IJCSN - International Journal of Computer Science and Network, Volume 7, Issue 6, December 2018
ISSN (Online): 2277-5420
www.IJCSN.org
Impact Factor: 1.5

Enhancing K-Means Clustering with Bio-Inspired Algorithms

[1] Doaa Abdullah; [2] Hala Abdel-Galil; [3] Ensaf Hussein

[1] Computer Science, Helwan, Faculty of Computers and Information
Helwan, Cairo, Egypt

[2] Computer Science, Helwan, Faculty of Computers and Information
Helwan, Cairo, Egypt

[3] Computer Science, Helwan, Faculty of Computers and Information
Helwan, Cairo, Egypt

Abstract - Data clustering is considered an important data analysis and data mining technique. It is included in a variety of disciplines such as machine learning, pattern recognition and bioinformatics. K-Means algorithm is a popular clustering algorithm but it suffers from its dependency on its initial centroid locations which fells the algorithm into the local optima. Bio-inspired algorithms are powerful in searching for the global optimal solutions. In this paper, the most recent bio-inspired algorithms; Crow search, Whale optimization, Grasshopper optimization and Salp swarm algorithms are integrated into the K-Means algorithm, to overcome the K-Means drawbacks. The proposed techniques are implemented and applied on eight numerical UCI datasets. Experimental results reveal the capability of the proposed algorithms to find the optimal initial centroid locations which achieve better clustering integrity. Moreover, the results show that the integration of the k-Means with the Crow search algorithm is superior compared to the others bio-inspired algorithms.

Keywords - *Crow Search Algorithm, Whale optimization Algorithm, Salp Swarm algorithm, Grasshopper Optimization Algorithm, K-Means Clustering Algorithm, Sum of Squared Errors (SSE).*

1. Introduction

Data clustering is one of the most important techniques in data analysis and data mining fields. It is the process of representing a set of data points in a form of groups. These groups are dissimilar from each other but the members inside each group are at a high level of similarity based on a similarity measure. Many fields make use of the clustering process to extract hidden and useful groups of data. Image segmentation, market segmentation, text clustering and geographic information systems are examples of these fields [10]. The K-Means clustering algorithm is a partitional clustering algorithm. It partitions a set of data points into groups through assigning each point to its nearest centroid, recalculates the centroid points and then reassigns the points based on their distance from the new centroid points. K-Means drawbacks come from its performance dependency on the initial centroid points which prevent the K-Means algorithm from reaching the global optimum solution and fall into the local optima and hence affects the clustering result. In the last two decades, bio-inspired computation has become popular and been applied in almost every area of science and engineering [21]. Bio-inspired algorithms imitate the social cooperative behaviour of the swarms of the natural creatures for catching a prey, foraging for food .. etc. The search process begins with a population which is generated randomly. This population is evolved over next generations. The strengths of these methods lie in the combinations of the best individuals to form the individuals of the next generations. This cause the population to be optimized over the course of generations. Simplicity (Easy to implement), Flexibility (dealing with various and different kinds of optimization problems), and Ergodicity (ability to search multimodal landscape and avoiding any local optimum) are considered the main factors of the bio-inspired computation that cause this popularity.

2. Related Work

Many researchers began to solve the clustering problem and overcome the drawbacks of the K-Means algorithm using bio-inspired algorithms to make use of the bio-inspired searching mechanism for the global optima and hence achieve better cluster homogeneity and better

Table 1 : Summery of the Bio-inspired Algorithms for Data Clustering

Paper Referred	*Algorithm*	*Fitness Fn*	*Performance metrics*	*Dataset*
[Huang & Zhou, (2011)][7]	GSO + K-Means (GSOCA+KM)	SSE	Accuracy	Iris, Synthetic data
[Karaboga & Ozturk, (2011)][11]	ABC	SSE	Error rate	Iris, Wine, Breast Cancer, Breast Cancer Int, Credit, Balance Scale, Glass, Heart, Horse, Thyroid, Ecoli, Dermatology, Diabetes
[Kwedlo, (2011)][12]	DE + K-Means (DE-KM)	SSE	SSE	Iris, TSP-LIB, Image segmentation
[Hassanzadeh & Meybodi, (2012)][8]	FA + K-Means (KFA)	SSE	Fitness, Error rate	Iris, Wine, Breast Cancer, Glass, Sonar
[Hatamlou & et al. (2012)][9]	GSA + K-Means (GSA-KM)	MSE	SSE, number of fitness evaluation	Iris, Wine, Glass, CMC, Breast Cancer
[Liu & et al. (2012)][13]	CSO + K-Means (KSACSOC)	SSE	Fitness, Success rate, run time	Iris, Wine, Breast Cancer, Glass, Vowel, Crude Oil, Data-52, Data-62
[Tang & et al. (2012)][20]	WSA + K-Means (C-wolf) (C-ant)(C-firefly)(C-bat)(C-Cuckoo)	SSE	SSE	Iris, Wine, Haberman's, Libras, Synthetic, Mouse
[Shah-Hosseini, (2013)][17]	IWE+K-Means (IWE-KM)	SSE	SSE	Iris, Wine, Breast Cancer, Thyroid, Ecoli, Image Segmentation
[Fong & et al. (2014)][5]	C-ACO, C-Firefly, C-Cuckoo, C-Bat	SSE	Fitness, CPU time	Iris, Wine, Haberman's, Image Segmentation Libras, Synthetic,
[Saida & et al. (2014)][18]	CS	SSE	Fitness, Convergence	Iris, Wine, Breast Cancer, Vowel
Chen & et al.(2014)][2]	Monkey, ABC , K-Means (ABC-MA)	SSE	SSE , SD , Convergence	Art1, Art2, Iris , TAE , Wine, Seeds, Glass, Heart, Haberman's survival, Balance scale, Cancer, CMC
[Gu & et al. (2015)][6]	Chaotic PSO , K-Means (CPSO-KM)	SSE	SSE , Rand Index , Jaccard coefficient and F-Measure	SynSet1, SynSet2, SynSet3 ,Iris , Haberman , Wine, Hayes-Roth
[Corrêa & et al. (2016) [3]	PSO , FSS , K-Means , K-Harmonic Means (KHM) K-FSS , K-PSO, KH-FSS , KH-PSO	SSE and KHM	Two internal cluster validity measures (SSW and SSB). Accuracy , SD , Serial execution time ,Parallel execution time	Iris, Breast Cancer , Glass, Wine , Olive, Grudi Oil , Diabetes, Egyptian Skulls, Heart Statlog, Ionosphere , Vehicle , Balance Scale , Sonar
Li and Liu, (2017)[14]	KHA + K-Means	SSE	SSE , Accuracy	Iris , Wine , Glass , Cancer , CMC

clusters heterogeneity. Table 1. presents a summary of the previous work for data clustering using bio-inspired algorithms. Proposed solutions in the literature present a reasonable results with respect to the clustering accuracy and the objective function value of the clustering integrity. In this Paper, We aim to introduce the most recent swarm intelligence algorithms to explore and test their applicability and performance in solving the clustering problem. The most recent swarm algorithms are Whale optimization algorithm, Crow search algorithm, Salp swarm algorithm and Grasshopper optimization algorithm. These algorithms have their own social behaviors and use special methods to produce new solutions from old ones. Another aim is to present a comparison among the aforementioned bio-inspired algorithms with respect to the performance, the convergence curve, and the optimization result .

The organization of the rest of the paper is as follows: Section 3 briefly explains the K-Means clustering algorithm and the bio-inspired algorithms. Section 4 presents the integration of the K-Means with the bio-inspired algorithms. Section 5 presents the experimental results of the integration methods. Finally, the conclusion is provided in section 6.

