IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 27, 2021, accepted June 12, 2021, date of publication June 21, 2021, date of current version July 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3091397

Automatic Data Clustering Framework Using Nature-Inspired Binary Optimization Algorithms

BEHNAZ MERIKHI AND M. R. SOLEYMANI
Department of Electrical and Computer Engineering, Concordia University, Montreal, QC H3G 1M8, Canada

Corresponding author : Behnaz Merikhi (behnaz.merikhi@concordia.ca)

This work did not involve human subjects or animals in its research.

ABSTRACT Cluster analysis using metaheuristic algorithms has earned increasing popularity over recent years due to the great success of these algorithms in finding high-quality clusters in complex real-world problems. This paper proposes a novel framework for automatic data clustering with the capability of generating clusters with approximately the same maximum distortion using nature-inspired binary optimization algorithms. The inherent problem with clustering using such algorithms is having a huge search space. Therefore, we have also proposed a binary encoding scheme for the particle representation to alleviate this problem. The proposed clustering solution requires no prior knowledge of the number of clusters and proceed with the process based on re-clustering, merging, and modifying the small clusters to compensate for the distortion gap between groups with different sizes. The proposed framework's performance has been evaluated over a wide range of synthetic, real-life, and higher dimensional datasets first by considering four different binary optimization algorithms for the optimizer module. Then, it has also been compared to multiple classical and new clustering solutions and two other automatic clustering techniques in continuous search space in terms of separation and compactness of the clusters by utilizing internal validity measures. The experimental results show the proposed solution is highly efficient in creating well-separated and compact clusters with approximately the same distortion in most datasets. Moreover, the application of the proposed framework to the correlated binary dataset is also reported as a case study. The presence of correlation in a dataset results from the similarity between data points in the same category, such as repeated measurements in remote sensing, crowdsourced multi-view video uploading, and augmented reality. Simplicity, customizability, and flexibility in adding extra conditions to the proposed solution and having a dynamic number of clusters are the advantages of the proposed framework.

INDEX TERMS Automatic data clustering, binary search space, binary clustering, cluster-level constraint, distortion, dynamic clusters, nature-inspired optimization algorithms.

I. INTRODUCTION

The increasing penetration of artificial intelligence technologies and the widespread usage of sensors, networking devices, and IoT smart applications in people's daily lives impose massive amounts of unstructured data into wireless networks every day. This massive volume of data is extremely beneficial in data analysis, social sciences, and many other modern applications if it is appropriately classified and analyzed in a meaningful way [1], [2]. Clustering is a popular unsupervised data analysis technique that captures the natural structure of a dataset and places similar objects into a set of groups to simplify the process of analyzing and understanding information coming from different sources. As a result, objects within a cluster are expected to be more similar to each other. Nowadays, clustering analysis methods are being widely used in many fields such as wireless sensor networks, mobile networks, image and video processing, and data summarization [3]. In some applications of wireless sensor networks, having clusters with approximately the same distortion (radius) is desirable, while there is no prior information to specify the exact number of clusters in most cases. One example is designing compression schemes for the nodes in distributed source coding problems [4], [5]. Distributed source coding problem considers the compression of multiple correlated information sources that are statistically dependent but physically distinct. Requiring an appropriate clustering solution in such applications motivated us to propose an

The associate editor coordinating the review of this manuscript and approving it for publication was Haruna Chiroma.

automatic clustering framework that can create compact and well-separated clusters with approximately the same distortion without requiring any extra information.

Assigning all objects correctly to different groups and determining the optimal number of clusters are the two fundamental challenges in automatic data clustering problems [6]. The number of combinations in assigning m objects into K groups is calculated by the Stirling number of the second kind:

$$S(m,K) = \frac{1}{K!}\sum\nolimits_{i=0}^{K} (-1)^{K-i}\binom{K}{i}(i)^{m} \tag{1}$$

On the other hand, the size of the search space for determining the optimal number of clusters is calculated by the Bell numbers described by:

$$B(m) = \sum\nolimits_{K=1}^{m} S(m,K) \tag{2}$$

Moreover, it has been shown [3], [6] that finding an optimal solution for the clustering problem is NP-hard when $K > 3$. These problems become evident, especially in the case of having a high-dimensional dataset or dealing with non-overlapping clusters having the potential of being different in density, size, and shape.

Over the years, an extensive number of clustering algorithms have been proposed for different types of data, algorithms, and applications. In general, partitional and hierarchal clustering algorithms are two main approaches for cluster analysis in the literature [3], [6]–[12]. The partitional algorithms can be performed in two different modes: hard and fuzzy. In the hard-clustering algorithm, each pattern only belongs to one cluster, while in the fuzzy clustering algorithm, different membership degrees are assigned to each pattern in a group. Partitional algorithms are often non-deterministic. These algorithms require a priori knowledge of the number of non-overlapping clusters [3], [7]. The well-known K-means method is a famous example of this type that is initialized with a random solution and tries to partition a given dataset into a predefined number of clusters. Although it is an efficient and robust algorithm, the results are strongly dependent on the initial random guesses. Furthermore, this algorithm computes the local minimum and cannot guarantee the global optimum solution [8]–[10].

The hierarchical clustering algorithm, on the other hand, develops a tree-based data structure to reach the exact number of clusters by splitting the tree at different levels. This algorithm creates a more deterministic and flexible mechanism compared to the partitional approach. However, the final grouping is static since each cluster's assigned objects cannot move to other groups. Besides, hierarchical clustering exhibits poor performance when the separation of overlapping clusters is carried out [3], [9], [11]. In this case, the fuzzy clustering algorithm can express the overlapping nature of clusters much better than the hierarchical model. The fuzzy C-mean (FCM) algorithm [12] is a widely used clustering algorithm that divides objects into a C number of clusters. This number is determined by trial and error in advance. Although FCM is a powerful method, it is highly dependent on initial guesses; thus, it can be easily entrapped within local optima [3], [7], [12].

Being sensitive to initial solutions, entrapping in local optima, and requiring a priori knowledge of the number of clusters in most classical clustering algorithms made it challenging to handle this task in some applications. On the other hand, most of the real-world clustering problems can be described as a typical optimization problem that tries to optimize a criterion and specify the clustering quality [9]. Therefore, the use of nature-inspired metaheuristic algorithms in automatic clustering has been proposed as a superb solution to work with different applications and datasets and overcome the weaknesses of the classical approaches in recent years [13], [14].

In a nutshell, metaheuristic algorithms can address the clustering problem via two main approaches. In one approach, the optimization algorithm tries to find the optimum centroids for the desired dataset. Then, it divides data points into predetermined clusters according to the identified centroids. This approach requires a priori knowledge of the exact number of clusters. In another approach, the optimization algorithm assigns each data point directly to a group and tries to reach the best possible solution over the course of iterations. The second approach is more convenient since the number of clusters is not required to be predefined, and a sufficient number of clusters evolve during the entire clustering process. However, it suffers from a huge search space, which makes the overall solution extremely difficult without providing additional insights.

The advantage of not requiring predefined information in the second approach motivated us to develop our clustering problem based on this solution. In this regard, we have formulated the clustering problem as an optimization problem and adopted a dynamic range of clusters in accordance with the input data to improve the convergence speed.

One of the main contributions of this paper is that the obtained clusters will have approximately the same distortion. Therefore, as mentioned earlier, it is highly beneficial in applications such as compression schemes that need to have relatively close values for the maximum distortion in each cluster. Another contribution is that we have considered a merge and modify step in the objective function to compensate for the distortion gap between clusters of different sizes without increasing the number of clusters. This step significantly improves the convergence speed of the proposed framework compared to similar solutions without this assumption. Example 1 can elaborate on this effect. Since we believe assigning a cluster number to each data point has a discrete nature, another contribution is proposing a binary encoding scheme for the particle representation. In most of the previous work, such as [15], [16], this problem is usually carried out by considering a continuous encoding algorithm requiring additional assumptions. Moreover, the proposed framework is a customizable solution that can effectively work considering other assumptions or using different datasets and distance measures. The correlation-aware clustering scheme discussed in section VI shows this capability.

The performance of the proposed framework has been evaluated over a wide range of synthetic, real-life, and higher-dimensional datasets. This evaluation has been performed in two parts; At first, the performance of the proposed framework has been examined by considering four different binary optimization algorithms, including binary bat algorithm (BBA) [17], binary particle swarm optimization algorithm (BPSO) [18], binary genetic algorithm (BGA) [19], and binary dragonfly algorithm (BDA) [20]. Then, the performance of the proposed solution has been compared to other classical and new clustering algorithms as well. Moreover, a correlation-aware clustering scheme for a binary dataset is proposed and discussed as a case study. In this sense, the proposed framework has been tailored to the binary case and provided a solution for the automatic clustering problem in applications with correlated binary sources. Binary data is the simplest case of categorical data in which only two possible values describe discrete attributes and can be reflected as a special case of quantitative data. Binary data clustering is a challenging task due to its high dimensionality and sparsity [21]. The correlated binary clustering solution is beneficial in various disciplines such as medical sciences, machine learning, big data, pattern recognition, image analysis [3], [7], and many other recent applications such as cache-aided networks and edge caching [22]. In such cases, taking advantage of the similarity between the sample sets in the clustering solution can improve efficiency and reduce the delivery load [23].

The rest of this paper is organized as follows. Section II presents a literature review on automatic clustering using nature-inspired optimization algorithms and cluster-level constraints. Section III briefly presents data clustering problem definitions. In Section IV, the proposed framework and the problem formulation are described in detail. Section V addresses the experimental results and discussions. Section VI discusses a correlation-aware clustering scheme for the binary dataset as one of the applications of the proposed solution. Finally, Section VII presents the conclusion and some directions for future work.

II. LITERATURE REVIEW

Over the years, many optimization algorithms have been proposed to overcome the problems caused by traditional algorithms in cluster analysis. Tabu search algorithm [24], the simulated annealing (SA) algorithm [25], and the particle swarm optimization (PSO) algorithm [26], which is one of the most powerful optimization algorithms for data clustering in complex problems, are some of the well-known examples in this area. Van der Merwe and Engelbrecht [27] have investigated the capability of several swarm intelligence algorithms in partitioning different types of datasets. They have also proposed a novel approach for clustering different datasets into an optimal number of clusters through an optimization process. In [28], the data clustering problem has been formulated as a single objective problem, and the standard *gbest* of the PSO algorithm has been used to identify the centroid of the clusters. Evolutionary algorithms have also been among the most frequently used metaheuristic algorithms for the clustering problem [29]. Hence, different types of this algorithm have been studied in the literature, ranging from a straightforward encoding, when the i^{th} gene coding for clustering membership of the i^{th} object, to a more sophisticated solution similar to Falkenauer's grouping genetic algorithm [30].

Although many optimization algorithms have been used to solve the traditional clustering problems, only a few studies have focused on applying nature-inspired metaheuristic techniques to solve automatic data clustering. In [13], the automatic data clustering problem has been addressed by utilizing a hybrid solution called FAPSO based on an improved firefly algorithm and the particle swarm optimization algorithm. The authors also investigate the applicability of the proposed solution in detecting the correct number of clusters according to the Davis-Bouldin index (DB-index) [31] and the compact-separated index (CS-index) [32] as the validity measure. The proposed algorithm's performance has been evaluated on thirteen benchmark datasets, and it has also been compared to other well-known clustering algorithms. The experimental results indicated that the FAPSO outperforms the comparative studies in most cases in terms of the accuracy of the results.

In [33], an automatic data clustering algorithm using an improved PSO algorithm (ACPSO) has been proposed to address the clustering problem. The focus of the proposed algorithm is determining the correct number of clusters and adjusting the centroids. The authors have considered the K-means algorithm and a sigmoid function to adjust the cluster centroids and manage the infeasible solutions. This algorithm has been evaluated by considering four benchmark datasets in terms of consistency and accuracy. Abraham *et al.* [34] proposed a kernel-based automatic clustering using a modified PSO algorithm. This approach employs a kernel-induced similarity measure instead of the sum of square distances. They believed that using a kernel function in this solution leads to clustering linearly non-separable data into homogenous clusters in a high dimensional feature space transformation. This algorithm has been evaluated by considering five synthetic and three real-life datasets in terms of convergence, accuracy, and robustness. In [35], Nanda and Panda proposed a multi-objective automatic clustering algorithm called MOIMPSO to classify actions of 3D human models. This algorithm provides a Pareto optimal archive for automatic clustering problems by considering a developed hybrid evolutionary algorithm immunized PSO and two objective functions. Besides, a single best solution from the Pareto optimal archive has been provided to satisfy the users' requirements. They have also evaluated the proposed algorithm on eleven benchmark datasets in terms of computation time and accuracy.

Liu *et al.* [36] proposed a solution based on the genetic algorithm with unknown K called AGCUK. They employed the DB-index as the validity measure of clusters. The performance of this algorithm has been evaluated on several artificial and real-life datasets in terms of determining the correct number of clusters and the accuracy of the clustering. Then, He and Tan [37] proposed a novel two-stage genetic algorithm called TGCA to solve the clustering problem. This algorithm uses the selection and the mutation operators of the original genetic algorithm. The TGCA algorithm attempts to gradually reach globally optimal cluster heads by focusing on determining the correct number of clusters for each input. The experimental results on four artificial and seven real-life datasets in this study indicate this algorithm shows high performance in determining the number of clusters and the clustering solution's accuracy.

In [38], the application of the differential evaluation (DE) algorithm is described for the clustering problem with an unlabelled large dataset. The proposed algorithm is called the ACDE algorithm and used an improved differential evaluation algorithm for the data clustering problem. The ACDE algorithm has been evaluated by considering five benchmark datasets via DB-measure and CS-measure. The authors also reported the application of the ACDE algorithm to the automatic segmentation of images. Then, in [39], a new hybrid algorithm based on differential evaluation and fuzzy clustering called ADEFC has been proposed to solve the automatic clustering problem. In this algorithm, the cluster heads are encoded in the vectors. The data points are then assigned to different clusters based on the Xie-Beni index, a validity measure for the clustering validation. The performance of the ADEFC algorithm has been evaluated on two synthetic and two real-life datasets. It has also been compared to the fuzzy C-mean algorithm and the variable-length genetic algorithm based on fuzzy clustering. The authors indicated that the ADEFC has the capability of being considered for micro-array data clustering. An improved differential evaluation algorithm with cluster number oscillation called ACDE-O has been proposed in [40]. Since poor initial guesses lead to inefficient clusters, a cluster number oscillation mechanism is used in this algorithm to improve searching and finding more possible clusters. This algorithm's efficiency has been evaluated on three real-life datasets compared to the ACDE algorithm and reported better performance. Kuo *et al.* [41] proposed automatic kernel clustering with bee colony optimization (AKC-BCO) to address the weaknesses of the automatic clustering problem in determining the number of clusters and the accuracy of the clustering. The authors employed a kernel function to increase the capability of the clustering algorithm. The performance of the AKC-BCO has been evaluated on several benchmark datasets compared to three other clustering algorithms. The experimental results indicate that the AKC-BCO algorithm demonstrates superior performance in terms of not trapping in local optima, convergence speed, and accurate and stable clustering results. Then, in [42], Kuo and Zulvia proposed a hybrid solution of an improved artificial bee colony optimization and K-means algorithm called iABC for the automatic clustering problem and the customer segmentation problem. In this study, the onlooker bee exploration in the original ABC algorithm is improved by guiding their movements to a better location, leading to a better initial centroid in the K-means algorithm. Then, to increase the algorithm's efficiency, only the worst cluster centroid will be improved through an updating process. The experimental results on several benchmark datasets show that the iABC algorithm provides better performance than the classical ABC algorithm. They mentioned that the average value of the computational time for some datasets is less than the original ABC algorithm. The reason is that the iABC algorithm generates better solutions compared to the original ABC algorithm. However, its performance is not faster than the PSO and GA-based algorithms.

