EAI Endorsed Transactions
on Context-aware Systems and Applications
Research Article EAI.EU

Automatic Data Clustering using Dynamic Crow Search Algorithm

Rajesh Ranjan* and Jitender Kumar Chhabra

Computer Science and Enginnering Department,
National Institute of Technology, Kurukshetra, Haryana, 136119 India

Abstract

This work proposes Automatic clustering using Dynamic Crow Search Algorithm, which updates its parameters dynamically. Crow Search is a recently proposed algorithm that imitates the working of crow. Clustering is an essential aspect of data analysis whose significance has increased manifold since the advancements of technology which has led to enormous data generation, which need to be analysed in real-time. Automatic clustering detects optimal cluster numbers and produces sustainable cluster centroids. ACDCSA uses Cluster Validity using Nearest Neighbour as an internal validity measure that acts as a fitness function to find the optimal cluster centres. The present work is compared with some well-known other meta-heuristic search algorithms like PSO, DE, WOA and GWO for the automatic clustering task over seven benchmark clustering datasets. Inter-cluster distance, intra-cluster distance and the optimal cluster number produced are used to assess the performance of ACDCSA.

Received on 12 March 2022; accepted on 13 May 2022; published on 17 May 2022

Keywords: CVNN, Data Clustering, Meta-heuristic Search Algorithm

doi:10.4108/eai.17-5-2022.173982

1. Introduction

Data clustering is defined as the segregation of data points having the same traits that are placed together in a cluster such that intra-cluster compactness within a cluster and inter-cluster sequestration between the different clusters should be optimal [1]. Data clustering groups together data points having similar traits, and these grouped data are different from other data points. It minimises the distance between the data points and the cluster centres within a cluster. Being an active area of research as it has widespread use in various scientific and research works. Various application areas such as data mining, bioinformatics, image analysis, satellite data, and real-life applications require unsupervised data clustering before actual data analysis [2]. In literature, the clustering task is widely grouped into two types 1. Hierarchical Clustering and 2. Partitional clustering [3].

Hierarchical clustering clusters the data by creating a tree-like structure called a dendrogram, where it tries to find the cluster number through the dendrogram and groups the data based on similar traits. Various authors have proposed several partition-based algorithms. Among them, the prominent one is the K-means clustering algorithm. Being easy to apply for different data types, it is the most favoured clustering algorithm. However, it has certain imperfections, such as its convergence towards local minima and the number of clusters, i.e., are known a-prior [4]. Clustering the data when the data-label is not known a-prior comes under unsupervised machine learning task. Most of the data obtained from various fields usually do not have data labels. These unlabelled data must first be clustered to analyse and find the hidden pattern. In such a scenario, automatic clustering plays a vital role. Automatic clustering is the technique where the actual cluster number or the data labels needed to be clustered is unknown. Usually, the real-life data sets are very complex and large, which require to be grouped in small clusters. So, considering the present scenario need for automatic clustering is becoming inevitable.

Different nature-inspired algorithms have solved several optimisation problems, widely known as

*Corresponding author. Email: iiitm.rajesh@gmail.com

Meta-heuristic search-based algorithms. These nature-inspired algorithms find optimal optimisation solutions while balancing global expedition and intense local exploration of the space search [5]. Since automatic clustering simultaneously tries to find cluster numbers and their cluster centroid, it also comes under an optimisation problem that tries to solve two objectives at a time. Several researchers have proposed and implemented these meta-heuristic search-based algorithms for the automatic clustering problem. Among them, the prominent evolution-based algorithms are Genetic Algorithm (GA) [6], Differential Evolution (DE) [7], most applied swarm-based algorithms are Particle Swarm Optimization (PSO) [8], Grey-Wolf Optimization (GWO) [9], Firefly Algorithm (FA) [10], Whale Optimization Algorithm (WOA) [11], and Physics-based algorithms are Gravitational Search Algorithm (GSA) and Harmony Search Algorithm (HS) [12]. These nature-inspired algorithms use a partition-based approach to solve the clustering problem, using internal cluster validity indexes (CVI) as the optimisation function for evaluation. The prominent CVI's proposed by different researchers are Dunn's Index, Davies-Bouldin Index, CS-Index, Silhouette Index, S_Dbw Index [13], and Cluster Validity Index using Nearest Neighbour (CVNN) [14].

In the present work, a modified Crow Search Algorithm is used as a clustering algorithm to solve the automatic clustering problem. CVNN is used as an optimising function to simultaneously find the optimal number of clusters and their best centroid. Crow Search Algorithm is a recently proposed nature-inspired search-based algorithm that mimics crow working [15]. In this work, an extension of crow search algorithm that dynamically updates its awareness probability and flight length based on the result obtained from the fitness value has been used for the automatic clustering task.

The rest of the paper is in the following sequences. A brief literature review of the automatic data clustering using metaheuristic search algorithms is presented in section-2. Crow Search Algorithm and its main drawbacks are discussed in section-3. The suggested improvement in the CSA and modified Dynamic Crow Search Algorithm is discussed in section-4. Section-5 presents the implementation of Automatic clustering using the Dynamic Crow Search Algorithm using a hybrid approach, describing the dataset used for data clustering and results obtained after simulation. It also compares the obtained results with other algorithms. Finally, section-6 concludes the work and proposes future work to extend the present work.

