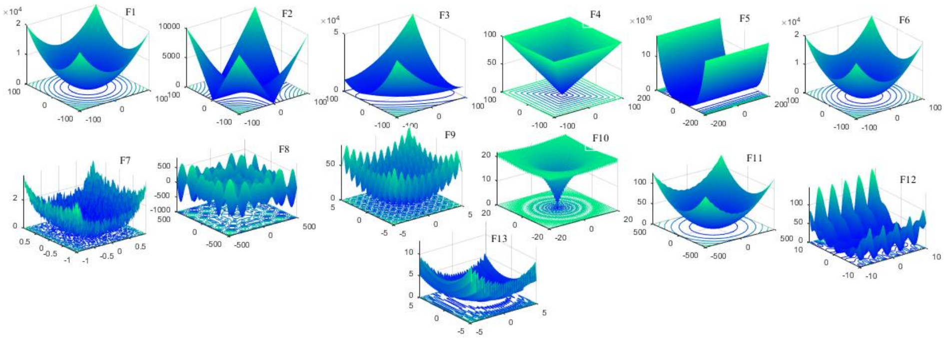


Figure 6. Graph of benchmark functions

Overall, when evaluated, the best performance is shown by the CWOA4 and CWOA8 algorithms, the performance of other algorithms is close to each other, and the performance of the exploitation phase is improved rather than the exploration phase.

4.1. Data Clustering Application

In this section, data clustering problem is solved by combining FCM and the proposed chaos based WOA algorithms. In addition to, the effect of changing the distance function of the FCM algorithm and of the normalization of chaos maps on the data clustering are evaluated.

Fuzzy C-Means Algorithm (FCM)

The FCM clustering divides a given set of n elements of $X = \{x_1, x_2, \dots, x_n\}$ data into c fuzzy sets [48]. A vector $v_i = [v_1, v_2, \dots, v_c]$, represents the *i*th cluster center. Each data sample has a membership degree represented by the membership matrix. The sum of the membership grades of all the clusters of a dataset should be 1. If data is closest to that cluster, then the membership level of the cluster will be larger. The membership matrix is represented as follows [48].

$$\sum_{i=1}^{c} U_{ij} = 1 \qquad j = 1,2, \dots n \tag{12}$$

The FCM algorithm is an objective function-based algorithm that tries to minimize the following objective function, which is the generalization of the least squares method [48].

$$J_m(U,V) = \sum_{i=1}^{n} \sum_{j=1}^{c} U_{ij}^m \left\| x_i - v_j \right\|^2 , \quad 1 \le m < \infty \tag{13}$$

The algorithm is initiated by the random assignment of the membership matrix. Then cluster centers are calculated according to Equation 3 [48].

$$v_j = \frac{\sum_{i=1}^{n} U_{ij}^m x_i}{\sum_{i=1}^{n} U_{ij}^m} \tag{14}$$

According to the calculated cluster centers, U matrix is updated using the following formula [48].

$$U_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|x_i - v_i\|}{\|x_i - v_k\|} \right)^{2/(m-1)}} \tag{15}$$

The above operations are repeated until the difference between the old matrix and the new matrix is smaller than stopping criteria (ε).

In FCM algorithm, distance of data to cluster centers is measured by Euclidean distance function which is the shortest distance between two points. The distance between point A and point B is calculated by the following formula, where $A(x_1, y_1)$ and $B(x_2, y_2)$ are two different points in a plane [52].

$$d_{oklid} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (16)$$

There are many different techniques for calculating the distance between two points. The suitability of the selected technique may vary according to the nature of the given data and the size of the data set [53]. Since the Euclidean distance is not always efficient in complex shapes [54], in this study Chebyshev distance is selected as the distance function of the FCM algorithm rather than Euclidean distance. The Chebyshev distance is the number of chess moves that must be made to pass another square in the chessboard. Thus, it is also known as the distance of the chessboard. The distance between the points $A(x_1, y_1)$ and $B(x_2, y_2)$ is calculated by the distance function Chebyshev as [52]:

$$d_{chebyshev} = max(|x_1 - x_2|, |y_1 - y_2|) \quad (17)$$

4.2. Data Clustering with FCM and CWOA Algorithms

Data clustering aims to cluster a data set that composed of a number of data rows in a certain number of clusters according to the ratio of their similarities to each other. A data row can include several features in its columns according to the properties of the data set. In this study the population matrix X for the FCM-CWOA algorithms are defined as follows:

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1,cxk} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{n,cxk} \end{bmatrix} \quad (18)$$

where n is the number of elements in the data set, k is the number of features in the k data set, and c is the number of clusters. Each row of the matrix X represents a candidate cluster centers and FCM-CWOA algorithms tries to minimize the FCM objective function. The pseudo code of the FCM-CWOA algorithms are given in Figure 7.

Start the whale population, X_i $(i = 1, 2, \ldots, n)$, with randomly generated cluster centers
Calculate FCM objective function for each candidate cluster center
X^*: the best cluster centers
***While** (t<maximum iteration)*
    ***For** each candidate cluster center*
        update a, A, C, l, P
        update membership matrix U
        update the location of the current cluster center according to p.
    end for
Check if that any candidate cluster center has exceeded the research space and correct it.
Calculate FCM objective function for each candidate cluster center
Update X^* if there is a better solution.
t=t+1
end while
back to X^*

Figure 7. The pseudo code of the FCM-CWOA Algorithms

4.3. Evaluation Criteria

In order to evaluate the performance of the proposed algorithm on solving data clustering problem, in this study the Rand and the Adjusted Rand Indexes are used.