In [43], the harmony search algorithm has been employed by Kumar *et al.* to present a parameter adaptive harmony search algorithm called ACPAHS for automatic data clustering problems. The number of clusters in the proposed algorithm is determined by using a real-coded variable-length harmony vector. The data points are assigned to clusters according to the developed weighted Euclidean distance. The authors also reported the application of the proposed algorithm to the automatic segmentation of images. The efficiency of the ACPAHS algorithm has been evaluated on several real-life datasets and compared to four other well-known clustering techniques in terms of the determined number of clusters and the accuracy of the clustering solution.

In [44], the problem of automatic data clustering has been solved based on an evolutionary metaheuristic algorithm known as invasive weed optimization (IWO). This algorithm can perform the clustering task without requiring any prior knowledge of the datasets and employs the genetic algorithm's fitness function as the validity measure. The algorithm's proficiency has been evaluated on nine artificial and four real-life datasets, and the results are compared to three other clustering algorithms. The experimental results have indicated that the IWO algorithm shows great performance in population size and computation time.

Qaddoura *et al.* [45] proposed an open-source and cross-platform framework called EvoCluster for data clustering. EvoCluster is a customizable framework that can employ various objective functions in addition to different well-known nature-inspired optimizers developed by other researchers to perform partitional clustering tasks. It can also evaluate the result according to different validity measures such as Purity, Entropy, the sum of squared error, and some other common validity measures. Since this framework covers a different set of algorithms and measures, it can be useful in different applications. Although it is a comprehensive framework, it does not overlap with what we have proposed since our main focus is elsewhere, and we are not in the same direction.

A comprehensive survey on data clustering using metaheuristic algorithms can also be found in [6], [14], [38], [46], [47].

Reaching clusters with the same radius can be discussed under the cluster-level constraint as well and has many practical applications. The cluster-level constraint considers some available information about the underlying cluster structures in the form of limitations [48], [49]. The facility location problem discussed in [50] is similar to the clustering problem with a cluster-level constraint. In this study, the authors propose two heuristics algorithms for the facility location problems that can be interpreted as a clustering problem with upper bounds on the radius of the clusters. In [51], the authors study two types of cluster-level constraints in a search-based agglomerative hierarchical clustering algorithm. This algorithm forms initial partitioning using the must-link constraints and then merges some groups by taking constraints into account to meet the stopping criteria. Finally, they mentioned creating a feasible dendrogram is intractable since solving the clustering problem with unspecified K is NP-complete under these constraints. One of the other exciting applications using cluster-level constraint approaches is discussed in [52] for a distributed sensor network. In such applications, each sensor has one master node, and the aim is to find balanced clusters of sensor nodes while attempting to minimize the distance between master and sensor nodes. The authors formulated the problem as a minimum cost flow problem and optimally solved it.

We have reviewed many related studies that focus on the clustering problem from different aspects in this section. Since cluster analysis is being exploited in diverse research fields, a unique algorithm cannot be a solution to all clustering scenarios due to the differences in the nature of the patterns and applications. Hence, considering this paper's main purpose, we remain focused on reaching clusters with approximately the same distortion value while the number of clusters is in accordance with the number of inputs. For this purpose, we have utilized nature-inspired binary optimization algorithms. This framework is highly beneficial in clustering correlated binary sources used in distributed source coding applications discussed as a case study in Section VI.

III. DATA CLUSTERING PROBLEM DEFINITION

Let $P_{m\times l}$ represents a set of m patterns, each having l features. Then, the data clustering problem considers a given dataset $P_{m\times l} = [P_1, P_2, P_3, \ldots, P_m]$ and attempts to partition it into K clusters $C = [C_1, C_2, C_3, \ldots, C_K]$ such that $K \leq m$. In such partitioning, the following properties should be maintained:

- $\forall i \in \{1, 2, 3, \ldots, K\}, C_i \neq \emptyset$
- $\bigcup_{i=1}^{K} C_i = P$
- $C_i \cap C_j = \emptyset, \forall i, j \in \{1, 2, \ldots, K\}$, and $i \neq j$

Given dataset $P_{m\times l}$, the fitness function f is defined as a partitioning adequacy measure to quantify the goodness of a partition based on the similarity of the patterns. Therefore, the clustering problem turns into finding optimal partitions among all other feasible solutions [7].

Euclidean distance, which has also been used in this paper, is widely used as a distance measure to evaluate the similarity between data points in clustering problems. The Euclidean distance between any two l-dimensional data point is given by:

$$d(P_n, P_m) = \sqrt{\sum\nolimits_{i=1}^{l} (P_n^i - P_m^i)^2} = \left\| \overrightarrow{P_n} - \overrightarrow{P_m} \right\| \tag{3}$$

IV. PROPOSED DATA CLUSTERING FRAMEWORK

A. METHODOLOGY

The main idea of designing the proposed framework is reaching a sufficient number of compact and well-separated clusters with relatively close values for the maximum distortion in each group without requiring any prior knowledge of the clusters. To this end, we consider the clustering problem as an optimization problem and solve it in a discrete search space using nature-inspired binary algorithms for the optimizer module. Furthermore, to overcome the inconvenience of having a huge search space as well as not having a priori knowledge of the adequate number of clusters, we propose a binary encoding scheme for the particle representation of the optimization algorithm in a predetermined range $[K_{min}, K_{max}]$, where $K_{min} = 1$, $K_{max} = \lfloor\sqrt{\mathrm{m}}\rfloor$, and m is the total number of data points.

K_{max} is considered as $\lfloor\sqrt{\mathrm{m}}\rfloor$ according to similar assumptions discussed in [53] for a clustering solution with fuzzy C-mean model and [6] for an automatic clustering using nature-inspired metaheuristics. It is mentioned in those studies that $K = \sqrt{\mathrm{m}}$ usually provides a decent answer for such clustering solutions as a rule of thumb. Hence, we applied this assumption to our dynamic range of clusters in the proposed solution. We observed that considering this assumption along with the proposed binary encoding scheme will provide excellent results in our proposed framework. Overall, the proposed framework works in two main steps: First, each data point will be directly equipped with an initial cluster number. Consequently, a primary clustering solution will be formed in this step. The obtained clusters will then be re-clustered, merged, and modified based on some conditions to compensate for the distortion gap between groups with different sizes and improve the result according to the desired objectives.

The optimizer module, the binary encoding scheme, and the objective function will be described in detail in the following.

B. OPTIMIZER MODULE

The optimizer stands at the highest level of the proposed approach and considers the clustering problem as a black box that needs to be optimized iteratively. In this regard, the automatic clustering problem will be formulated as an objective function based on the problem's desired goals. The optimizer takes a binary vector as input and calculates the corresponding output according to some merit factors while minimizing this function during the entire process. In other words, the optimizer checks combinations of the input to determine which input vector (cluster numbers) will yield the minimum output of the function. A wide range of binary optimization algorithms can be utilized for the optimizer. This paper considers the BPSO algorithm, the BBA algorithm, the BGA algorithm, and the BDA algorithm due to their excellent performance discussed in the literature.

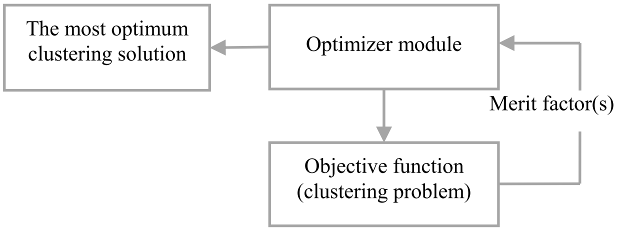

The most optimum clustering solution
Optimizer module
Merit factor(s)
Objective function (clustering problem)

FIGURE 1. The relation between the objective function and the optimizer module in the proposed clustering framework.

Fig. 1 shows how the optimizer and the objective function are related to come up with the optimum solution for the clustering problem.

C. BINARY ENCODING SCHEME

Similar to other iterative metaheuristic algorithms [6], our approach requires a representation of a solution, which is directly related to the objective function to be optimized. Therefore, a binary encoding scheme for the particle representation has also been proposed in this paper. The proposed binary encoding scheme assigns binary vectors with the length of $m \times L$ bits to each particle in the optimizer. L is the number of required bits to address each cluster number and calculates as $L = Log_2^{K_{\max}}$.

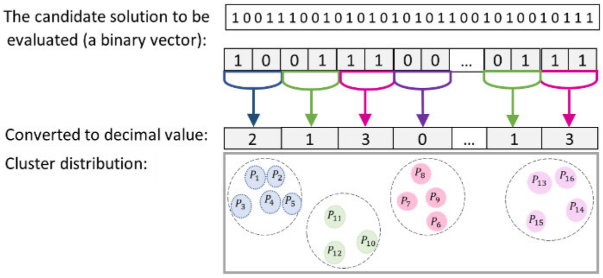

The candidate solution to be evaluated (a binary vector):
10011100101010101011001010010111
1 0 0 1 1 1 0 0 ... 0 1 1 1
Converted to decimal value:
2 1 3 0 ... 1 3
Cluster distribution:
P_1 P_2 P_3 P_4 P_5
P_{11} P_{10} P_{12}
P_8 P_7 P_9 P_6
P_{13} P_{16} P_{14} P_{15}

FIGURE 2. The proposed binary encoding scheme for the proposed clustering framework.

The suggested binary vectors will be considered as the candidate solution for the optimizer and evaluated by the algorithm in each iteration. For this purpose, every L bit of this vector will be converted to the decimal equivalent in sequential order. These decimal numbers are assigned to data points as the initial cluster numbers. Fig. 2 shows a simple example of the proposed encoding scheme for $m = 16$ data points. In this case, $K_{max} = 4$ and each particle is represented by a binary vector of length $N = m \times L = 32$ bits.

D. OBJECTIVE FUNCTION

We formulate the data clustering problem as a single objective optimization problem that needs to be minimized. More formally, let f be a single criterion function, and ψ be the set of all feasible ways of clustering a given dataset $\psi = \{C^1, C^2, \ldots, C^{S(m,K)}\}$.We aim to find the clustering solution $C^* = [C_1, C_2, \ldots, C_K]$ where $f(C^*) = min\ \{f(C)|C\ \ \psi\ \}$.

1) INITIAL CLUSTERING STAGE

Once the initial clustering solution is formed, a representative is selected for each cluster. The representative is defined as the nearest data point to the current centroid in each group. The centroid is the mean of all data points in each cluster. Then, data points will be re-clustered according to the identified representatives of the clusters. Consequently, the representatives will also be updated according to the recent changes. These two steps, re-clustering and updating the representatives, will be repeated several times until the representatives do not change anymore.

2) MERGING AND MODIFYING STAGE

Although the achieved clustering solution up to this stage consists of separated clusters, it is not well-organized yet and needs to be revised to satisfy our goals. To reach clusters with approximately the same maximum distortion[1] while keeping the number of clusters less than K_{max} the distortion gap between the largest and the smallest clusters must be compensated. Therefore, the proposed framework tries to find the smallest clusters and mark them as the defective ones that need to be merged appropriately with the adjacent clusters. To this end, first, the cluster that has the largest maximum distortion is determined. Then, the maximum distortion of other clusters will be compared to this value according to inequality (4). Those clusters that are not true in this inequality will be considered defective and selected to be merged with adjacent clusters two by two.

$$\delta^{C_i} \le 0.9 \max_{C \in \psi}\{\delta^{C_i}\} \tag{4}$$

where δ^{C_i} is the maximum distortion within the cluster C_i. Then, the process of re-clustering and updating the representatives will be repeated over again to stabilize the achieved clustering solution. In this stage, the optimizer takes the current clustering solution as the best result for the corresponding input vector and calculates the following parameters out of it; K_{max} as the maximum number of clusters, δ^{C_i} as the maximum intra-cluster distance within each cluster C_i, Δ as the distortion deviation over all clusters, $\bar{d}_C^{\max}$ as the average of the maximum distortion over all clusters, and $\bar{E}_C^{\min}$ as the average of minimum inter-cluster distances. The definition of these parameters is given as follows.

Definition 1: Let δ^{C_i} be the maximum distortion of cluster C_i, then $\bar{d}_C^{\max}$ is defined as the average of the maximum

[1] In this paper, the maximum distortion and the maximum intra-cluster distance are used interchangeably since both meaning the same in our application.

distortion over all clusters and is calculated as:

$$\bar{d}_C^{\max} = \frac{1}{K}\sum_{i=1}^{K}\delta^{C_i} \tag{5}$$

where $i \in \{1, 2, \ldots, K\}$, and $C \in [C_1, C_2, \ldots, C_K]$

Definition 2: The distortion deviation Δ is defined as the difference between the maximum and the minimum value of maximum distortion over all clusters and is defined as:

$$\Delta = \max_{C\in\psi}\{\delta^{C_i}\} - \min_{C\in\psi}\{\delta^{C_j}\} \tag{6}$$

where $i, j \in \{1, 2, \ldots, K\}$, $i \neq j$, and $C \in [C_1, C_2, \ldots, C_K]$

Definition 3: Let E_{ij}^{p} be the inter-cluster distance between data point p within cluster i to the cluster-head j, where $p \in C_i$, $i \in \{1, 2, \ldots, K\}$, $j \in \{1, 2, \ldots, K-1\}$ and $i \neq j$.