2. Related works

Several researchers have proposed different heuristic and meta-heuristic algorithms for clustering when considering the automatic clustering problem. Further, different researchers have applied many efforts in hybridizing two or more different algorithms for automatic clustering tasks. One of the studies shows that these nature-inspired algorithms have been applied extensively by several researchers to solve automatic clustering tasks [16]. Some of the meta-heuristic search-based algorithms used for solving clustering problems are discussed in the following section. Some of the significant works in automatic clustering are carried out using PSO, either fine-tuning its parameters or hybridizing it with other nature-inspired algorithms. Merwe et al. combined the PSO and K-means and used them for data clustering and hence paved the path of data clustering using the swarm-based algorithm. However, it failed to find the cluster number for unlabelled data [17]. Omran et al. [18] proposed Dynamic Clustering using PSO, used in image segmentation to cluster the image data automatically. Abraham et al. [19] proposed kernel MEPSO for automatic complex data clustering. Instead of using standard Euclidean distance measure in Cluster Validity Index, they induced Kernel-based similarity measure, and the obtained results were superior to the compared work. Recently, Alswaitti et al. [20] have proposed DPSO, a dynamic clustering algorithm, which solves the premature convergence of PSO and balances its intensified local search and diverse global search by combining a kernel-based density estimation technique. DPSO used Dunn's Index as a CVI to judge the robustness of the obtained result. Gao et al. [21] proposed a hybrid PSO-K-means algorithm, which uses the hybrid initialization technique using the K-means algorithm, and apply Lévy flight-based position update to avoid getting trapped into local minima. Sharma and Chhabra, in 2019, proposed AHPSOM, which uses a mutation operator into a hybrid PSO algorithm for solving automatic data clustering problems. AHPSOM is mainly applied for the continuously generated data, usually from different networks, having dynamic and heterogeneous features, with unknown cluster numbers [22]. Amol et al. [23] proposed the hybridized grey wolf optimizer with a whale optimization algorithm, each having a different hunting style to catch its prey and further applying it to the data clustering domain. The proposed work uses inter-cluster distance, intra-cluster distance and cluster density-based fitness measures to find the optimal centroid for the automatic clustering task. Ashish, in 2018 has proposed (MR-EGWO) which applied grey-wolf optimizer in the big-data environment using Map-reduce algorithm for clustering large-scale data sets in which the grey wolf is hybridized with binomial

crossover and Lévy flight-based searching is applied to elevate the searching capability [24]. Ibrahim et al. hybridized GWO with trajectory-based search algorithm TS for clustering to intensify the effectiveness and balance between exploring and exploiting the GWO algorithm. In this hybridized work, TS is used as an operator for GWO, which helps it find the leader's neighbourhood, thus giving stress to more localized search in cases with high chances of finding the solution [25]. Kuo et al. [26] proposed iABC, which combines the ABC with the k-means algorithm, where k-means help find the better initial centroid and thus direct the bees to better positions during further iterations. Here the K-means algorithm gives the initial centroid, which is used by the onlooker key to finding an optimized location nearby to the initial centroid. The algorithm is applied to real-life customer segmentation problems. Hussain et al. [27] proposed an ABC optimization-based algorithm for clustering large datasets having higher dimensions. The proposed method incorporates aspects of co-clustering by the ABC algorithm. Instead of using Euclidean distance in this work, the author has applied higher-order correlations to find the result. Also, the search space is explored in three different ways to have a better diversification result. Finally, the proposed work has shown scalability to parallel architecture in shared memory and distributed environments. Kumar et al. has proposed and implemented Gravitational Search Algorithm for automatic clustering problem and further applied it to image segmentation. The proposed work is known as ACGSA. The method used variable chromosome representation for cluster centroid encoding, and further weighted cluster centroids were applied to get the best centroid. The authors have introduced a new fitness function to achieve better and more stable cluster centroids [**?**]. Kazem et al. implemented a variant of harmony search algorithm (HS), called best–worst-mean harmony search for data clustering; it employs an enhanced memory consideration method to efficiently employ the collected insight and experience in harmony memory [28]. Tseng and Yang et al. has used, Genetic Algorithm for automatic data clustering problem, well known as CLUSTERING. It clustered the data at three levels to obtain the final clustering output, outperforming other algorithms used for comparison [29]. Vovan et al. proposed an Automatic Clustering for interval data using a Genetic Algorithm. The overlapped distance within data intervals helps determine the optimal clusters; the proposed algorithm has applications like clustering data with different characteristics and recognizing the images [30]. Das et al. proposed ACDE, Automatic Clustering using Differential Evolution for clustering unlabelled data, which, apart from standard data set, gave better results for high-dimensional data. Further to demonstrate the effectiveness of the present work, two cluster validity index CS Index and Davies Bouldin Index, is used to find the appropriate number of clusters and their centroids [7]. Chen et al. [31] implemented an elastic-differential evolution algorithm for automatic data clustering by adopting a variable particle encoding scheme where the population consists of changeable-length parameter vectors, each denoting a different number of clusters. Also, the mutation and crossover operators are designed accordingly.