Rand and Adjusted Rand Index

The Rand index, which calculates the similarity ratio between two clusters, is one of the most commonly used indices. It calculates the accuracy of clustering by finding how similar new clusters are to the actual clusters after clustering. The Rand Index is calculated using the following formula [55]:

$$RI = (n_s + n_d)/N \tag{19}$$

where n_s is the number of point pairs assigned to the same cluster, n_d is the number of pairs of points assigned to different clusters, and N is the number of all pairs of points in the dataset. If the two sets being compared are exactly the same, the Rand Index is 1 and if it is completely different, or if it contains a single element, the Rand Index is 0 [55].

The Adjusted Rand Index is the corrected version of the Rand Index. Similarity calculations based on estimation. The Adjusted Rand index gets -1 in the worst estimate and gets 1 the best estimate. The Adjusted Rand Index is calculated by the following formula [56].

$$ARI = \frac{a_i - b_i}{\max(a_i) - b_i} \tag{20}$$

where a_i is the current index value, b_i is the expected index value, and max (a_i) is the maximum index value.

4.4. Experimental Results

Each of the CWOA1, CWOA2…, CWOA10 algorithms were hybridized to the FCM algorithm and data clustering algorithms, named FCM-CWOA1, FCM-CWOA2..., FCM-CWOA10, were proposed. In section 3.1, it was mentioned that the chaos maps used were passed through an adaptive normalization process. For better understanding of effect of this normalization process, data clustering was performed with chaotic WOA algorithms integrated with non-normalized chaos maps. These hybrid algorithms were named CWOA1*, CWOA2*…, CWOA10*. In addition, to avoid the existing disadvantages of the Euclidean distance, all distances in the FCM algorithm was calculated with the Chebyshev distance function. And the revised FCM algorithm was integrated with chaotic WOA algorithms and proposed new hybrid algorithms called FCWOA1-c, FCWOA2-c..., FCWOA10-c. Clustering performance of the FCM-WOA, FCM-CWOA and FCWOA-c algorithms was evaluated with six different data sets selected from the UCI Machine Learning Repository. These data sets have the following characteristics; Iris dataset that has 150 data with 4 attributes, Balance Scale dataset has 625 data with 4 attributes, User Modeling dataset has 403 data with 5 attributes, Breast Cancer dataset has 699 data with 10 attributes, Seeds dataset has 210 data with 7 attributes, Fertility dataset has 100 data with 10 attributes. Proposed algorithms were compared with each other, K-Means, FCM, FCMALO, FCMGWO, FCMPSO and FCMSCA algorithms. ALO, GWO, PSO and GWO algorithms were hybridized with FCM in the same way as the WOA algorithm. All algorithms have been run 30 times. In the all of the algorithms parameter of m has been selected as 2, maximum number of iterations as 1000 and population size as 50. The closeness of the clustering results of algorithms to the real results was tested with two indexes, Rand Index and Adjusted Rand Index, which are frequently used in the literature. All the work is done under the same conditions as the Intel Core i7-7700HQ CPU with a 2.80GHz processor and 16 GB Ram on a computer with MATLAB R2017b program. The obtained results are shown in Table 4-6.

Table 4 shows that in most cases the proposed algorithms give better results than the compared algorithms. When the mean benchmark function, mean Rand Index and mean Adjusted Rand Index values for the Iris dataset are examined, it is seen that all of FCWOA-c algorithms give better results. Although the best maximum index values are obtained by FCMSCA algorithm, the average index values are low. The FCWOA-c algorithms for the Balance Scale dataset yield better results in terms of benchmark function, but only FCWOA2, FCWOA4, FCWOA6, and FCWOA7 algorithms perform well in terms of mean index values. Also, the clustering performance of the FCMWOA algorithm is significantly better than the FCM. For the User Modeling dataset, the FCMWOA algorithm yield better results than other algorithms. Although FCWOA-c algorithms have minimized the objective function better, it seems that this situation has no effect on clustering accuracy. In addition, the best clustering for this dataset is performed by FCM, FCMALO, FCMPSO, and FCMGWO algorithms. In the Breast Cancer dataset, although the FCM algorithm minimized the benchmark function better, K-means is observed to have higher clustering accuracy. In addition, the best result for maximum index values is obtained by FCWOA3-c algorithm. As for Seeds dataset, the maximum index values in all of the proposed algorithms are higher than the compared algorithms. However, FCWOA2-c, FCWOA7-c and FCWOA9-c algorithms have low mean index values. Lastly, in the Fertility dataset, all of the FCWOA-c algorithms are better than the other algorithms in terms of average benchmark function, maximum and mean index values. Though the aim is to minimize the benchmark function, it is important to note that the benchmark function is not critical in comparing the data clustering results since the distance function is changed here. While the benchmark function values are good, index values may be low.