We calculate this parameter for all data points in all clusters to determine the inter-cluster distances. Then, we extract the minimum of these distances for each cluster and denote it with $E_{C_i}^{\min}$ Next, we calculate $\bar{E}_C^{\min}$ as the average of the minimum inter-cluster distances over all clusters as:

$$\bar{E}_C^{\min} = \frac{1}{K}\sum_{i=1}^{K}E_{C_i}^{\min} \tag{7}$$

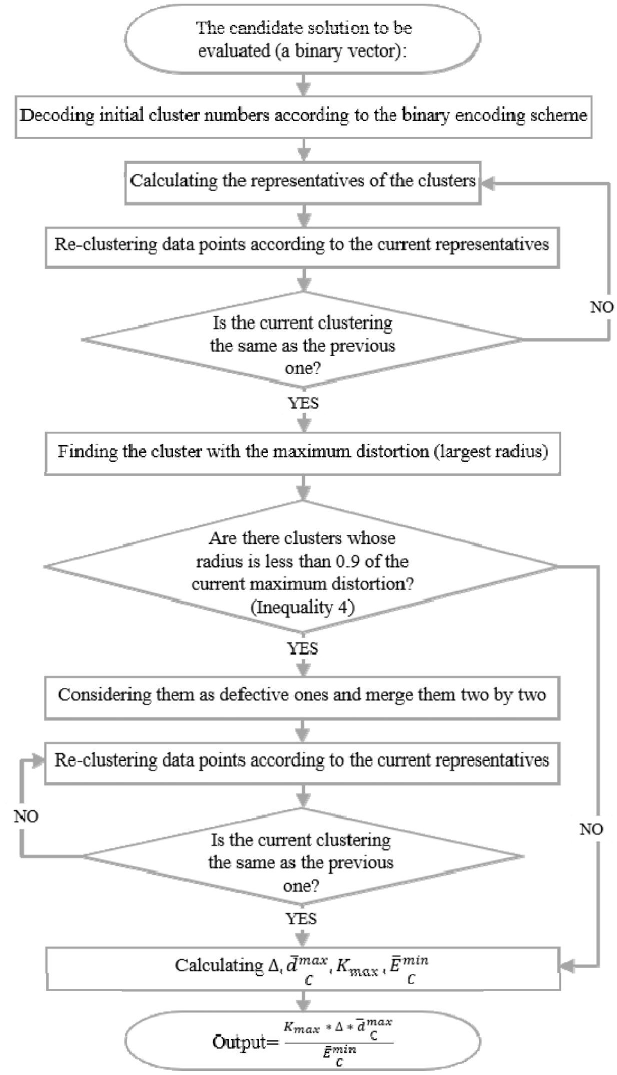
Finally, the output of the function f will be calculated according to the defined parameters, and the goal is to minimize it.

$$f = \frac{K_{\max} * \Delta * \bar{d}_C^{\max}}{\bar{E}_C^{\min}} \tag{8}$$

The entire process is simply described in the flowchart (1).

Also, example (1) illustrates the process of evolving clusters and migration of data points to other groups during different stages.[2] This evaluation has been performed on dataset R15 with m = 320 points (Fig. (3-a)), which is one of the standard datasets in UCI machine learning [54]. We have considered the BBA optimization algorithm with 100 agents and 200 iterations for this experiment. Fig. (3-b) shows the result of initial clustering, where an initial cluster number is assigned to each data point. Fig. (3-c) and (3-d) show the result of clustering after two levels of re-clustering. Although the clusters are distinguishable in this stage, the smallest cluster and the largest cluster still have a considerable size difference. During Stages (e) to (h), the process of merging and modifying the small clusters is performed, and the result is evaluated according to output. As can be seen, the proposed framework provides a sufficient number of compact and well-separated clusters with relatively close values for maximum distortion.

Fig. (4) illustrates the convergence curve of this example. As it is shown, the proposed framework demonstrates high convergence speed in a small number of iterations when we perform the merging and modifying steps to compensate for the distortion gap.


The candidate solution to be evaluated (a binary vector):
Decoding initial cluster numbers according to the binary encoding scheme
Calculating the representatives of the clusters
Re-clustering data points according to the current representatives
Is the current clustering the same as the previous one?
NO
YES
Finding the cluster with the maximum distortion (largest radius)
Are there clusters whose radius is less than 0.9 of the current maximum distortion? (Inequality 4)
YES
Considering them as defective ones and merge them two by two
Re-clustering data points according to the current representatives
NO
Is the current clustering the same as the previous one?
NO
YES
Calculating Δ, $\bar{d}_C^{max}$, K_{max}, $\bar{E}_C^{min}$
Output= $\frac{K_{max} * \Delta * \bar{d}_C^{max}}{\bar{E}_C^{min}}$

FLOWCHART 1. Flowchart of the objective function for the proposed clustering framework.

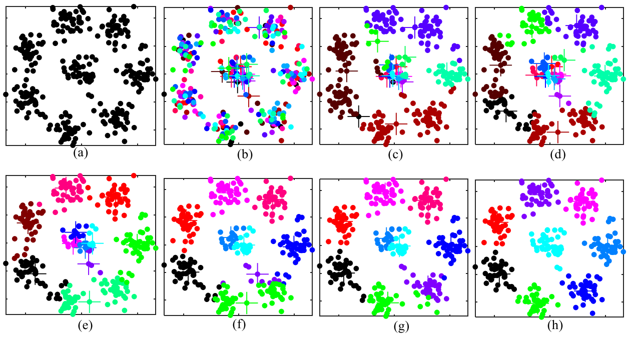


FIGURE 3. Process of evolving clusters and migration of data points to other clusters during different stages of the proposed clustering framework.

V. RESULTS AND DISCUSSION

In this section, the performance of the proposed automatic clustering framework has been evaluated on twenty-four

[2] Supplementary information is available for this example at https://github.com/BehnazMerikhi/Automatic-Framework from the corresponding author upon request.

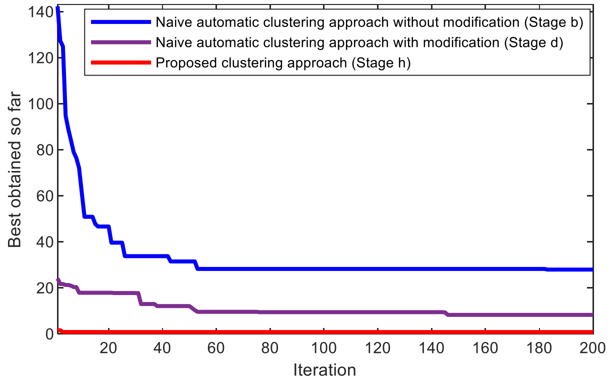


FIGURE 4. Convergence curve of example 1 in three different stages.

benchmark datasets. The details of the used datasets and the parameter settings for this performance evaluation have also been presented here. Then, simulation results and comparative study have been discussed. Finally, a correlation-aware clustering scheme for the correlated binary dataset has been reported as the application of the proposed clustering framework in the binary case, and the result has been evaluated on three artificial binary datasets.

A. DATASETS AND VALIDITY MEASURE

In this experiment, twenty-four different synthetic and real-world datasets are collected from the UCI Machine learning [54] and KEEL repositories [55]. This collection includes datasets with different shapes, densities, and dimensions ranging from 2 to 512, with diverse data points from 106 to 3100. Table 1 describes the summary of the datasets.

We have used the internal validity measures to examine the quality of the proposed clustering. In this regard, we have calculated the sum of intra-cluster and inter-cluster distance validity measures [3], [7] to ensure the compactness and the separation of the clusters. We have also evaluated the proposed framework by utilizing the DB-index validity measure [31], which describes a trade-off in maximizing intra-cluster compactness and inter-cluster separation. In the DB-index measure, the intra-cluster distances in the ith cluster and the inter-cluster distances between cluster ith and jth are defined as (9) and (10).

$$S_{i,q} = \left[\frac{1}{N_i}\sum\nolimits_{\vec{P}\in C_i}\left\|\vec{P}-\vec{m}_i\right\|_2^q\right]^{1/q} \tag{9}$$

$$d_{ij,t} = \{\sum\nolimits_{l=1}^{d}\left|m_{i,l}-m_{j,l}\right|^t\}^{1/t} = \left\|\vec{m}_i-\vec{m}j\right\|_t \tag{10}$$

$$R_{i,qt} = \max_{j\in K, j\neq i}\{\frac{S_{i,q}+S_{j,q}}{d_{ij,t}}\} \tag{11}$$

Finally, the DB-index is given by (12).

$$DB(K) = \frac{1}{K}\sum\nolimits_{i=1}^{K} R_{i,qt} \tag{12}$$

The smaller the value of the DB-index, the better the compactness and the separation.

TABLE 1. Summary of used datasets with different features.

Dataset		Number of features	Number of points	Number of Clusters
Shape dataset	Aggregation	2	788	6
	Compound	2	399	6
	D31	2	3100	31
	Flame	2	240	2
	Jain	2	373	2
	Pathbased	2	300	3
	R15	2	600	15
	Spiral	2	312	3
Real-World dataset	Appendicitis	7	106	2
	Dermatology	34	358	6
	Ecoli	7	336	8
	Glass	9	214	7
	Haberman	3	306	2
	Housevotes	16	232	2
	Ionosphere	33	351	2
	Iris	4	150	3
	Segment	19	2310	7
	Vehicle	18	846	4
	Wdbc	30	569	2
	Wine	13	178	3
Higher-dimensional dataset	Dime064	64	1024	16
	Dime128	128	1024	16
	Dime256	256	1024	16
	Dime512	512	1024	16

B. EXPERIMENTAL RESULTS AND PARAMETER SETTINGS

The experiments have been carried out on a PC with Windows 10 Professional 64-bit operating system, an Intel(R) Core ^{TM}i7-6700HQ processor, and 16 GB RAM using MATLAB software 2018 a. We have also used MATLAB packages,[3] Yarpize packages,[4] and NPIR source code[5] [59] with its required python packages for the comparative study during this experiment. Besides, the IBM SPSS Version 27 has been used for performing the statistical analysis test.

The comparison results are explained in two parts. The first set of results describe the effect of considering different binary optimization algorithms on the proposed framework. In this regard, we have evaluated the proposed framework performance using BBA, BPSO, BGA, and BDA algorithms in the optimizer module and presented the result for different cases. The parameter settings for the used algorithms are represented in Table 2. For this evaluation, twenty independent trials are performed on each dataset with each optimizer module. Then the best and the worst cost, the average cost, and the standard deviation are reported. These comparison results are described in Tables 3, 4, and 5. The best entries

[3] https ://www.mathworks.com/products/matlab.html.
[4] http://yarpiz.com/64/ypml101-evolutionary-clustering
[5] http://evo-ml.com/2019/10/28/npir/

TABLE 2. Parameter configuration of the utilized optimization algorithms in the proposed framework and two of the comparison studies.

BBA		BPSO		BGA		BDA		GCUK		DCPSO	
Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
Pop size	100	*Pop size*	100	*Pop size*	100	*Pop size*	100	*Pop size*	100	*Pop size*	100
Iter Max	200	*Iter Max*	200	*Iter Max*	200	*Iter Max*	200	*Iter Max*	200	*Iter Max*	200
K_{max}	$\sqrt{m}$	K_{max}	$\sqrt{m}$	K_{max}	$\sqrt{m}$	K_{max}	$\sqrt{m}$	K_{max}	$\sqrt{m}$	K_{max}	$\sqrt{m}$
K_{min}, K_{max}	0.2	C_1, C_2	2	μ_c	0.8	ω_{min}, ω_{max}	0.2, 0.9	μ_c	0.8	C_1, C_2	1.494
A	0.25	W	0.85	μ_m	0.001	α	0.01	μ_m	0.001	P_{ini}	0.75
r	0.5					β	0.09				
ε	[-1,1]										
α,γ	0.9										

TABLE 3. Comparison results for the shape dataset considering BBA, BPSO, BGA, and BDA optimizer module in the proposed method.

Dataset	BBA				BPSO				BGA				BDA			
	Best	Worst	Mean	Std	Best	Worst	Mean	Std	Best	Worst	Mean	Std	Best	Worst	Mean	Std
Aggregation	5.6286	6.7689	**6.5800**	0.2584	6.7689	8.1609	6.9238	0.3257	6.4781	7.2546	6.6879	0.1685	6.4781	7.6638	6.7724	0.2227
Compound	12.6988	22.4766	18.7042	2.9195	16.6346	24.0024	20.5652	2.3393	12.0882	21.8662	**16.0338**	2.5604	12.4404	24.0532	16.9529	2.8376
D31	2.1690	6.7443	**4.8710**	1.3991	3.6667	7.8899	6.5996	0.7892	2.1807	6.7345	5.4585	1.3507	2.1690	7.4051	6.2288	1.1587
Flame	3.5955	6.7825	**5.3221**	0.6692	5.1272	6.8180	6.0177	0.4533	3.5065	6.22141	5.3450	0.6802	5.4310	6.4252	5.9275	0.3365
Jain	6.0428	7.2892	6.2120	0.3579	6.0490	7.1225	6.1803	0.3209	5.9261	8.8739	6.5075	0.9012	5.7507	6.3269	**6.0480**	0.0935
Pathbased	8.5765	12.5244	**11.3455**	1.0841	8.9527	12.5116	11.7558	0.9447	10.0523	12.1479	11.6062	0.5856	9.5926	12.4449	11.8392	0.6363
R15	3.0021	3.8371	3.5388	0.2193	2.6340	3.8192	3.5582	0.2825	1.5466	3.8050	**2.9517**	0.7958	1.5466	3.8192	3.4506	0.5104
Spiral	11.1800	16.6888	14.6587	1.5984	12.677	17.0282	14.9723	1.3290	11.1276	15.9285	**14.1615**	1.3276	12.7713	16.6763	14.9223	1.1180

TABLE 4. Comparison results for the real-life dataset considering BBA, BPSO, BGA, and BDA optimizer module in the proposed method.

Dataset	BBA				BPSO				BGA				BDA			
	Best	Worst	Mean	Std	Best	Worst	Mean	Std	Best	Worst	Mean	Std	Best	Worst	Mean	Std
Appendicitis	1.6663	2.2850	**1.9597**	0.1371	2.2395	2.2395	2.0844	0.1163	1.6663	2.2405	2.0387	0.1402	1.6663	2.284	2.0043	0.1705
Dermatology	26.6937	41.5398	**36.3563**	4.1693	43.5930	43.5930	39.4908	2.8080	35.4513	44.1318	38.0146	2.0354	31.1669	44.0455	38.077	3.2976
Ecoli	367.5194	484.5737	**446.2215**	29.7053	501.2825	501.2825	466.992	27.1279	388.9087	487.4958	447.703	26.498	404.8281	524.5468	462.889	28.413
Glass	7.7872	8.1458	**8.1278**	0.0801	8.4200	8.4200	8.1869	0.1004	8.14580	8.4200	8.1732	0.0844	8.1458	8.4200	8.2006	0.1125
Haberman	44.9230	46.8628	45.7959	0.9901	46.8628	46.8628	**45.4079**	0.8617	44.9230	46.8628	**45.4079**	0.8617	44.9230	46.8628	45.6019	0.9492
Housevotes	2.2433	3.7942	2.9508	0.3882	5.4623	5.4623	3.5915	0.5192	2.0341	3.3358	**2.8132**	0.3898	2.3524	3.54983	2.9925	0.3449
Ionosphere	34.9838	48.3501	**41.4286**	3.1067	50.7202	50.7202	45.4032	3.3889	36.0527	47.5674	42.1577	3.5839	36.8054	48.6534	43.0474	3.4169
Iris	2.01661	3.2029	**2.6837**	0.4001	3.3367	3.3367	2.9749	0.2204	2.1990	3.2036	2.7937	0.3042	2.24264	3.5215	2.8303	0.3869
Segment	145.1097	148.7357	**145.291**	0.8107	166.3688	166.3688	146.7165	4.8107	145.1097	168.9339	150.734	9.6130	145.1097	168.9339	150.552	9.6866
Vehicle	140.6836	141.7339	141.488	0.4133	226.5138	226.5138	151.8188	23.6805	140.6836	141.7339	**141.330**	0.4871	140.6836	226.8402	148.115	21.234
Wdbc	1861.932	1869.163	**1864.61**	3.45	1858.820	1867.877	**1864.54**	2.398	1809.571	1928.531	1870.38	28.08	1857.946	1871.547	**1864.76**	3.451
Wine	139.2747	296.3782	210.934	47.5699	296.3782	296.3782	244.106	32.781	139.2747	296.3782	215.248	52.866	116.0226	296.3782	**178.401**	37.980

are shown in boldface in all tables. From the results, we can see the four binary optimization algorithms have reached a very competitive result. However, the performance of the proposed framework by considering the BBA algorithm in the optimizer module is more significant in most datasets. We have also provided the convergence curve of the shape

TABLE 5. Comparison results for the higher-dimensional dataset considering BBA, BPSO, BGA, and BDA optimizer module in The proposed method.

