IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 8 October 2022, accepted 2 November 2022, date of publication 7 November 2022, date of current version 10 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3219854

RESEARCH ARTICLE

Decision Making in Evolutionary Multiobjective Clustering: A Machine Learning Challenge

MARIO GARZA-FABRE[1], AARÓN L. SÁNCHEZ-MARTÍNEZ[1], EDWIN ALDANA-BOBADILLA[1,2], AND RICARDO LANDA[1]

[1]Center for Research and Advanced Studies, Cinvestav Campus Tamaulipas, Ciudad Victoria, 87130 Tamaulipas, Mexico
[2]National Council of Science and Technology of Mexico (CONACyT), 03940 Mexico City, Mexico

Corresponding author: Mario Garza-Fabre (mario.garza@cinvestav.mx)

ABSTRACT Evolutionary multiobjective algorithms have become a popular choice to tackle the clustering problem. On the one hand, the simultaneous optimization of complementary clustering criteria offers an increased robustness to changes in data characteristics. On the other hand, the evolutionary search is able to approximate the Pareto optimal front and deliver a set of trade-offs between these criteria in a single algorithm execution. Decision making is the concluding stage of the pipeline, having as its goal the selection of a single, final solution from the set of candidate trade-offs produced. This is a complex task for which a definitive answer does not seem to be available, as the underlying assumptions of existing techniques may not hold for all applications. In this paper, we investigate an alternative approach to address this challenge: posing it as a learning problem. The key idea is to build a model that, given a proper characterization of solutions and their context (defined by the full approximation solution set and the specific clustering task at hand), is able to estimate quality and facilitate the identification of the best choice. To evaluate the suitability of this approach, we conduct a series of experiments over diverse synthetic and real-world datasets, including comparisons against a range of representative decision-making strategies from the literature. Our proposal exhibits greater flexibility in dealing with problems of varying characteristics, consistently outperforming the reference methods considered. This study demonstrates that it is possible to learn from the decision-making process in example settings and generalize the acquired knowledge to new scenarios.

INDEX TERMS Clustering algorithms, multiobjective clustering, decision making, evolutionary computation, machine learning, pareto optimization.

I. INTRODUCTION

Clustering is a fundamental, unsupervised data analysis and machine learning task. Its goal is to determine the intrinsic organization of a set of elements into groups, such that this partition reflects the similarities and differences between them. *Evolutionary multiobjective clustering* (EMC) involves formulating this task as a multiobjective problem and adopting evolutionary algorithms as the optimization engine [1], [2], [3]. By exploiting multiple clustering criteria simultaneously, EMC methods are able to assess partition quality more comprehensively, which translates into an increased effectiveness and the ability to handle problems with a wider range of features. However, there is unlikely to be a single best solution for the resulting multiobjective formulation; due to the complementary but conflicting nature of the optimization criteria chosen, EMC methods generally produce a set of trade-offs between these criteria as output (this is illustrated in Figure 1 and further explained in Section II-B) [4]. Given that all the obtained trade-offs are considered equally good, i.e., they are all nondominated in the *Pareto-optimality* sense [5], how can one of them be selected and delivered as the final solution? The *decision making* process is concerned with this particular question, being the last step of the EMC pipeline. Identifying a single, promising solution may represent the ultimate goal in practice; this highlights the relevance of decision making, which is the specific focus of this study.

The associate editor coordinating the review of this manuscript and approving it for publication was Amir Masoud Rahmani.

The complexity of decision making has hindered the development of a definitive approach to carry out this process. Some of the existing methods employ an additional clustering criterion to induce an ordering (break ties) over the set of nondominated solution alternatives [6], [7], [8], [9]; this assumes compatibility between the criterion chosen and the clustering task at hand, being inconsistent with the motivations behind the adoption of a multiobjective problem formulation. Other methods rely on geometric considerations, analyzing the relative location of solutions in objective space [10], [11], [12], [13]; although such an approach is widely used in multiobjective optimization [14], we show later in this paper that it can lead to poor decisions in the specific context of EMC. Finally, there are also methods that construct a consensus solution from the set of nondominated candidates available [15], [16], [17], [18]; this approach assumes that all candidates are equally important, but the inclusion of low-quality partitions (despite being nondominated) can negatively affect the outcome of this process. Representative examples of the above three categories of decision-making methods and a detailed discussion of their limitations are provided in Section II-C.

