SN Computer Science (2025) 6:711
https://doi.org/10.1007/s42979-025-04248-y

ORIGINAL RESEARCH

A Split and Merge Strategy for Multi-objective Clustering Algorithms

Beatriz Flamia Azevedo[1,2,3] · **Ana Maria A. C. Rocha**[3] · **Ana I. Pereira**[1,2,3]

Received: 5 December 2024 / Accepted: 20 July 2025

Abstract
Complex real-world problems require advanced models for large datasets; combining optimization and machine learning methods can enhance solution effectiveness and efficiency. This work presents an automatic bio-inspired clustering algorithm named Multi-objective Clustering Algorithm II. Through an optimization process, the algorithm autonomously determines the number of clusters, their centroids, and the optimal distribution of their elements. Furthermore, the paper also presents a split and merge strategy for clustering algorithms, with a special focus on multi-objective ones. The proposed algorithms were executed on 10 benchmark datasets, yielding satisfactory results by accurately estimating the optimal number of clusters and providing appropriate dataset partitions. These results outstand the *k*-means and DBSCAN algorithms results, which were used as a comparison.

Keywords Multi-objective clustering · Collaborative system · Partitioning · Data analysis

Mathematics Subject Classification 90C29 · 90C30 · 90B50

Introduction

Clustering multi-objective problems with measure combination is a complex task, especially in real-world applications where decision-makers must navigate trade-offs and make informed choices based on multiple criteria. The approach's success depends on carefully selecting algorithms, appropriate measure combination techniques, and a deep understanding of the problem domain. Clustering algorithm is very useful in engineering, health science, humanities, economics, and other areas [1–6]. It consists of analyzing the intrinsic characteristics of each element of the dataset and grouping the elements with similarities in the same group and those with dissimilarities in other different groups [7, 8].

A multi-objective partitioning clustering algorithm considers different clustering measures to automatically define the optimal number of clusters by minimizing the intra-cluster distance, which is a distance measure computed within the boundary of a cluster, and simultaneously maximizing the inter-cluster distance, which refers to a distance measure computed between different clusters, by defining a bi-objective function [9, 10].

According to Dutta et. al [9], single-objective clustering algorithms are adequate for efficiently grouping linearly separable clusters, while multi-objective strategies are more suitable for complex cases. In this context, the authors proposed a Multi-objective Genetic Algorithm (MOGA) for automatic clustering, considering numerical and categorical features, that takes advantage of the local search ability of *k*-prototypes clustering algorithm with the global search ability of MOGA to find the optimal number of clustering division, *k*. The algorithm intends to minimize the intra-cluster distance and maximize the inter-cluster distance. The authors proposed randomly generating some population chromosomes, where the number

✉ Beatriz Flamia Azevedo
beatrizflamia@ipb.pt

Ana Maria A. C. Rocha
arocha@dps.uminho.pt

Ana I. Pereira
apereira@ipb.pt

1 Research Centre in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Campus Santa Apolónia, 5300-253 Bragança, Portugal

2 Laboratório Associado para a Sustentabilidade e Tecnologia em Regiões de Montanha, Instituto Politécnico de Bragança, Campus Santa Apolónia, 5300-253 Bragança, Portugal

3 ALGORITMI Research Centre / LASI, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal

Published online: 31 July 2025

of clusters can be from 2 to the round square root of the number of elements in the dataset. The algorithm selects the points lying on the Pareto-optimal front. The optimization problem is considered a maximization problem, and points on the Pareto-optimal front are selected as chromosomes for the next generation of MOGA. From the second generation, for each event generation, the populations are made from selected Pareto elements of previous generations. Each point provides a clustering solution. So, in the last generation, it is necessary to select the best solution from the selected set of solutions. This selection is based on a novel majority voting technique based on nine clustering validity indices to choose a solution from the set of solutions lying on the Pareto front. Dutta et al. [11, 12] also have other works with a similar approach considering the GA techniques but considering different objective functions, such as maximization of Homogeneity and Separation measures.

A hybrid method of gradient-based and improved nondominated sorting genetic algorithm is proposed in [6] to balance convergence and diversity in multi-objective algorithm domains. A new partition clustering method based on target space is presented in this case. Also, the finite-difference method is used to obtain gradient information for multiple objective functions, which are used to construct Pareto descent directions that can accelerate convergence.

Another contribution was developed by Kaur and Kumar [13], which presents a multi-objective clustering algorithm based on a vibrating particle system, considering as an objective function the intra-cluster variance and the connectedness; besides the vibrating particle system was used for optimizing the objectives to obtain good clustering results. Also, in this approach, it is required to indicate the number of clusters previously. The approach could effectively address a wide variety of clustering problems compared to other single and multi-objective algorithms in different datasets through several performance measures.

Binu et al. [14] presented a multi-objective approach for clustering to establish the relationship between inter-cluster and intra-cluster distances. Three objective functions were considered to simultaneously minimize the sum of the distance between the elements and their centroids, maximize the sum of the distance between the centroids, and minimize the sum of the distance between elements of the same cluster. Nevertheless, this approach requires the prior specification of the optimal number of centroids, and their final position is obtained randomly from the centroid position that generates the minimum sum of the distance between the elements and their centroids. The algorithms were tested in different benchmark datasets and compared with single objective clustering algorithms, demonstrating superior performance.

In [15], the first version of the Multi-objective Clustering Algorithm I (MCA-I) is presented, which evaluates intra- and inter-clustering measures to define the optimal number of centroids and their optimal position. It is achieved by a multi-objective optimization strategy that simultaneously minimizes intra-clustering distances and maximizes inter-clustering distances. In this approach, several pairs of intra- and inter-clustering measures are considered, generating the Hybrid Pareto front [16], which consists of a Pareto front generated after combining and comparing different non-dominated-solutions provided by a set of pairs of measures that are considered in the multi-objective approach.

This work is an extension of the work [16], which improves the algorithm presented in [15]. Here, a new version of the MCA is proposed, called the Multi-objective Clustering Algorithm II (MCA-II). This algorithm presents an improved mathematical modulation combined with a novel split and merge strategy named the Clustering Connected Strategy. Furthermore, a procedure to support selecting the most appropriate solution among the ones composing the Hybrid Pareto front is proposed, the Clustering Collaborative Indices Procedure, which combines the strength of five different clustering validated indices to evaluate the Hybrid Pareto front and support the final choice.

The MCA is developed by combining clustering methods with multi-objective procedures to automatically determine the optimal number of clusters and partition elements based on various clustering measures. The proposed algorithm was performed on 10 benchmark datasets, yielding satisfactory results by accurately estimating the optimal number of clusters and providing appropriate dataset partitions. These results outstand the *k*-means and DBSCAN algorithms results, which were used as a comparison.

This paper is organized as follows: after introduction, "Split and Merge Concepts and Strategies" section explains the main concepts of split and merge strategies and the state-of-the-art of the topic. Afterward, "Proposed Method" section defines the methods and algorithms proposed. The results are described and discussed in "MCA-II Results" section. Finally, "Conclusions" section concludes the paper and defines the future path of future research.

Split and Merge Concepts and Strategies

Given the challenges inherent in dividing datasets and calculating centroids accurately, additional methods are frequently employed to enhance the effectiveness of partitioning clustering algorithms. Among these approaches are the split and merge techniques, which are used to refine existing clustering solutions by iteratively adjusting the cluster structure of the dataset through splitting and merging operations guided by specific criteria.

The split strategy consists of finding the worst clusters based on quality measures and splitting them into several small clusters. In turn, the merge step behaves similarly to agglomerative hierarchical clustering, so the strategies consist of finding the clusters with the closest distance between them and merging them together [17]. Several split and merge strategies have been proposed in the literature.

Rehman et al. [18] proposes a new non-parametric clustering algorithm. In this case, the novelty is based on split and merge strategies. According to the authors, the merge process will only achieve a good cluster division if the data is well divided during the cluster division process. So, an optimized *k*-means algorithm is used to define and divide the dataset into the optimal number of clusters concerning each data dimension, and the intersection of all dimensions is used to calculate the total number of optimal sub-clusters in the data. After that, three steps establish the merge process: projection of the data, probability density estimation of the projected data, and calculation of the overlap region. The proposed algorithm was tested on 20 benchmark datasets of different shapes and densities, providing excellent abilities for working with different shapes and densities and dealing with noise and outliers data.

Another novelty named Split-Merge Evolutionary Cluster is proposed by [19]. This algorithm analyzes the correlation between two clustering solutions, and according to the recognized patterns, a split or merge strategy is used to update the clustering process. The proposed framework models two clusterings (the existing and the newly constructed one) as a bipartite graph that is decomposed into connected components (bi-cliques). In the first step, all bi-cliques of the graph are found, and after the evaluation of different scenarios, the clustered nodes are removed from the graph. The remaining bi-cliques are decomposed into split/merge sub-components in the second step. The performance of the algorithm was analyzed by three validation measures (Silhouette Index, F-measures, and Jaccard index) and also compared to the solutions produced by the PivotBiCluster [20] and Dynamic split and merge [21] algorithms. The proposed algorithm demonstrated the ability to deal with clustering problems and provide flexibility to update the existing clustering solutions. At the same time, the approach [22], which is an extension of the work [19], is applied to multi-view data applications. The proposed algorithm demonstrated abilities to process, analyze, and integrate multi-view streaming data. Moreover, it was able to perform vertical and horizontal data integration. Both algorithms update an existing clustering solution well when new data are added.

Traditionally, clustering algorithms have the problem of dealing with non-round shape data. Considering this, Chen and Lu [23] proposed the Minimum Spanning Tree clustering algorithm based on density estimation and split and merge methods. In the first stage, the densities of patterns are employed, supported by the shortest path graph strategy. The density estimation method is designed for the split stage, and the maximal connected subgraph is employed for the merge stage. So, the proposed clustering algorithm uses density to determine the similarity of two patterns. When the gap of the density of the edge's patterns is greater than a threshold, the edge should be split. After the split stage, if the number of the divided edges is equal to $k-1$, where k is the number of clusters of a dataset, the clusters do not merge. Thus, in the merge stage, the edges connecting two patterns belonging to a cluster need to be put back, but the edges connecting two patterns belonging to two clusters can not be put back. If the number of patterns in the two maximum connected subgraphs that contain one of the split edge's patterns is greater than a threshold, the edge can not be split in any event. Although the proposed methods demonstrated low ability in dealing with high dimensional datasets, the method presented a robust ability to deal with noises and recognize complex shapes.

Table 1 summarizes some of the split and merge clustering techniques found in the literature.

According to [18], the split and merge techniques play a crucial role in refining the partitioning of data in clustering algorithms, enabling the creation of more accurate and meaningful clusters. By iteratively adjusting the cluster structure based on specific criteria, these techniques contribute to the effectiveness and efficiency of partitioning clustering approaches.

Proposed Method

Measure combination in clustering algorithms refers to integrating multiple evaluation measures or criteria to assess the quality and performance of clustering solutions. In the context of multi-objective clustering, where multiple conflicting objectives exist, measure combination becomes essential for providing a comprehensive assessment of the algorithm's effectiveness.

A multi-objective partitioning clustering algorithm considers different clustering measures to define the optimal number of clusters automatically by minimizing the intra-cluster distance, it is a distance measure computed within the boundary of a cluster and simultaneously maximizes the inter-cluster distance that refers to a distance measure computed between different clusters, by defining a bi-objective function [9, 10].