Dataset	BBA				BPSO				BGA				BDA			
	Best	Worst	Mean	Std	Best	Worst	Mean	Std	Best	Worst	Mean	Std	Best	Worst	Mean	Std
Dime064	13.0690	13.0705	**13.0698**	0.0004	13.1015	13.1818	13.1471	0.0192	13.1451	13.1454	13.1452	0.0001	13.1002	13.1094	13.1046	0.0023
Dime128	12.9516	13.2021	13.0691	0.0512	13.0191	13.1438	13.0731	0.0365	12.9340	13.1293	**13.0395**	0.0418	13.6873	13.8159	13.7431	0.0386
Dime256	3.5678	5.1516	**4.24784**	0.4043	3.8465	4.8430	4.3647	0.2982	3.7436	4.9207	4.2541	0.2795	4.1234	5.0762	4.6114	0.2936
Dime512	3.0210	4.7812	**3.9556**	0.4858	3.5532	4.8709	4.1412	0.3805	3.2822	5.1432	4.2679	0.5051	3.0534	5.5435	4.2726	0.6290

and the real-world datasets by considering all four optimizer modules in Fig. (5). The convergence curve is a useful tool to visualize how an algorithm improved the *gbest* as the first best path iteratively to reach the global optimum solution starting from a random solution. In all these curves, the optimizer that reaches the minimum cost after passing all iterations is suitable for that specific dataset and can provide the best solution. Fig. (6) shows the result of performing the proposed framework on the shape datasets to visualize some results. As seen in the figures, the proposed approach can generate different well-separated clusters while keeping a trade-off between the sufficient number of clusters and the shape of the clusters.

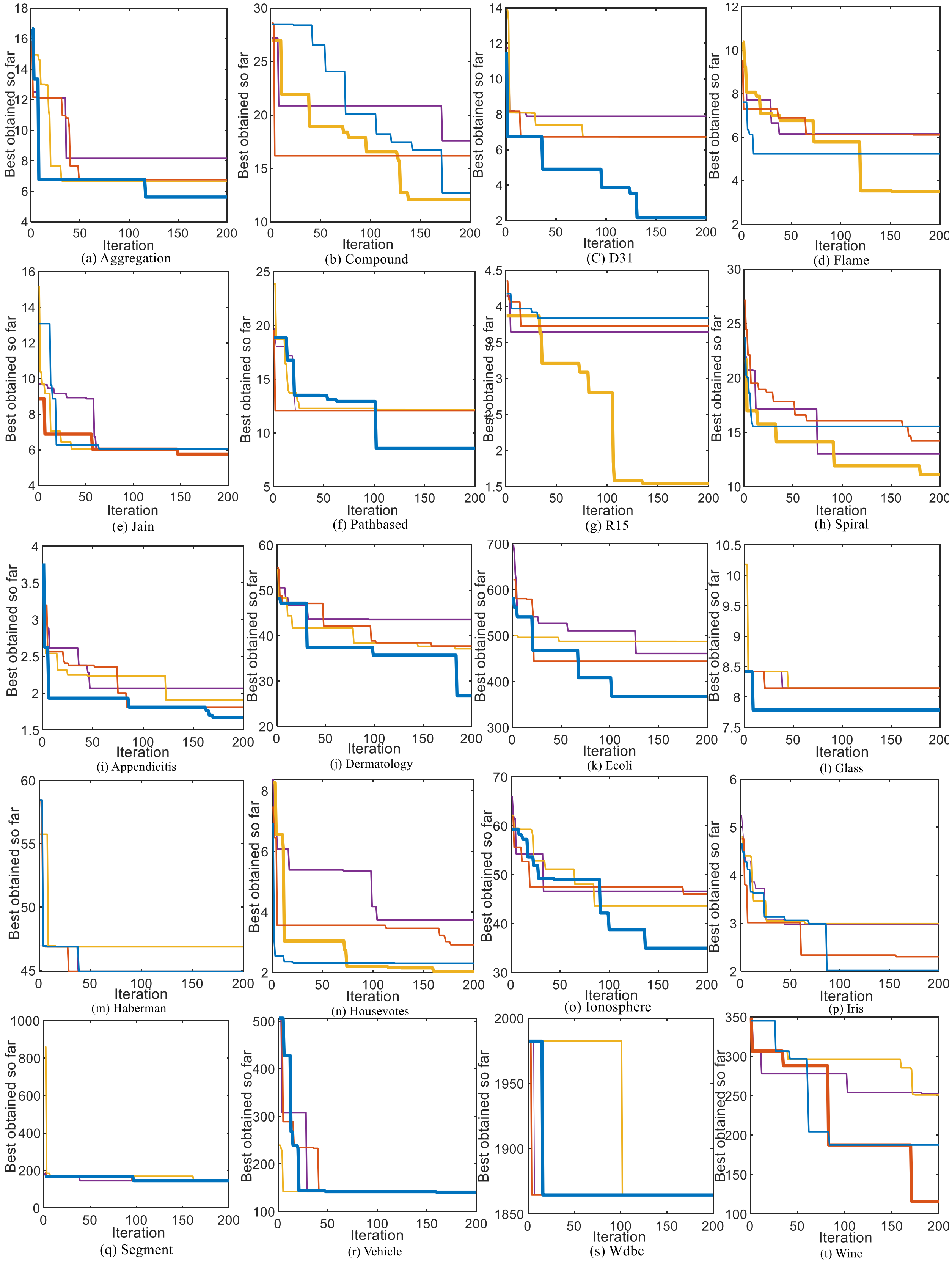
In the second part of the comparison results, the performance of the proposed framework has been compared with other classical and new clustering algorithms in terms of internal validity measures. To this end, we have considered the K-means ++ [56] as a representative of partitional clustering and the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [57] as a representative of density-based models. Also, we have considered the Expectation-Maximization (EM) algorithm [58] and the Nearest point with indexing ratio (NPIR) algorithm [59]. EM is a famous example of distribution-based clustering that employs a fixed number of Gaussian distributions to reach the distribution of the objects. NPIR is one of the latest clustering algorithms and works based on finding the nearest neighbors. Moreover, we have considered two well-known optimization algorithms in the continuous search space. These two algorithms are known as GCUK [60], a genetic-based clustering with an unknown number of clusters, and DCPSO [61], the dynamic PSO.

In this comparison, we have evaluated the sum of intra-cluster distances, the sum of inter-cluster distances, and the DB-index for twenty independent trials with each approach. We have also calculated the distortion deviation in all clustering solutions and compared the result. The distortion deviation, which is calculated in (6), shows the difference in the size (radius) of groups in a clustering approach. We wish to keep it minimized in our proposed method. Based on the described results in Tables 3 to 5, both BBA and BGA algorithms provide excellent performance as the optimizer module in most datasets. We have considered the BBA algorithm as the optimizer module for the rest of the experiment. For the comparative study, preliminary experiments have been done to determine the best settings for the required parameters to calculate internal validity measures. For GCUK, and DCPSO the parameter settings are described in Table 2. The EM algorithm needs to know the number of clusters in advance. The NPIR algorithm also requires prior knowledge of the number of clusters and the indexing ratio (IR) [59]. Therefore, we have performed multiple runs with various clusters in both algorithms and also have considered different IR values suggested by the authors for the NPIR to find the most appropriate parameters for each dataset in these algorithms. The DBSCAN algorithm forms clusters based on the density-based connectivity, and its performance is affected by MinPts and eps parameters. The MinPts can be selected based on dimensionality, and the eps can be specified based on the elbow in the k-distance graph [57]. Authors in [62] suggest using larger MinPts for a noisy and large dataset. Also, depending on the aim of clustering, you can decrease eps to avoid large clusters or increase it to avoid noise. Hence, we have run the DBSCAN with different MinPts and eps for each dataset to find the value leading to the best result in the mentioned validity measures.

The numerical results over twenty-four datasets have been summarized in Tables 6, 7, and 8.

It can be seen that in comparison with k-needed clustering methods (NPIR, K-Means++, and EM), the proposed framework has shown an excellent performance in most of the datasets in terms of the DB-Index and the distortion deviation measures as the main focus in our clustering method. In some cases, such as D31 and R15 in the shape datasets and Ionosphere, Iris, and vehicle in the Real-world datasets, the NPIR algorithm has shown better performance in the DB-index measure. However, the proposed method has been yielded a smaller distortion deviation in all these datasets.

The reason why the proposed framework has reached a bit higher DB-index compared to the other algorithms in some datasets can be explained as follows. The proposed framework focuses mainly on reaching clusters with approximately the same distortion while satisfying the other designed merit factors. This goal has been achieved by dividing the dataset into more or fewer groups compared to other algorithms



(a) Aggregation

(b) Compound

(C) D31

(d) Flame

(e) Jain

(f) Pathbased

(g) R15

(h) Spiral

(i) Appendicitis

(j) Dermatology

(k) Ecoli

(l) Glass

(m) Haberman

(n) Housevotes

(o) Ionosphere

(p) Iris

(q) Segment

(r) Vehicle

(s) Wdbc

(t) Wine

FIGURE 5. Convergence curve of the shape and the real-life datasets considering BBA, BPSO, BGA, and BDA optimizer module. The algorithm with the better performance is shown with thicker line. The color code for each algorithm is as follows: —— BPSO —— BGA —— BDA —— BBA.

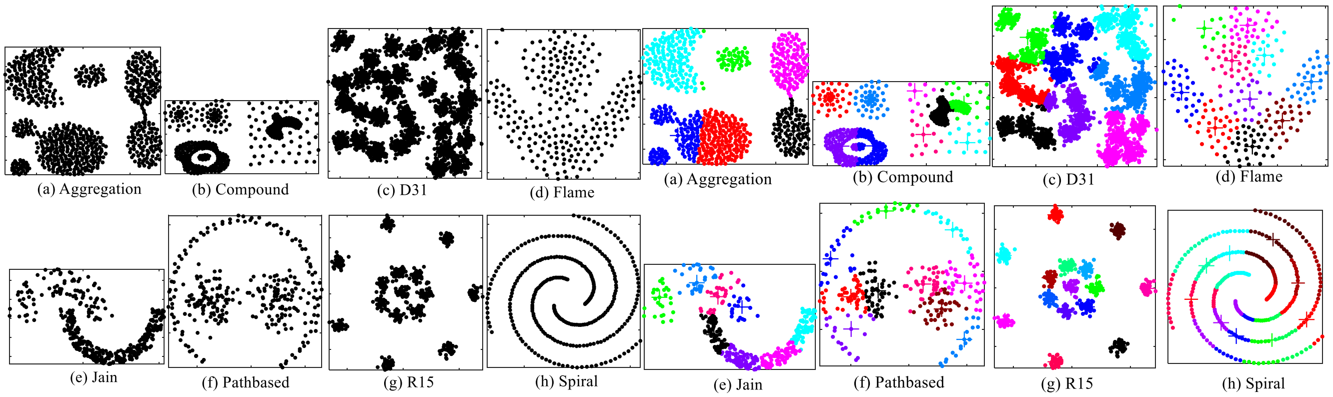


FIGURE 6. Visual results of performing the proposed clustering framework on the shape dataset.

TABLE 6. Mean and Standard Deviation of the Sum of Intra/Inter-Cluster Distances, DB-Index and Distortion Deviation Over 20 Independent Runs for the Proposed Framework, DCPSO, GCUK, K-means++, DBSCAN, EM, and NPIR for the shape datasets.