Table 4. The data clustering results of the proposed algorithms

Results	Used Algorithm	K-Means	FCM	FCMPSO	FCMALO	FCMGWO	FCMSCA	FCMWOA	FCWOA1-c	FCWOA2-c	FCWOA3-c	FCWOA4-c	FCWOA5-c	FCWOA6-c	FCWOA7-c	FCWOA8-c	FCWOA9-c	FCWOA10-c
Iris Dataset	Benchmark Function (Mean)	91,6319	60,5057	60,5057	60,5057	61,4973	105,1161	63,0240	37,2824	37,7335	37,7014	37,8305	37,3291	37,0239	37,4434	36,8744	37,3076	36,8381
	RI (Max)	0,8797	0,8797	0,8797	0,8797	0,8797	0,9412	0,9195	0,9195	0,9267	0,9055	0,9195	0,9124	0,9124	0,9124	0,9124	0,9341	0,9195
	RI (Mean)	0,8481	0,8797	0,8797	0,8797	0,8785	0,8359	0,8788	0,8923	0,8894	0,8860	0,8897	0,8894	0,8948	0,8893	0,8942	0,8936	0,8942
	ARI (Max)	0,7302	0,7294	0,7294	0,7294	0,7294	0,8672	0,8180	0,8176	0,8343	0,7860	0,8176	0,8015	0,8022	0,8015	0,8019	0,8512	0,8176
	ARI (Mean)	0,6708	0,7294	0,7294	0,7294	0,7268	0,6476	0,7283	0,7571	0,7523	0,7437	0,7529	0,7511	0,7624	0,7507	0,7609	0,7599	0,7611
Balance Scale Dataset	Benchmark Function (Mean)	3491,65	1666,66	1666,66	1666,6691	1666,67	1710,64	1667,03	756,64	757,17	755,40	757,69	756,78	757,74	756,51	757,18	759,91	758,74
	RI (Max)	0,6855	0,7078	0,7497	0,6954	0,7247	0,7158	0,7028	0,7448	0,7833	0,6484	0,7906	0,7883	0,7233	0,8777	0,6892	0,6494	0,7402
	RI (Mean)	0,5844	0,5758	0,5846	0,5827	0,5811	0,5858	0,6091	0,5823	0,5908	0,5686	0,6148	0,5748	0,5943	0,5904	0,5766	0,5784	0,5772
	ARI (Max)	0,3401	0,387	0,4935	0,3647	0,4270	0,4046	0,3794	0,4808	0,5589	0,2686	0,5646	0,5603	0,4282	0,7500	0,3584	0,2729	0,4590
	ARI (Mean)	0,1286	0,1169	0,1430	0,1277	0,1241	0,1322	0,1809	0,1364	0,1487	0,1024	0,2008	0,1178	0,1575	0,1506	0,1211	0,1262	0,1216
User Modeling Dataset	Benchmark Function (Mean)	25,1014	11,0557	11,0555	11,0557	11,0719	13,0972	11,0961	4,9792	4,9889	5,0116	4,9937	4,9990	5,0140	4,9937	4,9898	5,0035	4,9846
	RI (Max)	0,7557	0,7659	0,7546	0,7700	0,7698	0,7780	0,7973	0,7336	0,7358	0,7420	0,7477	0,7420	0,7352	0,7664	0,7332	0,7568	0,7291
	RI (Mean)	0,7171	0,7534	0,7501	0,7508	0,7514	0,6889	0,7610	0,7120	0,7054	0,7024	0,7094	0,7062	0,7054	0,7067	0,7054	0,7080	0,7091
	ARI (Max)	0,3625	0,3936	0,3574	0,4169	0,4107	0,4192	0,4732	0,3325	0,3611	0,3808	0,3760	0,3555	0,3683	0,4417	0,3189	0,3577	0,3294
	ARI (Mean)	0,2647	0,3555	0,3438	0,3464	0,3486	0,2048	0,3805	0,2813	0,2736	0,2713	0,2803	0,2769	0,2756	0,2786	0,2719	0,2789	0,2769
Breast Cancer Dataset	Benchmark Function (Mean)	19323,17	14916,68	14916,68	14916,68	14916,74	17796,01	14923,07	5395,92	5620,80	5429,89	5421,45	5397,81	5344,28	5491,01	5490,57	5456,20	5469,72
	RI (Max)	0,924	0,9159	0,9159	0,9159	0,9159	0,9403	0,9213	0,9132	0,9213	0,9458	0,9321	0,9321	0,9240	0,9294	0,9159	0,9294	0,9321
	RI (Mean)	0,924	0,9159	0,9159	0,9159	0,9159	0,9098	0,9166	0,8520	0,8411	0,8534	0,8177	0,8362	0,8690	0,8741	0,8426	0,8301	0,8729
	ARI (Max)	0,8465	0,83	0,8300	0,8300	0,8300	0,8797	0,8409	0,8247	0,8414	0,8909	0,8631	0,8631	0,8465	0,8577	0,8302	0,8575	0,8631
	ARI (Mean)	0,8465	0,83	0,8300	0,8300	0,8300	0,8175	0,8315	0,6970	0,6718	0,6984	0,6184	0,6604	0,7312	0,7414	0,6733	0,6515	0,7396
Seeds Dataset	Benchmark Function (Mean)	588,048	414,6639	414,6639	414,7314	414,9329	683,2851	461,0920	301,3388	304,1829	301,6898	301,9656	301,8779	302,1744	301,7212	301,2437	302,1187	301,2629
	RI (Max)	0,8714	0,8714	0,8714	0,8714	0,8714	0,8866	0,8870	0,8872	0,8874	0,8874	0,8872	0,8872	0,8867	0,8867	0,8922	0,8874	0,8872
	RI (Mean)	0,8698	0,8714	0,8714	0,8714	0,8714	0,8096	0,8700	0,8750	0,8694	0,8744	0,8754	0,8739	0,8730	0,8713	0,8751	0,8712	0,8758
	ARI (Max)	0,7093	0,7093	0,7093	0,7093	0,7093	0,7432	0,7443	0,7446	0,7454	0,7454	0,7446	0,7446	0,7434	0,7434	0,7560	0,7454	0,7446
	ARI (Mean)	0,7059	0,7093	0,7093	0,7093	0,7093	0,5786	0,7062	0,7170	0,7048	0,7158	0,7180	0,7145	0,7127	0,7089	0,7173	0,7087	0,7189
Fertility Dataset	Benchmark Function (Mean)	176,0189	114,661	114,3693	114,3835	114,7361	124,1806	114,9374	48,2492	47,7413	48,1061	48,2210	48,0398	47,6823	48,5084	48,1217	47,9559	47,6222
	RI (Max)	0,5533	0,5	0,5000	0,5113	0,5345	0,5467	0,5467	0,6422	0,6422	0,7149	0,6533	0,7568	0,6648	0,7568	0,6891	0,7149	0,7018
	RI (Mean)	0,526	0,5	0,5000	0,5005	0,5129	0,5113	0,5152	0,5309	0,5266	0,5272	0,5358	0,5485	0,5348	0,5444	0,5428	0,5366	0,5325
	ARI (Max)	0,0633	0,0037	0,0037	0,0172	0,0427	0,0559	0,0773	0,1450	0,1899	0,0979	0,0863	0,1301	0,1991	0,0984	0,0884	0,0795	0,0757
	ARI (Mean)	0,0273	0,0037	0,0037	0,0040	0,0082	0,0048	0,0172	0,0190	0,0113	0,0090	0,0070	0,0249	0,0223	0,0204	0,0102	0,0188	0,0192