This section presents the strategies developed to solve the complex clustering problems. Firstly, the concepts of a multi-objective clustering problem are defined, presenting the objective functions considered in the problem. Then, the proposed Multi-objective Clustering Algorithm II is described, where the Clustering Connected Strategy, aiming

Table 1 Split and merge clustering techniques

Split technique	Merge Technique	Reference
k-means algorithm	Projection of the data, probability density estimation of the projected data, and calculation of the overlap region	[18]
Correlation and bipartite graph	Correlation and bipartite graph	[19]
Cross entropy method and Gaussian densities	Cross entropy and Gaussian densities	[24, 25]
Density estimation	Shortest path graph	[23]
Min-max principle through similarity metrics	Min-max clustering principle through cohesion metrics	[26]
Bayesian information criterion	Bayesian information criterion	[27]
Density curve of projection	Density curve of projection	[28]
Gaussian mixtures of factor analyzers Log likelihood fuction	Gaussian mixture of factor analyzers Log likelihood function	[29]
Bayesian information criterion	Homogeneity criterion	[21]

to improve the clustering quality through a split and merge strategy, is explained.

Multi-objective Clustering Problem

A bi-objective clustering problem can be defined as

$$\min F = \{f_a, -g_b\} \tag{1}$$

where f_a, for $a = 1, \ldots, a_{max}$, is the intra-clustering measure and g_b, for $b = 1, \ldots, b_{max}$, is the inter-clustering measure. The intra-cluster measure refers to the distance among elements of a given cluster. In this work, a inter-clustering measure is used as the first objective function f_a, which must be minimized. The inter-cluster measure defines the measure between elements that belong to different clusters. Here, the inter-clustering measure g_b must be maximized, for this reason it is considered a negative signal in (1).

The solution of problem (1) consists of partitioning the dataset represented by $X = \{x_1, x_2, \ldots, x_m\}$, made up of x_i ($i = 1, \ldots, m$) elements, into k optimal clusters. In order to explain the intra- and inter-clustering measures, some notation should be introduced:

- $C = \{c_1, c_2, \ldots, c_k\}$ is the set of centroids, where c_j ($j = 1, \ldots, k$) is the centroid j;
- $C_j = \{x_1^j, x_2^j, \ldots, x_i^j\}$ defines the cluster j, whose element i is represented by x_i^j;
- $\#C_j$ is the number of elements of cluster C_j.

Thus, based on previous works [15, 16], the most prominent measures to be used as intra-clustering and inter-clustering measures are described below.

The sum of the distances between the elements x_i^j belonging to C_j to its centroid c_j, is denoted by Sxc_j. Thus, the intra-clustering measure $SAxc$ represents the average of the distance in each cluster from the element to its centroid, in terms of the number of elements belonging to each cluster set $\#C_j$, as defined in (2)

$$SAxc = \sum_{j=1}^{k} \frac{Sxc_j}{\#C_j}. \tag{2}$$

The intra-clustering measure FNc gives the sum of the distance of the furthest neighbor within the cluster. It evaluates the sum of the furthest neighbor distance of each cluster C_j, where x_i^j and x_l^j belong to the same cluster j, as presented in (3)

$$FNc = \sum_{j=1}^{k} \max\{D(x_i^j, x_l^j)\} \text{ for } i = 1, \ldots, \#C_j,\ l = 1, \ldots, \#C_j,\ i \neq l. \tag{3}$$

The inter-cluster measure Scc, defined in (4), represents the sum of the distance between centroids [30]

$$Scc = \sum_{\substack{t,j=1,\\ t \neq j}}^{k} D(c_t, c_j). \tag{4}$$

Thus, the average of the distance between centroids constitutes the inter-cluster measure Acc that is based on Scc, and is presented in (5),

$$Acc = \frac{Scc}{k}. \tag{5}$$

The sum of the furthest neighbor distance between elements of different clusters C_j defines the inter-cluster measure $FNcc$, known as the complete linkage [31], which is given by

$$FNcc = \sum_{j=1}^{k} \sum_{t>j}^{k} \max \{D(x_i^j, x_l^t)\} \quad \text{for } i = 1, \ldots, \#C_j, l = 1, .., \#C_t. \tag{6}$$

Thus, the inter-cluster measure *AFNcc* gives the average of the sum of the furthest neighbor distances between elements of k different clusters, as presented in (7)

$$AFNcc = \frac{FNcc}{k}. \tag{7}$$

Another inter-cluster measure considered is *NNcc*, that is the sum of the nearest neighbor distance between elements of different clusters, known as single linkage [30], is defined in (8),

$$NNcc = \sum_{j=1}^{k} \sum_{t>j}^{k} \min \{D(x_i^j, x_l^t)\} \quad \text{for } i = 1, \ldots, \#C_j, \ l = 1, .., \#C_t. \tag{8}$$

Finally, the inter-cluster measure denoted as *ANNcc*, is the average of the sum of the nearest neighbor distance between elements of k different clusters, defined in (9),

$$ANNcc = \frac{NNcc}{k}. \tag{9}$$

Multi-objective Clustering Algorithm II

The Multi-objective Clustering Algorithm II (MCA-II) can be defined in 10 stages, as shown in Fig. 1, and described below.

Stage 1—Input data: the algorithm starts with the input of the dataset X, and the definition of the values k_{min} and k_{max}, which represent the minimum, and maximum number

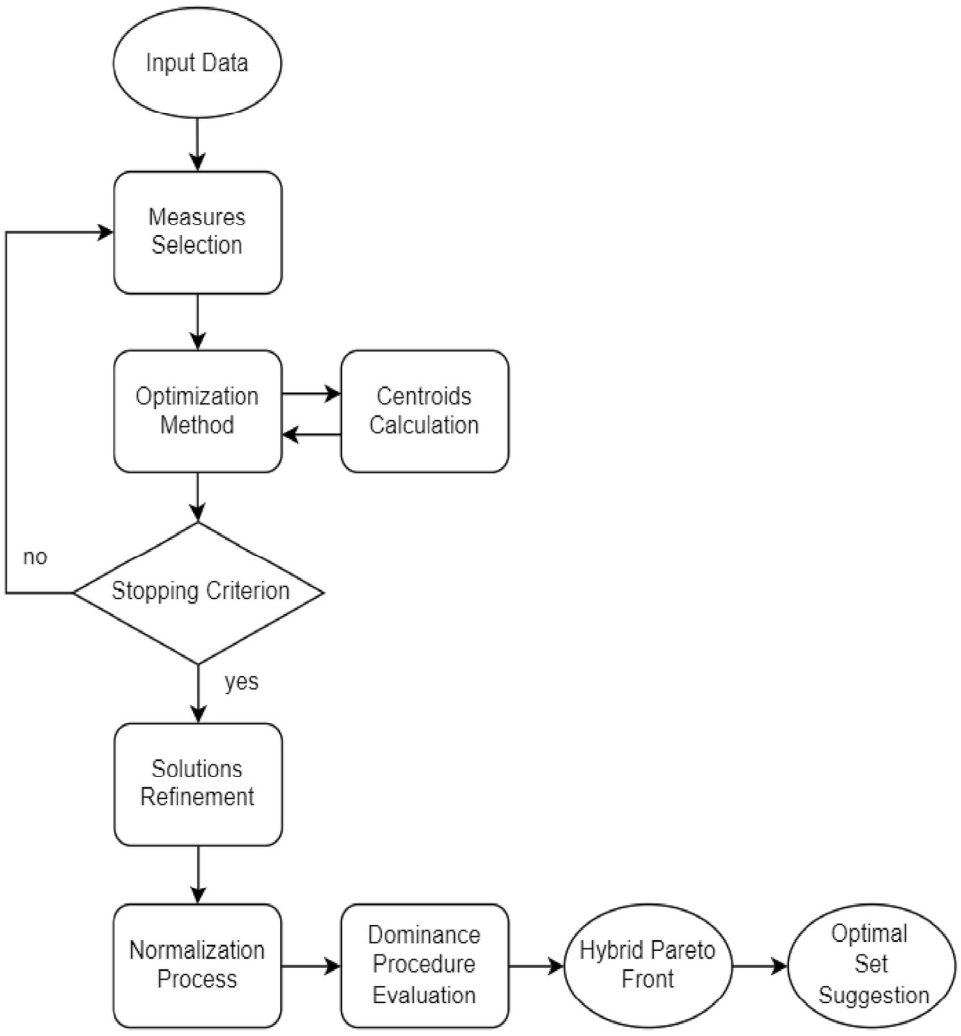

Fig. 1 Multi-objective Clustering Algorithm II (MCA-II)

of clusters that could be assigned. The MCA-II automatically defines the optimal number of cluster partitions in a range of possible partitions. The value k_{min} and k_{max} can be given by the user or considered by default $k_{min} = 2$ and $k_{max} = [\sqrt{m}]$ [32].

Stage 2—Measures selection: a pair of measures is automatically selected, one intra-measure and another inter-measure, in which f_a, is an intra-clustering measure, among $f_1 = SAxc$ $f_2 = FNc$ and g_b, is the inter-clustering measure among $g_1 = Scc$, $g_2 = Acc$, $g_3 = FNcc$, $g_4 = AFNcc$, $g_5 = NNcc$, and $g_6 = ANNcc$. More details can be seen in [15, 16].

Stage 3—Centroids calculation: after the measures selection it is necessary to define the centroids. The Centroids Calculation is an iterative procedure that randomly selects k candidates to be evaluated as possible centroids. The range of this k is an integer between k_{min} and k_{max}. Next, the Euclidean distance D between all the elements of X up to each centroid j is evaluated. The closest elements of each centroid j define a cluster set C. To avoid small cluster sets, a minimum number of elements per cluster is defined, as ζ, [33]. Thereby, the centroids j that have less than ζ associated elements are automatically removed from the set of centroids, and the elements become part of other remaining centroid, which is the closest one in terms of Euclidean distance of the elements considered. As default, it is considered $\zeta = [\sqrt{m}]$, with $\zeta \in \mathbb{N}$ [9]. After that, the remaining centroids are definitively denoted as the centroid of each subset c_j, in which X is partitioned.

In the end, the elements are associated with their centroid c_j, so adjusting a position in each j coordinate is necessary to improve the algorithm's performance. Thus, the coordinates of each centroid j assume the coordinates of their barycenter cluster c_j, composed of its x_i^j elements.

For a better understanding of the Centroids Calculation (CC), the Algorithm 1 is presented.

Algorithm 1 Centroids Calculation

Input: X, k_{min}, and k_{max}
  Define ζ
  Randomly define the number of cluster $k \in [k_{min}, k_{max}]$
  Randomly select $c \in \mathbb{R}^{k,d}$
  Evaluate $D(x_i, c_j)$, for all $c_j \in C$, $i \in X$
  Associate each x_i to the nearest c_j
  if $\#C_j > 0$ **then**
    the c_j is kept
  else
    c_j is eliminated
    Update k
  end if
  for $j = 1 : k$ **do**
    if $\#C_j < \zeta$ **then**
      c_j is eliminated, and associate element with the nearest cluster
      Update k
    end if
  end for
  Update c_j based on its cluster barycenter coordinates.
Output: The k, the centroids c_j, and $s^* = \{C_1^*, C_2^*, ..., C_k^*\}$.

Stage 4—Optimization method: to identify the Pareto front associated with that bi-objective function of the problem, it is necessary to use a multi-objective algorithm. In this case, the MCA-II uses a Multi-objective Particle Swarm Optimization (MOPSO) [34]. This iterative evolutionary process revisits Stage 3 as many times as necessary, refining and advancing the population algorithm.