Dataset	Algorithm Measure	Proposed	DCPSO	GCUK	K-means++	DBSCAN	EM	NPIR
Aggregation	Intra-C Distances	3182.75 ± 124.67	3232.18 ± 262.5	3794.60 ± 110.40	2936.75 ± 2.25	3807.66 ± 536.65	3250.73 ± 285.80	3158.3 ± 142.19
	Inter-C Distances	327.39 ± 7.62	312.59 ± 11.21	374.62 ± 482.97	270.68 ± 0.49	207.02 ± 96.88	271.12 ± 13.47	307.10 ± 52.86
	DB Index	**0.57 ± 0.01**	0.63 ± 0.05	0.69 ± 0.06	0.64 ± 0.001	0.65 ± 0.06	0.65 ± 0.07	0.85 ± 0.06
	Distortion Deviation	**1.69 ± 0.06**	4.8 ± 1.58	6.50 ± 1.45	1.72 ± 0.49	10.77 ± 1.28	7.06 ± 2.10	7.73 ± 0.53
Compound	Intra-C Distances	1449.76 ± 458.32	1507.7 ± 135.48	1491.43 ± 302.96	1158.82 ± 61.31	870.87 ± 0.00	1247.61 ± 79.60	1335.15 ± 41.53
	Inter-C Distances	483.56 ± 307.79	303.39 ± 42.11	798.29 ± 453.21	233.76 ±7.08	130.83 ± 0.00	222.48 ± 18.02	179.76 ± 37.48
	DB Index	**0.66 ± 0.07**	0.72 ± 0.07	0.74 ± 0.08	0.79 ± 0.06	3.39 ± 0.00	1.42 ± 0.74	1.06 ± 0.13
	Distortion Deviation	**1.83 ± 0.04**	3.54 ± 1.44	2.46 ± 0.75	4.81 ± 0.86	5.18 ± 0.00	6.52 ± 1.22	3.88 ± 0.14
D31	Intra-C Distances	9329.36 ± 186.35	6453.9 ± 302.71	7334.47 ± 373.03	3168.59 ± 121.41	11201.94 ± 784.87	3031.76 ± 200.29	2885.56 ± 5.86
	Inter-C Distances	394.20 ± 89.96	2791.8 ± 338.56	1870.69 ± 952.76	6169.09 ± 103.59	365.35 ± 89.62	6370.58 ± 152.51	6085.48 ± 97.31
	DB Index	0.75 ± 0.03	0.75 ± 0.02	0.79 ± 0.00	0.65 ± 0.04	0.76 ± 0.021	1.14 ± 0.18	**0.55 ± 0.01**
	Distortion Deviation	**0.37 + 0.01**	4.89 ± 0.72	4.13 ± 0.83	1.81 ± 0.25	7.54 ± 0.022	4.13 ± 0.90	1.76 ± 0.02
Flame	Intra-C Distances	318.17 ± 21.14	527.19 ± 142.83	411.27 ± 67.50	783.22 ± 5.87	730.98 ± 30.00	819.22 ± 12.69	766.06 ± 96.36
	Inter-C Distances	278.92 ± 12.02	158.44 ± 118.45	274.56 ± 156.82	5.83 ± 0.04	11.34 ± 10.51	5.54 ± 0.05	11.99 ± 13.57
	DB Index	0.76 ± 0.09	**0.69 ± 0.05**	0.76 ± 0.04	1.11 ± 0.003	2.03 ± 1.64	1.20 ± 0.02	1.10 ± 0.15
	Distortion Deviation	**0.48 ± 0.00**	2.14 ± 0.69	2.16 ± 0.69	0.93 ± 0.22	5.04 ± 1.44	1.20 ± 0.60	2.48 ± 1.25
Jain	Intra-C Distances	1006.77 ± 81.14	1819.49± 746.28	1427.06 ± 406.96	2622.99 ± 3.81	2808.77 ± 11.29	2739.39 ± 0.00	2647.30 ± 60.98
	Inter-C Distances	653.40 ± 34.02	268.53 ± 181.08	553.44 ± 374.15	18.00 ± 0.01	53.97 ± 0.09	17.57 ± 3.77	17.78 ± 0.05
	DB Index	**0.64 ± 0.03**	**0.64 ± 0.03**	0.69 ± 0.09	0.78 ± 0.00	0.78 ± 0.00	0.74 ± 1.39	0.76 ± 0.01
	Distortion Deviation	**1.04 ± 0.001**	4.03 ± 2.21	4.89 ± 2.30	2.41 ± 0.13	53.97 ± 0.09	9.32 ± 0.00	6.99 ± 1.59
Pathbsed	Intra-C Distances	930.41 ± 65.31	1266.95 ± 215.84	975.13 ± 184.00	1435.39 ± 1.37	1527.55 ± 106.33	1468.76 ± 0.00	1699.18 ± 1.45
	Inter-C Distances	664.53 ± 47.95	340.25 ± 205.02	908.05 ± 464.93	46.99 ± 0.07	46.63 ± 21.06	47.53 ± 0.00	15.04 ± 0.01
	DB Index	**0.66 ± 0.00**	**0.66 ± 0.03**	0.73 ± 0.03	**0.66 ± 0.00**	1.67 ± 0.41	0.68 ± 0.07	0.75 ± 0.00
	Distortion Deviation	**0.57 ± 0.18**	3.30 ± 1.54	4.35 ± 1.72	1.20 ± 0.15	9.82 ± 0.70	7.15 ± 0.00	0.60 ± 0.27
R15	Intra-C Distances	501.25 ± 11.01	508. 41 ± 41.02	512.16 ± 49.00	229.01± 11.82	249.52 ±3.07	273.20 ± 24.11	231.35 ± 7.18
	Inter-C Distances	655.50 ± 59.00	622.21 ± 48.15	655.51 ± 63.12	643.96 ±11.30	572.44 ± 0.27	674.47 ± 25.33	639.95 ± 0.39
	DB Index	0.36 ± 0.07	0.41 ± 0.04	0.42 ± 0.06	**0.33 ± 0.02**	0.37 ±0.00	0.51 ± 0.09	**0.33 ± 0.02**
	Distortion Deviation	**0.34 ± 00**	0.78 ± 0.10	0.71 ± 0.00	0.34 ± 0.18	0.84 ± 0.07	1.27 ± 0.27	0.57 ± 0.20
Spiral	Intra-C Distances	1781.72 ± 121.01	1823.39 ± 253.34	1968.14±197.68	1815.35 ± 0.12	2907.78 ± 31.75	1829.30 ± 0.00	1765.12 ± 70.26
	Inter-C Distances	310.41 ± 56.45	179.19 ± 79.41	123.28 ± 32.74	40.47 ± 0.04	9.94 ± 0.04	36.39 ± 0.00	64.25 ± 19.26
	DB Index	**0.74 ±0.02**	**0.74 ± 0.01**	**0.74 ± 0.00**	0.87 ± 0.003	5.94 ± 0.07	0.99 ± 0.00	0.94 ± 0.02
	Distortion Deviation	0.84 ± 0.21	2.15 ± 0.98	1.98 ± 0.99	**0.34 ± 0.09**	0.71 ± 0.08	0.83 ± 0.00	4.96 ± 2.64

in some cases; for example, D31, which contains several spherical clusters with high overlap. The proposed framework has divided D31 into fewer clusters compared to others. Consequently, the sum of intra-cluster distances has significantly increased, while inter-cluster distances have considerably decreased. As a result, we have reached a higher

TABLE 7. Mean and Standard Deviation of the Sum of Intra/Inter-Cluster Distances, DB-Index and Distortion Deviation Over 20 Independent Runs for the Proposed Framework, DCPSO, GCUK, K-means++, DBSCAN, EM, and NPIR for the Real-world Datasets.

Dataset	Algorithm Measure	Proposed	DCPSO	GCUK	K-means++	DBSCAN	EM	NPIR
Appendicitis	Intra-C Distances	66.26 ± 5.76	72.37 ± 5.91	65.30 ± 11.78	38.32 ± 0.07	36.23 ± 0.34	38.79 ± 0.10	36.86 ± 2.24
	Inter-C Distances	1.86 ± 0.11	1.58 ± 0.12	11.23 ± 9.46	0.75 ± 0.01	1.74 ± 0.01	0.73 ± 0.00	1.76 ± 0.77
	DB Index	**0.74 ± 0.02**	0.89 ± 0.02	0.93 ± 0.10	1.008 ± 0.03	0.97 ± 0.01	1.06 ± 0.01	0.97 ± 0.30
	Distortion Deviation	0.20 ± 0.09	0.59 ± 0.22	0.53 ± 0.26	0.20 ± 0.10	1.73 ± 0.00	**0.13 ± 0.01**	0.57 ± 0.08
Dermatology	Intra-C Distances	2977.37 ± 170.39	2987 ± 272.11	3172.57 ± 239.09	2038.78 ± 38.63	2332.97 ± 114.82	3554.73 ± 303.89	2282.80 ± 105.79
	Inter-C Distances	1205.72 ± 546.38	587.61 ± 130.39	450.65 ± 63.64	396.82 ± 18.68	771.54 ± 65.74	291.70 ± 31.54	262.38 ± 54.15
	DB Index	**1.01 ± 0.02**	1.06 ± 0.03	1.02 ± 0.05	**1.01 ± 0.02**	1.41 ± 0.11	2.56 ± 0.59	1.67 ± 0.66
	Distortion Deviation	**3.53 ± 0.87**	5.30 ± 0.78	4.05 ± 2.13	4.02 ± 1.30	19.27 ± 1.30	27.20 ± 4.27	12.02 ± 2.11
Ecoli	Intra-C Distances	13903.66 ± 1764.31	15728.53 ± 2307.41	14829.25 ± 2254.47	10028.6 ± 177.57	10047.59 ± 190.55	11503.37 ± 809.12	12695.86 ± 357.77
	Inter-C Distances	155.95 ± 11.72	1848.06 ± 1890.39	3036.63 ± 2337.28	950.00 ± 76.92	957.17 ± 71.11	1792.92 ± 72.61	105.71 ± 58.31
	DB Index	**0.90 ± 0.01**	0.91 ± 0.06	1.06 ± 0.06	1.35 ± 0.11	1.34 ± 0.02	2.20 ± 0.57	1.38 ± 0.19
	Distortion Deviation	**5.27 ± 3.44**	34.38 ± 17.27	30.87 ± 14.87	28.66 ± 5.73	18.19 ± 0.33	44.08 ± 10.60	18.77 ± 5.01
Glass	Intra-C Distances	375.61 ± 87.01	333.49 ± 26.36	498.65 ± 28.94	243.25 ± 6.30	187.94 ± 0.93	233.47 ± 13.11	226.16 ± 3.21
	Inter-C Distances	124.82 ± 23.72	154.08 ± 32.01	157.98 ± 29.66	134.86 ± 6.66	19.69 ± 2.79	109.75 ± 10.03	119.69 ± 0.12
	DB Index	0.68 ± 0.18	**0.60 ± 0.13**	0.88 ± 0.02	0.72 ± 0.09	0.83 ± 0.13	1.16 ± 0.22	1.50 ± 0.07
	Distortion Deviation	**3.30 ± 0.92**	4.31 ± 1.08	4.85 ± 1.06	4.31 ± 0.40	4.33 ± 0.04	4.27±1.02	3.78 ± 0.02
Haberman	Intra-C Distances	2001.48 ± 124.67	2964.39 ± 595.71	2599.60 ± 646.60	3053.93 ± 253.51	2472.11 ± 0.00	3335.36±0.00	2715.88 ± 0.00
	Inter-C Distances	152.19 ± 2.32	301.23 ± 281.28	699.66 ± 664.06	17.21 ± 0.34	11.19 ± 0.00	8.70±0.00	18.55 ± 0.00
	DB Index	**0.59 ± 0.00**	0.63 ± 0.05	0.72 ± 0.07	1.22 ± 0.15	1.13 ± 0.00	2.54±0.00	0.91 ± 0.00
	Distortion Deviation	7.49 ± 1.5	13.49 ± 4.58	11.53 ± 4.57	**5.61 ± 3.65**	21.18 ± 0.00	19.72±0.00	22.48 ± 0.00
Housevotes	Intra-C Distances	341.34 ± 19.54	399.69 ± 21.14	408.34 ± 20.25	332.74 ± 0.05	304.69 ± 0.09	332.97±0.00	449.96 ± 0.34
	Inter-C Distances	31.93 ± 0.14	34.54 ± 28.88	102.12 ± 34.43	2.54 ± 0.00	2.66 ± 0.00	2.54±0.00	0.67 ± 0.09
	DB Index	1.26 ± 0.06	1.44 ± 0.19	1.97 ± 0.07	1.13 ± 0.00	**1.04 ± 0.00**	1.13 ± 0.00	5.81 ± 0.91
	Distortion Deviation	**0.07 ± 0.4**	0.29 ± 0.12	0.39 ± 0.16	0.10 ± 0.01	0.13 ± 0.09	0.12 ± 0.00	0.18 ± 0.02
Ionosphere	Intra-C Distances	1209.76 ± 74.81	1427.34 ± 66.54	1421.71 ± 52.38	831.44 ± 71.84	478.61 ± 1.07	904.03 ± 0.73	845.40 ± 3.52
	Inter-C Distances	77.53 ± 0.34	85.16 ± 5672	261.72 ± 132.08	3.64 ± 1.16	2.40 ± 0.00	1.88 ± 0.00	11.08 ± 0.02
	DB Index	1.24 ± 0.02	1.34 ± 0.07	1.70 ± 0.04	1.30 ± 0.42	1.21 ± 0.00	2.82 ± 0.01	**1.17 ± 0.01**
	Distortion Deviation	**0.28 ± 0.14**	1.42 ± 0.31	1.42 ± 0.47	0.74 ± 0.38	3.45 ± 0.00	0.92 ± 0.08	2.16 ± 0.54
Iris	Intra-C Distances	156.96 ± 7.88	132.19 ± 10.37	138.85 ± 11.76	97.33 ± 0.00	90.76 ± 11.71	104.17± 8.66	128.14 ± 0.00
	Inter-C Distances	4.88 ± 0.16	10.95 ± 6.64	7.76 ± 8.37	10.15 ± 0.01	4.65 ± 1.85	9.30± 0.21	3.97 ± 0.01
	DB Index	0.43 ± 0.008	0.55 ± 0.9	0.44 ± 0.10	0.66 ± 0.00	0.40 ± 0.15	0.77±0.03	**0.38 ± 0.00**
	Distortion Deviation	**0.20 ± 0.05**	0.60 ± 0.28	0.31 ± 0.12	0.41 ± 0.00	0.57 ± 0.19	0.89±0.35	1.31 ± 0.00
Segment	Intra-C Distances	246834.2 ± 54297.3	373633.9 ± 72569.9	477924.2 ± 38190.21	270181.7 ± 24659	124128.85 ± 1293.78	204704.1 ± 9713.0	160144.48 ± 6911.76
	Inter-C Distances	6083. 25 ±2971.2	15215.4 ± 2472.0	25575.8 ± 8408.23	17401.93± 85.96	3529.28 ± 375.08	7184.62±2785.47	2736.35 ± 656.41
	DB Index	**0.50 ± 0.09**	0.54 ± 0.14	0.78 ± 0.05	0.69 ± 0.01	1.01 ± 0.02	2.97±1.11	1.32 ± 0.09
	Distortion Deviation	**110.01 ± 45.87**	361.76 ± 21.20	325.81 ± 38.03	122.02 ± 22.19	117.66 ± 3.35	791.22±271.72	1330.52 ± 6.37
Vehicle	Intra-C Distances	107956.2 ± 7782.0	73457.05 ± 6883.7	84066.13 ± 6907.29	57194.51±3885.5	16856.20 ± 56.29	55696.8 ± 1679.9	65103.75 ± 66.64
	Inter-C Distances	527.74 ± 41088	845.52 ± 1249.52	3894.19 ± 5314.35	2227.86 ± 160.46	1684.29 ± 5.92	1822.56±344.56	358.91 ± 0.35
	DB Index	0.54 ± 0.00	0.48 ± 0.03	0.72 ± 0.22	0.65 ± 0.017	0.63 ± 0.01	0.92 ± 0.09	**0.44 ± 0.00**
	Distortion Deviation	**22.40 ± 8.67**	22.49 ± 9.90	56.65 ± 13.90	81.15 ± 41.96	46.02 ± 1.73	167.30 ± 40.61	99.04 ± 0.66
Wdbc	Intra-C Distances	262664.86 ± 9839.3	188642.3 ± 52069.3	221180.21 ± 53610.8	152647.25 ± 0.00	93540.44 ± 1601.99	175895.6 ± 0.00	175454.44 ± 0.00
	Inter-C Distances	3655.04 ± 259.66	10456.4 ± 9420.0	12662.2 ± 16499.91	1331.33 ± 0.00	785.98 ± 9.00	1030.10±0.00	2112.74 ± 0.00
	DB Index	0.51 ± 0.00	0.52 ± 0.05	0.59 ± 0.13	0.50 ± 0.00	**0.40 ± 0.01**	0.70 ± 0.00	0.71 ± 0.00
	Distortion Deviation	**69.58 ± 35.89**	528.84 ± 171.05	438.02 ± 125.45	2215.79 ± 0.00	436.29 ± 22.41	2558.05 ± 0.00	3260.76 ± 0.00
Wine	Intra-C Distances	17220.70 ± 663.29	17252.8 ± 895.49	19242.17 ± 6756.86	17966.63 ± 835.77	21726.14 ± 6404.70	22874.2 ± 338.82	22534.65 ± 6781.72
	Inter-C Distances	7288.33 ± 21.43	7065.07 ± 6736.62	84142.17 ± 8529.20	1549.71 ± 44.71	1173.08 ± 355.95	1230.15 ± 17.19	1023.02 ± 511.78
	DB Index	**0.47 ± 0.01**	0.48 ± 0.01	0.55 ± 0.04	0.54 ± 0.006	0.59 ± 0.01	0.84 ± 0.02	0.58 ± 0.05
	Distortion Deviation	**16.96 ± 4.71**	53.33 ± 16.22	82.94 ± 37.62	205.32 ± 58.74	183.88 ± 9.15	387.50±147.17	423.99 ± 135.53

TABLE 8. Mean and Standard Deviation of the Sum of Intra/Inter-Cluster Distances, DB-Index and Distortion Deviation Over 20 Independent Runs for the Proposed Framework, DCPSO, GCUK, K-MENAS++, DBSCAN, EM, and NPIR for the Higher-Dimensional Datasets.