Acknowledging the importance of decision making and its complexity, and motivated by the limitations of existing techniques, we explore a novel approach to tackle this challenge. Specifically, we frame decision making as a supervised learning problem. Our approach relies on the construction of a predictive model that is able to associate characteristics of individual solutions and their context with a measure of partition quality (as explained in Section III, by context we refer to the full set of competing nondominated solutions as well as to the particular clustering problem they try to solve). In this way, the derived model can be used to estimate the quality of candidate partitions and guide the decision-making process. We investigate the suitability of this approach in terms of its ability to automate the selection of high-quality partitions from the nondominated solution sets produced by a state-of-the-art EMC algorithm. For this sake, our proposal is compared with respect to different baselines and a set of reference approaches that are representative of the three categories of existing methods discussed above. Our experimental analysis is conducted over a diverse collection of both synthetic and real-world data clustering problems.

The remaining of this paper is structured as follows. Section II introduces the necessary background and reviews the related literature. Then, Section III describes our proposed approach to decision making in detail. The experimental setup is described in Section IV. Section V discusses our results and main findings. Finally, Section VI concludes this study and highlights potential directions for future research.

II. BACKGROUND AND RELATED WORK

Below we introduce background concepts and review the relevant literature. First, Section II-A presents a formal definition of clustering and states it as an optimization problem. Then, the formulation of clustering as a multiobjective problem is discussed in Section II-B. Finally, Section II-C covers the central topic of this paper: decision making in EMC.

A. DATA CLUSTERING

Clustering is the task of finding the best way to partition a collection of samples into two or more disjoint subsets. Because of its unsupervised nature, this task is mostly based on the analysis of the similarities between the samples, and it heavily relies on a mechanism that supports the effective assessment of partition quality. Such a mechanism, known as *clustering criterion* or *cluster validity index* [19], allows us to frame the clustering task as an optimization problem and use a range of techniques to search for the best possible partition.

Let $X = \{x_1, \ldots, x_N\}$ be a set of N samples, $f : \Psi \rightarrow \mathbb{R}$ be a clustering criterion, and $\Psi = \{\{c_1, \ldots, c_k\} \mid c_i \in \mathcal{P}(X) \setminus \{\emptyset\} \text{ and } X = \bigcup c_i, \text{ for } i = 1, \ldots, k\}$ be the set of all possible partitions of X. Clustering can be stated, without loss of generality, as the following optimization problem:

$$\begin{aligned} &\text{Minimize } f(C) \ , \\ &\text{subject to } C \in \Omega \ , \end{aligned} \tag{1}$$

where $C = \{c_1, \ldots, c_k\}$ is a partition of X into k subsets, called *clusters*, and the constraint that C belongs to the feasible space $\Omega \subset \Psi$ implies that it must be a proper partition; that is, it must hold that $X = \bigcup c_i$, $c_i \neq \emptyset$, and $c_i \cap c_j = \emptyset$, for $i, j = 1, \ldots, k$ and $i \neq j$. Despite that many clustering methods require the specific value of k to be known in advance, frequently this information is not readily available in practice. The task of partitioning X without any prior knowledge of the correct value of this parameter is commonly referred to as *automatic clustering* in the literature [20].

B. MULTIOBJECTIVE CLUSTERING

From the above definition, the critical role of the clustering criterion f is evident, as we aim to find a partition that is optimal according to it. Thus, f needs to correctly evaluate the properties that determine a good partition, being responsible for guiding the search process towards high-quality solutions.