3. Experimental Method

3.1. Crow Search Algorithm

Crow Search Algorithm is a recently proposed nature-inspired algorithm motivated by the crow's nature, considered one of the most intelligent species. Crows have an exceptional memory and searching ability, allowing them to recognize their food hideout and, lit also follows other crows to plunder their food storage at their hiding place. If the crow is aware of the follower, it changes its hideout place to a random position to deceive the follower crow.

3.2. Standard CSA Algorithm

In standard CSA, a group of N numbers of crow search in n-dimensional search space. $M_{j,\ iter}$ denotes the hiding location of j^{th} crow during iteration *'iter'*. It is the best location searched by j^{th} crow up to *'iter'* number of iterations. Further in the successive iterations, if the best location improves, this memory and the position also improve. Now, if crow *'j'* visits its hideout location without knowing that it is being chased by crow *'i'*, *(i.e.,* $r_j > ap_{j,\ iter}$*)* in this situation, CASE-A occurs, and new position of crow *'i'* is given by,

$$X_{i,\ iter+1} = X_{i,\ iter} + r_i \times fl_{i,\ iter} \times \left(M_{j,\ iter} - X_{i,\ iter}\right) \quad (1)$$

Where r_i and r_j are random numbers lying between [0,1] and $fl_{i,\ iter}$ represents the flight length of crow *'i'* during the iteration *'iter'*. Flight length is an effective parameter that decides the searching capability of the crow. Suppose flight length is less than one *(i.e., 'fl<1')*, it directs to intensified local searching, and if flight length is greater than one *(i.e., 'fl>1')*, it guides to global searching, *i.e.*, searching at some random position. Awareness probability is another parameter of the CSA apart from the flight length *'fl'*, here $ap_{j,\ iter}$ represents the awareness probability of crow *'j'*at iteration *'iter'*. Suppose crow *'j'* is aware that another crow is chasing it; it goes to any random position to misguide the crow *'i'*. In this scenario, CASE-B occurs, which is given by,

$$X_{i,\ iter+1} = (u - l) \times Random\ Number + l \quad (2)$$

where, *'u'* and *'l'*denotes the lower and upper boundary limits. After generating the new locations either

through CASE A or CASE B, the crows in the flock update their memory and positions depending upon the fitness function (in case of minimization problem) as follows:

$$M_{i,\ iter+1} = \begin{cases} X_{i,\ iter+1} & if,\ Z(X_{i,\ iter+1}) < Z(M_{i,\ iter}) \\ M_{i,\ iter} & Otherwise \end{cases} \quad (3)$$

This procedure repeats itself till the maximum iteration $iter_{max}$ is reached.

3.3. Drawbacks in Standard CSA

Standard CSA has only two parameters, which helps the CSA maintain a balance between intensification over local search and exploring unique positions for global search. In standard CSA, both these parameters remain constant, due to which the solution space is not fully explored for the optimum solution. We have addressed these shortcomings of the CSA Dynamic Crow Search Algorithm (DCSA) [32], that solves the problem mentioned above and dynamically changes the parameters to solve complex problems like data clustering. We have further extended the DCSA to solve the automatic clustering problem.

3.4. Dynamic Crow Search Algorithm

DCSA updates its flight length and awareness probability according to the results obtained from the fitness function used. A ranking system based on the obtained results helps find the best and worst crow during each iteration. Besides this, memory-based ranking is also performed to find the crow that has fetched the best memory. These results help to fine-tune the parameters of the original crow search algorithm, updating the awareness probability and flight length according to the rank-based value obtained from the fitness function used for optimizing the clustering task. Also, instead of going to any random position in Case-B, it uses Lévy-flight based position update in the DCSA. In the improved version of DCSA, the authors have used the hybrid approach to update the crow's positions. In Dynamic Crow Search Algorithm, the significant changes that the authors have suggested are:

• In DCSA awareness probability 'ap,' which depends upon the value obtained from the objective function has been used.

$$ap_{j,\ iter} = c1 \times \left(\frac{Rank_{i,\ iter}}{N}\right) + c2 \times \left(\frac{Rank_{j,\ Memory}}{N}\right) \quad (4)$$

where, $c1\ and\ c2 \in (0.1,\ 0.4)$, and $Rank_{i,\ iter}$ is the rank obtained by i^{th} crow during $iteration$='$iter$' and $Rank_{j,\ memory}$ is the rank of the j^{th} crow's memory which is being chased by the i^{th} crow.

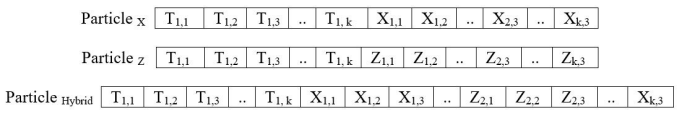


Figure 1. Hybrid Encoding Scheme.