Stage 5—Stopping criterion: the stages 2, 3, and 4 are repeated until all combinations of pairs of measures have been evaluated.

Stage 6—Solutions refinement: the solutions generated can be refined by a clustering refinement strategy, such as split and merge strategy. For the MCA-II, a split and merge strategy named Clustering Connected Strategy (CCS) was developed.

Stage 7—Normalization process: considering the different measures with varying orders of magnitude, normalizing the values is essential to facilitate a comprehensive comparison and ensure a fair and meaningful analysis of the results. Then, each Pareto front generated is normalized through the Min-Max scaling method [35]. That is, each solution s of the Pareto front is individually normalized between [0, 1], using (10),

$$f_i^* = \frac{f_i(s) - f_i^{min}}{f_i^{max} - f_i^{min}}, \tag{10}$$

where f_i represents the components of the bi-objective function F, f_i^{min} and f_i^{max} are respectively the smallest and the highest solution value, of the function i, belonging to the Pareto front considered. Thus, $f_i^* = (f_i(s^*))$ is a normalized Pareto front solution of each Pareto front considered considering a given set of measures.

Stage 8—Nondominated procedure evaluation: in this stage, all Pareto front normalized solutions are evaluated regarding nondominated criterion, and the nondominated solutions are selected to compose a Hybrid Pareto front (HPF).

Stage 9—Hybrid Pareto front: the hybrid Pareto front is the set of nondominated solutions, considering all the normalized solutions of the Pareto fronts obtained for each pair of measures.

Stage 10—Optimal set suggestion: in the last stage, the decision-maker can use their knowledge exclusively to select one solution in the optimal set or use a strategy to support their decision. As default, the strategy named Cluster Collaborative Indices Procedure (CCIP) supports the decision-maker choice.

The pseudocode of the MCA-II is presented in Algorithm 2.

Algorithm 2 Multi-objective Clustering Algorithm II

Input: X, k_{min} and k_{max}
  for $a = 1, ...$max number of intra-measures **do**
    for $b = 1, ...$max number of inter-measures **do**
      C $\leftarrow CC - II(X, k_{min}, k_{max})$ ▷ Centroids Calculation of MCA-II
      $PF^{*,a,b} \leftarrow$ MOPSO (X, f_a, g_b) ▷ Pareto front
    end for
  end for
  Apply CCS ▷ Refinement strategy
  NPF^{*,a,b} $\leftarrow$ normalize $PF^{*,a,b}$ ▷ normalize Pareto front
Output: HPF $\leftarrow$ nondominated solution $(\text{NPF}^{*,a,b}, a = 1, ..., a_{max}; b = 1, ..., b_{max})$ ▷ hybrid Pareto front
    $s^* \in HPF$, selected considering a given criteria.

The MCA-II is a block-structured algorithm, allowing some of its stages to operate independently of each other. Thus, Stages 4, 6, and 10 can be replaced or removed according to the convenience of the decision-making. Therefore, the MCA-II can be executed in its complete version, as previously presented in 10 stages, or if the decision-maker deems it necessary, it can replace the MOPSO multi-objective optimization process in Step 4 with another method, such as NSGA-II [36], Multi-Objective Grey Wolf Optimizer [37], or Multi-objective Genetic Algorithm [9], among others. Additionally, Stage 6 refers to the refinement of solutions for which a split and merge method was proposed, which can be eliminated or replaced by another refinement approach, such as outlier removal [38], ensemble techniques [39], among others. Finally, the last stage is used whenever the decision-maker desires a logical indication of the most appropriate solution to the problem. However, the decision-maker can employ their knowledge of the problem and stop the algorithm at Stage 9, autonomously choosing the final solution.

These substitutions or adaptations represent a significant advantage for customizing the algorithm to specific and complex problems. By allowing the individual manipulation of its steps, the MCA-II can easily adapt to problems that require high performance, velocity, and precision. Thus, flexibility is a significant differential of the MCA-II, making it effective for a wide range of problems. Additionally, the greater the number of selection measures used in Stage 2, the greater the variety of optimal solutions offered to the problem.

Clustering Connected Strategy

The Clustering Connected Strategy (CCS) aims to improve the solutions generated by the MCA-II base algorithm through a split and merge strategy. The CCS evaluates the clusters close to each other to explore if an element that belongs to one cluster should be moved to a nearby cluster. The main motivation of this strategy is to improve locally the obtained solution. Figure 2 illustrates the possible relation between two clusters.

In Fig. 2a–d, the clusters are geometrically connected, that is, they have a region of intersection. Specifically, there are some elements in the intersection between the clusters in Fig. 2b, c. Figure 2d represents a case in which a cluster is inside another cluster. Figure 2a has no elements in the

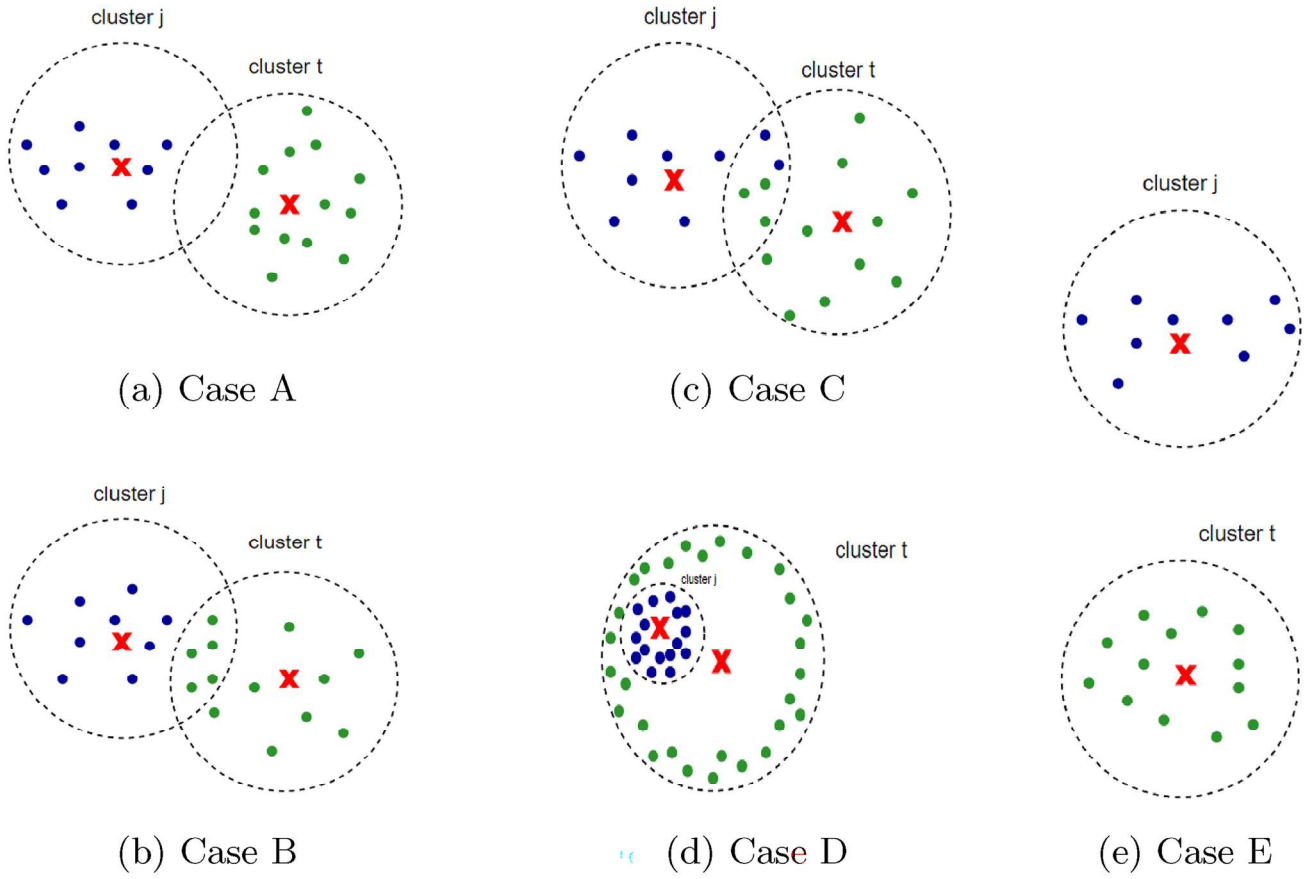

Fig. 2 Two clusters possible relationship

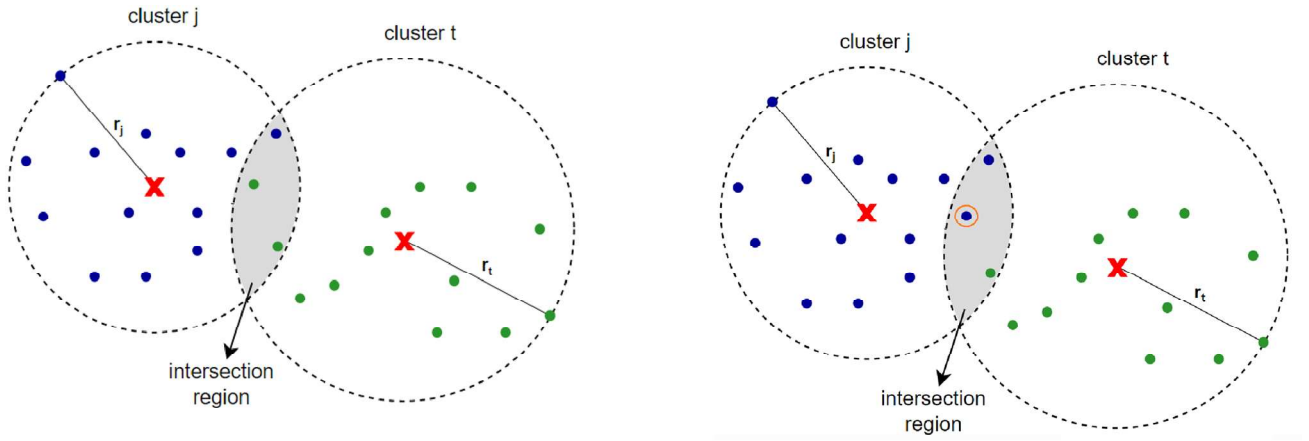

Fig. 3 Connected clusters

intersection region. Finally, Fig. 2e illustrates a case where the clusters are unconnected.

To better explain the CCS, consider a cluster C_j and a cluster C_t, as illustrated in Fig. 3. The elements associated with C_j are denoted as x_i^j (in blue); similarly, the elements related to cluster C_t are denoted as x_l^t (in green).

To geometrically define if two clusters are connected, it is necessary to evaluate each cluster's radius, r. The radius of a cluster is the furthest distance between an element and its centroids, as defined in (11)

$$r_j = \max_{i=1,\ldots,\#C_j} \{D(x_i^j, c_j)\} \tag{11}$$

If the sum of r_j and r_t is greater than the distance between the centroids c_j and c_t (see (12)), the cluster is considered connected (Fig. 2a–d). Otherwise, they are not connected (Fig. 2e).

$$r_j + r_t > D(c_j, c_t) \tag{12}$$

When two clusters are connected, it is necessary to evaluate the elements that belong to the intersection region. An element is considered in the intersection region if the condition considered at (13) is satisfied. If the distance between the element x_i^j until the centroid c_t is smaller than r_t, the element x_i^j is located at the intersection region, and can be checked with the following condition

$$D(x_i^j, c_t) < r_t. \tag{13}$$

Thereby, if no elements are in the intersection (Fig. 2a), no improvement is applied. However, if elements belong to the intersection region, these elements are temporarily removed from its cluster and allocated to a temporary list $L_{jt} = \{lx_1, \ldots, lx_f\}$. After that, the number of the remaining elements of each cluster is evaluated.

