www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 15(4), Nov 2024, 85-100. ISSN: 2007-1558. https://doi.org/10.61467/2007.1558.2024.v15i4.501

Bioinspired Hybrid and Incomplete Data Clustering

Claudia C. Tusell-Rey[1], Yenny Villuendas-Rey[2], Oscar Camacho-Nieto[2], and Viridiana Salinas-García[2]
[1] Instituto Politécnico Nacional, Centro de Investigación en Computación, Juan de Dios Bátiz s/n, GAM, CDMX 07738
[2] Instituto Politécnico Nacional, Centro de Innovación y Desarrollo Tecnológico en Cómputo, Juan de Dios Bátiz s/n, GAM, CDMX 07700
E-mails: clautusellrey2014@gmail.com; {yvilluendasr;ocamacho;vsalinasg}@ipn.mx

Abstract. Enhancing the results of data clustering for hybrid (numeric and categorical) and missing data is paramount for robust pattern recognition and decision-making in various fields. Traditional clustering algorithms often need help with heterogeneous data types and incomplete information, leading to suboptimal groupings and potentially biased insights. By addressing these challenges, advanced techniques, such as bioinspired algorithms, can provide more accurate and comprehensive clustering. We show the usefulness of using internal validity indices in obtaining high-quality clusters for hybrid and incomplete data. In addition, this work analyzes the influence of applying the recently proposed PAntSA on the performance of hybrid and incomplete clustering algorithms. To do this, the results of different algorithms are compared before and after applying PAntSA. The statistical analysis of the results provides experimental evidence that supports the PAntSA algorithm improves the quality of the clusters obtained by hybrid and incomplete data clustering methods.
Keywords: bioinspired algorithms; clustering; hybrid and incomplete data.

Article Info
Received Jul 20, 2024
Accepted Sep 11, 2024

1 Introduction

The process of clustering, which involves grouping a set of instances into several groups containing similar objects, has found significant practical applications. This technique has been successfully used in solving real-life problems such as speech recognition, image segmentation, computer vision, text mining, and computational biology for DNA analysis. The main goal of clustering algorithms is to find the "natural structure" of the data; that is, to find the underlying patterns that make certain instances belong to a certain cluster. Usually, the cluster objective is to obtain compact and well-separated clusters, assuming that the instances in a cluster must be very similar and dissimilar to those of other clusters.

Most clustering algorithms have been designed to work only with numerical or categorical data. However, in many cases, it is necessary to work with mixed (or hybrid) data, which includes attributes of different types, such as numerical, binary, discrete, and categorical. Moreover, the presence of incomplete data, where the value of a certain attribute is unknown, poses a significant challenge. This underscores the need to develop algorithms that can effectively handle incomplete data (Ruiz-Shulcloper, 2008).

The clustering of hybrid and incomplete data has traditionally been approached following classic paradigms such as hierarchical and partitional, although bio-inspired proposals have also appeared that have had good performance (González-Patiño, Villuendas-Rey, Saldaña-Pérez, & Argüelles-Cruz, 2023). Bioinspired clustering is based on optimizing a certain validity index. Several validity indexes have been proposed to compute the compactness and separation of a clustering result (Brun et al., 2007; Maulik & Bandyopadhyay, 2002; Rendón, Abundez, Arizmendi, & Quiroz, 2011). They somehow calculate the relation between the distances of the members of a given cluster (compactness) and the distance relations with respect to the members of other clusters (separation). However, although they have a similar underlying background, the results obtained by applying different validity indexes to the same clustering may be very dissimilar.

To overcome this drawback, we explore the use of several validity indexes in bioinspired clustering. In addition, the bio-inspired algorithm PAntSA that is based on Ant Tree algorithm (Azzag, Monmarche, Slimane, & Venturini, 2003) was published (Ingaramo, Errecalde, & Rosso, 2010), which takes the results obtained by a previous clustering algorithm and tries to refine them using the Silhouette index (Brun et al., 2007) and the definition of an attraction between groups. PAntSA improves the quality of results obtained by clustering algorithms on numerical data, particularly in document classification; however, there is a need to condict a study of the influence of PAntSA on hybrid and incomplete bio-inspired clustering.

The contributions of this paper are the following:

- We focus on a recently proposed bioinspired clustering based on Artificial Bee Colonies (Villuendas-Rey, Barroso-Cubas, Camacho-Nieto, & Yáñez-Márquez, 2021).
- We analyze the influence of PAntSA (Ingaramo et al., 2010) in improving the results of hybrid and incomplete clustering algorithms.
- We show the usefulness of using internal validity indices in obtaining high-quality clusters for hybrid and incomplete data.

The rest of the paper is organized as follows: Section 2 shows some work related to hybrid and incomplete data clustering, as well as internal cluster validity indexes. Section 3 presents the materials and method, including the PAntSA algorithm used to improve the groups obtained by other algorithms, and the bioinspired hybrid and incomplete data clustering based on Artificial Bee Colonies. Section 4 comprises the experimental results obtained and Section 5 offers the conclusions and avenues of future works.

2 Related Works on Hybrid and Incomplete Data Clustering

The existing clustering algorithms for hybrid and incomplete data, for the most part, are the product of extensions made to methods for handling homogeneous data types (numeric or non-numeric). These can be divided into several categories according to the procedure they use to group objects.

2.1 Algorithms for clustering hybrid and incomplete data

In 1967, MacQueen proposed k-means, a classical algorithm that is considered the archetype of the partitional model (MacQueen, 1967). For the k-means algorithm to work, it must know the number of groups to obtain (k). The idea is to locate k centroids and cluster objects by their closest centroid according to a distance function defined a priori. Iteratively, the centroids are updated as the average of the objects that belong to each group, until the centroids stop changing.

One of the first extensions to k-means to deal with DMI was k-Prototypes, published by Huang (Huang, 1997). This author based his proposal on the definition of a new function of dissimilarity between objects, which allows for the treatment of mixed descriptions. Let o be an instance, c_i be the center of the i-th cluster, and X_p be the p-th attribute. The dissimilarity between an instance and the cluster center is given by:

$$d(o, c_i) = \sum_{p \in Rn} (X_p(o) - X_p(c_i))^2 + \gamma_i \sum_{r \in Rc} \Gamma(X_r(o), X_r(c_i)) \quad (1)$$

where: γi is a user-defined parameter, Rn is the set of numeric attributes, Rc is the set of categorical attributes, and $\Gamma(X_r(O_j), X_r(c_i))$ is a dissimilarity function equal to zero if $X_r(O_j) = X_r(c_i)$ and equal to one otherwise. In addition, it makes a modification in the way of obtaining the centers of the groups taking as values the mean of the numerical attributes, and the mode of the categorical attributes.