Many criteria have been proposed so far [21], [22], each presenting a specific formulation to evaluate (either one or a combination of) properties such as intra-cluster homogeneity (compactness), connectedness, and inter-cluster separation. The diversity of existing clustering criteria highlights the lack of consensus on how to assess partition quality, and the fact that it is unlikely that a single solution can simultaneously satisfy all the desirable but (usually) conflicting properties [23]. The effectiveness of a clustering algorithm strongly depends on whether the underlying assumptions of the specific criterion adopted hold for the particular characteristics of the problem, which is in line with the *No-Free-Lunch theorems* [24], [25]. Acknowledging these facts, however, allows us to realize that reasonable trade-offs may be possible through the simultaneous consideration of multiple criteria,

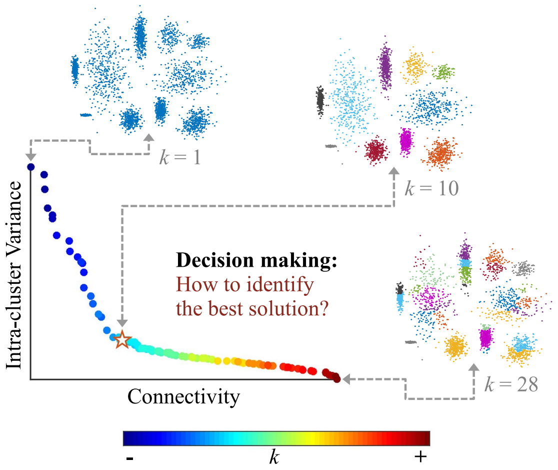


FIGURE 1. The clustering phase of EMC produces a Pareto front approximation (PFA). This PFA was obtained using algorithm Δ-*MOCK* [26], which simultaneously optimizes the connectivity and intra-cluster variance criteria (to be minimized). The PFA obtained includes trade-off partitions of varying quality and numbers of clusters (*k*). The decision-making phase is concerned with identifying a member from this set as the final solution.

an approach which may deal more effectively with a wider range of characteristics in the problem domain.

That is, rather than focusing on a single criterion, clustering can be stated as a multiobjective optimization problem:

$$\begin{aligned}&\text{Minimize}\, f(C)\ ,\\&\text{subject to}\ C \in \Omega\ ,\end{aligned} \qquad (2)$$

where $f(C) = [f_1(C), \ldots, f_m(C)]^T$ and $f_i : \Psi \rightarrow \mathbb{R}$ is the i-th criterion to be optimized. Due to the conflict that may exist between the m clustering criteria, we are now interested in identifying the set of the best possible trade-off solutions [5]: $P^* = \{C^* \in \Omega \mid \nexists C \in \Omega : C \prec C^*\}$.[1] P^* is referred to as the *Pareto-optimal set*, and the image of P^* in the objective function space is the so-called *Pareto front*.

The simultaneous optimization of multiple, complementary criteria commonly results in the identification of promising, high-quality partitions which may be unattainable through the optimization of a single criterion. Moreover, set P^* may include solutions with a range of potential k values if the criteria optimized present opposing biases regarding this parameter, which can be particularly useful in an automatic clustering setting. Figure 1 helps to illustrate the above behaviors in the context of two specific clustering criteria: *connectivity* and *intra-cluster variance* [26]. Whereas the optimization of the former tends to decrease k, the optimization of the latter tends to increase it; consequently, simultaneously optimizing these criteria produces a set of trade-off partitions that can vary greatly both in characteristics and k values.

Although the intrinsic multiobjective nature of clustering was acknowledge since the early work of Delattre and Hansen [27], it was not until the application of metaheuristics to this multiobjective task that the topic started to attract increasing attention [1], [2], [3], [4], [10], [28]. In particular, population-based metaheuristics such as *multiobjective evolutionary algorithms* offer a significant advantage: they are able to construct a Pareto front approximation (PFA) in a single execution. Such is the case of the recently reported algorithm Δ-*MOCK* [26], which produced the PFA of Figure 1. Δ-*MOCK* is a newer version of *MOCK* [10], one of the most representative EMC algorithms from the literature.