3. K-Means and Bio-inspired Algorithms

3.1 K-Means Algorithm

Given a set of data points $D = \{x_{1,2},...,x_n\}$, each data point xi is a p-dimensional vector. The following Pseudo code

show the method of the K-Means to cluster the dataset D into k clusters {C1,C2,...,Ck} considering the following conditions :

- $C_i = \ , i = 1, 2, \ldots K$.
- $C_i \cap C_j = 0, \ , j = 1, 2, \ldots , K, \ i \neq j$.
- $\bigcup_{i=1}^{k} C_i = \{x_1, x_2,\ldots,_n\}$.

Table 2 : K-Means Pseudo code

Input : *K* (the number of clusters) , *D* (dataset)
Output : a set of k clusters
Method :
1. Randomly select k points from D as the initial clusters centroid locations .
2.Repeat:
1. (re) assign each data point to the cluster to which the point is the most similar based
on the mean value of the data points in the clusters .
2. calculate the mean value of the data points for each cluster to update clusters centroid locations .
Until no change .

K-means algorithm is very sensitive to the selection of the initial centroid locations since different centroid points cause different clustering results. So, placing them far away as much as possible is a better choice.

3.2 The bio-inspired algorithms

3.2.1 Crow search algorithm

Crow search algorithm (CSA) [1] is a new bio-inspired algorithm proposed by Askarzadeh in 2016. The primary inspiration for CSA came from the search mechanism of crows to hide their food.

The four main principles of CSA are defined as follows:

- Crows live in crowds' form.
- Crows hide foods and save the hidden place of the food in their memory.
- Crows follow each other while doing thievery.
- Crows have an awareness probability against thievery so the ability to protect their catches depends on this probability.

The mathematical model of the tailing motion of CSA is formulated as follows:

$$x_{(i,t+1)} = \begin{cases} x_{(i,t)} + r_i * fl_{(i,t)} * [m_{(j,t)} - X_{(i,t)}] & r_j \geq AP_{(j,t)} . \\ a\ random\ position & otherwise . \end{cases} \quad (1)$$

where $x_{(i,t+1)}$ and $x_{(i,t)}$ is the next and current position of crow i. r is a random number uniformly distributed in the interval [0 , 1], $fl_{(i,t)}$ is the flight length of crow i at (t)th iteration. $m_{(j,t)}$ is the hidden food position of crow j. $AP_{(j,t)}$ is the awareness probability of crow j at (t)th iteration.

3.2.2. Salp swarm algorithm

Salp Swarm Algorithm (SSA) [16] is one of the natural-inspired algorithms recently proposed by Mirjalili in 2017. The swarming behaviour of salps is considered the main inspiration for SSA. The population of SSA is divided into two groups. These two groups are the leader search agent and the follower's search agents (followers follow each other and the leader directly or indirectly). The leader updates its position according to the following mathematical model :

$$x_j^1 = \begin{cases} F_j + c_1 \left((ub_j - lb_j)c_2 + lb_j \right) & c_3 \geq 0 . \\ F_j - c_1 \left((ub_j - lb_j)c_2 + lb_j \right) & c_3 < 0 . \end{cases} \quad (2)$$

here x_j^1 denotes as the position of the first salp (leader) at jth dimension, Fj is the target food position of the swarm at jth dimension, ubj and lbj is the upper and lower boundary of the search space at jth dimension. c2 and c3 are random values uniformly generated in the range [0 ,1]. c1 balance the exploration and exploitation . It is defined as following :

$$c_1 = 2e^{-(\frac{4l}{L})^2} . \quad (3)$$

where l is the current iteration number and L is the maximum number of iterations .

The followers update their position according to the following equation :

$$x_j^1 = \frac{1}{2}(x_j^i + x_j^{i-1}) . \quad (4)$$

where $i \geq 2$ and x_j^i denotes as the position of the ith salp at jth dimension .

3.2.3 Whale optimization algorithm

Whale optimization algorithm (WOA) [15] is a novel meta-heuristics algorithm proposed by Mirjalili and et. Al in 2016. WOA simulates bubble-net attacking method of the humpback whales when they hunt their prey. The humpback whales foraging behavior is done by creating special bubbles in a spiral shape. Humpback whales know the location of the prey so WOA considers the current candidate solution is the best-obtained solution and near the optimal one. The other whales try to update their positions towards the best solution.

Exploitation phase in WOA is achieved through designing two mechanisms to mathematically model The bubble-net attack method. These mechanisms are :

(1) Shrinking encircling mechanism.
(2) Spiral updating position mechanism.

These mechanisms are mathematically formulated by the following equation :

$$\overrightarrow{X}_{(t+1)} = \begin{cases} \overrightarrow{X^*}_{(t)} - \overrightarrow{A} . \overrightarrow{D} & if\ p < 0.5 . \\ \overrightarrow{D'} . e^{bl} . cos(2\pi l) + \overrightarrow{X^*}_{(t)} & p \geq 0.5 . \end{cases} \quad (5)$$

where p is a random value in the interval [0 ,1] . Based on the value of p , the whale choose between shrinking encircling mechanism in case $p < 0.5$ and spiral updating position mechanism in case $p \geq 0.5$. X^* is the position vector of the current best solution . $\overrightarrow{A}$ and $\overrightarrow{D}$ vectors are calculated from the following equations .

$$\overrightarrow{D} = \left| \overrightarrow{C} . \overrightarrow{X^*}_{(t)} - \overrightarrow{X}_{(t)} \right| . \quad (6)$$

$$\overrightarrow{A} = 2 \overrightarrow{d} . \overrightarrow{r} - \overrightarrow{a} . \quad (7)$$

$$\overrightarrow{C} = 2 . \overrightarrow{r} . \quad (8)$$

where a is a random vector decreased from 2 to 0 over a course of iterations and r is a random vector . b is a constant and l is a random number from [-1,1] .

The exploration phase in WOA is achieved through replacing the best search agent ($\overrightarrow{X^*}$) by a randomly chosen search agent $\overrightarrow{X}_{(\text{rand})}$based on the value of the vector A . if A > 1 or A< -1 the search agent follow a random chosen agent otherwise it follow the best one. The mathematical model is as follows :

$$\overrightarrow{D} = \left| \overrightarrow{C} . \overrightarrow{X}_{(rand)} - \overrightarrow{X}_{(t)} \right| . \quad (9)$$

$$\overrightarrow{X}_{(t+1)} = \overrightarrow{X}_{(rand)} - \overrightarrow{A} . \overrightarrow{D} . \quad (10)$$