Ahmad and Dey also proposed another modification to k-means (Ahmad & Dey, 2007, 2011). The modifications made consist of updating the dissimilarity function, considering each attribute's contribution to each group. This algorithm does not have a name, only a pseudocode is stated under the title *modified_kmean_subspace_clustering*. Other authors have proposed modifications to the k-means for dealing with hybrid and incomplete data, such as KMSF (Martínez Trinidad, Garcia Serrano, & Ayaquica Martínez, 2002).

Despite their low time complexity, k-means and its variants have many disadvantages: for example, the results depend on the initialization of the centroids. Furthermore, they are invalid for objects with categorical attributes due to the need to calculate the average and are inadequate for detecting non-convex groups and outliers (noisy objects).

Hierarchical algorithms, as the name suggests, construct a hierarchy of clustering, joining or dividing the groups according to a certain similarity/dissimilarity function between the groups. In other words, they build a tree of groups called a dendrogram. Such an approach allows data to be studied with different levels of granularity. Hierarchical clustering algorithms are categorized into agglomerative (bottom-up) and divisive (top-down). An agglomerative cluster generally begins with singleton clusters and recursively joins two or more appropriate groups. The process continues until some stopping criterion (frequently the number k of groups) is reached.

Reyes-González and Ruiz-Shulcloper proposed the first agglomerative algorithm for hybrid and incomplete data (Reyes-González & Ruiz-Shulcloper, 1999). The algorithm (which we will call AERE) forms a new level at each step until all objects are on the same level. A level is defined by the number of groups present in the level, the value of β0 (maximum similarity between two groups of the same level), and the set of possible partitions. As a distinctive element, it is deterministic (always obtains the same solution), and each group is made up of elements that are in the same connected component β0 in a graph of maximum similarity. Finally, the algorithm returns all possible structurings (set of partitions) for the desired level k of the formed hierarchy.

In addition, a hierarchical algorithm called HIMIC (HIerarchical MIxed type data Clustering algorithm) was proposed (Ahmed, Borah, Bhattacharyya, & Kalita, 2005). The algorithm is based on the use of a dissimilarity function between two groups c_i, c_j that considers the set of possible categorical values as follows:

$$d(c_i, c_j) = \sum_{p=1}^{n} S_p(c_i, c_j) \tag{2}$$

where

$$S_p(C_i, C_j) = \begin{cases} 1 - \left| \frac{1}{|C_i|} \sum_{o \in C_i} X_p(o) - \frac{1}{|C_j|} \sum_{o \in C_j} X_p(o) \right| & if\ p\ is\ numeric \\ \sum_{l=1}^{|D_p|} \frac{|\{o \in C_i | X_p(o) = v_l\}|}{|C_i|} * \frac{|\{o \in C_j | X_p(o) = v_l\}|}{|C_j|} & if\ p\ is\ categorical \end{cases} \tag{3}$$

Subsequently, a traditional agglomerative method is applied, using obtaining the desired number of groups k as a stopping criterion.

Among the advantages of hierarchical clustering algorithms are their flexibility with respect to the level of granularity, ease of handling, and application to any type of attribute. Among the disadvantages are the non-improvement of the groups that have been constructed due to the non-consideration of the objects already assigned and the sensitivity to noise. Furthermore, the computational cost for most of these algorithms is also at least $O(m^2)$, where m is the number of objects, which limits their application to large data sets.

The data clustering problem can be viewed as an optimization problem that locates the optimal centroids of clusters or finds the optimal partition of a set of objects. For this reason, different optimization techniques have been successfully used to help find the best solution or at least a good enough solution for a problem in a search space. In this sense, metaheuristics stand out as approximate algorithms that try to solve these problems, sacrificing the guarantee of finding the optimal one in exchange for finding a "good solution in a reasonable time," which is why they have been used for clustering large data sets.

A metaheuristic is a high-level strategy that uses different strategies to explore the search space. Due to its easy adaptation, simplicity, and efficiency, metaheuristics are among the most widely used approximate methods.

The AKGA algorithm (Roy & Sharma, 2010) consists of the use of a Genetic Algorithm (GA) to obtain groups without the need to exhaustively apply a clustering algorithm. The application of a GA allows us to leave local optima and search for a global optimum without too high a computational cost. As a representation scheme, each solution consists of an individual, represented by a string of size equal to the number of objects, and where each i-th element of the string represents the group to which the ith object is assigned (see Fig. 1).

2	1	2	1	1	2	1	2	1	2

Fig. 1. Example of an individual. In this case, there are 10 objects, grouped into two groups. Each position encodes the group to which said object belongs.

In the case of group centers, the authors use the scheme of Ahmad and Dey (Ahmad & Dey, 2007, 2011). They also use the dissimilarity function of objects to the centers [12] as part of the optimization function of the Genetic Algorithm.

Several bioinspired algorithms have been successfully used in clustering hybrid and incomplete data (Villuendas-Rey et al., 2021). One example is the CABC algorithms, based on the Artificial Bee Colony model (Karaboga & Basturk, 2007) for numerical optimization. CABC algorithm randomly generates n initial groupings that constitute the food sources. It then generates new sources using the mutation strategy defined in (Villuendas-Rey et al., 2021) and using HEOM (Wilson & Martinez, 1997) as a dissimilarity that allows dealing with hybrid and incomplete data. To evaluate food sources, an internal validation index is used as an objective function. Finally, after a number of iterations defined a priori, the food source (clustering) that optimized the objective function is returned.

2.2 Internal cluster validity indexes

Internal cluster validity indexes are metrics used to evaluate the quality of clusters formed by clustering algorithms based solely on the data and without external reference. These indexes assess how well clusters are separated from each other and how compact and well-defined individual clusters are. Popular internal validity indexes include the Davies-Bouldin index (Davies & Bouldin, 1979), which measures the average similarity between each cluster and its most similar neighboring cluster relative to their internal dispersion; the Dunn index (Brun et al., 2007), which evaluates the ratio of the minimum inter-cluster distance to the maximum intra-cluster distance; and the Silhouette coefficient (Brun et al., 2007), which quantifies how similar an object is to its own cluster compared to other clusters. These measures help select the optimal clustering and compare different clustering results to find the most meaningful partitioning of data. In the following, we review such indexes.

The Silhouette is the average of the silhouette width of the instances belonging to that cluster (Rendón et al., 2011). If x is an instance from cluster c_i and n_i is the total number of instances in c_i, then the silhouette of x is given by:

$$S(x) = \frac{b(x) - a(x)}{max\{a(x), b(x)\}} \tag{4}$$

where a(x) is the average distance to all instances in c_i, and b(x) is the minimum average dissimilarity between x and the instances from other clusters, as:

$$a(x) = \frac{1}{n_i - 1} \sum_{\substack{y \in c_i \\ y \neq x}} d(x, y), \tag{5}$$

$$b(x) = \min_{\substack{h=1..k \\ h \neq i}} \left\{ \frac{1}{n_h} \sum_{y \in c_h} d(x, y) \right\} \tag{6}$$

Finally, the global silhouette of a clustering partition is defined as:

$$S = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n_i}\sum_{x\in c_i} S(x) \tag{7}$$

For a certain instance, its silhouette width is in the [-1, 1] interval. The greater the Silhouette, the more compact and separated the clusters.