• The flight length '*fl*', used in the DCSA is also dynamic, whose value is static in the standard CSA.

$$if\left(\frac{Rank_{i,\ iter}}{N} < \frac{Rank_{j,\ Memory}}{N}\right):$$

$$fl_{i,\ iter} = c3 + 1.0 \times \left(\frac{Rank_{i,\ iter}}{N}\right) \quad (5)$$

$$else:$$

$$fl_{i,\ iter} = c4 + 1.0 \times \left(1 - \frac{Rank_{j,\ Memory}}{N}\right) \quad (6)$$

$where,\ c3\ and\ c4 \in (0.5,\ 1)$

• Lévy-flight based position updating, is incorporated in the DCSA.

$$X_{i,\ iter+1} = X_{i,\ iter} + r_i \times Lévy(d) \times (M_{j,\ iter} - Hybrid_{i,\ iter}) \quad (7)$$

4. Proposed Work

In the present work we have extended the Dynamic Crow Search Algorithm by using a two-point hybrid of the best two solutions. In the hybrid encoding scheme, one third (i.e.,$\frac{1}{3}$) of the solution space encoding is taken from the best solution, next one third (i.e.,$\frac{1}{3}$) is taken from 2^{nd} best solution, and finally, the last one third (i.e.,$\frac{1}{3}$) elements of the particles are again taken from the best solution. Also, the best position and 2nd best position keep changing after each iteration to maintain the diversification. The above discussed implementation of hybrid encoding scheme is depicted in Figure-1. Here best solutions are represented by $Particle_X$ and $Particle_Z$. $Particle_{Hybrid}$ is hybrid of $Particle_X$ and $Particle_Z$.

4.1. Automatic Data Clustering

Automatic clustering groups together similar data within a dataset when the data labels or cluster numbers are unknown. So basically, it is a multi-objective optimization process in which it simultaneously detects the number of clusters formed along with their best. The various researchers have given several internal cluster validity indices to find the optimal cluster centres; among them, the major validity indices are Dunn Index, Davies Bouldin Index, CH-Index, Xie-Beni index, CS-Index. The present work uses the CVNN index as a cluster validity index. CVNN index uses the

Algorithm 1: Dynamic Crow Search Algorithm

Input : A Dataset D and *Objective Function*
Output : *Optimal Solution*
Parameter: $iter_{max}$: *Maximum iterations*
N: *Number of crows*
Initial : $iter = 0$
Crow's Memory = Initial Position
Evaluate Crow's Fitness using Objective Function

while $iter < iter_{max}$ **do**
  for $i \leftarrow 1$ **to** N **do**
    // Randomly select any crow to follow (let's say j^{th} crow)
    $Rank_{i,\ iter} = i, where, i = 1, 2..N.$
    $Rank_{j,\ Memory} = j, where, j = 1.2..N.$
    $R_{i,\ iter} = \frac{Rank_{i,\ iter}}{N}$
    $R_{j,\ iter} = \frac{Rank_{j,\ Memory}}{N}$
    $ap_{j,\ iter} = c1 \times R_{i,\ iter} + c2 \times R_{j,\ iter}$
    if $(R_{i,\ iter} < R_{j,\ iter})$ **then**
      $fl_{i,\ iter} = c3 + 1.0 \times R_{i,\ iter}$
    end
    else
      $fl_{i,\ iter} = c4 + 1.0 \times (1.0\ -\ R_{j,\ iter})$
    end
    if $(random\ number > ap_{j,\ iter})$ **then**
      $X_{i,\ iter+1} = X_{i,\ iter} + r_i \times fl_{i,\ iter} \times \left(M_{j,\ iter} - X_{i,\ iter}\right)$
    end
    else
      $X_{i,\ iter+1} = X_{i,\ iter} + r_i \times Lévy(d) \times (M_{j,\ iter} - Hybrid_{i,\ iter})$
    end
  end
  if $(Z(X_{i,\ iter+1}) < Z(M_{i,\ iter}))$ **then**
    $M_{i,\ iter+1} = X_{i,\ iter+1}$
  end
  else
    $M_{i,\ iter+1} = M_{i,\ iter}$
  end
  $iter = iter + 1$
end
Return the best solution in terms of optimal memory

Algorithm 2: Automatic Clustering using DCSA

Input : *DCSA, Dataset D*
Output : *Clustered Data with optimal Cluster Number*
Parameter: $iter_{max}$:
Maximum number of iterations
K_{max}:10
K_{min}:2
Threshold Value:0.5
N: *Number of crows*

Encode and initialize each particle with K cluster centers where $K \in (K_{max}, K_{min})$
Encode the activation threshold of particles between $(0, 1)$
while $iter < iter_{max}$ **do**
  a. *Select cluster centres of the particle whose activation value is greater than* 0.5.
  b. *Calculate the distance between data point to the active cluster center calculated in Step* 1.
  c. *Assign the data points to that cluster centeres having minimum distance.*
  d. *Reinitialize the clusters if any of the clusters have less than* 3 *data points.*
  e. *Update the clusters using DCSA, where CVNN is used as the objective function.*
  f. *Check the boundary condition for initial threshold values and cluster centroid values.*
end

Nearest Neighbours concept given by [14]. As suggested by [33], a slightly modified version of CVNN is given in the present work.