Dataset	Algorithm Measure	Proposed	DCPSO	GCUK	K-means++	DBSCAN	EM	NPIR
Dime064	Intra-C Distances	122307.87 ± 136.66	21901.45 ± 516.94	21454.51 ± 599.04	12149.84 ± 246.00	11958.65 ± 202.65	12014.33 ± 277.24	11677.26 ± 933.38
	Inter-C Distances	143803.46 ± 1169.10	9475.02 ± 519.77	10331.19 ± 628.54	144819.30 ± 475.55	144915.87 ± 484.40	143968.13 ± 499.43	144074.93 ± 2259.14
	DB Index	**0.04 ± 0.00**	0.20 ± 0.04	0.14 ± 0.04	**0.04 ± 0.00**	0.08 ± 0.01	0.06 ± 0.01	0.09 ± 0.03
	Distortion Deviation	**26.82 ± 2.88**	45.81 ± 5.06	46.22 ± 4.42	35.46 ± 1.81	36.82 ± 3.29	39.61 ± 2.69	40.44 ± 2.15
Dime128	Intra-C Distances	138102.07 ± 227.48	11852.50 ± 683.21	12730.87 ± 703.25	13690.57 ± 358.88	13181.59 ± 309.60	13416.67 ± 467.11	12935.25 ± 1060.25
	Inter-C Distances	18488.69 ± 408.67	201642.50 ± 795.77	21486.41 ± 707.19	205577.92 ± 612.40	205765.65 ± 973.78	205785.09 ± 739.41	205649.62 ± 2010.96
	DB Index	0.07 ± 0.00	1.17 ± 0.23	1.42 ± 0.11	**0.04 ± 0.00**	0.10 ± 0.01	**0.04 ± 0.00**	1.15 ± 0.55
	Distortion Deviation	**30.55 ± 1.65**	78.37 ± 13.54	71.26 ± 14.56	46.91 ± 1.91	35.41 ± 4.16	47.85 ± 2.79	54.46 ± 6.07
Dime256	Intra-C Distances	14105.69 ± 220.82	18457.17 ± 1463.47	19928.46 ± 2129.47	13974.62 ± 164.43	13400.60 ± 242.16	13985.81 ± 226.55	14203.75 ± 920.49
	Inter-C Distances	27070.40 ± 914.45	19412.74 ± 1564.95	18546.85 ± 2180.76	296376.21 ± 765.56	296078.75 ± 1087.07	286227.20 ± 1104.29	276228.65 ± 1520.49
	DB Index	0.04 ± 0.00	0.10 ± 0.03	0.09 ± 0.04	**0.02 ± 0.00**	0.07 ± 0.01	0.04 ± 0.00	0.03 ± 0.00
	Distortion Deviation	**18.00 ± 3.64**	43.06 ± 7.39	41.84 ± 8.51	22.60 ± 2.67	24.83 ± 2.29	26.62 ± 3.03	25.72 ± 4.17
Dime512	Intra-C Distances	184041.45 ± 398.61	190629.83 ± 13235.42	198168.38 ± 14725.03	16022.35 ± 436.29	15106.74 ± 1798.74	152965.83 ± 939.93	151784.53 ± 1176.09
	Inter-C Distances	38553.45 ± 1364.09	384915.26 ± 17197.38	395860.59 ± 13171.77	418384.62 ± 684.90	420203.11 ± 2817.58	461964.34 ± 3321.19	471126.27 ± 2768.98
	DB Index	0.15 ± 0.05	1.01 ± 0.03	2.16 ± 0.13	**0.02 ± 0.01**	0.41 ± 0.07	0.68 ± 0.07	0.67 ± 0.09
	Distortion Deviation	**16.78 ± 5.30**	78.61 ± 4.49	75.12 ± 6.42	27.71 ± 0.74	40.41 ± 3.42	38.12 ± 5.97	35.85 ± 4.28

DB-index. Yet, we have demonstrated the best performance in terms of the distortion deviation in this dataset.

K-means++ has shown highly competitive performance in the Pathbased and the dermatology datasets in terms of the DB-index. It has also been able to achieve better performance in distortion deviation on spiral and Haberman datasets. However, by increasing the dimension and the data points as shown in higher-dimensional datasets, the proposed framework outperforms the K-means++ in terms of the distortion deviation measure. The proposed algorithm has achieved the best DB-index in the Appendicitis dataset, but the EM algorithm reached the best distortion deviation in this case.

On the other hand, the proposed framework has shown highly successful results in the automatic k-determination clustering methods (DBSCAN, DCPSO, and GUCK) in most datasets. DBSCAN algorithm has reached the best value of DB-index in the Housevotes and WDBC dataset. Nevertheless, the proposed method has obtained the best distortion deviation among all other algorithms in both datasets. It should be mentioned that DBSCAN is fundamentally different from center-based clustering methods. Although DB-index may not be considered a fair validity measure for this clustering method, we needed to evaluate the proposed solution in terms of the DB-index with other algorithms for the purpose of this paper. Aside from this point, the algorithm has not performed well in reaching the minimum possible value of distortion deviation compared to the other algorithms.

In a few datasets (such as Jain, Pathbased, and Spiral), the DCPSO and the GCUK algorithms have reached the same value as the proposed solution in the DB index. Nevertheless, the distortion deviation, which is a primary goal in this paper, still yields lower figures in these datasets for the proposed framework.

To validate the above numerical results, we have performed a non-parametric statistical test called the Friedman test [63], [64]. This test which is similar to the ANOVA [13], can point out significant differences between the behavior of two or more algorithms. Table 9 describes the achieved ranks by the Friedman test in the proposed framework considering different optimizer modules. The ranks indicate that all four optimizer modules have shown competitive performance, especially among the BBA, BGA, and BPSO. However, BBA and BGA optimizer modules have been ranked better in most datasets with our proposed framework. Then, we have performed another statistical test called the Wilcoxon rank-sum test [65], [66] to draw a more meaningful conclusion on the results. The Wilcoxon rank-sum test for equal medians establishes a proper pairwise comparison between the algorithms. It compares the null hypothesis that two values are samples from a continuous distribution with equal medians against the alternative that they are not. For evaluating the first set of comparison results, the Wilcoxon test has

TABLE 9. Achieved Ranks by the Friedman Test for the Proposed Framework Considering four Different Optimizer Modules.

Datasets	BBA	BPSO	BGA	BDA
Aggregation	**1.65**	2.65	2.08	3.63
Compound	2.75	2.23	**1.58**	3.45
D31	**1.70**	2.70	2.13	3.48
Flame	2.10	2.78	**1.93**	3.20
Jain	2.45	**2.08**	2.75	2.73
Pathbased	**2.10**	2.60	2.25	3.05
R15	2.90	2.55	**1.68**	2.88
Spiral	2.45	2.65	**2.10**	2.80
Appendicitis	**2.00**	2.35	2.70	2.95
Dermatology	**1.98**	2.48	2.35	3.20
Ecoli	2.25	2.75	**2.10**	2.90
Glass	**2.20**	2.70	2.50	2.60
Haberman	2.75	2.55	**2.35**	2.35
Housevotes	2.15	2.40	**1.85**	3.60
Ionosphere	**2.05**	2.55	2.20	3.20
Iris	**2.30**	2.40	2.45	2.85
Segment	**2.20**	2.60	2.68	2.53
Vehicle	2.38	2.78	**1.83**	3.03
Wdbc	**2.20**	2.55	2.65	2.60
Wine	2.30	**1.63**	2.65	3.43
Dim64	**2.03**	2.43	2.73	2.83
Dim128	**2.08**	2.48	2.75	2.70
Dim256	**2.40**	2.50	2.50	2.60
Dime512	**2.30**	**2.40**	**2.55**	**2.75**

TABLE 10. Achieved P-values by the Wilcoxon Rank-Sum Test for the Proposed Framework Considering Different Optimizer Module.

Datasets	BBA	BPSO	BGA	BDA
Aggregation	1	0.0077	0.4849	0.0000
Compound	0.0000	0.2235	1	0.0000
D31	1	0.0006	0.2439	0.0000
Flame	0.6073	0.0032	1	0.0005
Jain	0.2288	1	0.0711	0.0265
Pathbased	1	0.0496	0.6262	0.0149
R15	0.0023	0.0110	1	0.0009
Spiral	0.2733	0.0909	1	0.0468
Appendicitis	1	0.4092	0.0482	0.0058
Dermatology	1	0.2789	0.4986	0.0087
Ecoli	1	0.1636	0.8817	0.0133
Glass	1	0.0251	0.0914	0.0482
Haberman	0.1960	0.5065	1	1.0000
Housevotes	0.3104	0.1988	1.0000	0.0000
Ionosphere	1	0.1895	0.4249	0.0005
Iris	1	0.2972	0.5883	0.0138
Segment	1	0.0671	0.0335	0.1573
Vehicle	0.1410	0.0129	1	0.0029
Wdbc	1	0.9246	0.6750	0.7972
Wine	0.0441	1.0000	0.0088	0.0000
Dim64	1	0.1774	0.0094	0.0019
Dim128	1	0.1794	0.0097	0.0226
Dim256	1	0.3421	0.3421	0.1624
Dime512	1	0.2733	0.0565	0.0962

been applied to the optimizer module with the best average performance against the rest of the modules according to Tables 3 to 5. The significance level is considered as 0.05, which gives strong evidence against the null hypothesis. The achieved p-values by the Wilcoxon test for the first set of comparison results considering different optimizer modules have been reported in Table 10. As can be seen in some datasets (i.e., R15, Wine), the p-value of one of the optimizer modules has a significant difference from others. On the other hand, the achieved p-values by two or three optimizer modules are not significantly different in some other datasets (i.e., Pathbased, Dermatology, Haberman, Housevotes, Wdbc, Dim256). It means the proposed framework utilizing all these optimizer modules exhibits similar performance for that dataset.

We have also applied these non-parametric statistical tests to the second set of comparison results to compare the proposed framework with other algorithms statistically. During this experiment, we have implemented the proposed framework considering the BBA optimizer. We have applied the statistical tests to the DB-index and distortion deviation measures achieved in the second set of comparison results. The achieved ranks by the Friedman test and the p-values by the Wilcoxon test for the second set of comparison results have been reported in Tables 11 and 12. As shown, the proposed framework has been able to reach the best or the second-best DB-index rank in multiple datasets. Besides, the best distortion deviation rank in almost all datasets has been achieved by the proposed framework, which is a great success. The reason why we have not achieved the lowest DB-index rank in a few datasets lies in how the problem has been formulated in our model. In line with the initial motivation of this paper, reaching the minimum distortion deviation has been prioritized in our proposed model. Therefore, there might be cases where the proposed framework achieves the minimum distortion by dividing the data points into more or fewer groups; thus, it also affects the DB-index.

We have observed the performance of the proposed framework in some of the datasets which have higher data points and larger dimensions. (e.g., D31, Segment, and Dim datasets). From our perspective, the proposed framework effectively works with small to mid-size datasets from different dimensions. We come to this conclusion that having a dataset with high data points can slightly affect the performance as it is directly related to the binary encoding scheme and can increase the search space size while increasing the dimension only affects distance calculation, which is normally expected to happen. Also, as discussed in [59], it is worth mentioning that using Euclidean distance is not a proper distance measure in higher-dimensional datasets. Hence, we can substitute this measure with other appropriate distance measures for further investigation.

TABLE 11. Achieved Ranks by the Friedman Test on the DB-index and Distortion Deviation Measures for the Proposed Framework Compared to Other Algorithms.

Dataset	DB-index ranks							Distortion deviation ranks						
	Proposed	DCPSO	GCUK	K-means++	DBSCAN	EM	NPIR	Proposed	DCPSO	GCUK	K-means++	DBSCAN	EM	NPIR
Aggregation	**1.2**	3.6	4.8	4.4	3.95	3.2	6.85	**1.45**	3.35	4.65	1.55	6.85	4.85	5.3
Compound	**1.65**	2.5	2.6	3.4	7	5.3	5.55	**1.4**	3.25	2.15	5.35	5.5	6.7	3.65
D31	3.95	3.9	5.75	1.95	4.4	7	**1.05**	**1**	6.65	6.05	3.45	2.1	5.3	3.45
Flame	2.35	**1.5**	2.35	4.6	5.25	6.75	5.2	**1.05**	4.95	4.55	2.55	6.85	2.8	5.25
Jain	**1.6**	1.85	3.55	6.35	6.25	3.8	4.6	**1.1**	3.35	3.55	2.25	7	5.9	4.85
Pathbased	**1.9**	2.35	4.95	3.25	7	3	5.55	**1.45**	4.25	4.75	3	6.95	5.95	1.65
R15	3.05	5	5.35	2.05	3.7	6.65	**2.2**	2.15	4.95	3.9	**1.4**	5.45	6.7	3.45
Spiral	2.35	1.9	**1.75**	4	7	6	5	3.5	5.45	5.3	**1.15**	2.85	3.65	6.1
Appendicitis	**1.45**	2.8	3.75	4.9	4.45	6.35	4.3	2.25	5.1	4.65	2.35	7	**1.8**	4.85
Dermatology	2.35	3.45	2.35	**1.85**	5.65	6.65	5.7	**1.95**	3.3	2.35	2.4	6.05	6.95	5
Ecoli	**1.4**	1.65	2.95	5.05	5.1	7	4.85	**1.05**	5.2	4.65	4.9	2.7	6.25	3.25
Glass	2.55	**1.7**	4.55	2.65	3.65	6	6.9	**2**	4.25	5.15	4.5	4.95	4.6	2.55
Haberman	**1.15**	2	2.85	5.75	5.25	7	4	**2.3**	3.7	2.95	1.3	5.9	4.85	7
Housevotes	3.9	4.75	6	3	**1.05**	2.3	7	**1.9**	5.85	6.4	2.25	3.5	3.25	4.85
Ionosphere	3.2	3.95	6	4.2	2.45	7	**1.2**	**1.2**	4.55	4.2	2.35	7	3	5.7
Iris	3.45	4.65	3.3	5.7	1.85	7	**2.05**	**1.5**	4.25	2.45	3.2	4.1	5.7	6.8
Segment	**1.4**	1.85	3.9	2.85	5	7	6	1.85	4.75	4.25	2.4	**1.75**	6	7
Vehicle	3.25	2.1	4.95	5.3	4.5	6.8	**1.1**	**1.3**	1.7	4	5.25	3.25	6.75	5.75
Wdbc	3.65	3.5	4.55	2.6	**1.1**	5.8	6.8	**1**	3.3	3.05	5	2.65	6	7
Wine	**1.5**	1.7	3.9	3.875	5.35	7	4.675	**1**	2.15	2.85	4.375	4.75	6.25	6.625
Dim64	**1.45**	6.85	5.95	1.75	4.40	3.30	4.30	**1.00**	6.00	6.40	2.60	2.70	4.30	5.00
Dim128	3.85	6.20	6.80	2.30	4.95	**1.25**	2.65	**1.10**	6.50	6.25	3.50	1.90	3.70	5.05
Dim256	3.70	6.25	5.75	**1.00**	5.55	3.70	2.05	**1.30**	6.50	6.40	2.55	3.35	4.20	3.70
Dim512	2.00	6.30	6.70	**1.00**	3.00	4.50	4.50	**1.00**	6.60	6.40	2.05	4.45	3.95	3.55

VI. CORRELATION-AWARE CLUSTERING SCHEME

Without loss of generality, the proposed clustering framework has been applied to a set of correlated binary datasets as a case study in this section. The presence of correlation in a binary dataset can be realized as the relevance of content files in the same category, such as the repeated measurements in remote sensing [52], the updated versions of dynamic content, augmented reality, news updates, etc. [3], [22]. Crowd-sourced multi-view video uploading is another application that can benefit from correlation among binary datasets [23]. Moreover, correlated binary data clustering is widely used in medical studies, such as dental and radiologic studies. In such cases, the observations are taken from multiple representations of the same subject [67].