The Davies – Bouldin index is a well-known unsupervised cluster validity index (Maulik & Bandyopadhyay, 2002). It considers the dispersion of instances in a cluster $\Delta(c_i)$ and the dissimilarity of clusters $diss(c_i, c_j)$. The Davies – Bouldin index measures the average similarity between each cluster and its most similar cluster. The lower the values of the Davies – Bouldin index, the more compact and separated are the clusters. Formally, Davies – Bouldin index is defined as:

$$DB = \frac{1}{k}\sum_{i=1..k}\max_{j=1..k, j\neq i} R_{i,j} \tag{8}$$

where $R_{i,j}$ is usually defined as:

$$R_{i,j} = \frac{\Delta(c_i) + \Delta(c_j)}{diss(c_i, c_j)} \tag{9}$$

Several inter-cluster dissimilarities and measures of cluster size are defined. In this research, we used the centroid dissimilarity as an inter-cluster measure and the centroid measure for cluster size.

$$diss(c_i, c_j) = diss(\bar{c_i}, \bar{c_j}) \tag{10}$$

$$\Delta(c_i) = \sum_{p\in c_i}\frac{diss(p, \bar{c_i})}{|c_i|} \tag{11}$$

Dunn´s index for clustering considers the ratio between the minimum distance between two clusters and the size of the bigger cluster. Similar to the Silhouette index, greater values correspond to better clustering (Rendón et al., 2011).

$$D = \min_{1\leq i\leq k}\left\{\min_{\substack{1\leq j\leq k \\ i\neq j}}\left\{\frac{diss(c_i, c_j)}{\max_{1\leq l\leq k}\{\Delta(c_l)\}}\right\}\right\} \tag{12}$$

It has been assumed so far that there is a dissimilarity measure to compute the dissimilarity between instances in the clustering process. In fact, the fundamental of several clustering is to select a set of centroid instances and then assign each object to the less dissimilar (or closest) centroid. In this research, we used the HEOM dissimilarity to compare instances in such a way. Wilson and Martínez introduced the HEOM dissimilarity to compare objects with mixed numerical and categorical descriptions. Let be two instances p and q, the HEOM dissimilarity is defined as (Wilson & Martinez, 1997):

$$HEOM(p,q) = \sqrt{\sum_{A_i \in A} d_i(p_i,q_i)^2} \tag{13}$$

$$d_i(p_i,q_i) = \begin{cases} 1 & \text{if } p_i \vee q_i \text{ are missing} \\ overlap(p_i,q_i) & \text{if } A_i \text{ is categoric} \\ rn\_diff(p_i,q_i) & \text{if } A_i \text{ is numeric} \end{cases} \tag{14}$$

$$overlap(p_i,q_i) = \begin{cases} 0 & \text{if } p_i = q_i \\ 1 & \text{otherwise} \end{cases} \tag{15}$$

$$rn\_diff = \frac{|p_i - q_i|}{max_i - min_i} \tag{16}$$

By using the HEOM dissimilarity, we are able to directly compare mixed and incomplete instances, carry out the clustering process, and evaluate the resulting clustering with the selected optimization functions. In addition, due to its simplicity and low computational cost, the HEOM dissimilarity is a feasible choice as an optimization function in evolutionary algorithms.

3 Materials and Methods

3.1 Bioinspired Clustering Based on Artificial Bee Colonies

The Clustering using the Artificial Bee Colony (CABC) algorithm was recently proposed for clustering hybrid and incomplete data (Villuendas-Rey et al., 2021). It is based on the Artificial Bee Colony optimization model (Karaboga & Basturk, 2007). The keys of CABC are its cluster representation, its center computation, and its updating strategy.

Regarding representation, CABC adopts a simple approach: each solution is an array of cluster centers. Each bee (candidate clustering) S_i is represented as $S_i = [\bar{c_1}, \bar{c_2}, \ldots, \bar{c_k}]$ where $\bar{c_i}$ is the selected center for the i-th cluster. Initially, the cluster centers are assigned randomly and the instances are assigned to the nearest center. After that, the centers need to be updated. To do so, CABC avoids constructing artificial data, instead, for each cluster, a new cluster center $\bar{c_i}$ is selected as:

$$\bar{c_i} = \arg \min_{\substack{p,q \in c_l \\ p \neq q}} \{diss(p,q)\} \tag{17}$$

After the new centers are obtained, the iterative process begins. In Artificial Bee Colonies, the employees and onlooker bees search for a solution closest to the current one and then compare both solutions, retaining the best of them. The CABC mimics this process by randomly changing a cluster center in the current solution. A cluster is selected randomly, and a new instance is also randomly selected to replace the cluster center. The new solution is then compared according to the fitness function and replaces the old one if best. Figure 2 presents the pseudocode of the CABC algorithm.

CABC algorithm

Inputs: X: dataset of instances
k: cluster number
I: number of iterations
η: population size
L limit of food sources

Output: $C = \{c_1, \dots c_k\}$: clustering of instances

Steps:

1. Send a scout bee for random generation of η food sources
2. it = 1
3. **While** $it < I$
    a. For each employed bee assigned to a food source S_i
        i. Generate a new food source S_{new}, closer to the current source, as *Updating*(S_i)
            ii. If S_{new} is better than S_i, then $S_i \leftarrow S_{new}$
        else increase limit of S_i
    b. For each onlooker bee
        i. Fly to a food source with good nectar amount (S_i)
        ii. Generate a new food source S_{new}, closer to the current source, as *Updating*(S_{new})
        i. If S_{new} is better than S_i, then $S_i \leftarrow S_{new}$
        else increase limit of S_i
    c. Send the scout bee to find the food sources that have reached the limit L, and replace them with randomly generated food sources
    d. it+
4. Create a cluster C by assigning the instances in X to its closest centers, considering the centers of the best food source
5. Return C

Fig. 2. Pseudocode of CABC.

In (Villuendas-Rey et al., 2021), the fitness functions used for CABC were the Dunn's index, the Davies-Bouldin index, and the Silhouette coefficient. In addition, the suggested parameters were using 10 bees and 50 iterations.