If the number of remaining elements in one cluster j or t is smaller than ζ, the clusters j and t are merged. The new centroid is defined as the cluster barycenter. After that, it is necessary to evaluate the distance between the elements $lx_i \in L_{jt}$ to the nearest element of the clusters C_j and C_t, it is $D(lx_i, x_i^j)$ and $D(lx_i, x_i^t)$, respectively. The element lx_i should be associated within the same cluster of the nearest neighbor element that does not belong to L_{jt}. Thereafter, the centroid of the cluster set that receives the new element is updated based on the cluster barycenter. This process occurs until the list is empty.

Considering the example illustrated in Fig. 3a , there are three elements in the intersection region. In the example, two elements belong to cluster j and one to cluster t. After the CCS performance, the element highlighted that initially belonged to cluster t is replaced in the cluster j since its nearest neighbor belongs to cluster j (Fig. 3b).

The CCS algorithm is an iterative process, and it stops when the maximum number of iterations τ is achieved or when

no improvement is verified between the solution of the previous iteration and the current one. The Algorithm 3 presents the pseudocode of the CCS strategy. The input data is s_i, where s_i is one solution obtained by MOPSO. To highlight that s_i represents one distribution with k clusters of the data X.

Algorithm 3 Clustering Connected Strategy (CCS)

Consider the input parameters: $s_i \in HPF$, τ.
Define ζ
while NIT $< \tau \vee s_i$ is not change in the previous iteration **do**
  for $j = 1, ..., k-1$ **do**
    for $t = j+1, ..., k$ **do**
      $r_j = \max_{i=1,...,\#C_j} D(x_i^j, c_j)$
      $r_t = \max_{i=1,...,\#C_t} D(x_i^t, c_t)$
      if $r_j + r_t > D(c_j, c_t)$ **then**
        Define $L_{jt} \leftarrow \{x_i^j, x_i^t\}$ all point of the intersection region.
        $C_j = C_j \setminus \{x_i^j\}$
        $C_t = C_t \setminus \{x_i^t\}$
        if $\#C_j < \zeta$ or $\#C_t < \zeta$ **then**
          $C_t = C_t \cup C_j \cup L_{jt}$
          Eliminate C_j
          Update C_t and c_t
        else
          while $\#L_{jt} \neq 0$ **do**
            Select $lx_i \in L_{jt}$
            Found x_i^j and x_l^t that are the nearest of lx_i
            if $\min_{l=1,...,\#C_j} D(lx_i, x_i^j) < \min_{l=1,...,\#C_t} D(lx_i, x_i^t)$ **then**
              $C_j = C_j \cup \{lx_i\}$
            else
              $C_t = C_t \cup \{lx_i\}$
            end if
            Update $L_{ij} = L_{ij} \setminus \{lx_i\}$
          end while
        end if
        Update k
      end if
    end for
  end for
  Update NIT
end while
Output: The k, the centroids c_j, and $s_i = \{C_1, C_2, ..., C_k\}$.

Clustering Collaborative Indices Procedure

To evaluate the quality of the solutions generated by the Hybrid Pareto front, the Cluster Collaborative Indices Procedure (CCIP) was developed. Cluster validity Indices (CVI) are quantitative measures used to evaluate the quality or validity of clustering results. These indices provide a way to compare different clustering solutions or algorithms objectively and select the most appropriate one based on certain criteria. The outcome of the CVI only depends on the partition provided by the clustering algorithm given a specific number of clusters [40].

Numerous indices have been developed to evaluate the performance of several clustering algorithms. However, each index has its limitations and biases. These indices yield evaluations across different ranges, and the recommendations derived from them frequently conflict. Consequently, the challenge of selecting the most suitable CVI persists, along with the determination of the optimal clustering algorithm [41, 42]. An optimal solution for one specific CVI could not be optimal for another CVI [41]. Considering this, the CCIP evaluates each Hybrid Pareto front solution according to each CVI and defines the best solution of each CVI; it is the one that most satisfies the CVI criterion. Therefore, the solution that receives the most indications, s^*, is the most appropriate to be selected by the decision-maker. In case of a tie, the set of solutions indicated is considered the most appropriate for the problem, and it is up to the decision-maker to make the final choice.

Here, five CVI were chosen, the classical ones according to the literature and they are described below.

The Davies-Bouldin index (DB) [43], estimates the cohesion based on the distance from the elements x_i^j in a cluster to its centroid c_j and the separation based on the distance between centroids $D(c_j, c_t)$. The smallest DB indicates the optimal partition. First, it is necessary to evaluate an intra-cluster measure represented by the mean distance between each element within the cluster x_i^j and its centroid c_j, which is a dispersion parameter $S(c_k)$, as Eq. (14),

$$S(c_j) = \sum_{i=1}^{\#C_j} \frac{D(x_i^j, c_j)}{\#C_j}. \tag{14}$$

And, the DB index is given by Eq. (15),

$$DB = \frac{1}{k} \sum_{j=1}^{k} \max_{\substack{t = 1, \ldots, k, \\ j \neq t}} \left\{ \frac{S(c_j) + S(c_t)}{D(c_j, c_t)} \right\}. \tag{15}$$

The Dunn index (DI) [44] is a ratio-type index where the cohesion is estimated by the nearest neighbor distance and the separation by the maximum cluster diameter. Thus, a higher DI will indicate compact, well-separated clusters, while a lower index will indicate less compact or less well-separated clusters [44]. The higher the value of the Dunn index, the better the quality of the clusters. So, DI is defined as the rate between the minimum distance between elements of different clusters, and the largest distance between elements of the same cluster (sometimes called cluster diameter), as defined in Eq. (16).

$$DI = \frac{\min\limits_{\substack{j, t = 1, \ldots, k, \\ j \neq t}} \{D(x_i^j, x_l^t), \forall i, l\}}{\max\limits_{j=1,\ldots,k} \{D(x_i^j, x_l^j)\ \forall i, l\}} \tag{16}$$

The Calinski-Harabasz (CH) [45] is a ratio-type index where the cohesion is estimated based on the distance from the elements in a cluster to its centroid [43, 45]. First, it is necessary to calculate the inter-cluster dispersion (BGSS), which measures the weighted sum of squared distance between the centroids of a cluster, c_j, and the barycenter of the X dataset, denoted as $\overline{X}$, as represented in Eq. (17),

$$\overline{X} = \frac{1}{m} \sum_{i=1}^{m} x_i. \tag{17}$$

The BGSS is defined as Eq. (18)

$$BGSS = \sum_{j=1}^{k} \#C_j \times D^2(c_j, \overline{X}). \tag{18}$$

The second step is to calculate the intra-cluster dispersion for each cluster j, evaluated by the sum of the squared distance between the elements x_i^j until its centroid c_j, as defined in Eq. (19).

$$WCSS = \sum_{j=1}^{k} \sum_{i=1}^{\#C_j} D^2(x_i^j, c_j) \tag{19}$$

Table 2 Benchmark datasets characteristics

Name	N. of elements	Dimension	Optimal k	References
Mydata	300	2	3	[48]
Sphere6	300	2	6	[49]
Jain	373	2	2	[50]
Breast	699	9	2	[51]
Aggregation	788	2	7	[52]
Tetra	400	3	4	[53, 54]
Spheres3D	400	3	4	[49]
Iris	150	4	3	[55]
Thyroid	215	5	2	[56]
Yeast	1484	8	10	[57]

The CH index seeks to maximize the dispersion between clusters (increasing BGSS) while minimizing the dispersion within clusters (decreasing WCSS). The higher the value of the CH, the better the separation between clusters. Thus, the CH index is defined as Eq. (20):

$$CH = \frac{m-k}{k-1} \times \frac{BGSS}{WCSS} \tag{20}$$

The CS index [46] is a ratio-type index that estimates the cohesion by the cluster diameters and the separation by the nearest neighbor distance. This measure is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. The smallest CS, defined by Eq. (21) indicates a valid optimal partition [46].

$$CS = \frac{\sum_{j=1}^{k} \frac{1}{\#C_j} \sum_{i=1}^{\#C_j} \max_{\substack{l=1,\ldots,\#c_j \\ l \neq i}} \{D(x_i^j, x_l^j)\}}{\sum_{j=1}^{k} \min_{\substack{t=1,\ldots,k, \\ t \neq j}} \{D(c_j, c_t)\}} \tag{21}$$

The Xie-Beni index (XB) [47] is an index of fuzzy clustering but also applies to crisp clustering. It is defined as the quotient between the mean quadratic error WCSS, as presented in Eq. (19), and the minimal of the minimum squared distances between different centroids. The smallest XB, defined by Eq. (22) indicates an optimal partition.

$$XB = \frac{1}{m} \frac{WCSS}{\min_{i \neq j} \{D^2(c_j, c_t)\}} \tag{22}$$

The pseudo-code of the CCIP is presented in Algorithm 4.

Algorithm 4 Cluster Collaborative Indices Procedure

Input: $CVI = \{CVI_1, ..., CVI_n\}$
Consider the Hybrid Pareto front $HPF = \{s^1, ..., s^f\}$
for i=1,...,#CVI **do**
Calculate for each s^* the CVI_i value
Defines the best solution for each CVI_i
end for
Output: $s_i \leftarrow s^*$, the solution most indicated as the best solution.

Table 3 CVI results for Mydata HPF with CCS

Solution	Pareto front Measures	k	Dunn	DB	CS	CH	XieBeni
–	Literature	3	0.096	0.548	0.876	653.559	10.23
1	SAxc—FNcc	3	0.054	**0.525**	**0.852**	**666.917**	**9.847**
2	SAxc—AFNcc	3	**0.096**	0.548	0.876	653.559	10.23
3	SAxc—NNcc	2	0.062	0.737	1.118	317.850	31.406

The bold values indicate the optimum solution

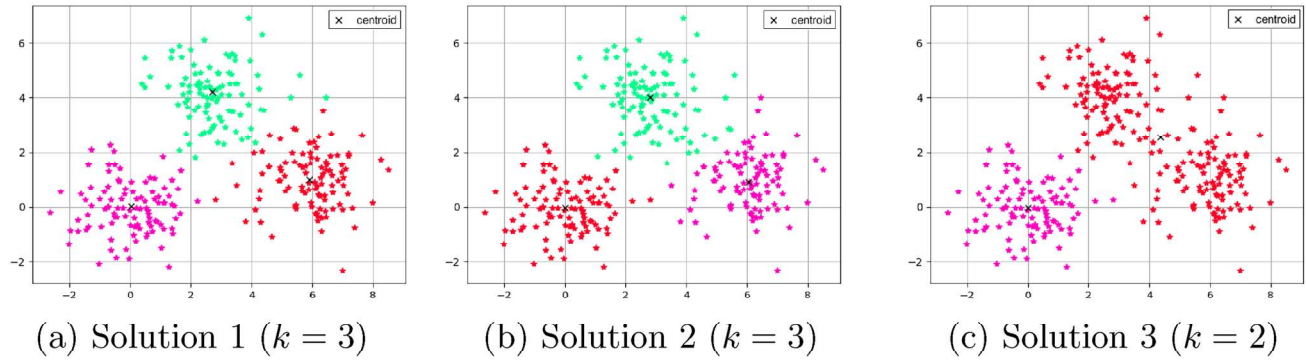

Fig. 4 Mydata HPF solutions with CCS

MCA-II Results

To validate the proposed algorithm, MCA-II, 10 clustering benchmark datasets were selected based on their properties. Each dataset was chosen to cover a diverse range of characteristics, ensuring a comprehensive evaluation of the algorithms' capabilities across different scenarios. The well-known datasets in the literature are presented in Table 2.