Therefore, Rand and Adjust Rand index values are considered as priorities when evaluating the results. In order to better observe the effect of the distance function on data clustering Table 4, Table 5 and Table 6 should be considered together. 5 out of the FCM-CWOA algorithms for the Iris dataset yield better than FCM, while all of FCWOA-c algorithms are better than FCM. 8 out of the FCM-CWOA algorithms for Balance Scale dataset perform better than FCM but with a minor difference. 4 out of FCWOA-c algorithms are significantly better than FCM for this dataset. In User Modeling dataset, all FCM-CWOA algorithms are better than FCM, however, FCWOA-c algorithms are far behind the FCM in terms of clustering performance. No performance improvement is observed in the FCM-CWOA algorithms for the Seeds dataset. Contrary to this, 7 out of the FCWOA-c algorithms yield better result than existing algorithms for this dataset. The FCM-CWOA algorithms in the Fertility dataset are not better than K-Means, but perform slightly better than FCM. Namely, it observed that performance of FCWOA-c algorithms is better than FCM-CWOA algorithms.

As a result, it can be said that changing the distance function has a positive effect on the clustering performance. Comparison of clustering results of FCM-CWOA and FCM-CWOA* algorithms are given in Table 5 and Table 6. When the results are examined, it is seen that normalization of chaos functions increases the data clustering performance of algorithms. For example, FCM-CWOA* algorithms in Iris and Breast Cancer datasets have very good results in maximum index values but average index values are low. That is, algorithms can achieve good results in only a few of 30 consecutive runs. It is observed that normalization of chaos functions increases the number of successful results by making this situation more stable. Although there is a similar case for the User Modeling and Seeds dataset, FCM-CWOA algorithms are by far better than FCM-CWOA* algorithms. In contrast to these examples, the use of non-normalized chaos functions in the Balance Scale and Fertility datasets are more useful and FCM-CWOA* algorithms perform better than FCM-CWOA algorithms. As a result, it can be concluded that the proposed hybrid algorithms successfully clusters most of the dataset tested and show better clustering performance than the compared algorithms.