3.2 Partitional AntSA

The PAntSA algorithm (Ingaramo et al., 2010) is based on the Ant Tree algorithm, proposed in (Azzag et al., 2003). For a better understanding of PAntSA, we will first explain in detail the operation of the Ant Tree algorithm. This algorithm is a pioneer in the application of modeling the construction of nests by ants to Artificial Intelligence problems. The Ant Tree is based on modeling the ability of ants to build living structures with their bodies and to discover, in a distributed and unsupervised way, a tree structure that organizes a set of data. This hierarchical structure can be interpreted in several ways: as a partition of the data or as a hierarchical structuring of the data.

The fundamental principle of the Ant Tree is the following: each ant represents a node in the tree that will be built (i.e., the objects that will be grouped) and there is a similarity function between two objects $Sim(i,j)$. Based on a fictitious root node a0, which represents the support on which the tree will be built, each ant ai will gradually be attached to the initial node and successively to the ants already fixed until all the ants are in the structure. All movements and fixations in the structure will depend on the value of $Sim(i,j)$ and the neighborhood where the ants move.

Taking as inspiration the Ant Tree and particularly the AntSA, an algorithm derived from it, Ingaramo et al. propose PAntSA (Partitional AntSA) (Ingaramo et al., 2010), specifically to improve the results of any text clustering algorithm. In PAntSA each connected ant represents a group, and those connected to it form a simple list. Thus, when an ant a_i joins the group of an ant a+ (the ant most similar to a_i) denoted Ga+, the attraction is implemented simply by adding the ant to the corresponding group. The pseudocode of the algorithm is presented below (Fig. 3).

PAntSA Algorithm

Inputs:
T: the set of instances
A: clustering algorithm
k: number of clusters
Steps:
1. Apply the algorithm *A* to the instances from *T*.
2. Build *k* rows (one for each cluster obtained in step 1) and sort them according to their Silhouette coefficient.
3. Create a cluster for the first ant of each row, where such ant is the representative of the cluster R_j (the ant is the instance with higher Silhouette value).
4. Joint the rows in a new row *F* by iteratively taking the ants in non-empty rows, until all rows are empty.
   For each ant a_i in *F*:
   a. For each ant R_j:
      i. Fin the set of ants in the cluster of R_j, and denote them by Ga_+
      ii. Compute the attraction of each ant $att(a_i, Ga_+) = \sum_{a \in Ga_+} Sim(a_i, a)/|Ga_+|$.
   Connect the ant a_i to the cluster having maximum $att(a_i, Ga_+)$.
5. Return the obtained clusters.

Fig. 3. Pseudocode of PAntSA.

Although Ingaramo et al. tested the efficiency of PAntSA and showed that it is capable of improving clustering algorithms for numerical data, to our knowledge there is no study of whether it is capable of improving the results of clustering algorithms for mixed data, which is the objective of this work.

4 Results and Discussion

In the following, we present the datasets used in the experimental analysis and the performance measures used to assess the quality of the different clustering obtained. Then, we present the results of the influence of the PAntSA on the performance of hybrid and incomplete data clustering.

4.1 Datasets

In the experiments, we used 15 mixed and/or incomplete datasets from the University of California at Irvine (UCI) machine learning repository (Kelly, Longjohn, & Nottingham, 2023). Table 1 describes the datasets considered.

Table 1. Description of the datasets used in the numerical experiments.

Dataset	Instances	Numeric attributes	Categorical attributes	Classes	Missing Values
anneal	898	6	32	5	x
autos	205	15	10	6	x
cmc	1473	2	8	3	x
colic.ORIG	368	7	21	3	x
credit-a	690	6	9	2	x
credit-g	1000	7	13	2	
dermatology	366	1	33	6	x
heart-c	303	6	7	2	x
hepatitis	155	6	13	2	x
labor	57	8	8	2	x
lymph	148	3	15	4	
postoperative	90	0	9	3	x
tae	151	3	3	2	x
vowel	990	10	3	11	
zoo	101	1	16	7	

The selected datasets are labeled instances; thus, it is possible to compare the results obtained by the clustering algorithms with respect to the true labels of the data. Although in several types of research, the cluster number is greater than the number of classes (ex., having k equal to 50 clusters and only two or three true classes (Ahmad & Dey, 2011)), we want to explore the use of bioinspired algorithms for finding the natural structure of data.

In the experiments, we consider that the natural structure of data is the class labels. Thus, in our experiments, the instances of the same class should belong to the same cluster. We are not dealing with the issue of possible mislabeled or noisy instances. That is why, in our experiments, the number of clusters to obtain was set as the number of classes in the corresponding dataset (column "Classes" of Table 1).

4.2 Performance Measures

In this paper, we used two performance measures to evaluate the quality of the obtained clusters. The first performance measure used was Entropy, which measures the degree of disorder of the model clustering (AM) in the obtained clusters using the following equation:

$$E = \sum_{k \in AE} \frac{|k|}{N} \left[\frac{1}{log(|AM|)} \sum_{m \in AM} \frac{n_k^m}{|k|} log \frac{n_k^m}{|k|} \right] \quad (18)$$

where AE represents the clustering to be evaluated, |k| is the total number of objects in cluster k, |AM| is the total number of clusters in the model clustering, and n_k^m is the total number of objects in cluster k that belongs to the AM clustering. The lower the Entropy, the better the quality of the clustering.

The second validation index used was the Cluster Error, another of the internal validation indexes most used in experimental comparisons. Its operation is based on minimizing the objects that are assigned to a different clusters from the model clustering (AM) in the clusreting to be evaluated (AE) and its definition is given by:

$$CE = \sum_{k \in AE} \frac{1}{N} \min_{m \in AM} \{P_k^m\} \quad (19)$$

where P_k^m is the total number of instances from the cluster *k* not belonging to the cluster *m* in AM. The lower Cluster Error the better the clustering.

4.3 Algorithms under comparison

In the experimental comparison, six algorithms were used to cluster hybrid and incomplete data. The results obtained were evaluated before and after applying the PAntSA. Thus, we compare algorithms of the following approaches to clustering: partitional (kPrototypes and KMSF), hierarchical (HIMIC), ensemble (CEBMDC), evolutionary (AGKA), and bioinspired (CABC). Regarding CABC, we computed three different versions using different fitness functions as in the original article (Villuendas-Rey et al., 2021).

The parameters with which the algorithms will be executed are an important aspect of the experiment design (see Table 2). As all algorithms require knowing the number of groups to form, the value assigned to this parameter will coincide, for each dataset, with the number of classes. With this, the classes are taken as a model clustering against which to evaluate the clustering obtained by applying the algorithms. The rest of the parameters were chosen based on studies that recommended certain values for better performance. Furthermore, the common parameters of the different algorithms were supplied with the same value. This allowed us to achieve a certain homogeneity and reduce a possible imbalance in the performance of one algorithm compared to another due to the use of different values for the same parameter. In the case of dissimilarity, the HEOM function (Wilson & Martinez, 1997) was used for all algorithms. Table 3 shows the Entropy results of the compared algorithms without postprocessing.