Due to their different characteristics, the results of the Jain, Mydata, and Sphere6 datasets are presented in detail

Table 4 CVI results for Sphere6 HPF with CCS

Solution	Pareto front Measures	k	Dunn	DB	CS	CH	XieBeni
–	Literature	6	0.515	0.355	0.405	2713.652	2.747
1	SAxc—Scc	4	**0.658**	0.361	0.413	1043.097	3.261
2	FNc—Scc	6	0.515	**0.355**	**0.405**	**2713.652**	**2.747**

The bold values indicate the optimum solution

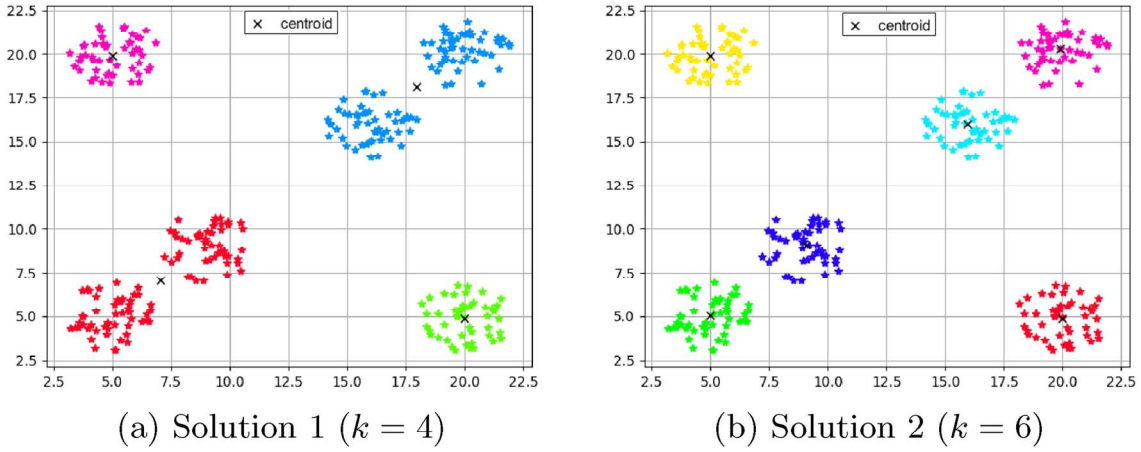

Fig. 5 Sphere6 HPF solutions with CCS

Table 5 CVI results for Jain HPF with CCS

Solution	Pareto front measures	k	Dunn	DB	CS	CH	XieBeni
–	Literature	2	0.092	0.899	1.237	279.48	47.639
1	FNc—ANNcc	2	**0.092**	0.899	1.237	279.48	47.639
2	SAxc—FNcc	3	0.010	**0.669**	1.055	**495.286**	28.032
3	FNc—AFNcc	4	0.045	0.735	**0.976**	479.661	**16.710**
4	SAxc—Acc	2	0.038	0.753	1.066	405.897	39.492

The bold values indicate the optimum solution

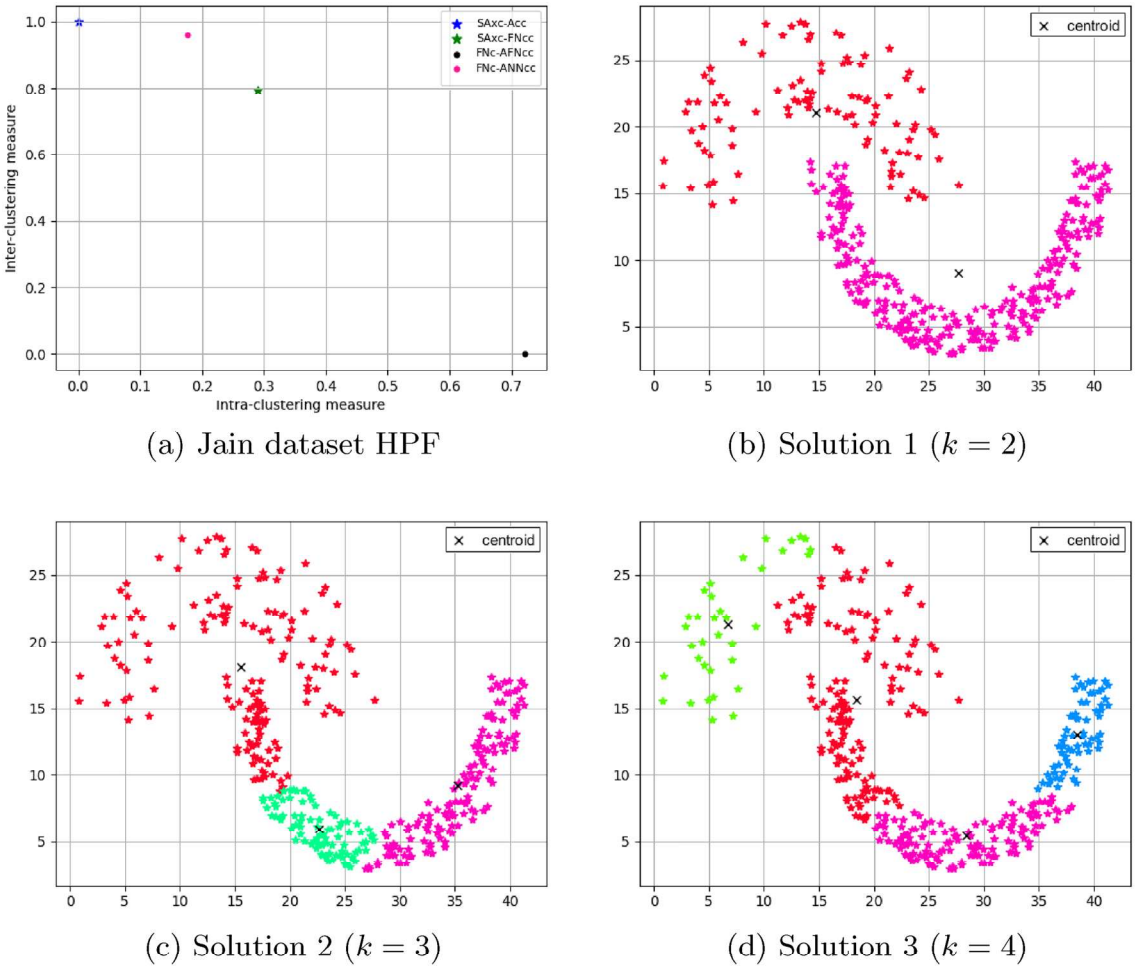

Fig. 6 Jain HPF and solutions with CCS

below and then the results achieved with the other datasets are summarized.

MCA-II performed one run for each of the combinations of measures (2 intra-clustering measures and 6 inter-clustering measures, making a total of 12), since the final solution integrates the results of all measure combinations. It was considered a population of 100 particles for all datasets and each run of MCA-II had 100 iterations as the stopping criterion. In the CCS algorithm, the maximum number of iterations τ equals 30. The same parameters will be used across all datasets.

Mydata Dataset Results

The Mydata dataset [48], available in Matlab library [58], is composed of 300 elements divided into 3 clusters, according to the literature. Therefore, considering the previously mentioned MCA-II parameters, the possible number of clusters can vary between $k_{min} = 2$ and $k_{max} = 17$. And $\zeta = 17$ is the minimum number of elements per cluster, as the algorithm default.

In order to analyze the quality of the solutions generated and also to support the decision-maker in choosing the best solution, the Algorithm CCIP presented in Algorithm 4 is applied. Remember that the higher the Dunn and the CH indices, the better the clustering solutions, and, the lower the DB, CS, and XieBeni indices, the better the clustering solutions.

The HPF is composed of 3 different solutions generated by the combination of measures *SAxc* – *FNcc* (3 clusters), *Saxc* – *AFNcc* (3 clusters), and *SAxc* – *NNcc* (2 clusters).

In this specific case, for each of the three pairs of measures, a single solution remained, and these are different from each other, as described in Table 3. When normalized independently, the three solutions are located in the point (0, 0) on the HPF and thus overlap. Figure 4a–c illustrate the HPF solutions.

Solution 1 considers 3 clusters, but the distribution of the elements is different from the solution 2, which is identical to the solution proposed by the literature, as can be seen by comparing the CVI values of both solutions.

Table 6 MCA-II Benchmark datasets results with CCS

Dataset	HPF number of solutions	HPF different measures combination	Min num. of clusters	Max num. of clusters	Literature num. of cluster
Breast	3	3	2	4	2
Aggregation	7	6	3	7	7
Tetra	2	2	4	4	4
Spheres3D	5	4	2	4	4
Iris	2	2	2	3	3
Thyroid	19	5	2	5	2
Yeast	30	6	2	10	10

The difference between the element's position in the clusters could be interesting for the decision-maker in a real context. Finally, solution 3 is composed of 2 clusters, which can also be interesting for decision-making depending on the preference.

Sphere6 Dataset Results

The Spheres6 dataset is a benchmark dataset defined by two variables, available in [49], which comprises 300 elements, divided into 6 clusters [49]. Therefore, the k_{min} and k_{max} parameters are respectively 2 and 17, and $\zeta = 17$.

The Sphere6 HPF is composed of a 2 overlap solution, each of them provided from a different pair of measures that remained after the normalization process. The first solution was provided by the measures $SAxc - Scc$, which indicates 4 clusters and illustrated in Fig. 5a, and the other solution from $FNc - Scc$, which indicates 6 clusters as described in Table 4 and illustrated in Fig. 5b , equal to the literature documentation.

Also for this dataset, the solution proposed by the literature is presented in the final set of solutions as seen in Table 4. When comparing the two solutions using the CVI, the solution of 6 clusters stands out, being considered the best by four CVI (DB, CS, CH, and XieBeni). Even though the literature solution indicates 6 clusters, if this were a real dataset and the decision-maker had prior knowledge to define a better division, the solution of 4 cluster could be the most appropriate.

Jain Dataset Results

According to the literature documentation, the Jain dataset comprises 373 elements divided into 2 clusters. Therefore, considering the previously mentioned MCA-II parameters, the possible number of clusters can vary between $k_{min} = 2$ and $k_{max} = 19$. And, $\zeta = 19$ is the minimum number of elements per cluster, as the algorithm default.

The Jain dataset has a complex geometry for partitioning clustering since it does not exhibit a rounded geometry [59, 60]. The preliminary results of the MCA-II without the CCS resemble some found by other algorithms in the literature and are inappropriate for this dataset [61, 62]. Therefore, using CCS is crucial for the MCA-II to find the highest-quality solutions.

The Jain MCA-II results are presented in Table 5, and some of them are illustrated in Fig. 6. The HPF is composed of 4 different solutions provided by 4 pairs of measures, one solution each: $SAxc - Acc$, $SAxc - FNcc$, $FNc - AFNcc$, and $FNc - ANNcc$.