4.2. Solution Space Encoding

A string having the size $K_{max} + K_{max} \times d$ represents each particle where, max is the maximum number of clusters chosen, i.e., in this case, $K_{max} = 10$ and $d = no.\ of\ attributes$ presents in a particular data set. The initial K_{max} of solution space encoding represents the threshold values having a range [0,1]. In this work, as most researchers suggested, the threshold value is chosen as 0.5. If the value is more significant than 0.5 for i^{th} position from initial K_{max}, the corresponding centre will be selected for clustering; otherwise, it will be rejected. Now, in this work, stress has been given that during initialization, each K value from [2,10] will get an equal number of chances to participate in the clustering task. For example, if the number of particles taken is 27, then for each K value from [2,10], three particles will be assigned for each K. K_{max} data points from the data set without replacement are selected for each particle during initialization. Now depending upon the threshold value of the initial K_{max} position from the solution space encoding, these data points get activated and participate in the clustering task. In this work, Euclidean distance measure is used as a distance measurement. Initially, for the selected centroid values of each particle, Euclidean distance is measured for all the data points from the data set. The data point having a minimum distance from a particular centroid value is assigned to that centroid. While calculating the distance, it may be possible that some of the clusters may have two or less than two elements. In such cases, reinitialization of such particles is done such that none of the active centroids has less than or equal two elements.

4.3. Internal Cluster Validity Index

In the domain of automatic clustering, the cluster internal validity measure plays a dominant role. It helps detect the number of clusters that could be possibly present in the given data set. The optimal value (i.e., maximum or minimum) of the chosen internal cluster validity measure over a set of $k \in (K_{min}, K_{max})$ decides the best cluster number for the given dataset. The Internal cluster validity mostly depends upon two properties:

a. Compactness. It measures the cohesiveness of the data points present within a cluster. Several measures are based on the distance, which estimates the compactness of the given cluster, such as maximum or average pairwise distance in a cluster or maximum or average centre-based distance.

b. Separation. It measures how different one cluster is from the other cluster. Here again, the distance measures play a crucial role in deciding the dissimilarity between the two clusters. For example, a pairwise minimum distance between data points in different clusters or between the centres of two clusters is widely used to measure separation.

The most prominent internal cluster measures used by researchers are Dunn's index (DI), Calinski-Harabasz Index (CH Index), Davies-Bouldin Index (DB Index), Silhouette Index (S), Compact-Separated Measure (CS Measure) and I-index [13].

4.4. Clustering Validation Index Based on Nearest Neighbours (CVNN)

Liu et al. [14] have proposed CVNN in which Nearest neighbour-based separation instead of directly distance-based separation measurement between the clusters in a given data set has been used. In CVNN, the separation part is added to a straightforward compactness measure based on dissimilarity values. To properly balance these two measures against each other's, the author proposes to divide both of them by their maximum over clustering with different numbers of clusters $K = \{K_{min}, .., K_{max}\}$. The aim is to find the best number of groups given a clustering method.

$$Sep_n(C_K) = \max_{\{j=1,\ldots,K\}} \left(\frac{1}{n_j} \sum_{x \in C_j} \frac{q_n(x)}{n} \right) \tag{8}$$

Where, $q_n(x)$ is the number data points among the 'n' nearest neighbors of x that are not in the same cluster. n_j is the number of data points in j^{th} cluster. The compactness statistic is just the average within-cluster dissimilarity,

$$Com(C_K) = \frac{\sum_{j=1}^{K} \sum_{x_h \neq x_i \in c_j} d(x_h, x_i)}{\sum_{j=1}^{K} n_j \left(n_j - 1\right)} \tag{9}$$

$$CVNN_n(C_K) = \frac{Sep_n(C_K)}{max_{C \in K} Sep_n(C)} + \frac{Com(C_K)}{max_{C \in K} Com(C)} \tag{10}$$

A lower value of both the separation and compactness gives better results.

4.5. Performance Metrics

The following performance metrics have been used in the present work to evaluate the performance of DCSA for data clustering problem:

a. Objective Function- In this work, Cluster Validity using Nearest Neighbour is used as an objective function. A lower value of CVNN represents a better cluster. The number of nearest neighbours as required by CVNN is taken as ten except in the case of the Wine dataset, where five nearest neighbours have been taken.

b. Optimal Cluster Number- As in automatic clustering, the essential task is to find the optimal cluster number, so it is also used as a performance metric to find the accuracy of the algorithm in finding the optimal number of clusters in particular datasets.

c. Intra-cluster compactness- It finds the cohesion between data points in the cluster formed after applying clustering algorithms. A lower value of cohesion denotes a better cluster formed. The Sum of squared error (SSE) is a commonly used measure to find the Intra-cluster distance. Mathematically it is given as:

$$Intra - Cluster\ Distance = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|c_i - Xj\| \tag{11}$$

where, k denotes optimal number of clusters obtained and c_i denotes centroid of cluster C_i.

d. Inter-cluster separation- It finds the separation between different cluster centres. A higher value of separation denotes a better result. Mathematically it is the distance of cluster centroids from the mean of the dataset used for clustering. It is given as:

$$Inter - Cluster\ Distance = \sum_{i=1}^{k} \|X_{mean} - c_i\| \tag{12}$$

where, k represents optimal number of clusters obtained, X_{mean} denotes the mean of the dataset used for clustering and c_i represents cluster centroid of cluster C_i.