Table 2. Parameters used by the algorithms under study

Family	Algorithm	Parameters
Partitional	KMSF	None
	k-prototypes	None
Hierarchical	HIMIC	None
Ensemble	CEBMDC	Similarity threshold: 0.00
Evolutive	AGKA	Population size: 10; Iterations: 50; Mutation probability: 0.05; Crossover probability: 1
Bioinspired	CABC-DB	Food sources: 10; Iterations: 50; Scout bees: 1; Limit of food sources: 2; Fitness function: Davies-Bouldin index
	CABC-DN	Food sources: 10; Iterations: 50; Scout bees: 1; Limit of food sources: 2; Fitness function: Dunn's index
	CABC-ST	Food sources: 10; Iterations: 50; Scout bees: 1; Limit of food sources: 2; Fitness function: Silhouette coefficient

Table 3. Entropy for compared algorithms before postprocessing. The best results for each dataset are in bold.

Datasets	KMSF	kPrototypes	HIMIC	CEBMDC	AGKA	CABC-DB	CABC-DN	CABC-ST
anneal	0.94	1.14	1.13	1.19	1.25	0.88	0.80	**0.72**
autos	1.86	1.95	2.03	2.17	2.13	1.77	1.71	**1.67**
cmc	1.51	1.51	1.49	1.50	1.53	**1.46**	1.49	1.48
colic.ORIG	0.94	0.93	0.95	0.95	0.95	**0.81**	0.84	0.82
credit-a	0.88	0.99	0.99	**0.73**	0.99	0.98	0.89	0.96
credit-g	0.88	0.88	0.88	0.87	0.87	**0.86**	**0.86**	0.87
dermatology	1.52	2.29	2.13	2.08	2.41	1.16	**0.65**	0.76
heart-c	0.82	0.98	0.99	0.93	1.00	0.80	0.68	**0.64**
hepatitis	0.72	0.73	0.73	0.73	0.73	0.72	0.67	**0.58**
labor	0.91	0.90	0.92	0.93	0.94	**0.53**	0.80	0.80
lymph	0.98	1.02	0.86	1.20	1.15	**0.84**	0.86	0.71
postoperative	1.02	1.01	**0.95**	1.01	0.98	1.22	1.19	1.20
tae	1.52	1.55	1.53	1.52	1.57	1.47	1.48	**1.45**
vowel	3.21	**2.36**	2.38	3.46	3.33	3.09	3.16	3.32
zoo	0.48	0.77	0.52	2.18	2.01	0.46	**0.20**	0.33

Table 4. Cluster Error for compared algorithms before postprocessing. The best results for each dataset are in bold.

Datasets	KMSF	kPrototypes	HIMIC	CEBMDC	AGKA	CABC-DB	CABC-DN	CABC-ST
anneal	0.24	0.24	0.23	0.24	0.26	0.22	0.21	**0.18**
autos	0.59	0.59	0.62	0.67	0.62	0.50	**0.45**	0.50
cmc	0.57	0.57	0.56	0.56	0.58	**0.53**	0.57	0.57
colic.ORIG	0.37	0.37	0.37	0.37	0.37	**0.26**	0.34	0.29
credit-a	0.34	0.44	0.44	**0.21**	0.43	0.43	0.34	0.44
credit-g	0.30	0.30	0.30	0.30	**0.29**	0.30	**0.29**	0.30
dermatology	0.42	0.69	0.66	0.57	0.73	0.36	**0.19**	0.21
heart-c	0.25	0.42	0.46	0.37	0.47	0.25	**0.18**	0.17
hepatitis	0.21	0.21	0.21	0.21	0.21	0.20	**0.18**	0.20
labor	0.35	0.35	0.35	0.35	0.37	**0.14**	0.28	0.25
lymph	0.31	0.32	0.24	0.45	0.42	**0.22**	0.24	**0.22**
postoperative	**0.30**	**0.30**	**0.30**	**0.30**	0.34	0.35	0.33	0.34
tae	0.58	0.58	0.60	0.59	0.62	0.54	0.54	**0.53**
vowel	0.83	**0.68**	0.69	0.91	0.84	0.82	0.84	0.88
zoo	0.12	0.21	0.14	0.59	0.54	0.13	**0.07**	0.12

As shown in Table 3, the best-performed algorithm is CABC. It demonstrates the usefulness of bioinspired algorithms in clustering hybrid and incomplete data. In addition, it supports the idea of using different internal validity indexes as fitness functions to guide the bioinspired optimization process. Similar results were obtained for the Cluster Error measure (Table 4).

4.4 Assessing the Influence of PAntSA

To assess the influence of PAntSA, the experiments were conducted as follows. The PAntSA (Ingaramo et al., 2010) postprocessing was applied to each algorithm's result, obtaining a new clustering. The Entropy was also calculated for these new results (Table 5). In each case, the number of clusters was established as the number of classes in each of the databases.

Table 5. Entropy for compared algorithms after postprocessing. The best results for each dataset are in bold.

Datasets	KMSF	kPrototypes	HIMIC	CEBMDC	AGKA	CABC-DB	CABC-DN	CABC-ST
anneal	0.84	**0.67**	0.94	1.19	1.04	0.75	0.68	0.69
autos	1.56	1.61	1.83	2.21	1.87	1.69	**1.54**	1.58
cmc	1.49	1.49	1.51	1.49	1.50	**1.48**	**1.48**	**1.48**
colic.ORIG	0.88	0.84	0.86	0.92	0.77	**0.74**	0.86	0.86
credit-a	0.94	0.86	0.99	0.69	0.96	**0.66**	0.74	0.97
credit-g	0.88	0.88	0.87	0.88	0.87	**0.82**	0.87	0.87
dermatology	1.02	0.79	0.78	1.57	1.13	0.42	0.40	**0.31**
heart-c	0.67	0.66	0.99	0.74	0.67	**0.65**	0.68	**0.65**
hepatitis	0.60	0.73	0.73	0.70	0.63	0.56	0.61	**0.54**
labor	0.90	0.88	0.80	0.90	0.72	**0.59**	0.69	0.69
lymph	0.77	1.01	0.80	1.20	0.95	0.77	0.78	**0.60**
postoperative	1.03	1.02	**0.97**	1.03	0.99	1.22	1.18	1.19
tae	1.51	1.49	1.47	1.52	**1.43**	1.46	1.46	1.44
vowel	3.41	3.42	3.33	3.46	3.41	**3.21**	3.34	3.28
zoo	0.27	0.51	0.29	0.99	0.34	0.34	**0.25**	**0.25**

This procedure was repeated for the Cluster Error measure (Table 6). Again, as shown in Tables 5 and 6, the best-performed algorithm is CABC. It is important to note that postprocessing benefited all algorithms.