As can be observed, the solutions offer a good distribution among the elements. Note that some solutions divide the dataset into 2 clusters, while others suggest to divide into 3 or 4 clusters. The solution mentioned in the literature, which is the same as described in Fig. 6b (solution 1) is found by the pairs of measures $FNC - ANNcc$. However, when analyzing the results in Table 5, it is found that other solutions have better CVI results than those presented in the literature.

In the specific case of the Jain dataset, due to its geometry, even with solutions enhanced by CCS, the CVI does not tend to select a single solution. For the DB and CH indices, the solution depicted in Fig. 6c is the most appropriate, while for the CS and XieBeni, the solution from Fig. 6d is the most recommended. And, for the Dunn index, the solution in Fig. 6b, identical to the literature, is the most suitable.

In this case, the final answer from the complete version of MCA-II, would be solutions 2 and 3 from Fig. 6, excluding the literature solution, referred to as solution 1. Numerically, this is correct, but knowing that the literature solution is solution 1, it is up to the decision-maker to make the final choice. With MCA-II, this is not necessary, as one simply needs to choose among the HPF solutions that best fit the problem at hand.

MCA-II Results on Other Datasets and General Discussion

Table 6 presents the MCA-II results in seven other datasets. The greater the complexity of the dataset, the higher the variability of solutions in the HPF. For less complex

and rounded geometries datasets like Mydata, Sphere6, Sphere3D, and Tetra, the number of solutions in the hybrid Pareto front tends to be lower than in more complex datasets such as Thyroid and Yeast.

The HPF of every dataset considered contains different pairs of measures. Thus, this result reinforces the conclusion that there is no single pair of measures capable of finding the optimal solution for any dataset. And, even for a given dataset, there is always more than one pair of measures capable of generating an optimal solution. Therefore, using a set of measures to provide variability among solutions is valuable.

Table 7 DBSCAN number of clusters results

Dataset	$\epsilon = 1$ $mp = 5$	$\epsilon = 1$ $mp = 10$	$\epsilon = 0.5$ $mp = 5$	$\epsilon = 0.5$ $mp = 10$	$\epsilon = 5$ $mp = 10$	$\epsilon = 5$ $mp = 5$
My data	1 (5)	2 (10)	2 (27)	3 (58)	1	1
Sphere6	6 (5)	6 (12)	6 (12)	10 (32)	5	4
Jain	9 (99)	7 (267)	17 (216)	-1	4 (54)	4 (12)

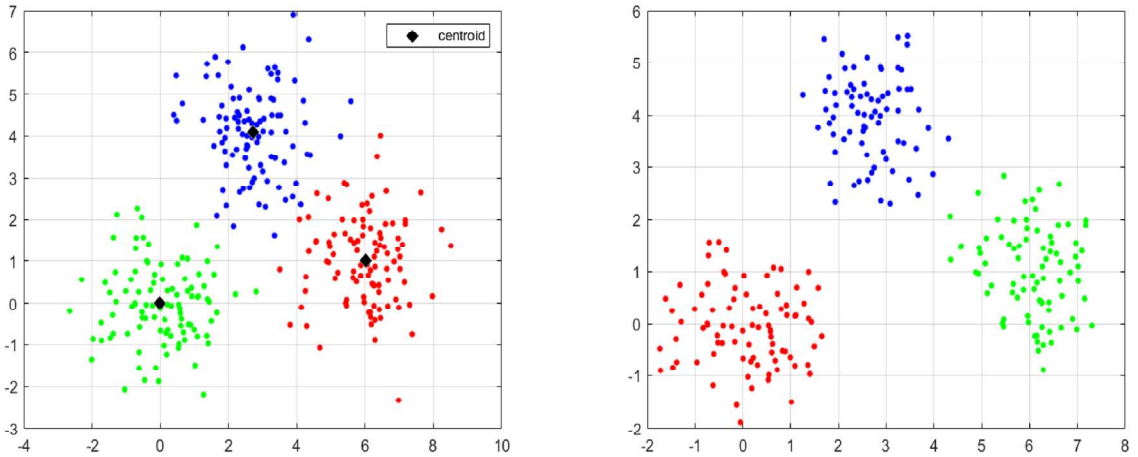

Fig. 7 Mydata results for k-means and DBSCAN algorithms

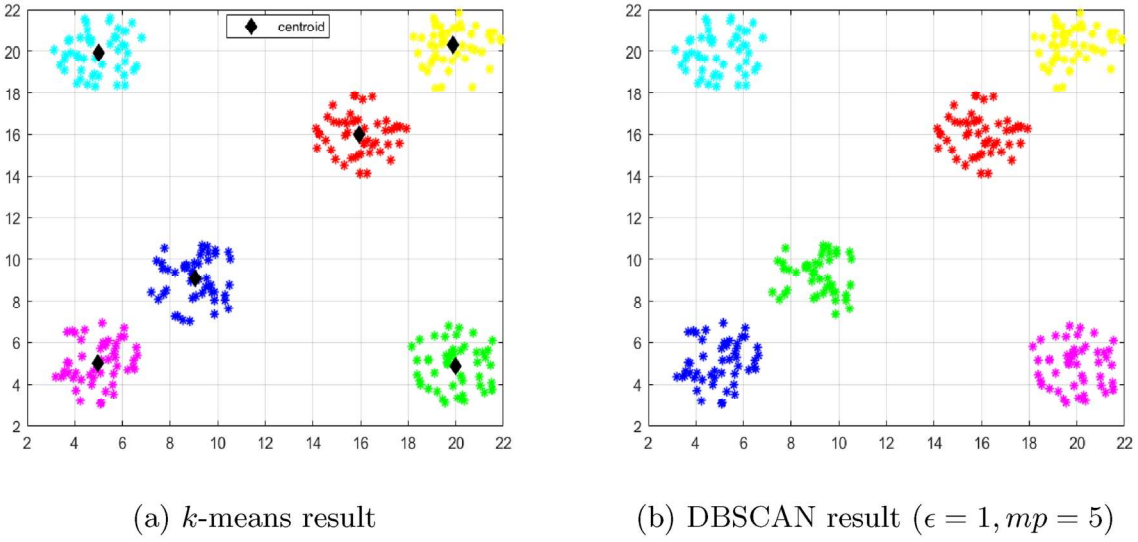

Fig. 8 Sphere6 results for k-means and DBSCAN algorithms

In general, the analysis of MCA-II results on benchmark datasets reveals that the algorithm not only manages to find the optimal solution as indicated by the literature but also proposes other solutions that belong to the Pareto front set. These alternatives can be even more interesting for the decision-maker, depending on the context in which they are working.

Therefore, through the results presented, it can be stated that in the MCA-II (with the CCS), it is possible to verify the literature solution in the HPF. Furthermore, the algorithm results go further by offering other optimal solutions to the decision-maker. In this way, MCA-II contributes to the decision-support process, allowing the decision-maker to rethink the choice of the most appropriate solution for the problem in question.

The CCS technique has proven effective in improving the quality of generated solutions. This approach allows for a more robust splitting and merging of clusters. The CCS in MCA-II provides a more precise way of identifying data clusters, ensuring a more faithful representation of the solution space. By dividing clusters into smaller subgroups and then merging them, the algorithm can find optimal solutions more efficiently and accurately.

Furthermore, MCA-II demonstrates strong scalability by efficiently handling increasing data dimensionality and size, specially in Breast and Yeast datasets, maintaining performance and solution diversity where traditional clustering algorithms like *k*-means and DBSCAN often degrade or fail.

Finally, from the range and variability of the HPF generated, it is possible to perceive the impact of combining different measures to solve a problem. In this way, if only one pair of measures was considered in the model, the solution would be restricted to the optimum provided by one combination of measures and could be inappropriate for the decision-maker. So, the HPF strategy enriches the model's final solution. Furthermore, the MCA-II does not require the prior indication of the cluster number, which is a common complaint in the literature regarding *k*-means and DBSCAN, and other clustering algorithms [63].

MCA-II Comparison with k-means and DBSCAN Algorithms

There are few works on multi-objective clustering in the literature, making it very difficult to find open codes that allow the comparison of multi-objective approaches. Thus, to compare the MCA-II results with some classical clustering algorithms of the literature, three datasets were tested with the *k*-means [64] and the DBSCAN algorithm [65]. More comparisons can be consulted in [15, 16].

The *k*-means, is one of the most well-known and simple clustering algorithms. It consists of separating samples into groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares [64]. The *k*-means is not an automatic clustering algorithm; thus, it depends on the initial estimation of the parameter *k*, which represents the number of clusters in which the dataset is divided. The four datasets use the *k*-means, considering the *k* as the literature indicates, and 100 algorithm executions.

In turn, the Density-based Spatial Clustering of Applications with Noise (DBSCAN) is a popular clustering algorithm for clustering spatial data points based on their density distribution. DBSCAN is particularly effective in identifying clusters of arbitrary shapes and handling noise in the data. DBSCAN operates by grouping points that are closely packed, forming high-density regions. It defines clusters as continuous regions of high density separated by regions of

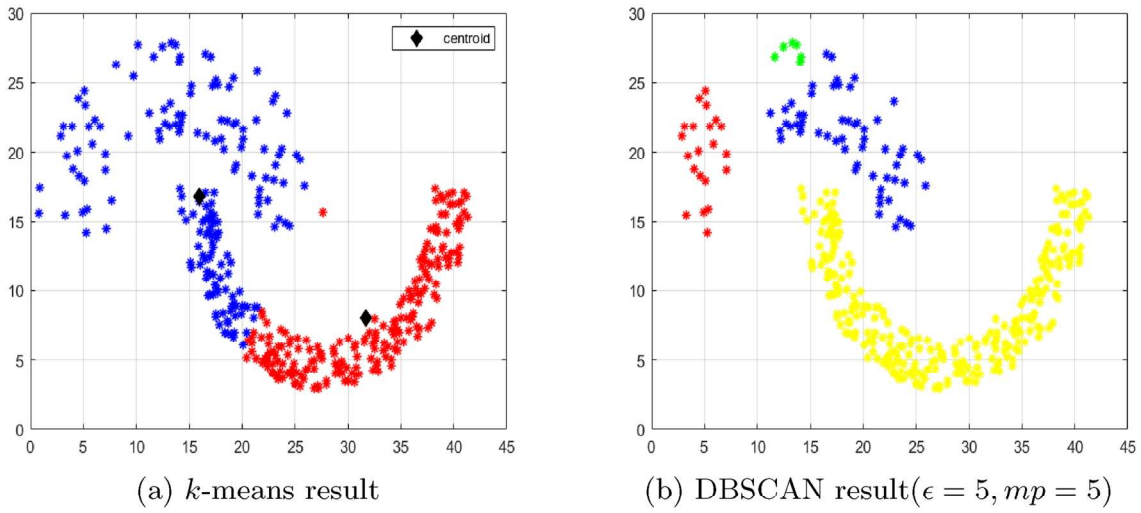

Fig. 9 Jain results for *k*-means and DBSCAN algorithms

low density [65]. This algorithm is based on a threshold for a neighborhood search radius ϵ and a minimum number of neighbors mp required to identify a core point and define the clustering division [65].