5. Results and Discussion

5.1. ACDCSA Experimental Results

In this section, implementation and obtained experimental results of DCSA for automatic clustering are discussed. DCSA is implemented via Python 3.6 using ubuntu 20.04 operating system having i7 processor and 16 GB RAM.

Table 1. Dataset used for Automatic Clustering

Name	*Instances*	*Attributes*	*Classes*	*Neighbour*
Iris	150	4	3	10
CMC	1473	9	3	10
Cancer	683	9	2	10
Seed	210	7	3	10
Thyroid	215	5	3	10
Wine	178	13	3	5
Vowel	871	3	6	10

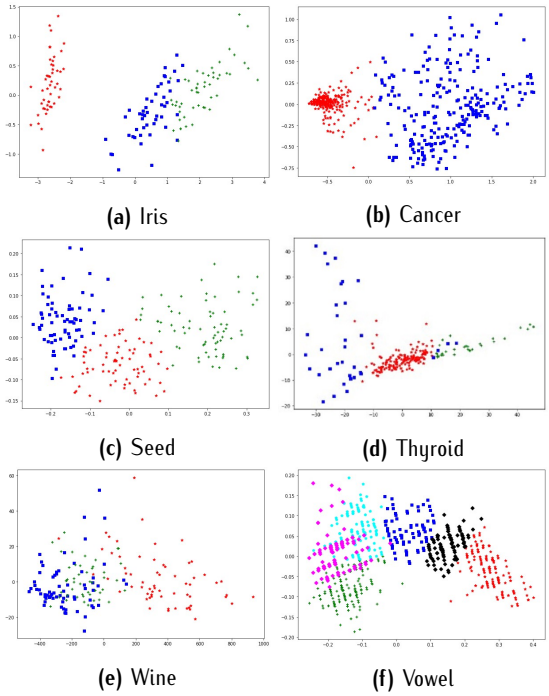
5.2. Dataset Used

To analyse the performance of the DCSA over automatic data clustering, the author has experimented with seven standard datasets obtained from the UCI repository. Datasets details are given in Table 1.

5.3. Simulation Result

Further, the results obtained from ACDCSA are compared with four other search algorithms such as GWO, PSO, WOA and DE that were previously applied for the automatic data clustering problem. Furthermore, CVNN, which is discussed in the methodology section, has been used as a fitness function while simulating ACDCSA and other algorithms used for comparison. The parameters used for automatic clustering by different metaheuristic search algorithms are given in Table 2.

Table 3, 4 and 5 represent the results obtained after simulating ACDCSA and the other four algorithms for the automatic data clustering problem. Table 3 represents the mean of an optimal number of clusters obtained and their standard deviation. From the results present in the table, it is clear that ACDCSA has produced better results than other algorithms. The number of clusters produced by ACDCSA is nearly equal to the number of clusters present in those particular datasets. Table 4 and Table 5 represent the mean and standard deviations of intra-cluster and inter-cluster distance. In the case of intra-cluster distance, a minimum obtained value represents better cluster formed, whereas, in the case of inter-cluster distance, a maximum value obtained signifies better cluster formed. It is clear from Table 4 and Table 5 that ACDCSA has produced better clusters in terms of intra-cluster and inter-cluster calculations. Figure-2 represents clustering results obtained using ACDCSA using CVNN as a cluster validity index. The figure shows that the algorithm has produced excellent and robust clusters.



(a) Iris **(b)** Cancer **(c)** Seed **(d)** Thyroid **(e)** Wine **(f)** Vowel

Figure 2. Clustering results produced by ACDCSA

6. Conclusion and Future Work

In the present work, an automatic clustering task is implemented using Dynamic Crow Search Algorithm, and the obtained results were compared with other nature-inspired algorithms over inter-cluster, intra-cluster distance and optimal clusters obtained. The performance of ACDCSA is better than different algorithms used for comparison.

The present work can be extended to automatic document and image clustering, as most of the data produced are in the form of text and images. Further, ACDCSA can be extended to automatic feature selection and clustering simultaneously. ACDCSA can be used to cluster over the live stream and spatiotemporal data and improve the above algorithm for a distributed environment. In today's digital era., these data are produced extensively. Applying clustering techniques over these data gives insight into other data-analysis fields. Apart from the data science domain, the proposed work can be further applied to various other disciplines where the main task is problem optimization.

Further, ACDCSA can be tested and compared with other internal cluster validity indexes, which multiple researchers have already proposed. We can say that ACDCSA has multiple scopes in the data-science domain and further research and real-life optimization problems.