In the binary case, each data point is denoted by an n-bit vector with i.i.d binary random symbols. Therefore, the n-dimensional binary dataset with m data points is indicated by A.

$$A = \begin{bmatrix} p_{1,1}, p_{1,2} & \cdots & , p_{1,n} \\ \vdots & \vdots & \vdots \\ p_{m,1}, p_{m,2} & \cdots & , p_{m,n} \end{bmatrix}_{m\times n} \tag{13}$$

Since the proposed clustering approach has been designed as a general framework, it can also successfully cope with binary datasets. Hence, all the steps are similar to the general framework. However, the definition of *distance measure* and *representative selection* has been tailored with binary space.

The distance measure is considered as the Hamming distance [3], which is defined as (14).

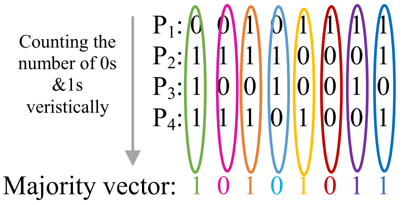
$$d^H_{i,j} = \begin{cases} 1 & \text{if } \mathrm{p}_{i,n} \neq \mathrm{p}_{j,n}, \quad \forall i \neq j \\ 0 & \text{Otherwise} \end{cases} \tag{14}$$

In this case, the maximum distortion of each cluster is calculated according to the Hamming distance measure between each data point P and its representative Y_i within each cluster C_i where $C_i \in C$ and $i = [1,2,\ldots,K]$.

$$\delta^{C_i} = \max_{C_i} \mathrm{d}^H(P, Y_i) \tag{15}$$

Selecting cluster representative in the binary case is carried out in two steps. First, the centroid of each group is identified by performing the majority rule. Then, the data point with the least hamming distance to the centroid is selected as the representative. According to the majority rule, a decision is made based on the majority of alternatives. The following example shows how a centroid is determined based on the majority rule for a group of 4 binary data points.

Example 2:



Counting the number of 0s &1s veristically

P_1: 0 0 1 0 1 1 1 1
P_2: 1 1 1 1 0 0 0 1
P_3: 1 0 0 1 0 0 1 0
P_4: 1 1 1 0 1 0 0 1

Majority vector: 1 0 1 0 1 0 1 1

TABLE 12. Achieved P-values by the Wilcoxon Rank-Sum Test on the DB-index and Distortion Deviation Measures for the Proposed Framework Against Other Algorithms.

Dataset	DB-index						Distortion deviation					
	DCPSO	GCUK	K-means++	DBSCAN	EM	NPIR	DCPSO	GCUK	K-means++	DBSCAN	EM	NPIR
Aggregation	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.5801	0.0000	0.0000	0.0000
Compound	0.0071	0.0026	0.0000	0.0000	0.0000	0.0000	0.0005	0.0003	0.0000	0.0000	0.0000	0.0000
D31	0.0000	0.0001	0.0000	0.0000	0.0960	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Flame	0.0060	0.9892	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Jain	0.7353	0.0077	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Pathbased	0.8392	0.0000	0.0574	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.7338
R15	0.0043	0.0040	0.0970	0.2067	0.0000	0.2493	0.0000	0.0000	0.0000	0.0000	0.0000	0.0005
Spiral	0.0294	0.0215	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0148	0.8168	0.0001
Appendicitis	0.0000	0.0000	0.0000	0.0000	0.0000	0.5953	0.0000	0.0000	0.6488	0.0000	0.3304	0.0000
Dermatology	0.0000	0.7972	0.2499	0.0000	0.0000	0.0000	0.0000	0.6554	0.1070	0.0000	0.0000	0.0000
Ecoli	0.2287	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Glass	0.1806	0.0001	0.2731	0.0175	0.0000	0.0000	0.0060	0.0000	0.0000	0.0002	0.0016	0.0010
Haberman	0.2534	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0040	0.0000	0.0000	0.0000	0.0000
Housevotes	0.0020	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0127	0.0625	0.0001	0.0000
Ionosphere	0.0001	0.0000	0.0010	0.0001	0.0000	0.0000	0.0000	0.0000	0.0010	0.0000	0.0000	0.0000
Iris	0.0000	0.4570	0.0000	0.0002	0.0000	0.0000	0.0000	0.0006	0.0000	0.0000	0.0000	0.0000
Segment	0.6949	0.0000	0.0009	0.0000	0.0000	0.0000	0.0000	0.0000	0.1961	0.6749	0.0000	0.0000
Vehicle	0.0000	0.0040	0.0000	0.0000	0.0000	0.0000	0.6359	0.0000	0.0000	0.0000	0.0000	0.0000
Wdbc	0.4903	0.0083	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Wine	0.2534	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Dim64	0.0000	0.0000	0.3548	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Dim128	0.0000	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000
Dim256	0.0000	0.0002	0.0000	0.0000	0.3734	0.0000	0.0000	0.0000	0.0001	0.0000	0.0000	0.0000
Dim512	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

The similarity between correlated vectors is extracted by a statistic measure named the simple matching coefficient (SMC) [58], [68], which is closely related to the definition of the hamming distance on bit strings.

Let P_i and P_j be two different n-bit binary vectors in a cluster. Let n_{11} and n_{00} represent the number of bits that are 1 or 0 simultaneously among two vectors, while n_{01} and n_{10} represent the number of bits that are not the same in each position. Then SMC is defined as follows.

$$\mathrm{SMC} = \frac{n_{00} + n_{01}}{n_{00} + n_{01} + n_{10} + n_{11}} \tag{16}$$

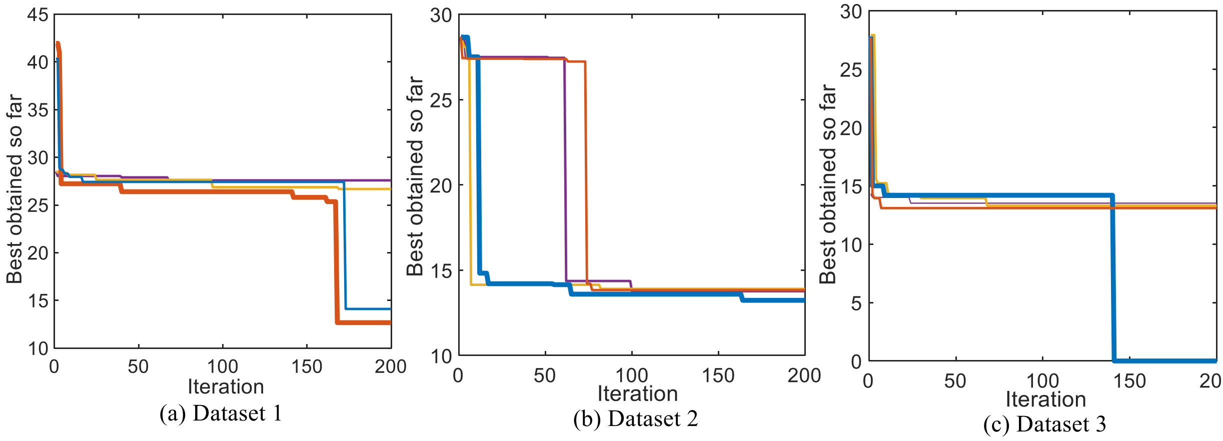
Similar to the discussed experiment in the general framework, the goal is partitioning $\mathcal{A}_{m \times n}$ into K number of compact and well-separated clusters with relatively close values for the maximum distortion in each group, such that $K \leq m$.

A. RESULTS AND DISCUSSION ON CORRELATION-AWARE CLUSTERING SCHEME

In this section, the performance of the proposed correlation-aware clustering scheme has been analyzed on a correlated binary dataset. For this purpose, based on assumptions considered in [22] for generating correlated binary vectors, three synthetic binary datasets are generated with dimensions 128 × 100 bits and a maximum similarity of 50%, 60%, and 70%, respectively. Similar to the general framework, the correlation-aware clustering scheme has been evaluated under the presence of the BBA, BPSO, BGA, and BDA optimizer modules. For each algorithm, twenty independent trials have been performed on each dataset. The best and worst cost, the average cost, and the standard deviation have been reported in Table 13. The convergence curve of binary datasets by considering all four optimizer modules is also presented in Fig. 7. Finally, the statistical analysis has been described in Table 14. According to the experimental results, all four optimizer modules show high capabilities in performing the proposed correlation-aware clustering scheme and present very competitive results in this case. However, both the BBA and BDA optimizer modules provide superb performance in solving such a binary clustering problem since the fastest convergence rate belongs to the BBA optimizer module, followed by the BDA optimizer. The BBA algorithm can take advantage of the loudness and pulse emission balance between the exploration and exploitation to accelerate the convergence rate toward the global optimum and not trap in local minima over the course of iterations. Besides, the V-shaped transfer function in the BBA algorithm helps the particles not go through the unpromising area of the search space, and therefore it contributes to having a fast convergence rate in this case. The BDA optimizer also inherits high exploration and exploitation from the DA algorithm and provide an excellent result. Furthermore, the convergence curves of this experiment show that by increasing the similarity among data points, the convergence speed significantly increases. Consequently, reaching the minimum cost can be achievable in fewer iterations. The reason is that, as the

TABLE 13. Comparison Results of the Proposed Correlation-Aware Clustering Scheme Considering BBA, BPSO, BGA, and BDA Optimizer Modules.

Dataset	BBA				BPSO				BGA				BDA			
	Best	Worst	Mean	Std	Best	Worst	Mean	Std	Best	Worst	Mean	Std	Best	Worst	Mean	Std
Dataset 1	13.2253	27.5974	17.4428	5.7571	27.5978	27.5978	21.5311	6.5529	27.6555	27.6555	18.8102	6.1719	12.6746	27.4833	**17.0180**	5.6712
Dataset 2	13.2332	13.9870	**13.7247**	0.2223	14.1375	14.1375	13.9374	0.1246	14.2618	14.2618	13.9195	0.1841	13.2595	14.0283	13.7915	0.1739
Dataset 3	0.0000	13.9319	**12.1649**	4.1725	13.9808	13.9808	13.7809	0.1579	13.9668	13.9668	13.6544	0.1850	12.9543	13.8418	13.5725	0.2520



(a) Dataset 1

(b) Dataset 2

(c) Dataset 3

FIGURE 7. Convergence curve for the correlated-binary datasets considering BBA, BPSO, BGA, and BDA algorithms for the optimizer module in the correlation-aware clustering scheme. The algorithm with the better performance is shown with thicker line. The color code for each algorithm is as follows: — BPSO — BGA — BDA — BBA .

TABLE 14. Statistical Results of the Proposed Correlation-Aware Clustering Scheme Considering four Different Optimizer Modules.

Dataset	Friedman's Rank				P-values			
	BBA	BPSO	BGA	BDA	BBA	BPSO	BGA	BDA
DATASET 1	2.35	3.25	2.7	**1.7**	0.1333	0.0003	0.0071	1
DATASET 2	**2.05**	3.05	2.8	2.1	1	0.0026	0.0051	0.4249
DATASET 3	**1.85**	3.5	2.45	2.2	1	0.0016	0.2184	0.4093

correlation among datasets increases, the similarity between data points becomes very large. As a result, the maximum Hamming distance between data points decreases; therefore, the clustering problem becomes a much simpler problem that can even be solved in less than half of the iterations. In such cases, the maximum distortion in each cluster will be decreased as well.

VII. CONCLUSION AND FUTURE WORK

Clustering algorithms are developed as a powerful tool to analyze the massive amount of data produced by modern applications. Over the years, various meta-heuristic searching techniques have been proposed to achieve optimal or near-optimal solutions due to the challenges such as defining a suitable objective function and ambiguity in data clustering definition. In this paper, the clustering problem has been formulated as an optimization problem with the motivation to reach well-separated clusters with approximately the same maximum distortion. Such clustering solution is highly beneficial in applications such as compression schemes that need relatively close values for the maximum distortion in each cluster. The proposed framework employs binary optimization algorithms as the optimizer module. It also adopts a dynamic range of clusters in accordance with the input data to tackle the problem of determining the correct number of clusters in advance. Hence, users do not need a priori knowledge of the number of clusters. We have also proposed a binary encoding scheme for the particle representation in the proposed framework. Moreover, a correlation-aware clustering scheme is reported as the application of the proposed framework for the correlated binary datasets. The binary correlation-aware clustering scheme is useful in a wide range of practical applications such as the repeated measurements in remote sensing, medical studies, cache-aided networks with dynamic content, augmented reality, and crowdsourced multi-view video uploading. The experimental results show that the proposed framework exhibits superior performance in all binary datasets and most of the typical datasets by utilizing the considered binary optimizer module. According to the results, we have successfully reached a proper number of well-separated clusters with approximately the same maximum distortion value for each cluster in most datasets. This paper can be considered the opening to conduct further research to improve the distortion gap between clusters in other applications. Future studies can consider the proposed automatic clustering framework with approximately the same maximum distortion in each cluster as a multi-objective optimization algorithm problem and consider the maximum distortion of the clusters as an objective to possibly improved the distortion deviation gap. Besides, improving the efficiency of the proposed framework in the case of having a large dataset can be considered a future work.

REFERENCES

[1] M. Sharma and J. K. Chhabra, ''Sustainable automatic data clustering using hybrid PSO algorithm with mutation,'' *Sustain. Comput. Informat. Syst.*, vol. 23, pp. 144–157, Sep. 2019, doi: 10.1016/j.suscom.2019.07.009.

[2] X. Cao, L. Liu, Y. Cheng, and X. Shen, ''Towards energy-efficient wireless networking in the big data era: A survey,'' *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 303–332, 1st Quart., 2018, doi: 10.1109/COMST.2017.2771534.