3.2.4 Grasshopper optimization algorithm

Grasshopper optimization algorithm (GOA) [19] is a novel meta-heuristics algorithm proposed by Mirjalili and et. al in 2017. GOA simulates the grasshopper swarms and their social interaction. For source seeking grasshopper swarms divide the search process into two tendencies: exploration and exploitation. In GOA each grasshopper represents a solution in the search space. The mathematical model that formulate the social behaviour of the grasshopper swarm is as follows :

$$X_i^d = c \left(\sum_{\substack{j=1 \\ j \neq i}}^{N} c \frac{ub_d - lb_d}{2} s\left(\left| X_j^d - X_i^d \right| \right) \frac{X_j - X_i}{d_{ij}} \right) + \widehat{T_d} . \quad (11)$$

where X_i^d is the new position in the D^{th} dimension, ub_d and lb_d is the upper and lower bounds in the dimension d . x_i^d is the current position of the grasshopper i in the D^{th} dimension and x_j^d is the current position of the grasshopper j in the D^{th} dimension. $\widehat{T_d}$ is the value of the D^{th} dimension of the best solution found so far and c is a decreasing coefficient calculated as following :

$$c = \text{cmax} - \text{l} \frac{cmax - cmin}{L} . \quad (12)$$

where l is the current iteration and L is the maximum number of iterations. where $s\left(\left| X_j^d - X_i^d \right| \right)$ is a function represent the strengths of the attraction and repulsion social forces and is defined as :

$$s(r) = f e^{\frac{-r}{l}} - e^{-r} . \quad (13)$$

4. The Proposed Bio-Inspired Clustering Algorithms

K-means algorithm always converges but it may converge to the local optima instead of the global optimal solution. The first issue to achieve a better clustering using K-Means, good initial centroid locations have to be chosen at the beginning. Therefore, the initial centroid locations should be distributed as far as possible from each other and in a way that can form clusters with global optimal value. The second issue, a global optima exploring mechanism have to be applied. In clustering, a global optimal solution can be expressed as the maximum inter dissimilarities and the minimum intra dissimilarities. K-means suffers from the ability to explore the search space for the global optima due to its sensitivity to the number of clusters and their centroid locations which are comparatively stuck in the search space. Only small modifications of the initial centroid positions are done during the clustering process. Running k-means several times with different initiations doesn't guarantee that the solution is the global optima since each trial is independent of the other trials. This paper shows integration techniques between the original k-means algorithm and the bio-inspired algorithms. Figures [1 - 4] are the flowcharts of the integrated versions C-Salp, C-Crow, C-Whale, C-Grasshopper respectively.

These techniques enhance the k-means algorithm by adding an exploration function. The exploration function optimizes the current solution through exploring regions outside the vicinity of the current solution, and if a new better solution than the current best one is found, the search agents move towards the best solution. The exploration process continues until some stopping criteria are satisfied. Bio-inspired algorithms are meta heuristic algorithms so the exploration is done heuristically without the need to process the entire combinatorial search space. The integration methods are based on representing the search agents as a combination of centroid locations, then the search agents explore the search space for the best solution (i.e. The best initial centroid locations).

So the Integration method is divided into 4 basic steps (explained in section 4.1 to section 4.4) :

(1) Initialization step.
(2) Cluster assignment and fitness calculation step.
(3) Exploration and Centroid locations update step.
(4) Termination step.

Here, all the bio-inspired algorithms that are used in the integration with K-Means employ the same initialization

step, cluster assignment and fitness Calculation step and the termination step but each algorithm has its unique heuristic method to explore the search space to find the optimal solution.