Table 2. Parameters of different algorithms used for Simulation of Data Clustering

Algorithms	*Parameters : Values*	*Algorithms*	*Parameters : Values*
GWO	Flock Size: 27, Iterations: 200 Amax: 2, Amin: 0	WOA	Flock Size: 27, Iterations: 200 Amax, Amin: 2, 0 , b: 1
PSO	Flock Size: 27, Iterations: 200 Omega1, Omega2: 1.1, 1.5 Inertial weight: 0.7-0.4		A2max, A2min: -1, -2 Lmax, Lmin: 1, -1
DE	Flock Size: 27, Iterations: 200 Crossover probability: 0.8, Mutation: 0.3	ACDCSA	Flock Size:27, Iterations:200 AP Constants (c1, c2): (0.2,0.1)

Table 3. Average [Standard Deviation] of Number of Clusters generated by ACDCSA

Algorithms	*IRIS*	*CMC*	*CANCER*	*SEED*	*THYROID*	*WINE*	*VOWEL*
GWO	2.85 [0.366]	3.3 [0.470]	2 [0]	3 [0]	3.1 [0.447]	2.3 [0.470]	4.6 [1.142]
PSO	3.25 [0.550]	3.3 [0.47]	2 [0]	3 [0]	3.8 [0.767]	2.75 [0.444]	4.95 [1.05]
WOA	3.1 [0.307]	3.45 [0.604]	2 [0]	2.75 [0.444]	2.7 [0.656]	2.55 [0.510]	4.9 [1.165]
DE	3 [0]	3.7 [0.470]	2 [0]	3 [0]	3.45 [0.510]	2.9 [0.307]	5.2 [0.767]
ACDCSA	3 [0]	3.3 [0.470]	2 [0]	3 [0]	2.9 [0.307]	2.95 [0.223]	5.2 [0.951]

Table 4. Average [Standard Deviation] Intra-cluster distance generated by ACDCSA

Algorithms	*IRIS*	*CMC*	*CANCER*	*SEED*	*THYROID*	*WINE*	*VOWEL*
GWO	1.8081 [0.4086]	5.546 [0.71447]	6.5114 [0.6846]	3.1869 [0.4685]	23.3586 [9.1084]	0.9954 [0.1109]	257.9789 [41.5684]
PSO	1.1397 [0.2887]	4.8739 [0.5509]	6.0967 [0.8468]	2.3119 [0.3043]	10.9688 [1.3067]	0.7116 [0.0881]	286.2913 [95.8538]
WOA	1.6353 [0.7416]	5.179 [1.0438]	5.8036 [0.5382]	2.58 [1.00037]	38.8941 [10.5195]	0.6916 [0.0549]	233.0254 [29.3389]
DE	1.5814 [1.8786]	4.8112 [0.4484]	6.4906 [0.9177]	2.5263 [0.3517]	12.5726 [2.3679]	0.8402 [0.0896]	317.0387 [57.05553]
ACDCSA	1.1642 [0.4166]	4.4792 [0.4258]	5.9467 [0.8880]	2.1115 [0.3557]	32.1426 [3.8762]	0.6321 [0.0802]	251.592 [47.6612]

Table 5. Average [Standard Deviation] Inter-cluster distance generated by ACDCSA

Algorithms	*IRIS*	*CMC*	*CANCER*	*SEED*	*THYROID*	*WINE*	*VOWEL*
GWO	3.8566 [0.5292]	16.1716 [1.4921]	8.462 [0.87892]	7.2162 [1.1238]	37.917 [10.8852]	1.0153 [0.2627]	2135.975 [255.8602]
PSO	3.5645 [0.6469]	14.79808 [1.0500]	8.9292 [1.4579]	6.6381 [1.1841]	25.5818 [9.0621]	1.1978 [0.3031]	2234.526 [511.8010]
WOA	3.3416 [0.5444]	14.9725 [1.10184]	7.7904 [1.1339]	5.9655 [1.1132]	41.5909 [10.2056]	1.0755 [0.1845]	2127.374 [322.8464]
DE	4.1984 [0.6055]	15.6082 [1.95981]	8.454 [0.8461]	6.5097 [1.1134]	27.4341 [5.9063]	1.379 [0.2580]	2529.333 [380.8382]
ACDCSA	4.9521 [1.4926]	16.6781 [1.9107]	9.76 [3.0549]	7.3545 [2.3824]	47.7413 [12.9715]	1.6298 [0.6031]	2358.521 [472.9768]

References

[1] Jain, A.K., Murty, M.N. and Flynn, P.J. (1999) Data clustering: a review. *ACM computing surveys (CSUR)* **31**(3): 264–323.

[2] Roberts, S.J. (1997) Parametric and non-parametric unsupervised cluster analysis. *Pattern Recognition* **30**(2): 261–272.

[3] Jain, A.K. (2010) Data clustering: 50 years beyond k-means. *Pattern recognition letters* **31**(8): 651–666.

[4] Gan, G., Ma, C. and Wu, J. (2020) *Data clustering: theory, algorithms, and applications* (SIAM).

[5] Talbi, E.G. (2009) *Metaheuristics: from design to implementation*, **74** (John Wiley & Sons).

[6] Bandyopadhyay, S. and Maulik, U. (2002) Genetic clustering for automatic evolution of clusters and application to image classification. *Pattern recognition* **35**(6): 1197–1208.

[7] Das, S., Abraham, A. and Konar, A. (2007) Automatic clustering using an improved differential evolution algorithm. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans* **38**(1): 218–237.