Table 5. Comparison of clustering results of FCM-CWOA and FCM_CWOA* algorithms

Clustering Results	Iris Dataset					Balance Scale Dataset					User Modeling Dataset				
Used Algorithm	Benchmark Function (Mean)	RI (Max)	RI (Mean)	ARI (Max)	ARI (Mean)	Benchmark Function (Mean)	RI (Max)	RI (Mean)	ARI (Max)	ARI (Mean)	Benchmark Function (Mean)	RI (Max)	RI (Mean)	ARI (Max)	ARI (Mean)
FCM-CWOA1	**60,8670**	0,8859	**0,8820**	0,7430	**0,7344**	**1667,0587**	0,6937	0,5887	0,3574	0,1379	**11,0851**	**0,8175**	**0,7574**	**0,5223**	**0,3747**
FCM-CWOA1*	88,6650	**0,9407**	0,8428	**0,8658**	0,6624	1667,3975	**0,7201**	**0,5997**	**0,4131**	**0,1633**	11,5920	0,7586	0,6910	0,3620	0,1973
FCM-CWOA2	**61,7569**	0,8859	**0,8785**	0,7430	**0,7270**	**1667,0190**	0,6736	0,5896	0,3152	0,1404	**11,0809**	**0,8022**	**0,7620**	**0,4834**	**0,3815**
FCM-CWOA2*	95,7630	**0,9055**	0,8278	**0,7874**	0,6346	1668,4020	**0,7061**	**0,5963**	**0,3842**	**0,1555**	11,5951	0,7641	0,6959	0,3822	0,2076
FCM-CWOA3	**60,9428**	0,8859	**0,8796**	0,7430	**0,7291**	**1667,0225**	**0,7170**	**0,5867**	**0,4062**	**0,1338**	**11,0873**	**0,7937**	**0,7535**	**0,4709**	**0,3601**
FCM-CWOA3*	91,2052	**0,9251**	0,8471	**0,8305**	0,6705	1679,6410	0,6778	0,5719	0,3240	0,1030	11,8353	0,7529	0,6904	0,3450	0,1902
FCM-CWOA4	**62,6398**	0,8859	**0,8777**	0,7430	**0,7255**	**1667,0123**	0,6791	**0,5985**	0,3266	**0,1593**	**11,1069**	**0,8060**	**0,7620**	**0,4947**	**0,3844**
FCM-CWOA4*	93,7571	**0,9570**	0,8370	**0,9025**	0,6499	1667,7986	**0,6885**	0,5851	**0,3478**	0,1316	11,6144	0,7570	0,6729	0,3848	0,1589
FCM-CWOA5	**61,9013**	0,9124	**0,8806**	0,8017	**0,7316**	**1667,0198**	0,6763	0,5894	0,3229	0,1399	**11,0884**	**0,7983**	**0,7589**	**0,4715**	**0,3775**
FCM-CWOA5*	89,5217	**0,9575**	0,8419	**0,9037**	0,6607	1667,3871	**0,8345**	**0,6061**	**0,6575**	**0,1760**	11,6246	0,7787	0,6927	0,4189	0,2037
FCM-CWOA6	**60,8410**	0,8859	**0,8818**	0,7430	**0,7340**	**1667,0216**	**0,6942**	**0,5904**	**0,3602**	**0,1434**	**11,0998**	**0,8068**	**0,7602**	**0,4957**	**0,3787**
FCM-CWOA6*	92,7244	**0,9491**	0,8182	**0,8847**	0,6145	1667,8758	0,6839	0,5841	0,3371	0,1292	11,6477	0,7554	0,6861	0,3745	0,1870
FCM-CWOA7	**62,4198**	0,9055	**0,8783**	0,7874	**0,7267**	**1667,0350**	**0,7165**	0,5888	**0,4053**	0,1385	**11,0948**	**0,7988**	**0,7613**	**0,4820**	**0,3825**
FCM-CWOA7*	88,9898	**0,9412**	0,8317	**0,8668**	0,6424	1667,5606	0,6873	**0,5982**	0,3492	**0,1596**	11,6191	0,7729	0,6942	0,4029	0,2091
FCM-CWOA8	**61,0186**	**0,9195**	**0,8817**	**0,8176**	**0,7337**	**1667,0207**	0,6433	0,5802	0,2514	0,1211	**11,0835**	**0,8066**	**0,7625**	**0,4926**	**0,3838**
FCM-CWOA8*	92,3732	0,9055	0,8234	0,7874	0,6256	1667,3573	**0,7218**	**0,5996**	**0,4173**	**0,1632**	11,6422	0,7850	0,6948	0,4449	0,2063
FCM-CWOA9	**61,8857**	0,8859	**0,8779**	0,7437	**0,7258**	**1667,0041**	0,7025	0,5860	0,3773	0,1329	**11,0904**	**0,7996**	**0,7618**	**0,4846**	**0,3813**
FCM-CWOA9*	89,0862	**0,9571**	0,8476	**0,9029**	0,6703	1667,4367	**0,7033**	**0,6006**	**0,3801**	**0,1654**	11,5918	0,7542	0,6886	0,3599	0,1978
FCM-CWOA10	60,8742	0,8859	**0,8810**	0,7437	**0,7322**	**1667,0201**	0,7050	0,5798	**0,3808**	0,1205	**11,1072**	**0,8218**	**0,7613**	**0,5383**	**0,3852**
FCM-CWOA10*	**86,6334**	**0,9328**	0,8480	**0,8480**	0,6704	1667,7593	**0,6724**	**0,5814**	0,3126	**0,1232**	11,6719	0,7581	0,6923	0,3862	0,1995