Table 6. Cluster Error for compared algorithms after postprocessing. The best results for each dataset are in bold.

Datasets	KMSF	kPrototypes	HIMIC	CEBMDC	AGKA	CABC-DB	CABC-DN	CABC-ST
anneal	0.24	**0.16**	0.23	0.24	0.26	0.22	0.17	0.17
autos	0.46	0.47	0.57	0.67	0.56	0.49	**0.45**	**0.45**
cmc	**0.57**	**0.57**	**0.57**	**0.57**	0.58	**0.57**	**0.57**	**0.57**
colic.ORIG	0.37	0.33	0.33	0.37	0.27	**0.21**	0.35	0.35
credit-a	0.41	0.29	0.44	0.20	0.43	**0.17**	0.21	0.44
credit-g	0.30	0.30	0.30	0.30	**0.29**	0.30	0.30	0.30
dermatology	0.34	0.29	0.29	0.50	0.33	0.15	0.15	**0.07**
heart-c	**0.17**	**0.17**	0.46	0.21	0.18	**0.17**	0.18	**0.17**
hepatitis	0.21	0.21	0.21	0.21	0.21	**0.19**	0.20	0.20
labor	0.35	0.35	0.35	0.35	0.22	**0.16**	0.19	0.19
lymph	0.21	0.33	0.23	0.45	0.26	0.19	0.19	**0.14**
postoperative	**0.30**	**0.30**	**0.30**	**0.30**	0.35	0.35	0.33	0.36
tae	0.56	0.55	0.53	0.59	**0.49**	**0.49**	0.53	0.53
vowel	0.89	0.89	**0.87**	0.91	**0.87**	**0.87**	0.88	0.88
zoo	**0.08**	0.17	**0.08**	0.27	0.14	0.12	**0.08**	**0.08**

Figures 4 and 5 present the individual results of the algorithms before (BP) and after (AP) the PAntSA preprocessing according to Entropy and Cluster Error, respectively.

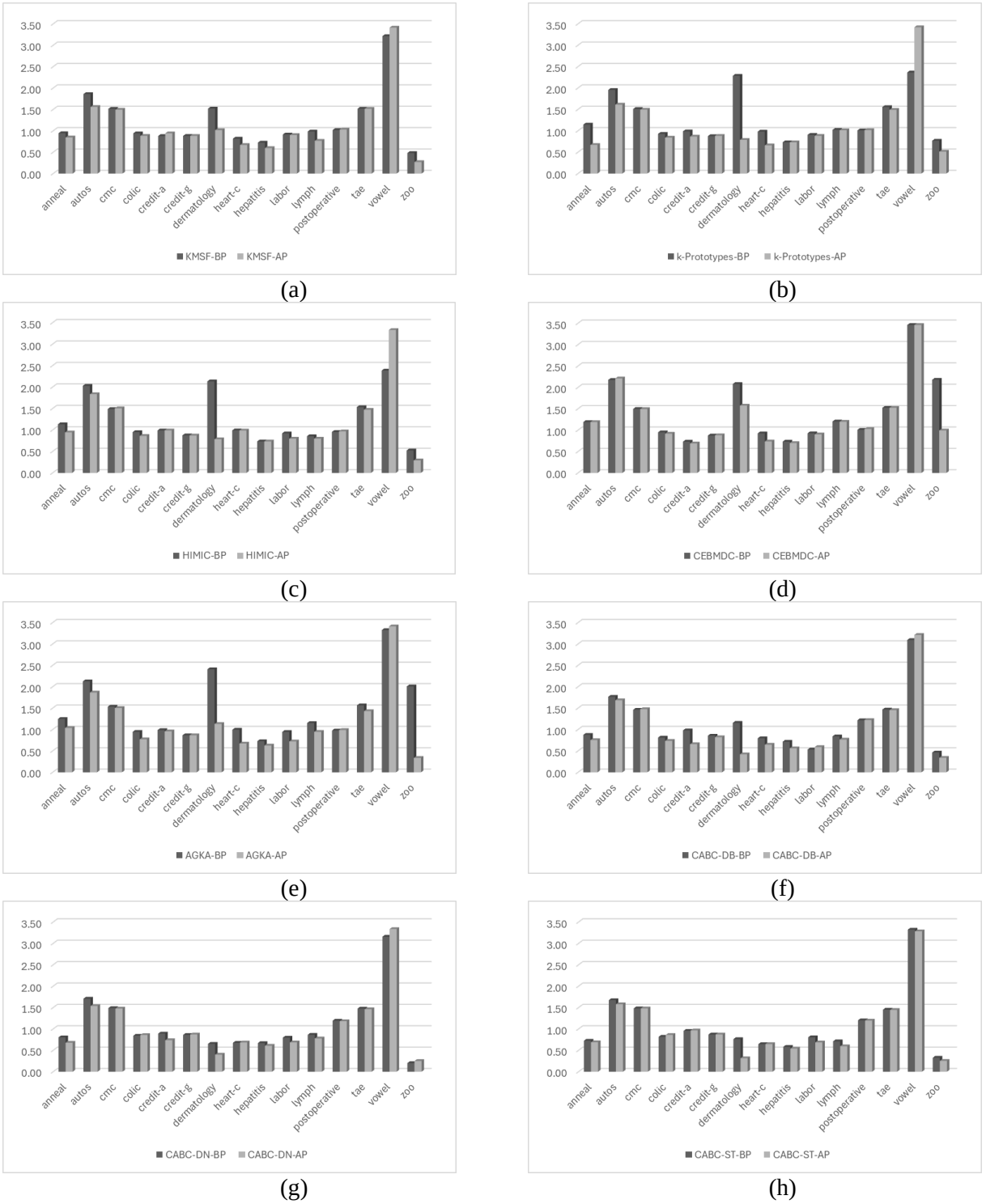

Fig. 4. Entropy of the algorithms before (BP) and after (AP) PAntSA (a) KMSF (b) kProtptypes (c) HIMIC (d) CEBMDC (e) AGKA (f) CABC-DB (g) CABC-DN, and (h) CABC-ST.