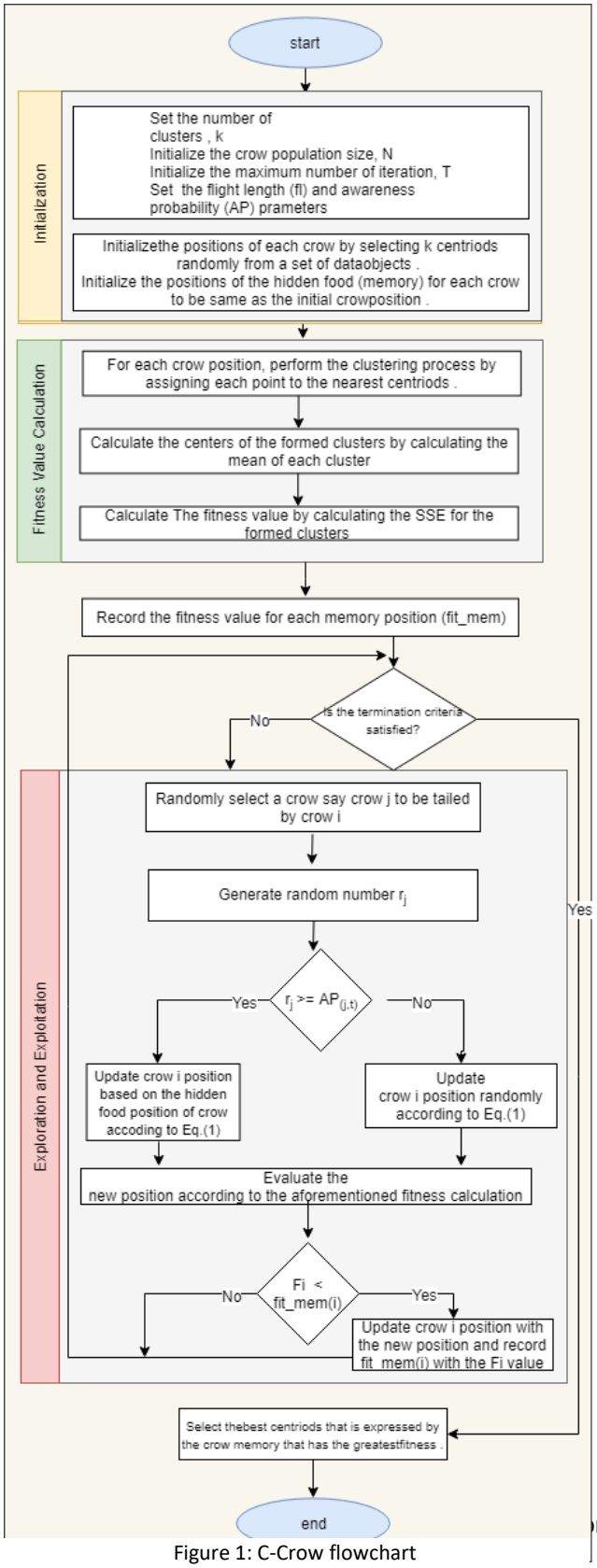

Figure 1: C-Crow flowchart

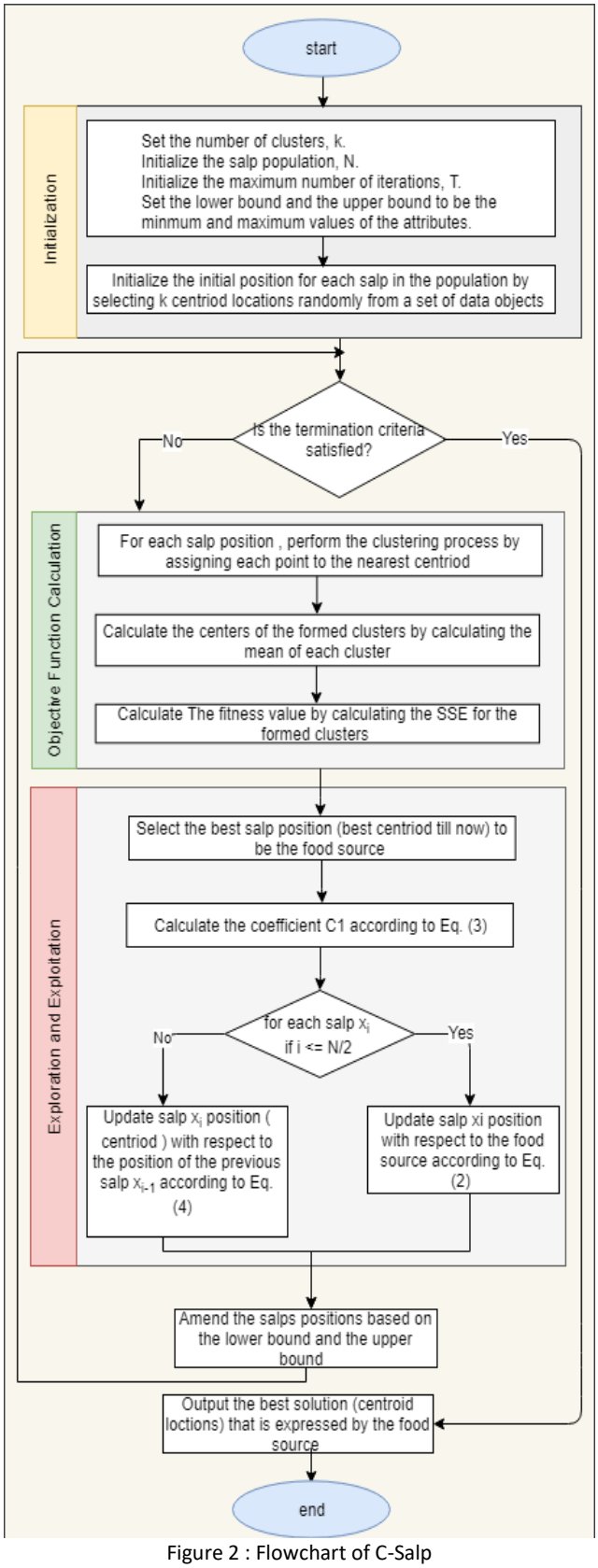

Figure 2 : Flowchart of C-Salp

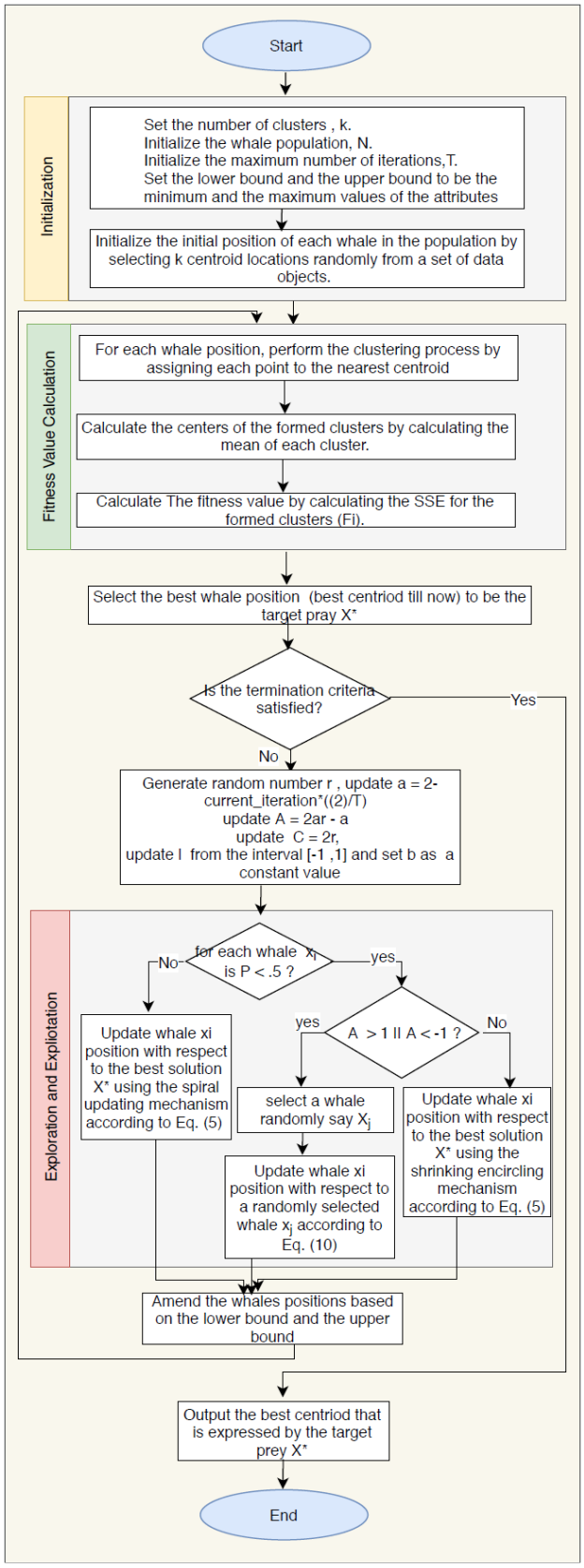

Figure 3 : C-Whale Flowchart

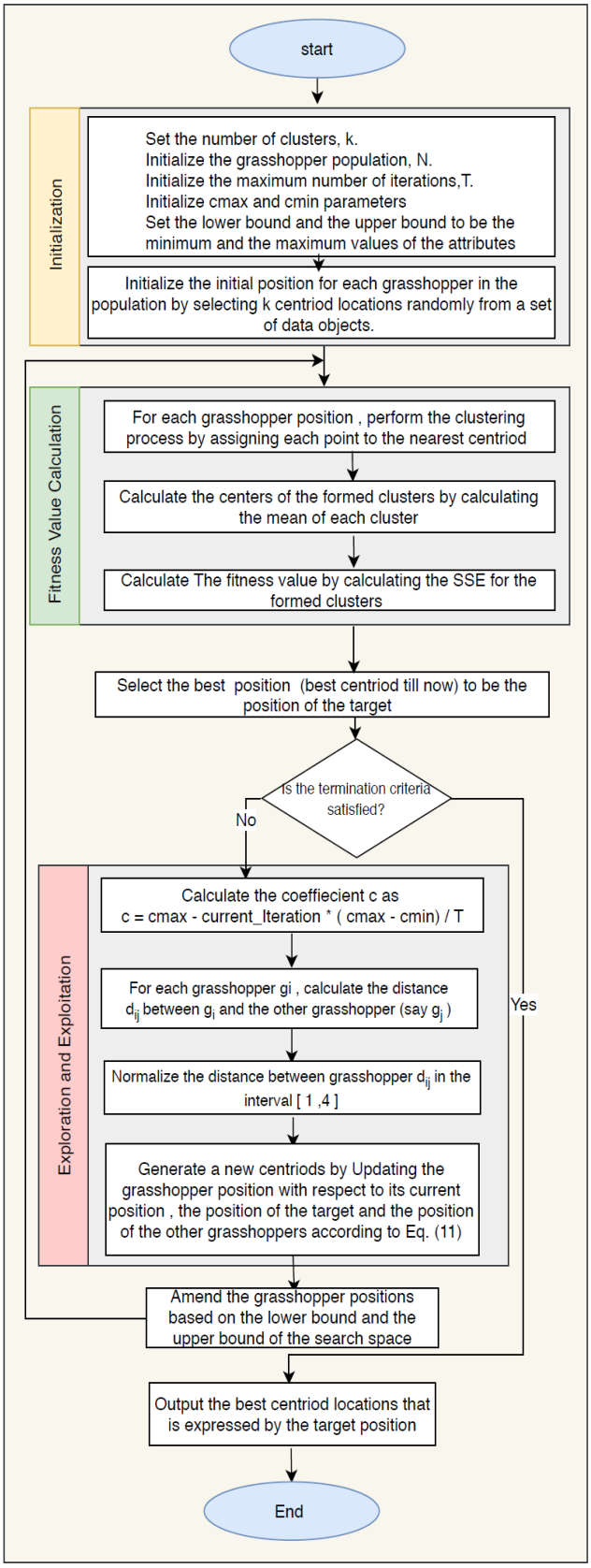

Figure 4 : C-Grasshopper Flowchart

4.1. The initialization step

In the initialization step, the population size and the maximum number of iterations is initialized. Also, the upper bound and the lower bound of each search agent is set based on the maximum and the minimum value of the dataset attributes. Each search agent is initialized with a set of k centroid locations (k is a user-defined number of clusters) that is going to form the clusters of the dataset following the same rule of the original k-means. These k centroid locations are data points selected randomly from the search space. C-Crow, C-Salp, C-whale, and C-grasshopper has their unique functional parameters that affect the performance of the exploration stage so they have to be initialized carefully otherwise it may cause failure in finding the global optimal solution.