Table 6. Comparison of clustering results of FCM-CWOA and FCM_CWOA* algorithms

Clustering Results	Breast Cancer Dataset					Seeds Dataset					Fertility Dataset				
Used Algorithm	Benchmark Function (Mean)	RI (Max)	RI (Mean)	ARI (Max)	ARI (Mean)	Benchmark Function (Mean)	RI (Max)	RI (Mean)	ARI (Max)	ARI (Mean)	Benchmark Function (Mean)	RI (Max)	RI (Mean)	ARI (Max)	ARI (Mean)
FCM-CWOA1	**14923,8657**	0,9186	**0,9165**	0,8355	**0,8313**	**463,1330**	**0,8815**	**0,8695**	**0,7317**	**0,7051**	**114,5881**	0,5345	0,5112	0,0458	**0,0154**
FCM-CWOA1*	22438,1048	**0,9348**	0,8631	**0,8686**	0,7241	575,2815	0,8765	0,8124	0,7207	0,5814	121,1339	**0,5404**	**0,5119**	**0,0526**	0,0088
FCM-CWOA2	**14924,9914**	0,9186	**0,9162**	0,8355	0,0499	**466,7510**	0,8771	**0,8679**	0,7223	**0,7015**	**114,7031**	0,5345	0,5125	0,0526	0,0167
FCM-CWOA2*	21496,3161	**0,9348**	0,9032	**0,8685**	**0,8046**	567,5886	**0,8773**	0,8096	**0,7226**	0,5776	120,5363	**0,5533**	**0,5156**	**0,0676**	**0,0218**
FCM-CWOA3	**14925,4253**	0,9213	**0,9171**	0,8409	0,0521	**466,0527**	0,8765	**0,8698**	0,7207	**0,7057**	**114,5549**	0,5533	0,5089	0,0633	0,0109
FCM-CWOA3*	20531,2940	**0,9403**	0,9026	**0,8797**	**0,8033**	558,0882	0,8765	0,8174	0,7207	0,5939	122,1372	0,5533	**0,5111**	**0,0758**	**0,0141**
FCM-CWOA4	**14924,6271**	0,9186	**0,9167**	0,8355	0,0713	**469,2325**	**0,8771**	**0,8688**	**0,7223**	**0,7035**	**114,6061**	0,5467	0,5109	0,0606	0,0151
FCM-CWOA4*	20791,3852	**0,9376**	0,9135	**0,8742**	**0,8251**	568,1740	0,8770	0,8063	0,7216	0,5683	120,7442	**0,5840**	**0,5215**	**0,1059**	**0,0195**
FCM-CWOA5	14923,7750	0,9186	**0,9168**	0,8355	0,0553	**462,3616**	**0,8780**	**0,8686**	**0,7243**	**0,7029**	**114,6183**	0,5467	0,5147	**0,0676**	**0,0199**
FCM-CWOA5*	**21177,1687**	**0,9348**	0,8847	**0,8686**	**0,7657**	585,7823	0,8765	0,8009	0,7207	0,5595	120,9781	0,5467	**0,5159**	0,0559	0,0167
FCM-CWOA6	14923,3290	0,9186	**0,9165**	0,8355	0,0633	**460,8005**	0,8822	**0,8672**	0,7336	**0,6998**	**114,5884**	0,5404	0,5109	0,0501	0,0154
FCM-CWOA6*	**22354,1561**	**0,9458**	0,8796	**0,8909**	**0,7557**	567,3689	**0,8879**	0,8162	**0,7467**	0,5892	121,7493	**0,5604**	**0,5168**	**0,0845**	**0,0170**
FCM-CWOA7	14924,2188	0,9186	**0,9168**	0,8355	0,0613	**460,3581**	0,8765	**0,8690**	0,7207	**0,7038**	**114,6854**	0,5467	0,5093	**0,0773**	0,0162
FCM-CWOA7*	**21365,4721**	**0,9403**	0,9093	**0,8797**	**0,8167**	544,2650	**0,8864**	0,8289	**0,7428**	0,6161	119,8872	**0,5533**	**0,5184**	0,0633	**0,0166**
FCM-CWOA8	14926,9324	0,9213	**0,9165**	0,8409	0,0553	**464,0310**	**0,8870**	**0,8691**	**0,7443**	**0,7040**	**114,6718**	0,5404	**0,5098**	0,0491	**0,0148**
FCM-CWOA8*	**21282,4620**	**0,9403**	0,9097	**0,8797**	**0,8176**	576,1721	0,8735	0,8149	0,7147	0,5876	119,5880	**0,5679**	0,5075	**0,0687**	0,0053
FCM-CWOA9	14926,3224	0,9213	**0,9170**	0,8409	0,0392	**300,7128**	**0,8819**	**0,8722**	**0,7327**	**0,7108**	**115,0477**	0,5345	0,5105	0,0529	0,0103
FCM-CWOA9*	**21955,8029**	**0,9458**	0,8989	**0,8908**	**0,7949**	**562,4678**	0,8781	0,8178	0,7250	0,5942	121,2436	**0,5604**	**0,5200**	**0,0959**	**0,0194**
FCM-CWOA10	14924,0029	0,9213	**0,9167**	0,8409	0,0511	**464,4170**	0,8870	**0,8707**	0,7443	**0,7078**	**114,5994**	0,5345	**0,5135**	**0,0606**	**0,0197**
FCM-CWOA10*	**22181,2992**	**0,9403**	0,8638	**0,8796**	**0,7244**	570,7439	**0,8922**	0,8265	**0,7558**	0,6135	121,4080	**0,5467**	0,5124	0,0559	0,0068