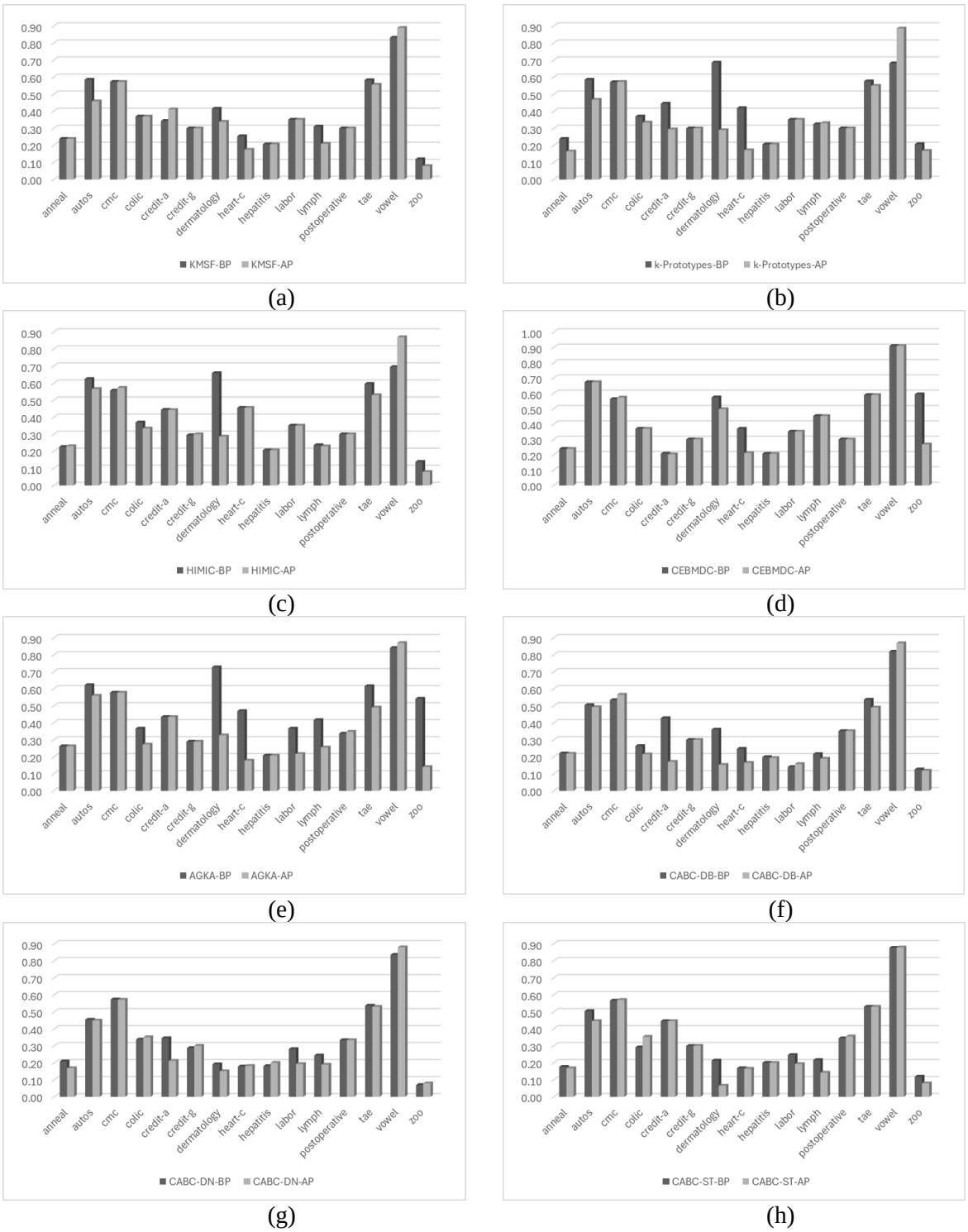

Fig. 5. Cluster Error of the algorithms before (BP) and after (AP) PAntSA (a) KMSF (b) kProtptypes (c) HIMIC (d) CEBMDC (e) AGKA (f) CABC-DB (g) CABC-DN, and (h) CABC-ST.

Figure 6 summarizes the average Entropy (left axis) and Cluster Error (right axis) of each method before (BP) and after (AP) postprocessing, showing a huge improvement in the performance of the clustering algorithms.

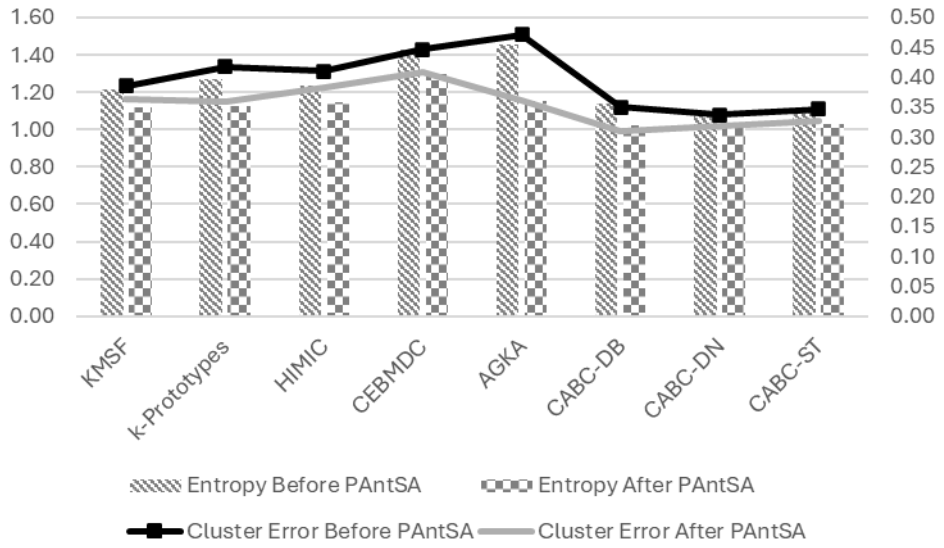

Fig. 6. Averaged Entropy (right axis) and Cluster Error (left axis) of the compared algorithms before (BP) and after (AP) PAntSA preprocessing.

Despite the obvious numerical differences in the algorithms after and before the PAntSa postprocessing, we conducted a non-parametric statistical analysis. The Wilcoxon test allows us to clarify whether PAntSA actually significantly improves or not the results obtained by the grouping methods. For this, a confidence level of 95% was established. Thus, for probability values greater than 0.05, it is considered that there are no differences in the quality of the groups after applying the PAntSA. For this, the symbol 😐 is chosen to facilitate understanding. However, for values less than 0.1, it is necessary to determine if this method improves or worsens the results obtained by the grouping method, using the symbols ☺ and ☹, respectively.

Table 7. Wilcoxon tests for Entropy measure for compared algorithms after and before postprocessing.

Pair	R+	R-	p-value	Decision	PAntSA performance
KMSF-AP vs. KMSF-BP	87.5	17.5	0.019352	Reject H0	☺
kPrototypes-AP v. kPrototypes-BP	101.0	19.0	0.016803	Reject H0	☺
HIMIC-AP vs. HIMIC-BP	81.5	23.5	0.061964	Do not Reject H0	😐
CEBMDC-AP vs. CEBMDC-BP	91.0	29.0	0.067547	Do not Reject H0	😐
AGKA-AP vs. AGKA-BP	100.0	5.0	0.002389	Reject H0	☺
CABC-DB-AP vs. CABC-DB-BP	90.0	15.0	0.017056	Reject H0	☺
CABC-DN-AP vs. CABC-DN-BP	79.5	25.5	0.072359	Do not Reject H0	😐
CABC-ST-AP vs. CABC-ST-BP	100.0	20.0	0.012592	Reject H0	☺

Table 8. Wilcoxon tests for Cluster Error measure for compared algorithms after and before postprocessing.