Table 7 summarizes the results achieved with the DBSCAN algorithm considering different values of ϵ and mp. The value −1 indicates that the DBSCAN considered all elements as outliers, and the algorithm is not able to define the clusters with that parameterization. The number inside the parentheses indicates the number of elements considered outliers for each parametrization considered.

Figure 7a presents the k-means results for the Mydata. Although this solution resembles those found in the literature, it is not identical. Therefore, if it does not meet the decision-maker's requirements, they will need to rerun the k-means algorithm with the same number of clusters or select another parametrization. On the other hand, the DBSCAN solution provided by $\epsilon = 0.5$ and $mp = 10$ is presented in Fig. 7b, but 58 were considered outliers and were removed from the dataset. This last solution was chosen since it is the closest solution to the one suggested by the literature.

Figure 8a shows the k-means results for the Sphere6 dataset, while Fig. 8b illustrates the DBSCAN results considering $\epsilon = 1$ and $mp = 5$. Although both results are similar, the DBSCAN considered 5 elements as outliers. Analyzing the other results of Table 7, DBSCAN can achieve other results composed of 4 or 5 clusters without eliminating any dataset element. However, all solutions composed of 6 clusters consider a set of elements to be outliers. This example illustrates the challenges of working with algorithms that heavily rely on parameterizations, as even minor changes can drastically alter the results. This complexity significantly increase decision-making work.

Finally, Fig. 9a, b present the k-means and DBSCAN results for the Jain dataset, respectively. As previously said, the Jain dataset has a more complex geometry than the previous ones. Although the k-means could identify the 2 clusters, the solution is unsuitable. With 6 different parameters set, the DBSCAN could not find a suitable solution. However, this algorithm can find the literature solution by knowing the correct parameters or using a hybrid technique to define the optimum parametrization as an optimization algorithm strategy.

The k-means algorithm is simple to implement and works well for round, low-dimensional datasets. This algorithm is heuristic and uses only one metric to define its optimal solution. Thus, the solution can change from run to run. It is important to note that k-means requires the decision-maker to specify the number of clusters, which is often difficult to determine, especially in real and complex datasets. If the execution result is not satisfactory for the decision-maker, they can rerun the algorithm with the same parameter of k and find a different solution, or another option is to change the k value until a satisfactory solution is found. Meanwhile, the MCA-II provides all these options at once.

In relation to DBSCAN which is known as an algorithm that works well in complex geometries, it tends to have difficulty in high-dimensional sets, and in addition, DBSCAN is very dependent on the ϵ and the mp parameters, requiring running the algorithm with different parameters until finding the most suitable solution or using another algorithm to define the values of the required parameters. In the sets presented in this work, DBSCAN failed several times to identify clusters and ended up categorizing the data as outliers. This reinforces the advantage of MCA-II in providing more robust and interpretable clustering results, particularly in high-dimensional contexts where traditional algorithms tend to struggle.

Conclusions

In recent decades, problem-solving domain has witnessed the emergence and proliferation of several machine learning and evolutionary algorithms. These algorithms have proven effective in tackling a wide range of challenges spanning different domains. However, as problems grow in complexity and intricacy, the requirement to merge methodologies and techniques through hybrid methods arises. Such an integration establishes a robust and powerful framework capable of delivering reliable and efficient solutions faster.

Considering this, the Multi-objective Clustering Algorithm II provides significant contributions, leveraging a multi-objective strategy and combining several measures to determine the optimal number of cluster sets and their element partitioning. When a single-objective approach is used, only one criterion is considered, making the solution dependent exclusively on that criterion and may not find the global optimum of the problem. In contrast, a multi-objective approach offers significant advantages by solving complex problems and identifying trade-offs between objectives.

The MCA-II offers a notable advantage through its application of multi-objective optimization, presenting a set of optimal solutions. This variety empowers decision-makers to select the most suitable solution based on their knowledge or preferences. The Hybrid Pareto front enhances diversity and robustness by considering different measures. Thus, the MCA-II stands out for its application of multi-objective optimization, providing a versatile set of optimal solutions. This flexibility proves valuable, especially in addressing challenges where encapsulating certain information within a mathematical model is difficult. This approach grants decision-makers the flexibility of choice and facilitates the integration of crucial insights through a human-in-the-loop collaborative system.

In general, analysis of MCA-II results on benchmark datasets reveals that the algorithm not only finds the optimal solution as indicated by the literature but also proposes other solutions that belong to the Pareto front set. Depending on the context in which it is working, these alternatives can be even more interesting for the decision-maker.

In benchmark datasets, solutions are generally validated, and it is relatively easy to identify the optimal solution. Therefore, based on the results presented, it can be stated that in the MCA-II with the CCS, it is possible to verify the literature solution in the HPF. Furthermore, the algorithm results go further by offering other optimal solutions to the decision-maker. In this way, MCA-II contributes to the decision support process, allowing the decision-maker to rethink the choice of the most appropriate solution for the problem in question. Moreover, from the range and variability of the HPF generated, it is possible to perceive the impact of combining different measures to solve a problem. In this way, if only one pair of measures was considered in the model, the solution would be restricted to the optimum provided by one combination of measures and could be inappropriate for the decision-maker.

The CCS in MCA-II provides a more precise way of identifying data clusters, ensuring a more faithful representation of the solution space, specially in complex and high-dimentional problems. By dividing clusters into smaller subgroups and then merging them, the algorithm can find optimal solutions more efficiently and accurately. Thus, implementing the CCS in MCA-II represents a significant advancement, providing higher-quality and more reliable solutions for a wide range of multi-objective decision-making problems.

The future of hybrid methods involving bio-inspired algorithms and machine learning lies in exploring new combinations, improving scalability and efficiency, enhancing explainability, and developing novel algorithms that can tackle complex and large-scale problems across various domains [66, 67].

In future approach in essential dedicate to hybrid methods comparison, since most publications have focused on comparing bio-inspired or hybrid methods with traditional ones through mathematical analysis of runtime, convergence guarantee, and parameter configurations. Few of these studies systematically compared the performance of different bio-inspired algorithms in machine learning tasks or different machine learning techniques in bio-inspired optimization algorithms. This leads to a lack of experimental results in selecting the most suitable method for a particular combination. The unavailability of such surveys may be due to the lack of publicly available source code, variation of encoding techniques, different objective functions, and evolutionary operators. As a result, there is a vast amount of published work since numerous metaheuristic algorithms can be combined with machine learning. Still, there is immense difficulty in pointing out which combinations are most appropriate or even why one is more advantageous than the other.

Furthermore, exploring the combination of machine learning in multi/many-objective could be a valuable and fruitful source of innovation, since real-world problems often involve multiple objectives [68]. Moreover, in the same scope, techniques for choosing the Pareto front solution can be explored through inspiration in machine learning techniques [69].

In summary, the future expectation of this research lies in their comparison with other methods and algorithm, and also the integration of hybrid algorithms with deep learning, increased explainability, meta-learning, cross-domain applications, real-time adaptation, and integration of human expertise. These trends aim to improve hybrid algorithms' performance, efficiency, and versatility in tackling complex and high-dimensional real-world problems.

Author contributions Beatriz Flamia Azevedo: Conceptualization, Methodology, Data curation, Writing-Original draft preparation, Visualization and Investigation. Ana M. A. C. Rocha: Conceptualization, Validation, Supervision, Writing-Reviewing and Editing. Ana I. Pereira: Conceptualization, Methodology, Visualization, Investigation, Supervision, Validation, Writing-Reviewing and Editing.

Funding Open access funding provided by FCT|FCCN (b-on). This work has been supported by FCT Fundação para a Ciência e Tecnologia within the R&D Units Project Scope UIDB/00319, UIDB/05757 (DOI: 10.54499/UIDB/05757, UIDP/05757) (DOI: 10.54499/UIDP/05757) and Erasmus Plus KA2 within the project 2021-1-PT01-KA220-HED-000023288.

Data availability The datasets are available at [48–57].

Code availability Available upon request from the authors.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Consent to participate Not applicable.

Consent for publication Not applicable.

Ethical approval Not applicable.