4.2. Cluster assignment and fitness calculation step

In our design cluster assignment step follows the same rule of the original k-means. Each search agent is represented as a three dimensional matrix of size 1* k*D where D is the number of attributes of the dataset and therefore it is the dimension of the search space. and the ith search agent will look like Ai = [c1 , c2] where c1 , c2 are vectors of size 1x3 that represent the clusters centroid locations for example, c1 = [xi1 , xi2 , xi3] where xi is a data point in the search space . The population is a three-dimensional matrix of size N * K * D where N is the population size. To perform cluster assignment we calculate the Euclidean distance between every data point in the dataset and each cluster centroid in the search agent , then each data point is assigned to the closest centroid. In our design we used Euclidean distance which is represented as the following :

$$d\,(x_i, c_j) = \sqrt{\sum_{v=1}^{D}(x_{iv} - c_{jv})} \quad , \text{i = 1.. n , j = 1..k .} \qquad (14)$$

So, a data point xi is assigned to cluster j only when the distance from point xi to cluster j centroid is at its minimum and hence each point in the same cluster is as close as possible to this centroid. Our objective function is to find the best group of centroid locations that form clusters with the minimum summation of the intra-cluster distance for all clusters. Therefore, we used the sum of the squared error function (SSE) to be our objective function. The objective function is given as follows:

$$f_{obj} = \sum_{j=1}^{k}\sum_{i=1}^{N} w_{ij} * \left\|x_i - c_j\right\|^2. \qquad (15)$$

$$w_{ij} = \begin{cases} 1 \;\; if \; xi \; is \; a \; member \; in \; cluster \; j \\ 0 \;, otherwise \end{cases} \qquad (16)$$

Generally, The objective function shows to what extent the contribution of every variable to the value to be optimized for the problem. In our situation, the objective function variable is the position of the centroid . In bio-inspired algorithms, this variable is relocated many times aiming to catch a position that achieves the best objective value (i.e the minimum value of the SSE function) of the clustering process.

4.3. Exploration and centroid update step

From the flowcharts fig. [1-4] , we can find that each one of the proposed algorithms are divided into four constructs . These constructs have common operations but the main difference lies in how to perform exploration for the global optimal centroid locations . In C-Crow , for instance the flight length (fl)parameter affects the search capability. Larger values of the (fl) parameter relinquish a crow away from the local proximity and hence diverse the results .but smaller values intense the results in the vicinity of another crow (performing local search).

In C-Salp, The new salp position is dictated by three random parameters C1 , C2 and C3 . C3 directs it towards the positive infinity or the negative infinity. C2 controls the step size and C1 balances the exploitation and exploration. Since the leading salp update its position only with respect to the food source which is the best obtained position so it constantly explores and exploits the vicinity of the food source. The follower salps move gradually towards the leader and update their positions with respect to each other . Moving gradually prevents the algorithm from falling into the local optima . In C-Whale, Choosing the coefficient vector A with random values less than -1 or greater than 1 enforce a whale agent to update its position far away from a reference whale . also choosing the reference whale randomly instead of the best search agent found yet, emphasize search space exploration and allow performing global search . In C-Grasshopper, each grasshopper update its position with respect to its current position , target position and the other grasshopper positions . the exploration is guided by the value of the adaptive parameter C and the social forces that indicate if a grasshopper should be attracted to the target or should be rebelled from.

4.4. Termination and output step

The optimization process terminates when it reaches the maximum number of iterations or when the best solution is found. In our design, the algorithm terminate when it reaches the maximum number of iterations . After termination the output will be the best search agent with the highest fitness value.

5. Experimental results and discussion

5.1. Dataset description

The new proposed algorithms are tested over eight datasets downloaded from the UCI machine learning repository [4]. A brief description of each dataset is presented in Table 2.

Table 3. Datasets Description

Dataset	*No. of Instances*	*No. of attributes*	*No. of Clusters*
Wine	178	13	3
Iris	150	4	3
CMC	1473	9	3
Ionosphere	351	34	2
Glass	214	10	6
Haberman's	306	3	2
Libras	360	91	15
Sonar	208	60	2

5.2. Performance metrics

In this subsection, four different measurements are adopted. These measurements are the best, the worst, the mean objective function values, and the standard deviation over all experiment. Also, a rank based on the mean metric is adopted to compare the performance of the integrated bio-inspired clustering algorithms.

$$\text{Worst Value}=\max_{i=1}^{N} BS_i \quad (17)$$

$$\text{Best Value}=\min_{i=1}^{N} BS_i \quad (18)$$

$$SD=\sqrt{\frac{\sum_{i=1}^{N}(BS_i-\mu)^{\wedge}2}{N}} \quad (19)$$

$$\text{Mean Value}=\frac{1}{N}\sum_{i=1}^{N} BS_i \quad (20)$$

Where BS is the best score obtained after each experiment and N is the number of experiments.

5.3. Analysis and discussion

In this subsection, various experiments are conducted on eight datasets. These experiments aim to evaluate the performance of the proposed integrated clustering algorithms. All experiments are implemented on the same platform with the following hardware specification : Inter(R) Core(TM) i5-4200U CPU and 8 GB RAM . Each Algorithm has its own parameters that have to be initialized at the beginning. Table 3. display the parameters set for all algorithms.

5.3.1. The performance of the proposed algorithms :

The main objective of the following experiments is to evaluate and compare the performance of the proposed algorithms. To find the average of the best objective function values, all algorithms were executed 10 times with the same parameters. To perform fairly comparison, all algorithms started with the same initial population for the same dataset. The best result among the tested algorithms is underlined. We compared the performance of the algorithms from various aspects. Firstly, we compared the

Table 4. Parameters of the algorithms

Algorithm	*Parameters*	*Values*
K-means	*K*	Based on the dataset
C-Crow	Awareness probability (*AP*) Flight length (*fl*)	.1 2
C-Whale	*P*	1
C-grasshopper	*cmax* *cmin*	1 .00001
C-Salp	No parameters	
For All Algorithms	Population size Maximum number of iterations	25 200

best, the worst and the mean values of the objective function in tables [4 - 11]. Secondly, we compared the convergence curve of all algorithms for all datasets in figures [5-12]. Thirdly, we compared the clustering accuracy of each algorithm in table 14. Finally, we compared the standard deviation of the best score obtained for each dataset from all the experiments in table 13.

From the Tables. [4 -11], we can observe that all the proposed algorithms achieve superior performance than the classical k-means algorithm (in terms of clustering compactness) except for the Libras dataset. According to the best metric, the C-Crow produced the minimum objective value (SSE value) for most of the datasets except for the CMC , Ionospher and Libras datasets. For CMC, The C-grasshopper produced better objective value than the others. For Ionosphere, The C-Salp algorithm produced better objective value than the others. According to the worst metric, the C-Crow produced values less worst than the others for most of the datasets except for the Iris, Haberman and libras datasets. For Iris, Haberman and Libras datasets, the C-Salp produced values less worst than the others. Table. 12 ranks the algorithms based on the mean values of the objective function. According to the ranking, we found that C-Crow occupies the first place in the optimization performance comparison for all datasets and C-Salp occupies the second place but the classical K-Means occupies the last place.