[3] C. Aggarwal and C. K. Reddy, *Data Clustering: Algorithms and Applications*. Boca Raton, FL, USA: CRC Press, 2015.

[4] S. Wang, Y. Fang, and S. Cheng, *Distributed Source Coding: Theory and Practice*. Hoboken, NJ, USA: Wiley, 2017.

[5] Z. Xiong, A. D. Liveris, and S. Cheng, ''Compression Via channel coding–distributed source coding for sensor networks,'' *IEEE Signal Process. Mag.*, vol. 21, no. 5, pp. 80–94, Sep. 2004, doi: 10.1109/MSP.2004.1328091.

[6] A. José-García and W. Gómez-Flores, ''Automatic clustering using nature-inspired metaheuristics: A survey,'' *Appl. Soft Comput.*, vol. 41, pp. 192–213, Apr. 2016, doi: 10.1016/j.asoc.2015.12.001.

[7] R. Xu and D. Wunsch, ''Survey of clustering algorithms,'' *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005, doi: 10.1109/TNN.2005.845141.

[8] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*. Philadelphia, PA, USA: SIAM, 2007.

[9] C. K. Reddy and B. Vinzamuri, ''A survey of partitional and hierarchical clustering algorithms,'' in *Proc. Data Clustering Algorithms Appl.*, 2013, pp. 87–110.

[10] C. Boutsidis, P. Drineas, and M. W. Mahoney, ''Unsupervised feature selection for the k-means clustering problem,'' in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 153–161.

[11] C. Ding and X. He, ''Cluster merging and splitting in hierarchical clustering algorithms,'' in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2002, pp. 139–146.

[12] Z. Huang and M. K. Ng, ''A fuzzy k-modes algorithm for clustering categorical data,'' *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 4, pp. 446–452, Aug. 1999.

[13] M. B. Agbaje, A. E. Ezugwu, and R. Els, ''Automatic data clustering using hybrid firefly particle swarm optimization algorithm,'' *IEEE Access*, vol. 7, pp. 184963–184984, 2019.

[14] A. E. Ezugwu, ''Nature-inspired Metaheuristic techniques for automatic clustering: A survey and performance study,'' *Social Netw. Appl. Sci.*, vol. 2, no. 2, p. 273, Feb. 2020.

[15] C. Gong, H. Chen, W. He, and Z. Zhang, ''Improved multi-objective clustering algorithm using particle swarm optimization,'' *PLoS ONE*, vol. 12, no. 12, Dec. 2017, Art. no. e0188815, doi: 10.1371/journal.pone.0188815.

[16] S. Zhu, L. Xu, and E. D. Goodman, ''Evolutionary multi-objective automatic clustering enhanced with quality metrics and ensemble strategy,'' *Knowl.-Based Syst.*, vol. 188, Jan. 2020, Art. no. 105018, doi: 10.1016/j.knosys.2019.105018.

[17] S. Mirjalili, S. M. Mirjalili, and X.-S. Yang, ''Binary bat algorithm,'' *Neural Comput. Appl.*, vol. 25, nos. 3–4, pp. 663–681, Sep. 2014, doi: 10.1007/s00521-013-1525-5.

[18] S. Mirjalili and A. Lewis, ''S-shaped versus V-shaped transfer functions for binary particle swarm optimization,'' *Swarm Evol. Comput.*, vol. 9, pp. 1–14, Apr. 2013.

[19] S. Mirjalili, ''Genetic algorithm,'' in *Evolutionary Algorithms and Neural Networks: Theory and Applications*, S. Mirjalili, Ed. Cham, Switzerland: Springer, 2019, pp. 43–55, doi: 10.1007/978-3-319-93025-1_4.

[20] S. Mirjalili, ''Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems,'' *Neural Comput. Appl.*, vol. 27, no. 4, pp. 1053–1073, May 2016, doi: 10.1007/s00521-015-1920-1.

[21] S. J. Nanda, R. Raman, S. Vijay, and A. Bhardwaj, ''A new density based clustering algorithm for binary data sets,'' in *Proc. Int. Conf. High Perform. Comput. Appl. (ICHPCA)*, Dec. 2014, pp. 1–6.

[22] P. Hassanzadeh, A. Tulino, J. Llorca, and E. Erkip, ''Cache-aided coded multicast for correlated sources,'' in *Proc. 9th Int. Symp. Turbo Codes Iterative Inf. Process. (ISTC)*, Sep. 2016, pp. 360–364.

[23] T. T. Nu, T. Fujihashi, and T. Watanabe, ''Content-aware efficient video uploading for crowdsourced multi-view video streaming,'' in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Mar. 2018, pp. 98–104.

[24] K. S. Al-Sultan, ''A tabu search approach to the clustering problem,'' *Pattern Recognit.*, vol. 28, no. 9, pp. 1443–1451, Sep. 1995.

[25] S. Z. Selim and K. Alsultan, ''A simulated annealing algorithm for the clustering problem,'' *Pattern Recognit.*, vol. 24, no. 10, pp. 1003–1008, Jan. 1991.

[26] R. Poll, J. Kennedy, and T. Blackwell, ''Particle swarm optimization: An overview,'' *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, 2007.

[27] D. W. van der Merwe and A. P. Engelbrecht, ''Data clustering using particle swarm optimization,'' in *Proc. Congr. Evol. Comput. (CEC)*, vol. 1, 2003, pp. 215–220.

[28] A. Abraham, S. Das, and S. Roy, ''Swarm intelligence algorithms for data clustering,'' in *Soft Computing for Knowledge Discovery and Data Mining*, O. Maimon and L. Rokach, Eds. Boston, MA, USA: Springer, 2008, pp. 279–313, doi: 10.1007/978-0-387-69935-6_12.

[29] J. Handl and J. Knowles, ''An evolutionary approach to multiobjective clustering,'' *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 56–76, Feb. 2007, doi: 10.1109/TEVC.2006.877146.

[30] E. Falkenauer, *Genetic Algorithms and Grouping Problems*. Hoboken, NJ, USA: Wiley, 1998.

[31] D. L. Davies and D. W. Bouldin, ''A cluster separation measure,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979, doi: 10.1109/TPAMI.1979.4766909.

[32] C.-H. Chou, M.-C. Su, and E. Lai, ''A new cluster validity measure and its application to image compression,'' *Pattern Anal. Appl.*, vol. 7, no. 2, pp. 205–220, Jul. 2004, doi: 10.1007/s10044-004-0218-1.

[33] R. J. Kuo and F. E. Zulvia, ''Automatic clustering using an improved particle swarm optimization,'' *J. Ind. Intell. Inf.*, vol. 1, no. 1, pp. 46–51, 2013.

[34] A. Abraham, S. Das, and A. Konar, ''Kernel based automatic clustering using modified particle swarm optimization algorithm,'' in *Proc. 9th Annu. Conf. Genetic Evol. Comput. (GECCO)*, New York, NY, USA, 2007, pp. 2–9, doi: 10.1145/1276958.1276960.

[35] S. J. Nanda and G. Panda, ''Automatic clustering algorithm based on multi-objective immunized PSO to classify actions of 3D human models,'' *Eng. Appl. Artif. Intell.*, vol. 26, nos. 5–6, pp. 1429–1441, May 2013, doi: 10.1016/j.engappai.2012.11.008.

[36] Y. Liu, X. Wu, and Y. Shen, ''Automatic clustering using genetic algorithms,'' *Appl. Math. Comput.*, vol. 218, no. 4, pp. 1267–1279, Oct. 2011, doi: 10.1016/j.amc.2011.06.007.

[37] H. He and Y. Tan, ''A two-stage genetic algorithm for automatic clustering,'' *Neurocomputing*, vol. 81, pp. 49–59, Apr. 2012, doi: 10.1016/j.neucom.2011.11.001.

[38] S. Das, A. Abraham, and A. Konar, ''Automatic clustering using an improved differential evolution algorithm,'' *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 1, pp. 218–237, Jan. 2008.

[39] I. Saha, U. Maulik, and S. Bandyopadhyay, ''A new differential evolution based fuzzy clustering for automatic cluster evolution,'' in *Proc. IEEE Int. Advance Comput. Conf.*, Mar. 2009, pp. 706–711, doi: 10.1109/IADCC.2009.4809099.

[40] W.-P. Lee and S.-W. Chen, ''Automatic clustering with differential evolution using cluster number oscillation method,'' in *Proc. 2nd Int. Workshop Intell. Syst. Appl.*, May 2010, pp. 1–4, doi: 10.1109/IWISA.2010.5473289.

[41] R. J. Kuo, Y. D. Huang, C.-C. Lin, Y.-H. Wu, and F. E. Zulvia, ''Automatic kernel clustering with bee colony optimization algorithm,'' *Inf. Sci.*, vol. 283, pp. 107–122, Nov. 2014.

[42] R. J. Kuo and F. E. Zulvia, ''Automatic clustering using an improved artificial bee colony optimization for customer segmentation,'' *Knowl. Inf. Syst.*, vol. 57, no. 2, pp. 331–357, Nov. 2018, doi: 10.1007/s10115-018-1162-5.

[43] V. Kumar, J. K. Chhabra, and D. Kumar, "Automatic data clustering using parameter adaptive harmony search algorithm and its application to image segmentation," *J. Intell. Syst.*, vol. 25, no. 4, pp. 595–610, Oct. 2016, doi: 10.1515/jisys-2015-0004.

[44] A. Chowdhury, S. Bose, and S. Das, "Automatic clustering based on invasive weed optimization algorithm," in *Swarm, Evolutionary, and Memetic Computing*. Berlin, Germany: Springer, 2011, pp. 105–112, doi: 10.1007/978-3-642-27242-4_13.

[45] R. Qaddoura, H. Faris, I. Aljarah, and P. A. Castillo, "EvoCluster: An open-source nature-inspired optimization clustering framework in Python," in *Applications of Evolutionary Computation*. Cham, Switzerland: Springer, 2020, pp. 20–36, doi: 10.1007/978-3-030-43722-0_2.

[46] S. J. Nanda and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering," *Swarm Evol. Comput.*, vol. 16, pp. 1–18, Jun. 2014.

[47] S. Das, A. Abraham, and A. Konar, "Metaheuristic pattern clustering—An overview," in *Metaheuristic Clustering*, S. Das, A. Abraham, and A. Konar, Eds. Berlin, Germany: Springer, 2009, pp. 1–62, doi: 10.1007/978-3-540-93964-1_1.

[48] P. Gançarski, B. Crémilleux, G. Forestier, and T. Lampert, "Constrained clustering: Current and new trends," in *A Guided Tour of Artificial Intelligence Research*. Springer, 2020, pp. 447–484.

[49] D. Dinler and M. K. Tural, "A survey of constrained clustering," in *Unsupervised Learning Algorithms*, M. E. Celebi and K. Aydin, Eds. Cham, Switzerland: Springer, 2016, pp. 207–235, doi: 10.1007/978-3-319-24211-8_9.

[50] Z. Drezner, "The p-centre problem-heuristic and optimal algorithms," *J. Oper. Res. Soc.*, vol. 35, no. 8, pp. 741–748, 1984.

[51] I. Davidson and S. S. Ravi, "Agglomerative hierarchical clustering with constraints: Theoretical and empirical results," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*, 2005, pp. 59–70.

[52] S. Ghiasi, A. Srivastava, X. Yang, and M. Sarrafzadeh, "Optimal energy aware clustering in sensor networks," *Sensors*, vol. 2, no. 7, pp. 258–269, Jul. 2002, doi: 10.3390/s20700258.

[53] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 370–379, Aug. 1995, doi: 10.1109/91.413225.

[54] D. Dua and C. Graff, "UCI machine learning repository," Univ. California School Inf. Comput. Sci., Irvine, CA, USA, 2019. [Online]. Available: http://archive.ics.uci.edu/ml

[55] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Multiple-Valued Log. Soft Comput.*, vol. 17, nos. 2–3, pp. 255–287, 2011.

[56] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proc. SODA*, Jan. 2007, pp. 1027–1035.

[57] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 1996, vol. 96, no. 34, pp. 226–231.

[58] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques* (Morgan Kaufmann Series in Data Management Systems), 3rd ed. San Mateo, CA, USA: Morgan Kaufmann, 2011.

[59] R. Qaddoura, H. Faris, and I. Aljarah, "An efficient clustering algorithm based on the k-nearest neighbors with an indexing ratio," *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 3, pp. 675–714, Mar. 2020, doi: 10.1007/s13042-019-01027-z.

[60] S. Bandyopadhyay and U. Maulik, "Genetic clustering for automatic evolution of clusters and application to image classification," *Pattern Recognit.*, vol. 35, no. 6, pp. 1197–1208, Jun. 2002.

[61] M. G. H. Omran, A. Salman, and A. P. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in image segmentation," *Pattern Anal Appl.*, vol. 8, no. 4, p. 332, Nov. 2005, doi: 10.1007/s10044-005-0015-5.

[62] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, no. 3, pp. 19:1–19:21, Jul. 2017, doi: 10.1145/3068335.

[63] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *Ann. Math. Statist.*, vol. 11, no. 1, pp. 86–92, 1940.

[64] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Stat. Assoc.*, vol. 32, no. 200, pp. 675–701, Dec. 1937.

[65] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 3–18, Mar. 2011.

[66] S. García, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, p. 617, May 2008.

[67] C. Ahn, F. Hu, and W. R. Schucany, "Sample size calculation for clustered binary data with sign tests using different weighting schemes," *Statist. Biopharmaceutical Res.*, vol. 3, no. 1, pp. 65–72, Feb. 2011.

[68] V. Verma and R. K. Aggarwal, "A new similarity measure based on simple matching coefficient for improving the accuracy of collaborative recommendations," *Int. J. Inf. Technol. Comput. Sci.*, vol. 11, no. 6, pp. 37–49, Jun. 2019, doi: 10.5815/ijitcs.2019.06.05.

BEHNAZ MERIKHI received the B.Sc. degree in electrical engineering and the M.Sc. degree in computer engineering from Shahid Beheshti University, Tehran, Iran, in 2010 and 2015, respectively. She is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, Canada. Having done her master's thesis on the IoT applications and wireless networks combined with over five years' experience as a network engineer (from 2010 to 2015) made her passionate to do more research on communication networks and its applications in the next generation of communication systems. Her current research interests include wireless communication networks and source coding, artificial intelligence, the Internet of Things, and their applications in future communication networks.

M. R. SOLEYMANI received the B.S. degree from the University of Tehran, in 1976, the M.S. degree from San Jose State University, in 1977, and the Ph.D. degree from Concordia University, in 1988, all in electrical engineering. From 1987 to 1990, he was an Assistant Professor with the Department of Electrical Engineering, McGill University. From 1990 to 1998, he was with Spar Aerospace Ltd. (presently called MDA), where he had leading role in the design and development of several satellite communications systems. In 1998, he joined the Department of Electrical and Computer Engineering, Concordia University, where he is currently a Professor. His current research interests include digital communications, satellite communications, digital broadcasting, communication networks, information theory and coding, data compression, and source coding.