5. CONCLUSION

In this paper, the Whale Optimization Algorithm (WOA), a global optimization algorithm inspired by the hunting behavior of humpback whales, has been hybridized with the Fuzzy C-Means (FCM) algorithm after its performance was improved with chaos maps using an adaptive normalization method. To improve the performance of the WOA algorithm, a randomly selected parameter of the algorithm (a) was modified with 10 different chaos maps which each collected with a value after normalization process. And chaotic WOA algorithms were proposed. The performances of these algorithms were evaluated in terms of mean benchmark function, standard deviation, and Wilcoxon Sign Rank Test at 0.05 significance level and they were tested with 13 different benchmark functions. In addition, hybrid data clustering algorithms were developed by integrating FCM with proposed chaotic algorithms. In order to increase the data clustering performance of the proposed hybrid algorithms, all the distances in the FCM algorithm were calculated by using the Chebyshev distance function instead of Euclidean and the new hybrid clustering algorithms were proposed. The clustering performances of the hybrid data clustering algorithms were measured with the benchmark function, Rand Index and Adjusted Rand Index values for 7 different datasets selected from the UCI Repository database and then compared with the K-means, FCM, FCMWOA, FCMPSO, FCMALO, FCMGWO and FCMSCA algorithms. Also, the effect of changing the distance function of the FCM algorithm and of the normalization of chaos maps on the data clustering were evaluated. As a result, it has been seen that chaos functions improve the optimization performance of WOA algorithm, integrating chaotic WOA algorithms with FCM algorithm improves disadvantages of FCM algorithm, changing distance function increases clustering performance of algorithms.

REFERENCES

[1] S. Mirjalili and A. Lewis (2016) The Whale Optimization Algorithm. Adv. Eng. Softw 95:51–67.
[2] D. H. Wolpert and W. G. Macready (1997) No free lunch theorems for optimization IEEE Trans. Evol. Comput. 1(1): 67–82.
[3] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi (2007) Optimization by Simulated Annealing Science, New Series 220(4598):671–680.
[4] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi (2009) GSA: A Gravitational Search Algorithm Information Sciences 179(13):2232–2248.
[5] O. K. Erol and I. Eksin (2006) A new optimization method: Big Bang-Big Crunch Advance Engineering Software 37(2):106–111.
[6] B. Webster and P. Bernhard (2003) A local search optimization algorithm based on natural principles of gravitation Melbourne, FL. Florida Institute of Technology, pp 18.
[7] A. Hatamlou (2013) Black hole: A new heuristic optimization approach for data clustering Information Science 222:175–184.
[8] K.-C. Wu and C.-J. Ting (2010) A beam search algorithm for minimizing reshuffle operations at container yards Int. Conf. Logist. Marit. Syst. 2000:703–710.
[9] J. H. Holland (1992) Genetic Algorithms Scientific American 66–72.
[10] J. R. Koza and R. Poll (2005) GENETIC PROGRAMMING Introductory Tutorials in Optimization, Decision Support and Search Methodology, Chapter 5, Kluwer Press, pp. 127–164.
[11] Eiben A.E., Smith J.E. (2003) Evolution Strategies. In: Introduction to Evolutionary Computing. Natural Computing Series. Springer, Berlin, Heidelberg.
[12] Baluja S (1994), Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning, Cmu-Cs-94-163, 1–41.
[13] D. Simon (2008) Biogeography-based optimization IEEE Trans. Evol. Comput 12(6):702–713.
[14] J. Kennedy and R. Eberhart (1995) Particle swarm optimization Neural Networks IEEE Int. Conf. 4:1942–1948.
[15] M. Dorigo, M. Birattari, and T. Stutzle (2006) Ant colony optimization IEEE Computational Intelligence Magazine 1(4):28–39.
[16] Dervis Karaboga (2010) Artificial bee colony algorithm. Scholarpedia, 5(3):6915.
[17] S. Mirjalili, S. M. Mirjalili, and A. Lewis (2014) Grey Wolf Optimizer Advance Engineering Software 69:46–61.
[18] Pan, X., Xue, L. & Li, R. Neural Comput & Applic (2018). https://doi.org/10.1007/s00521-018-3449-6 Accessed 18 February 2019.
[19] S. Mirjalili (2015) Advances in Engineering Software The Ant Lion Optimizer Adv. Eng. Softw. 83:80–98.
[20] S. Mirjalili (2016) Knowledge-Based Systems SCA : A Sine Cosine Algorithm for solving optimization problems Knowledge-Based Systems 96:120–133.
[21] E. Alba and B. Dorronsoro (2005) The exploration/exploitation tradeoff in dynamic cellular genetic algorithms IEEE Trans. Evol. Comput. 9(2):126–142.
[22] S. Mirjalili and A. H. Gandomi (2017) Chaotic gravitational constants for the gravitational search algorithm Appl. Soft Comput. J. 53:407–419.