Table 4 : Wine Objective Values

Algorithm	*Best*	*Worst*	*Mean*
K-means	16555.68	18436.95	17276.8
C-Salp	16299.53	<u>16312.76</u>	16310.91
C-Crow	<u>16298.13</u>	<u>16312.76</u>	<u>16304.73</u>
C-Whale	16303.27	16321.02	16313.53
C-Grassh	16298.35	<u>16312.76</u>	16309.88

Table 5. Iris Objective Values

Algorithm	*Best*	*Worst*	*Mean*
K-means	97.32592	122.4787	102.3666
C-Salp	96.95239	97.05206	97.02202
C-Crow	96.87586	97.06686	97.0051
C-Whale	96.93948	97.10718	97.0445
C-Grassh	96.95303	97.08699	97.02642

Table 6. CMC Objective Values

Algorithm	*Best*	*Worst*	*Mean*
K-means	5542.182	5545.333	5543.443
C-Salp	5539.486	5547.049	5542.255
C-Crow	5539.703	5540.691	5540.181
C-Whale	5539.684	5548.601	5543.21
C-Grassh	5539.244	5547.66	5541.607

Table 7. Ionosphere Objective Values

Algorithm	*Best*	*Worst*	*Mean*
K-means	796.3271	796.4667	796.4248
C-Salp	795.3681	797.1697	795.8413
C-Crow	795.3708	795.5047	795.4036
C-Whale	795.3721	796.138	795.5142
C-Grassh	795.4014	803.9438	796.6722

Table 8. Glass Objective Values

Algorithm	*Best*	*Worst*	*Mean*
K-means	215.6775	253.0302	229.2246
C-Salp	213.5481	222.2202	217.7399
C-Crow	213.117	216.0545	214.6342
C-Whale	213.6116	222.5173	216.3476
C-Grassh	213.4915	222.2789	216.8694

Table 9. Haberman Objective Values

Algorithm	*Best*	*Worst*	*Mean*
K-means	2625.108	3196.592	2684.16
C-Salp	2624.266	2624.827	2624.41
C-Crow	2624.266	2624.84	2624.329
C-Whale	2624.294	2625.108	2624.911
C-Grassh	2624.266	2625.317	2624.928

Table 10. Libras Objective Values

Algorithm	*Best*	*Worst*	*Mean*
K-means	322.3441	339.6949	328.8571
C-Salp	327.6302	337.9027	332.7917
C-Crow	326.7352	339.2793	331.5356
C-Whale	328.9837	344.6736	334.7812
C-Grassh	333.0902	350.9515	341.8811

Table 11. Sonar Objective Values

Algorithm	*Best*	*Worst*	*Mean*
K-means	234.7671	235.2066	235.1192
C-Salp	234.6469	234.912	234.7616
C-Crow	234.5923	234.6821	234.6289
C-Whale	234.6111	234.7411	234.6611
C-Grassh	234.715	235.243	234.933

Table 12. Rank of the Mean values of the objective function

Dataset	*Metric*	*K-means*	*C-Salp*	*C-Crow*	*C-Whale*	*C-Grasshopper*
Wine	**Mean**	17276.7	16310.9	16304.7*	16313.5	16309.87
	Rank	5	3	1	4	2
Iris	**Mean**	102.36	97.02	97.00*	97.04	97.02
	Rank	5	2	1	4	3
CMC	**Mean**	5543.44	5542.25	5540.18*	5543.21	5541.60
	Rank	5	3	1	4	2
Ionosphere	**Mean**	796.42	795.84	795.40*	795.51	796.67
	Rank	4	3	1	2	5
Glass	**Mean**	229.22	217.73	214.63*	216.34	216.86
	Rank	5	4	1	2	3
Haberman	**Mean**	2684.16	2624.41	2624.32*	2624.91	2624.92
	Rank	5	2	1	3	4
Libras	**Mean**	328.85	332.79	331.53*	334.78	341.88
	Rank	1	3	2	4	5
Sonar	**Mean**	235.11	234.76	234.62*	234.66	234.93
	Rank	5	3	1	2	4
Mean Rank		4.37	2.87	1.12	3.12	3.50
Final Rank		5	2	1	3	4

In table 13, the standard deviation of the best score for all experiments is given. From the mean ranking, C-Crow is the best algorithm to minimize the SSE function. C-Crow standard deviation is smaller than the others except for Wine, Iris, Libras . It also tends to zero for most of the datasets except for the Wine and Libras datasets. Therefore we can infer that the results produced by the C-Crow algorithm from each experiment is close to the best result and hence it converges to the optimal solution in most cases . The standard deviation of the k-means algorithm is high for most datasets. This confirms its dependency on its initial centroid locations.

Table 13. SD for all algorithms

Dataset	*K-means*	*C-Salp*	*C-Crow*	*C-Whale*	*C-Grassh*
Wine	935.3	4.02	6.62	4.81	4.88
Iris	10.6	0.02	0.05	0.04	0.05
CMC	1.62	2.38	0.34	3.19	2.53
Ionosphere	0.06	0.53	0.03	0.22	2.57
Glass	15.26	2.91	1.14	2.43	2.58
Haberman	180	0.22	0.18	0.24	0.28
Libras	4.85	3.82	4.42	4.23	4.98
Sonar	0.18	0.09	0.02	0.04	0.18

5.3.2. The convergence curve

The following graphs present a comparison of the convergence speed (i.e. the number of iterations required for each algorithm to converge). For the wine dataset, we found that C-Whale converged at about 10th iterations but the others convergence is slow at about 130 iterations for C-Salp and C-Crow and at about 180 iterations for C-Grasshopper. For the Iris dataset, the convergence of all the algorithms is slow but the fastest one is the C-Grasshopper which converged at about 60th iterations. C-Salp converged at about 75 iterations. C-Whale converged at 100th iterations. C-Crow Converged at about 192th iterations. For the Ionosphere dataset, C-Grasshopper converged very fast and get trapped into a local optimum. C-Salp and C-Whale converged with the same speed at 10th iterations. C-Crow at about 38th iterations. For the CMC dataset, C-Salp, C-Whale, and C-Grasshopper converged almost with the same speed at about 115, 115 and 120 iterations, respectively but C-Crow convergence speed is the slowest at about 190th iterations. For the Glass data, The C-Whale converged faster than the others at about 20th iterations but C-Grasshopper converged at about 100th iteration. C-Salp and C-Crow converged at about 130 iterations. For the Haberman dataset, C-Whale and C-Crow algorithms converged at about 60 iterations. C-Grasshopper and C-Salp converged at about 110th and 115th iterations respectively. For the sonar dataset, C-Grasshopper gets trapped into the local optima and achieve very fast convergence. C-Crow achieve faster convergence than C-Salp and C-Whale, it converged at about 35th iterations but C-Salp converged at about 65th iteration and C-Whale converged at about 100th iterations. For the Libras dataset, C-Grasshopper gets trapped into the local optima and achieved very fast convergence. C-Whale converged at about 80th iterations and C-Salp converged at about 135th iterations. C-Crow convergence speed is very slow at about 190th iteration.

Finally, we can conclude that all the algorithms suffer from slow convergence speed. Although C-Crow produced the best score but it suffers from slower convergence speed than the other algorithms for almost all the datasets except for the Sonar and Haberman datasets. Also, the performance of the C-Grasshopper is not as good as the others since it gets trapped into the local optima for 3 datasets.