References

1. Albarakati N, Obradovic Z. Multi-domain and multi-view networks model for clustering hospital admissions from the emergency department. Int J Data Sci Anal. 2019;8:385–403. https://doi.org/10.1007/s41060-018-0147-5.
2. Azevedo BF, Amoura Y, Rocha AMAC, Fernandes FP, Pacheco MF, Pereira AI. Analyzing the mathe platform through clustering algorithms. In: Gervasi O, Murgante B, Misra S, Rocha AMAC, Garau C, editors. Computational science and its applications—ICCSA 2022 workshops. Cham: Springer; 2022. p. 201–18. https://doi.org/10.1007/978-3-031-10562-3-15.
3. Bi X, Hu X, Wu H, Wang Y. Multimodal data analysis of Alzheimer's disease based on clustering evolutionary random forest. IEEE J Biomed Health Inform. 2020;24:2973–83. https://doi.org/10.1109/JBHI.2020.2973324.
4. Chen J, Qi X, Chen L, Chen F, Cheng G. Quantum-inspired ant lion optimized hybrid k-means for cluster analysis and intrusion detection. Knowl-Based Syst. 2020;203:106167. https://doi.org/10.1016/j.knosys.2020.106167.
5. Zhou Y, Wu H, Luo Q, Abdel-Baset M. Automatic data clustering using nature-inspired symbiotic organism search algorithm. Knowl-Based Syst. 2019;163:546–57. https://doi.org/10.1016/j.knosys.2018.09.013.
6. Yang D, Fan Q. Gradient-based hybrid method for multi-objective optimization problems. Expert Syst Appl. 2025;272:126675. https://doi.org/10.1016/j.eswa.2025.126675.
7. Iglesias F, Zseby T, Zimek A. Clustering refinement. Int J Data Sci Anal. 2021;12:333–53.
8. Shalev-Shwartz S, Ben-David S. Understanding machine learning: from theory to algorithms. Cambridge: Cambridge University Press; 2014.
9. Dutta D, Sil J, Dutta P. Automatic clustering by multi-objective genetic algorithm with numeric and categorical features. Expert Syst Appl. 2019;137:357–79. https://doi.org/10.1016/j.eswa.2019.06.056.
10. Liu C, Liu J, Peng D, Wu C. A general multiobjective clustering approach based on multiple distance measures. IEEE Access. 2018;6:41706–19. https://doi.org/10.1109/ACCESS.2018.2860791.
11. Dutta D, Dutta P, Sil J. Clustering by multi objective genetic algorithm. In: 2012 1st International Conference on Recent Advances in Information Technology (RAIT). 2012. p. 548–53. https://doi.org/10.1109/RAIT.2012.6194619.
12. Dutta D, Dutta P, Sil J. Clustering data set with categorical feature using multi objective genetic algorithm. In: 2012 International Conference on Data Science & Engineering (ICDSE). 2012. p. 103–8. https://doi.org/10.1109/ICDSE.2012.6281897.
13. Kaur A, Kumar Y. A multi-objective vibrating particle system algorithm for data clustering. Pattern Anal Appl. 2022;25(1):209–39. https://doi.org/10.1007/s10044-021-01052-1.
14. Binu Jose A, Das P. A multi-objective approach for inter-cluster and intra-cluster distance analysis for numeric data. In: Kumar R, Ahn CW, Sharma TK, Verma OP, Agarwal A, editors. Soft computing: theories and applications. Singapore: Springer; 2022. p. 319–32. https://doi.org/10.1007/978-981-19-0707-4_30.
15. Azevedo BF, Rocha AMAC, Pereira AI. A multi-objective clustering approach based on different clustering measures combinations. Comput Appl Math. 2024. https://doi.org/10.1007/s40314-024-03004-x.
16. Azevedo BF, Rocha AMAC, Fernandes FP, Pacheco MF, Pereira AI. Comparison between single and multi-objective clustering algorithms: mathe case study. In: Pereira AI, Fernandes FP, Coelho JP, Teixeira JP, Lima J, Pacheco MF, Lopes RP, Álvarez ST, editors. Optimization, learning algorithms and applications. Tenerife - Spain: Springer; 2024. p. 65–80. https://doi.org/10.1007/978-3-031-77426-3_5.
17. Wang M, Huang VA, Bosneag A-MC. A novel split-merge-evolve k clustering algorithm. In: 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService). 2018. p. 229–36. https://doi.org/10.1109/BigDataService.2018.00041.
18. Rehman AU, Belhaouari SB. Divide well to merge better: a novel clustering algorithm. Pattern Recognit. 2022;122:108305. https://doi.org/10.1016/j.patcog.2021.108305.
19. Boeva V, Angelova M, Devagiri VM, Tsiporkova E. Bipartite split-merge evolutionary clustering. In: Agents and artificial intelligence: 11th International Conference, ICAART 2019, Prague, Czech Republic, February 19–21, 2019, Revised Selected Papers. Berlin: Springer; 2019. p. 204– 23. https://doi.org/10.1007/978-3-030-37494-5_11.
20. Ailon N, Avigdor-Elgrabli N, Liberty E, Zuylen A. Improved approximation algorithms for bipartite correlation clustering. SIAM J Comput. 2012;41(5):1110–21. https://doi.org/10.1137/110848712.
21. Lughofer ED. A dynamic split-and-merge approach for evolving cluster models. Evol Syst. 2012;3:135–51. https://doi.org/10.1007/s12530-012-9046-5.
22. Devagiri VM, Boeva V, Tsiporkova E. Split-merge evolutionary clustering for multi-view streaming data. Procedia Comput Sci. 2020;176:460–9. https://doi.org/10.1016/j.procs.2020.08.048. (**Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 24th International Conference KES2020**).
23. Chen J, Lu J. A clustering algorithm based on minimum spanning tree and density. In: 2019 IEEE 4th International Conference on Big Data Analytics (ICBDA). 2019. p. 1– 4. https://doi.org/10.1109/ICBDA.2019.8713247.
24. Hajto K, Kamieniecki K, Misztal K, Spurek P. Split-and-merge tweak in cross entropy clustering. In: Saeed K, Homenda W, Chaki R, editors. Computer information systems and industrial management. Lecture notes in computer science, vol. 10244. Cham: Springer; 2017. https://doi.org/10.1007/978-3-319-59105-6_17.
25. Tabor J, Spurek P. Cross-entropy clustering. Pattern Recognit. 2014;47(9):3046–59. https://doi.org/10.1016/j.patcog.2014.03.006.
26. Ding C, He X. Cluster merging and splitting in hierarchical clustering algorithms. In: 2002 IEEE International Conference on Data Mining, 2002. Proceedings. 2002. p. 139– 46. https://doi.org/10.1109/ICDM.2002.1183896.
27. Muhr M, Granitzer M. Automatic cluster number selection using a split and merge k-means approach. In: 2009 20th international workshop on database and expert systems application. 2009. p. 363–7. https://doi.org/10.1109/DEXA.2009.39.
28. Cheng M, Ma T, Liu Y. A projection-based split-and-merge clustering algorithm. Expert Syst Appl. 2019;116:121–30. https://doi.org/10.1016/j.eswa.2018.09.018.
29. Ueda N, Nakano R, Ghahramani Z, Hinton GE. Smem algorithm for mixture models. Neural Comput. 2000;12(9):2109–28. https://doi.org/10.1162/089976600300015088.
30. Sokal RR, Michener CD. A statistical method for evaluating systematic relationships. Univ Kans Sci Bull. 1958;38:1409–38.

31. Sorensen TA. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. Biol Skar. 1948;5:1–34.
32. Pal NR, Bezdek JC. On cluster validity for the fuzzy c-means model. IEEE Trans Fuzzy Syst. 1995;3(3):370–9. https://doi.org/10.1109/91.413225.
33. Memarsadeghi N, Mount D, Netanyahu N, Moigne J. A fast implementation of the isodata clustering algorithm. Int J Comput Geom Appl. 2007;17:71–103. https://doi.org/10.1142/S0218195907002252.
34. Coello-Coello CA, Lechuga MS. Mopso: a proposal for multiple objective particle swarm optimization. In Proceedings of the 2002 Congress on Evolutionary Computation CEC'02 (Cat No02TH8600), vol 2. 2002. p. 1051–10562. https://doi.org/10.1109/CEC.2002.1004388.
35. Müller AC, Guido S. Introduction to machine learning with Python: a guide for data scientists. USA: O'Reilly Media, Inc.; 2016.
36. Kok J, González FC, Kelson N, Périaux J. An fpga-based approach to multi-objective evolutionary algorithm for multi-disciplinary design optimisation. 2011.
37. Mirjalili S, Saremi S, Mirjalili SM, Coelho, L S. Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. Expert Syst Appl. 2016;47:106–19. https://doi.org/10.1016/j.eswa.2015.10.039.
38. Ahmed M, Mahmood AN. A novel approach for outlier detection and clustering improvement. In: 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA). 2013. p. 577–82. https://doi.org/10.1109/ICIEA.2013.6566435.
39. Alqurashi T, Wang W. Clustering ensemble method. Int J Mach Learn Cybern. 2019;10:1227–46. https://doi.org/10.1007/s13042-017-0756-7.
40. Gurrutxaga I, Muguerza J, Arbelaitz O, Pérez JM, Martín JI. Towards a standard methodology to evaluate internal cluster validity indices. Pattern Recognit Lett. 2011;32(3):505–15. https://doi.org/10.1016/j.patrec.2010.11.006.
41. Jain M, Jain M, AlSkaif T, Dev S. Which internal validation indices to use while clustering electric load demand profiles? Sustain Energy Grids Netw. 2022;32:100849. https://doi.org/10.1016/j.segan.2022.100849.
42. Jain M, AlSkaif T, Dev S. Validating clustering frameworks for electric load demand profiles. IEEE Trans Ind Inform. 2021;17(12):8057–65. https://doi.org/10.1109/TII.2021.3061470.
43. Arbelaitz O, Gurrutxaga I, Muguerza J, Pérez JM, Perona I. An extensive comparative study of cluster validity indices. Pattern Recognit. 2013;46(1):243–56. https://doi.org/10.1016/j.patcog.2012.07.021.
44. Dunn JC. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. J Cybern. 1973;3(3):32–57. https://doi.org/10.1080/01969727308546046.
45. Caliński T, Harabasz J. A dendrite method for cluster analysis. Commun Stat. 1974;3(1):1–27. https://doi.org/10.1080/03610927408827101.
46. Chou C-H, Su M-C, Lai E. A new cluster validity measure and its application to image compression. Pattern Anal Appl. 2004;7:205–20. https://doi.org/10.1007/s10044-004-0218-1.
47. Xie XL, Beni G. A validity measure for fuzzy clustering. IEEE Trans Pattern Anal Mach Intell. 1991;13(8):841–7. https://doi.org/10.1109/34.85677.
48. Heris MK. Evolutionary Data Clustering in MATLAB. 2015. https://yarpiz.com/64/ypml101-evolutionary-clustering.
49. Bandyopadhyay S, Maulik U. Nonparametric genetic clustering: comparison of validity indices. IEEE Trans Syst Man Cybern Part C (Appl Rev). 2001;31(1):120–5. https://doi.org/10.1109/5326.923275.
50. Jain A, Law M. Data clustering: a user's dilemma. Lect Notes Comput Sci. 2005;3776:1–10. https://doi.org/10.1007/11590316_1.
51. Zwitter M, Soklic M. Breast cancer. UCI Mach Learn Repository. 1988. https://doi.org/10.24432/C51P4M.
52. Gionis A, Mannila H, Tsaparas P. Clustering aggregation. ACM Trans Knowl Disco Data (TKDD). 2007;1(1):1–30.
53. Ultsch A. Kohonen's self organizing feature maps for exploratory data analysis. INNC'90. 1990.
54. Ultsch A. Self-organizing neural networks for visualisation and classification. In: Information and classification: concepts, methods and applications proceedings of the 16th annual conference of the "Gesellschaft Für Klassifikation eV" University of Dortmund, April 1–3, 1992. Springer; 1993. p. 307–13. https://doi.org/10.1007/978-3-642-50974-2_31.
55. Fisher RA. Iris. UCI Mach Learn Repository. 1936. https://doi.org/10.24432/C56C76.
56. Quinlan R. Thyroid disease. UCI Mach Learn Repository. 1986. https://doi.org/10.24432/C5D010.
57. Nakai K. Yeast. UCI Mach Learn Repository. 1991. https://doi.org/10.24432/C5KG68.
58. MATLAB. The MathWorks Inc. 2019. https://www.mathworks.com/products/matlab.html
59. Han J, Kamber M, Pei J. Data mining: concepts and techniques. USA: Morgan Kaufmann Publishers; 2012.
60. Mehta V, Bawa S, Singh J. Analytical review of clustering techniques and proximity measures. Artif Intell Rev. 2020. https://doi.org/10.1007/s10462-020-09840-7.
61. Wang L, Hao Z, Sun W. A novel self-adaptive affinity propagation clustering algorithm based on density peak theory and weighted similarity. IEEE Access. 2019;7:175106–15. https://doi.org/10.1109/ACCESS.2019.2956963.
62. Xu L, Zhao J, Yao Z, et al. Density peak clustering based on cumulative nearest neighbors degree and micro cluster merging. J Signal Process Syst. 2019;91:1219–36. https://doi.org/10.1007/s11265-019-01459-4.
63. Azevedo BF, Rocha AMAC, Pereira AI. Hybrid approaches to optimization and machine learning methods: a systematic literature review. J Mach Learn. 2024. https://doi.org/10.1007/s10994-023-06467-x.
64. Arthur D, Vassilvitskii S. K-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. SODA '07. Society for Industrial and Applied Mathematics, USA; 2007. p. 1027–35. https://doi.org/10.1145/1283383.1283494.
65. Ester M, Kriegel H-P, Sander J, Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery in Databases and Data Mining, Portland, OR. AAAI Press; 1996. p. 226–31. https://doi.org/10.5555/3001460.3001507.
66. Pourpanah F, Wang R, Lim C, et al. A review of artificial fish swarm algorithms: recent advances and applications. Artif Intell Rev. 2023;56:1867–903. https://doi.org/10.1007/s10462-022-10214-4.
67. Telikani A, Tahmassebi A, Banzhaf W, Gandomi A. Evolutionary machine learning: a survey. ACM Comput Surv. 2021. https://doi.org/10.1145/3467477.
68. Kang Y, Xie W, Wang X, Wang H, Wang X, Li J. Mopisde: a collaborative multi-objective information-sharing de algorithm for software clustering. Expert Syst Appl. 2023;226:120207. https://doi.org/10.1016/j.eswa.2023.120207.
69. Wang X, Zhang F, Yao M. A many-objective evolutionary algorithm with estimating the convexity-concavity of pareto fronts and clustering. Inf Sci. 2023;644:119289. https://doi.org/10.1016/j.ins.2023.119289.