C. DECISION MAKING

EMC can be seen as a two-phase process. First, the *clustering phase* is concerned with the generation of a PFA of candidate partitions. Then, the *decision-making phase* focuses on the selection of one of the PFA members as the final solution. Whilst obtaining the full PFA can be useful in some cases, obtaining a single solution more likely corresponds to the ultimate goal in practice. The relevance of decision making is even more evident in automatic clustering, where the PFA may offer a diversity of choices with respect to the value of k, as shown in Figure 1; the selection of a final solution is the step that actually materializes a decision on this parameter.

Decision making in EMC has been accomplished using strategies that fall into three broad categories, as discussed in Section I. These categories and representative works for each of them are separately described below, followed by a discussion of the limitations and areas of opportunity motivating our new proposal introduced later in Section III.

1) DECISION MAKING BASED ON ADDITIONAL CLUSTERING CRITERIA

Perhaps the most widely adopted approach to select a final solution in EMC corresponds to the use of an additional criterion to discriminate between the (otherwise incomparable) nondominated partitions in the PFA. In algorithm *MOGAC* (*multiobjective genetic algorithm for clustering*) [28], for example, decision making relies on the use of index $\mathcal{I}$ [29]. In a closely related work [6], however, the authors replace index $\mathcal{I}$, adopting the *silhouette index* instead [30]. Other examples of the use of the silhouette index for decision-making purposes include algorithm *MOVGA* (*multiobjective variable string length genetic fuzzy clustering*) [31] and algorithm *MVMC* (*multi-view multiobjective clustering*) [7].

Algorithm *MOKGA* (*multiobjective k-means genetic algorithm*) considers both indices *Davies-Bouldin* [32] and *SD* [33] at the decision-making phase [34]. In a separate study [35], the authors use six different criteria for decision making: the silhouette, *C* [36], *Dunn* [37], Davies-Bouldin, SD, and *S_Dbw* [38] indices. In the work of Garcia-Piquer et al. [8], the silhouette, Dunn, Davies-Bouldin, and *Calinski-Harabasz* [39] indices are all explored as alternatives to guide decision making, within the context of algorithm *CAOS* (*clustering algorithm based on multiobjective strategies*) [40].

[1] Symbol $\prec$ refers to the *Pareto-dominance* relation. Solution C is said to *dominate* solution C' (which is denoted by $C \prec C'$) if and only if $\forall i : f_i(C) \leq f_i(C') \wedge \exists j : f_j(C) < f_j(C')$, $i, j = 1, \ldots, m$. All solutions in set P^* are said to be *nondominated* with respect to each other.

In particular, it is shown that the effectiveness of these indices in identifying a final solution can be improved by filtering solutions at the extreme regions of the PFA (hence, this proposal relates also to the strategies discussed in Section II-C2, considering geometric aspects of the PFA).

Besides the use of additional clustering criteria in an individual manner, decision making has also been assisted by index combinations. In a recent study [9], Zhu et al. report that a linear combination of the Calinski-Harabasz, Davies-Bouldin, and silhouette indices (denoted CH+DB+SIL) in most cases outperforms the use of several individual approaches. Their analysis initially centers on the application of decision making to PFAs generated by algorithm Δ-*MOCK* [26], but combination CH+DB+SIL is later explored in the context of the consensus-based decision-making strategy of algorithm *MOAC* (*multi-objective automatic clustering*) [9], which is further discussed in Section II-C3.