5.3.3 Clustering accuracy

In table 14, the clustering accuracies of the k-means and the other bio-inspired clustering algorithms are given. Table 14, shows The C-Crow achieve higher accuracy than K-means algorithm in Wine, Iris, Glass and Sonar datasets but they achieve equal accuracy in the CMC dataset. C-Salp achieves better accuracy than the k-means algorithm in all datasets except for CMC, Ionosphere and Libras datasets. The C-Whale algorithm achieves higher accuracy than k-means in the Iris, Glass and Sonar datasets. The C-Grasshopper achieves better accuracy than k-means in the Wine, Iris, Glass, Haberman and Sonar datasets. Although C-Crow produced the minimum objective function value in most of the datasets, it doesn't achieve higher accuracy than the others in most of them .

Table 14. Best Accuracy for all runs

Dataset	*K-Means*	*C-Salp*	*C-Crow*	*C-Whale*	*C-Grassh*
Wine	70.22	71.91	70.78	50	75.28
Iris	89.33	92.66	91.33	92.66	90.66
CMC	40.12	39.51	40.12	38.01	39.44
Ionosphere	71.22	70.08	68.66	68.66	70.37
Glass	41.12	54.67	54.67	54.2	53.27
Haber-	52.28	51.96	51.96	51.96	53.26
Libras	22.22	10.55	9.72	11.11	11.11
Sonar	54.32	57.69	56.25	56.25	57.21

The experiment reveals that minimizing SSE function in solving the clustering problem does not ensure higher accuracy. If the main goal is to promote the accuracy, more similarity measures that are related to accuracy could be applied as the objective function.

6. Conclusion

In this paper, we presented an integration of the k-Means algorithm with each one of the most recent bio-inspired algorithms to overcome the drawback of the K-means algorithm which is falling in the local optima and to maximize clusters integrity. C-Crow search algorithm, C-

Salp algorithm, C-Whale search algorithm and C-Grasshopper optimization algorithm are proposed and validated over eight datasets. six different evaluation criteria are adopted in this study. These criteria are the best, the worst, and the mean fitness value, the mean rank, the SD, and the Accuracy. The experimental results show that the proposed algorithms outperform the standard K-means algorithm in terms of the best, the worst and the mean fitness value. Moreover, the results show that the C-Crow can significantly enhance the clustering integrity