Pair	R+	R-	p-value	Decision	PAntSA performance
KMSF-AP vs. KMSF-BP	75.5	29.5	0.140146	Do not Reject H0	😐
kPrototypes-AP v. kPrototypes-BP	83.0	22.0	0.049825	Reject H0	☺
HIMIC-AP vs. HIMIC-BP	74.5	30.5	0.157811	Do not Reject H0	😐
CEBMDC-AP vs. CEBMDC-BP	81.5	38.5	0.211476	Do not Reject H0	😐
AGKA-AP vs. AGKA-BP	89.0	16.0	0.020194	Reject H0	☺
CABC-DB-AP vs. CABC-DB-BP	79.0	26.0	0.077701	Do not Reject H0	😐
CABC-DN-AP vs. CABC-DN-BP	77.5	44.5	0.360067	Do not Reject H0	😐
CABC-ST-AP vs. CABC-ST-BP	75.0	30.0	0.145592	Do not Reject H0	😐

The symbols ☺ and ☹ mean that PAntSA significantly improved/worsened the results of the corresponding algorithm, respectively, while the symbol 😐 means that no significant differences were evident between the algorithm's results before and after applying PAntSA. No evidence was found that PAntSA worsened the results obtained. As can be seen, PAntSA is capable of significantly improving the clustering results according to Entropy obtained by all algorithms except for HIMIC, CEBMDC, and CABC-DN which did not show significant differences at 95% of confidence. However, the p-values were lower than 0.1; therefore, at 90% confidence, we can assure PAntSA improved the results for such algorithms.

According to the Cluster Error measure, using PAntSA significantly improved kPrototypes and AGKA with 95% confidence and CABC-DB with 90% confidence. The remaining algorithms did not present significant improvements, although the postprocessing results correspond to higher ranks. In no scenario did using the PAntSA postprocessing worsen the results obtained by the clustering methods.

5 Conclusions

Obtaining high-quality clustering on hybrid and incomplete data is of particular importance. The study carried out allows us to assert that the results obtained by clustering methods of diverse nature (partitional, hierarchical, bioinspired and others) can be refined by applying post-processing strategies. The PAntSA algorithm, in all cases, improved or maintained the quality of the groups analyzed, and in no case did its application imply a detriment to it. On the other hand, the use of internal validation indices opens new lines of research regarding the quality of the clustering since it is considered in addition to the properties of compactness and separability. The limitations of this study are given by the amount of data and algorithms studied, so in the future it is intended to carry out a more extensive study.

Aknowledgemets

The authors would like to thank the Instituto Politécnico Nacional (Secretaría Académica, Comisión de Operación y Fomento de Actividades Académicas, Secretaría de Investigación y Posgrado, Centro de Innovación y Desarrollo Tecnológico en Cómputo, and Centro de Investigación en Computación), the Consejo Nacional de Humanidades, Ciencia y Tecnología (CONAHCYT), and Sistema Nacional de Investigadores (SNII) for their support developing this work.

References

Ahmad, A., & Dey, L. (2007). A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering, 63*(2), 503-527.

Ahmad, A., & Dey, L. (2011). A k-means type clustering algorithm for subspace clustering of mixed numeric and categorical datasets. *Pattern recognition letters, 32*(7), 1062-1069.

Ahmed, R., Borah, B., Bhattacharyya, D., & Kalita, J. (2005). HIMIC: A Hierarchical Mixed Type Data Clustering Algorithm. *Department of Computer Science and Information Technology*.

Azzag, H., Monmarche, N., Slimane, M., & Venturini, G. (2003). *AntTree: A new model for clustering with artificial ants.* Paper presented at the The 2003 Congress on Evolutionary Computation, 2003. CEC'03.

Brun, M., Sima, C., Hua, J., Lowey, J., Carroll, B., Suh, E., & Dougherty, E. R. (2007). Model-based evaluation of clustering validation measures. *Pattern recognition, 40*(3), 807-824.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*(2), 224-227.

González-Patiño, D., Villuendas-Rey, Y., Saldaña-Pérez, M., & Argüelles-Cruz, A.-J. (2023). A Novel Bioinspired Algorithm for Mixed and Incomplete Breast Cancer Data Classification. *International Journal of Environmental Research and Public Health, 20*(4), 3240.

Huang, Z. (1997). *Clustering large data sets with mixed numeric and categorical values.* Paper presented at the Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining,(PAKDD).

Ingaramo, D., Errecalde, M., & Rosso, P. (2010). *A general bio-inspired method to improve the short-text clustering task.* Paper presented at the International Conference on Intelligent Text Processing and Computational Linguistics.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization, 39*(3), 459-471.

Kelly, M., Longjohn, R., & Nottingham, K. (2023). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. .

MacQueen, J. (1967). *Some methods for classification and analysis of multivariate observations.* Paper presented at the Proceedings of the fifth Berkeley symposium on mathematical statistics and probability.

Martínez Trinidad, J. F., Garcia Serrano, J. R., & Ayaquica Martínez, I. O. (2002). C-means algorithm whit similarity functions. *Computación y Sistemas, 5*(4), 241-246.

Maulik, U., & Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(12), 1650-1654.

Rendón, E., Abundez, I., Arizmendi, A., & Quiroz, E. M. (2011). Internal versus external cluster validation indexes. *International Journal of computers and communications, 5*(1), 27-34.

Reyes-González, R., & Ruiz-Shulcloper, J. (1999). *An algorithm for restrictive space structuralization*. Paper presented at the Iberomerican Congress on Pattern Recognition (CIARP), La Habana, Cuba.

Roy, D. K., & Sharma, L. K. (2010). Genetic k-Means clustering algorithm for mixed numeric and categorical datasets. *International Journal of Artificial Intelligence & Applications, 1*(2), 23-28.

Ruiz-Shulcloper, J. (2008). Pattern recognition with mixed and incomplete data. *Pattern Recognition and Image Analysis, 18*(4), 563-576.

Villuendas-Rey, Y., Barroso-Cubas, E., Camacho-Nieto, O., & Yáñez-Márquez, C. (2021). A general framework for mixed and incomplete data clustering based on swarm intelligence algorithms. *Mathematics, 9*(7), 786.

Wilson, D. R., & Martinez, T. R. (1997). Improved heterogeneous distance functions. *Journal of artificial intelligence research*, 1-34.