[8] Kuo, R., Syu, Y., Chen, Z.Y. and Tien, F.C. (2012) Integration of particle swarm optimization and genetic algorithm for dynamic clustering. *Information Sciences* **195**: 124–140.

[9] Kumar, V., Chhabra, J.K. and Kumar, D. (2017) Grey wolf algorithm-based clustering technique. *Journal of Intelligent Systems* **26**(1): 153–168.

[10] Kaushik, K., Arora, V. *et al.* (2015) A hybrid data clustering using firefly algorithm based improved genetic algorithm. *Procedia Computer Science* **58**: 249–256.

[11] Nasiri, J. and Khiyabani, F.M. (2018) A whale optimization algorithm (woa) approach for clustering. *Cogent Mathematics & Statistics* **5**(1): 1483565.

[12] Chhabra, J.K. *et al.* (2017) Harmony search based remodularization for object-oriented software systems. *Computer Languages, Systems & Structures* **47**: 153–169.

[13] Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J.M. and Perona, I. (2013) An extensive comparative study of cluster validity indices. *Pattern recognition* **46**(1): 243–256.

[14] Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J. and Wu, S. (2013) Understanding and enhancement of internal clustering validation measures. *IEEE transactions on cybernetics* **43**(3): 982–994.

[15] Askarzadeh, A. (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. *Computers & Structures* **169**: 1–12.

[16] Ezugwu, A.E., Shukla, A.K., Agbaje, M.B., Oyelade, O.N., José-García, A. and Agushaka, J.O. (2021) Automatic clustering algorithms: a systematic review and bibliometric analysis of relevant literature. *Neural Computing and Applications* **33**(11): 6247–6306.

[17] Van der Merwe, D. and Engelbrecht, A.P. (2003) Data clustering using particle swarm optimization. In *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.* (IEEE), **1**: 215–220.

[18] Omran, M., Salman, A. and Engelbrecht, A. (2005) Dynamic clustering using particle swarm optimization with application in unsupervised image classification. In *Fifth World Enformatika Conference (ICCI 2005), Prague, Czech Republic*: 199–204.

[19] Das, S., Abraham, A. and Konar, A. (2008) Automatic kernel clustering with a multi-elitist particle swarm optimization algorithm. *Pattern recognition letters* **29**(5): 688–699.

[20] Alswaitti, M., Albughdadi, M. and Isa, N.A.M. (2018) Density-based particle swarm optimization algorithm for data clustering. *Expert Systems with Applications* **91**: 170–186.

[21] Gao, H., Li, Y., Kabalyants, P., Xu, H. and Martinez-Bejar, R. (2020) A novel hybrid pso-k-means clustering algorithm using gaussian estimation of distribution method and lévy flight. *IEEE access* **8**: 122848–122863.

[22] Sharma, M. and Chhabra, J.K. (2019) Sustainable automatic data clustering using hybrid pso algorithm with mutation. *Sustainable Computing: Informatics and Systems* **23**: 144–157.

[23] Jadhav, A.N. and Gomathi, N. (2018) Wgc: Hybridization of exponential grey wolf optimizer with whale optimization for data clustering. *Alexandria engineering journal* **57**(3): 1569–1584.

[24] Tripathi, A.K., Sharma, K. and Bala, M. (2018) A novel clustering method using enhanced grey wolf optimizer and mapreduce. *Big data research* **14**: 93–100.

[25] Aljarah, I., Mafarja, M., Heidari, A.A., Faris, H. and Mirjalili, S. (2020) Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach. *Knowledge and Information Systems* **62**(2): 507–539.

[26] Kuo, R.J. and Zulvia, F.E. (2018) Automatic clustering using an improved artificial bee colony optimization for customer segmentation. *Knowledge and Information Systems* **57**(2): 331–357.

[27] Hussain, S.F., Pervez, A. and Hussain, M. (2020) Co-clustering optimization using artificial bee colony (abc) algorithm. *Applied Soft Computing* **97**: 106725.

[28] Talaei, K., Rahati, A. and Idoumghar, L. (2020) A novel harmony search algorithm and its application to data clustering. *Applied Soft Computing* **92**: 106273.

[29] Tseng, L.Y. and Yang, S.B. (2001) A genetic approach to the automatic clustering problem. *Pattern recognition* **34**(2): 415–424.

[30] Vovan, T., Phamtoan, D., Tuan, L.H. and Nguyentrang, T. (2021) An automatic clustering for interval data using the genetic algorithm. *Annals of Operations Research* **303**(1): 359–380.

[31] Chen, J.X., Gong, Y.J., Chen, W.N., Li, M. and Zhang, J. (2019) Elastic differential evolution for automatic data clustering. *IEEE Transactions on Cybernetics* **51**(8): 4134–4147.

[32] Ranjan, R. and Chhabra, J.K. (2022) A dynamic crow search algorithm and its application in data clustering. *Kuwait Journal of Science* [Manuscript submitted for publication].

[33] Hennig, C., Meila, M., Murtagh, F. and Rocci, R. (2015) *Handbook of cluster analysis* (CRC Press).