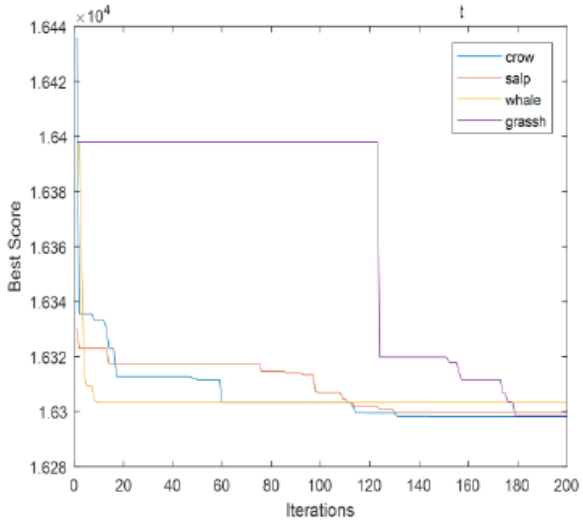

Figure 5. Wine dataset Convergence Curve

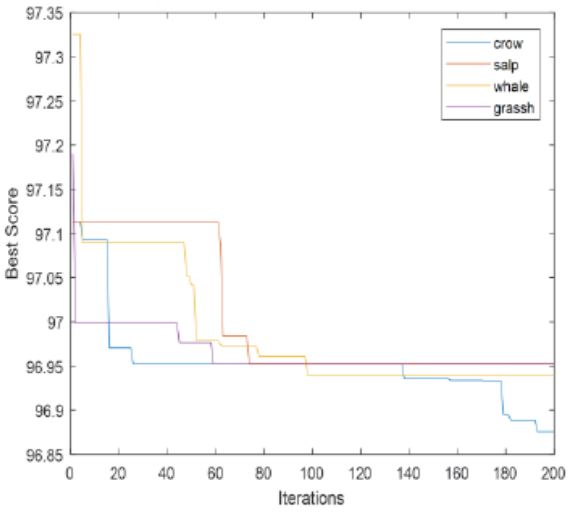

Figure 6. Iris dataset convergence curve

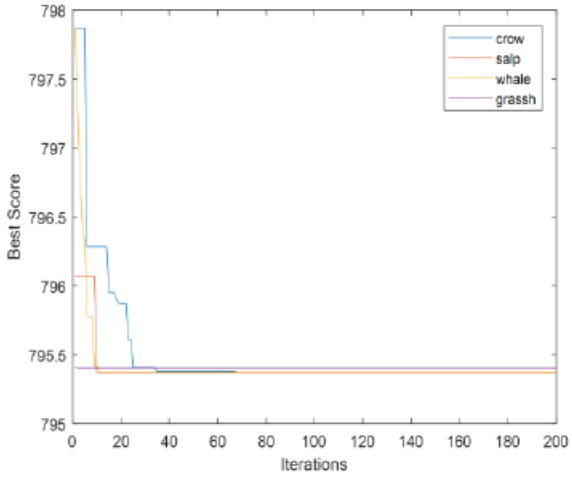

Figure 7. Ionosphere dataset convergence curve

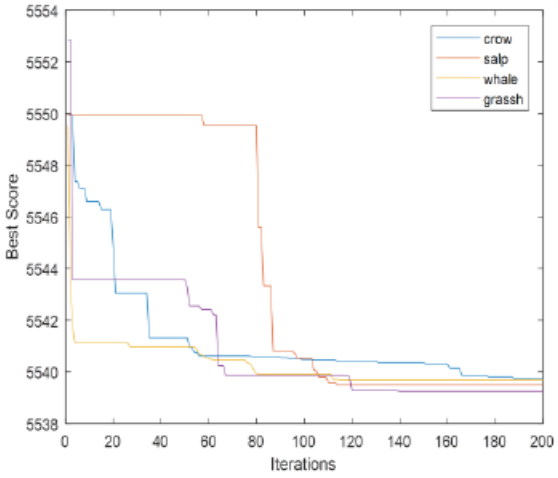

Figure 8. CMC dataset convergence curve

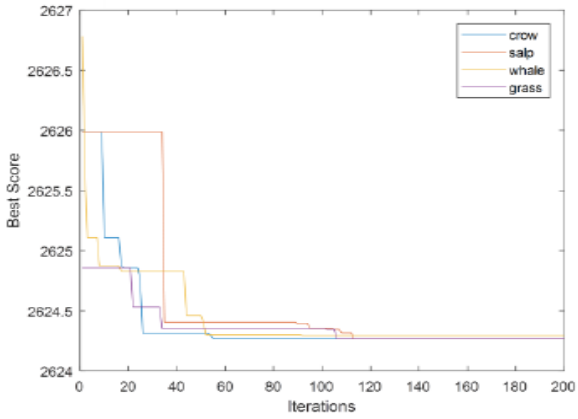

Figure 9. Haberman dataset convergence curve

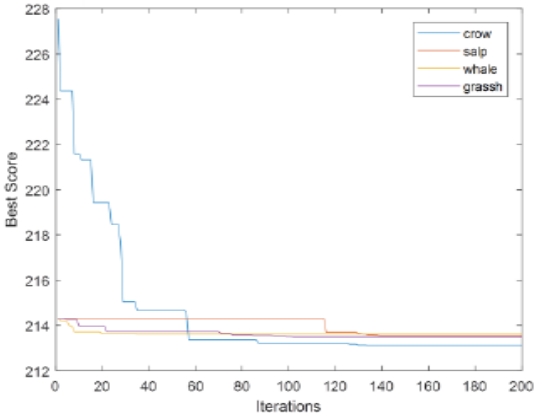

Figure 10. Class dataset convergence curve

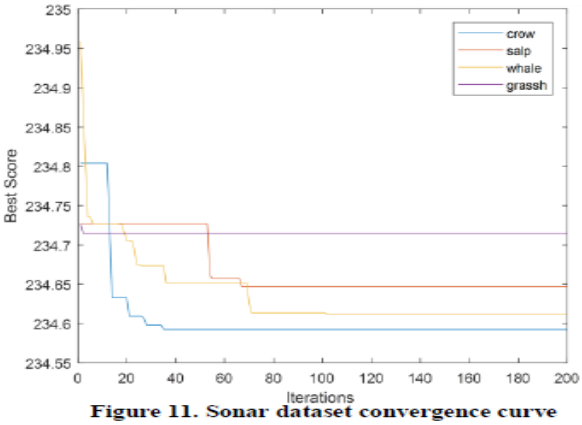

Figure 11. Sonar dataset convergence curve

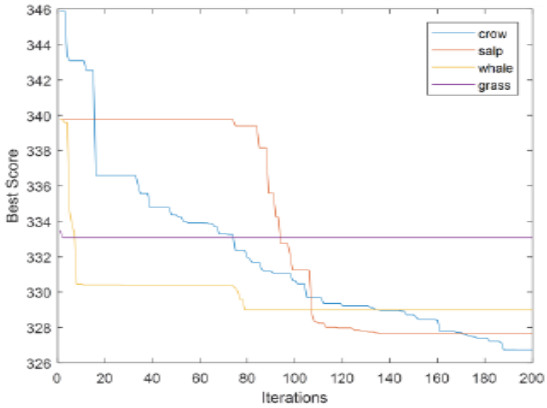

Figure 12. Libras dataset convergence curve

more than the other bio-inspired clustering algorithms. In terms of the convergence speed, all the proposed algorithms suffer from slow convergence in solving the clustering problem but the slowest one is the C-Crow algorithm in almost all the dataset.

7. Future Work

Further acceleration of the convergence speed of the C-Crow, C-Salp, C-Whale, and C-grasshopper will be considered. Also, we will adopt these algorithms with multi-objective clustering to enhance the accuracy of the bio-inspired clustering algorithms. Also, comparing the results of the above-mentioned algorithms with different clustering algorithms like DBSCAN.

References

[1] Askarzadeh, A, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," Computers & Structures, vol. 169, (2016) , pp. 1-12.
[2] Chen, X. ; Zhou, Y. ; Luo, Q.,"A Hybrid Monkey Search Algorithm for Clustering Analysis," The Scientific World Journal, (2014) , p. 16.
[3] Corrêa, G. S., and et al. ,"Combining K-Means and K-Harmonic with Fish School Search Algorithm for data clustering task on graphics processing units," Applied Soft Computing, vol. 41, (2016) , pp. 290-304.
[4] Dua, D. ;Karra Taniskidou, E. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. (2017).
[5] Fong, S. , and et al. ,"Towards Enhancement of Performance of K-Means Clustering Using Nature-Inspired Optimization Algorithms," The Scientific World Journal, vol. 2014, p. 16.
[6] Gu, C. ; Chen, Q. ; Tao, Q. ,"An Improved K-Means Algorithm Combined with Chaotic Particle Swarm Optimization Algorithm," Journal of Information & Computational Science, (2015), p. 12.
[7] Huang, Z. ; Zhou, Y. ,"Using glowworm swarm optimization algorithm for clustering analysis," Journal of Convergence Information Technology 6(2), (2011) , p. 78–85.
[8] Hassanzadeh, T.;Meybodi, M. R.,"A New Hybrid Approach for Data Clustering using Firefly Algorithm and K-means," Proceedings of the CSI International Symposium on Artificial Intelligence and Signal Processing, (2012) , pp. 7-11.
[9] Hatamlou, A. ; Abdullah, S. ; Nezamabadi-pour, H.,"A combined approach for clustering based on K-means and gravitational search algorithms," Swarm and Evolutionary Computation, vol. 6, (2012) , p. 47–52.
[10] Inkaya, T.; Kayalıgil, S.; Özdemirel, N. E. "Swarm Intelligence-Based Clustering Algorithms: A Survey," in Unsupervised Learning Algorithms, Springer, Cham, 30. (2016).
[11] Karaboga, D. ; Ozturk, C.,"A novel clustering approach: Artificial Bee Colony (ABC) algorithm.," Applied Soft Computing, vol. 11, no. 1, (2011) , p. 652–657.
[12] Kwedlo, W., "A clustering method combining differential evolution with the K-means algorithm," Pattern Recognition Letters, vol. 32, no. 12, (2011), p. 1613–1621.
[13] Liu, Y. ; Wu, X. ; Shen, Y. "Cat swarm optimization clustering (KSACSOC): A cat swarm optimization clustering algorithm," Scientific Research and Essays, vol. 7, no. 49, (2012) ,pp. 4176-4185.
[14] Li, Q. ; Liu, B., "Clustering using an Improved Krill Herd Algorithm," Algorithms, vol. 10, no. 2 , (2017).
[15] Mirjalili, S. ; Lewis, A."The Whale Optimization Algorithm," Advances in Engineering Software, vol. 95, (2016), pp. 51-67.
[16] Mirjalili, S., and et al. "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems," Advances in Engineering Software, vol. 114, (2017) ,pp. 163-191.
[17] Shah-Hosseini, H."Improving K-means clustering algorithm with the intelligent water drops (IWD) algorithm," International Journal of Data Mining, Modeling and Management, vol. 5, no. 4, (2013), pp. 301-317.
[18] Saida, I. B. ; Nadjet, K. ; Omar, B. "A New Algorithm for Data Clustering Based on Cuckoo Search Optimization," Genetic and Evolutionary Computing, vol. 238, (2014) , p. 55–64.
[19] Saremi, S. ; Mirjalili, A. ; Lewisa, A."Grasshopper Optimization Algorithm: Theory and application," Advances in Engineering Software, vol. 105, (2017) ,pp. 30-47.
[20] Tang, R., and et al. ,"Integrating nature inspired optimization algorithms," in Proceedings of the 7th International Conference on Digital Information Management (ICDIM '12) (2012).
[21] Yang, X. S., and et al." Swarm Intelligence and Bio-Inspired Computation: Theory and Applications, Elsevier Science Publishers B. V. Amsterdam, The Netherlands, (2013) .

Authors -

First Author Eng. Doaa Abdullah Abdel-Mohsen is a teaching assistant in Computer Science department, Faculty of Computers and Information, Helwan University, Cairo, Egypt. She holds a bachelor in Computer Science with honors degree.

Second Author Dr. Hala Abdel-Galil is associate professor of Computer Science, and head of the Computer Science Department, Faculty of Computers and information, Helwan University, Cairo, Egypt. She has skills and expertise in Image Processing, Pattern Recognition, Classification, Neural Networks and Artificial Intelligence, Computational Intelligence, Pattern Classification, Applied Artificial Intelligence and Machine Intelligence.

Third Author Dr. Ensaf Hussein Mohamed received her Ph.D. in Computer Science, Faculty of Computers and Information, Helwan University, Cairo, Egypt, 2013. Her recent Research focuses on Natural Language Processing, Text Mining, and Machine Learning. Currently, she is an assistant professor, Faculty of Computers and Information, Helwan University, Cairo, Egypt.