[23] J. Zhang, Y. Yang, and Q. Zhang (2009) The Particle Swarm Optimization Algorithm Based on Dynamic Chaotic Perturbations and its Application to K-Means 2009 Int. Conf. Comput. Intell. Secur. pp. 282–286.
[24] Y. Wang and M. Yao (2009) A new Hybrid Genetic Algorithm Based on Chaos and PSO IEEE International Conference on Intelligent Computing and Intelligent Systems, Shanghai, China, https://doi.org/10.1109/ICICISYS.2009.5357766 Accessed 18 February 2019.
[25] B. Alatas, E. Akin, and A. B. Ozer (2009) Chaos embedded particle swarm optimization algorithms Chaos, Solitons and Fractals 40(4)4:1715–1734.
[26] B. Alatas (2010) Chaotic bee colony algorithms for global numerical optimization Expert Syst. Appl. 37(8):5682–5687.
[27] B. Alatas (2010) Chaotic harmony search algorithms Appl. Math. Comput. 216(9):2687–2699.
[28] H. Yan, Z. Lv, Y. Zhao, G. Qiao, L. Xiao, and Z. Yang (2014) Chaos Genetic Algorithm Optimization Design Based on Linear Motor Chaos Genet. Algorithm Optim. Des. Based Linear Mot. 2:2265–2268.
[29] M. Javidi and R. Hosseinpourfard (2015) Chaos genetic algorithm instead genetic algorithm Int. Arab J. Inf. Technol. 12(2):163–168.
[30] L. D. S. Coelho and V. C. Mariani (2012) Firefly algorithm approach based on chaotic Tinkerbell map applied to multivariable PID controller tuning Comput. Math. with Appl. 64(8):2371–2382.
[31] J. Mingjun and T. Huanwen (2004) Application of chaos in simulated annealing Chaos, Solitons and Fractals 21(4):933–941.
[32] G. Zhenyu, C. Bo, Y. Min, and C. Binggang (2006) Self-adaptive chaos differential evolution Adv. Nat. Comput. 1:972–975.
[33] G. G. Wang, L. Guo, A. H. Gandomi, G. S. Hao, and H. Wang (2014) Chaotic Krill Herd algorithm Inf. Sci. (Ny). 274:17–34.
[34] Tanyıldızı E, Cigal T (2017) Kaotik Haritalı Balina Optimizasyon Algoritmaları 29(1):309–319.
[35] W. Z. Sun and J. S. Wang (2017) Elman Neural Network Soft-Sensor Model of Conversion Velocity in Polymerization Process Optimized by Chaos Whale Optimization Algorithm IEEE Access https://doi.org/10.1109/ACCESS.2017.2723610 Accessed 18 February 2019.
[36] D. Oliva, M. Abd El Aziz, and A. Ella Hassanien (2017) Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm Appl. Energy 200:141–154.
[37] S. Karthikeyan and T. Christopher (2014) A Hybrid Clustering Approach using Artificial Bee Colony (ABC) and Particle Swarm Optimization 100(15):1–6.
[38] X. H. Han, L. Quan, X. Y. Xiong, M. Almeter, J. Xiang, and Y. Lan (2017) A novel data clustering algorithm based on modified gravitational search algorithm Eng. Appl. Artif. Intell. 61:1–7.
[39] J. C. Dunn (1973) A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters J. Cybern. 3(3):32–57.
[40] J. C. Bezdek, R. Ehrlich, and W. Full (1984) FCM: The fuzzy c-means clustering algorithm Comput. Geosci. 10(2–3):191–203.
[41] S. J. Nanda and G. Panda (2014) A survey on nature inspired metaheuristic algorithms for partitional clustering Swarm Evol. Comput.16:1–18.
[42] G. Gan, J. Wu, and Z. Yang (2009) A genetic fuzzy k-Modes algorithm for clustering categorical data Expert Syst. Appl. 36(2) PART 1:1615–1620.
[43] U. Maulik and I. Saha (2010) Automatic fuzzy clustering using modified differential evolution for image classification IEEE Trans. Geosci. Remote Sens. 48(9):3503–3510.

[44] B. Zhao (2007) An ant colony clustering algorithm Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, China, pp. 19–22.
[45] T. a. Runkler and C. Katz (2006) Fuzzy Clustering by Particle Swarm Optimization 2006 IEEE Int. Conf. Fuzzy Syst. 3:601–608.
[46] W. Zhu, J. Jiang, C. Song, and L. Bao (2012) Clustering algorithm based on fuzzy c-means and artificial fish swarm Procedia Eng. 29:3307–3311.
[47] H. Izakian and A. Abraham (2011) Fuzzy C-means and fuzzy swarm for fuzzy clustering problem Expert Syst. Appl. 38(3):1835–1838.
[48] E. Esme and B. Karlik (2016) Fuzzy c-means based support vector machines classifier for perfume recognitio Appl. Soft Comput. J. 46:452–458.
[49] M. Lichman UCI Machine Learning Repository http://archive.ics.uci.edu/ml. Accessed: 18 February 2019.
[50] J. Feng, J. Zhang, X. Zhu, and W. Lian (2017) A novel chaos optimization algorithm Multimedia Tools and Applications 76(16):17405–17436.
[51] J. Derrac, S. García, D. Molina, and F. Herrera (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms Swarm Evol. Comput. 1(1):3–18.
[52] M. Kabak, F. Sağlam, and A. Aktaş (2017) Farkli Uzaklik Hesaplama Yaklaşimlarinin Topsis Üzeri Kullanilabilirliğinin İncelenmesi Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi 32(1):35–43.
[53] D. J. Bora and A. K. Gupta (2014) Effect of Different Distance Measures on the Performance of K-Means Algorithm : An Experimental Study in Matlab Int. J. Comput. Sci. Inf. Technol. 5(2):2501–2506.
[54] J. Arora (2016) Hybrid FCM PSO Algorithm with CityBlock Distance pp. 2609–2614.
[55] W. M. Rand (1971) Objective Criteria for the Evaluation of Clustering Methods Author (s): William M . Rand Source : Journal of the American Statistical Association 66 (336):846.
[56] L. Hubert and P. Arabie (1985) Comparing partitions J. Classif. 2(1):193–218.