2) DECISION MAKING BASED ON THE SHAPE OF THE PARETO FRONT

In the general area of multiobjective optimization, selecting a final solution based on geometric considerations is a popular strategy, which is inherited to the specific domain of EMC. The prominent regions of the Pareto front, commonly referred to as *knees* [14], [41], are considered particularly relevant. These regions tend to offer the most interesting trade-offs between the optimization criteria, where a minor improvement in one dimension causes a more significant deterioration in another. In the absence of any explicit preference information, knees are usually assumed to correspond to the likely choices of a decision maker. Thus, some of the strategies that have been proposed for decision making in EMC are based on identifying knee regions in the PFA and selecting promising trade-off partitions from such regions.

A representative example of the methods in this category is algorithm *MOCK* [10]. To locate a knee in the PFA, this method computes a number of control fronts by applying its clustering strategy (i.e., the same strategy used initially to obtain the PFA) to randomly generated data. Then, a potential knee is identified and selected as the final result based on the distance of candidate solutions with respect to such control fronts. Alternative schemes have also been proposed with the aim of lowering the computational complexity of *MOCK*'s strategy. One of them relies on a simple heuristic: the solution minimizing the sum of (normalized) objective values is always selected [12]. Another proposal reduces the cardinality of the PFA, removing solutions not complying with the assumptions of convexity; then, knee-like solutions are identified by analyzing the adjacent angles of PFA members with respect to their immediate neighbors [11].

Recently, two multiobjective fuzzy clustering methods were proposed [13]: *ECM-NSGA-II* (*entropy c-means-nondominated sorting genetic algorithm II*) and *ECM-MOEA/D* (*entropy c-means-multiobjective evolutionary algorithm based on decomposition*). These algorithms apply certain rules regarding the location of solutions with respect to a reference line connecting the extreme points of the PFA. Starting at the extreme point where cluster compactness (first objective) reaches its lowest value, the position of the next points along the PFA, relative to the reference line, determine the solution to be selected. If the points fall above the line, the decision is simply to keep the extreme point. Otherwise, when the points fall below the reference line, the one maximizing the distance with respect to this line is identified as the knee and delivered as the final partition.

3) DECISION MAKING BASED ON ENSEMBLE CLUSTERING

In *ensemble clustering* [42], the goal is to derive a new, consensus partition by integrating the information contained in a collection of base partitions. This concept has been adopted by several EMC methods, where the solutions in the PFA are taken as the ensemble members and a consensus is constructed on the basis of them. The motivation behind this approach is that every nondominated solution contains useful information on the cluster structure of the data, and the incorporation of ensemble techniques thus provides a means to exploit all this information in delivering a final answer.

Mukhopadhyay et al. [15] study approaches *MOGAC-CSPA*, *MOGAC-HGPA*, and *MOGAC-MCLA*, which combine algorithm *MOGAC* (discussed in Section II-C1) with the following ensemble techniques [43]: *CSPA* (*cluster-based similarity partitioning algorithm*), *HGPA* (*hypergraph partitioning algorithm*), and *MCLA* (*meta-clustering algorithm*). *MECEA* (*multiobjective evolutionary clustering ensemble algorithm*) is another example of the methods in this category [16]; it employs the aforementioned *MCLA* ensemble strategy at the decision-making phase, but the clustering phase relies on algorithm *MOCK* [10] rather than *MOGAC*.

Consensus-based approaches have also been combined with the use of a classifier [17], [18]. The resulting technique initially produces a consensus partition by means of a voting strategy, which assigns samples to clusters whenever the majority of the PFA members agree on the assignment. This is likely to result in a partial clustering, as some samples can be left unassigned due to the lack of agreement across voters. Hence, the complementary step exploits this partial consensus solution for the purposes of training a classifier, which is later employed to determine the cluster membership of the remaining (originally unassigned) samples. It is unclear, however, whether the above approach is applicable when the PFA involves solutions with different numbers of clusters.

Decision making in algorithm *MOAC* relies on both, the generation of consensus solutions and the use of additional criteria [9]. First, the cardinality of the PFA is reduced based on an indicator which takes into account the quality and diversity of solutions. Then, a new set of consensus partitions with a range of numbers of clusters is generated, from which a final solution is chosen by means of index combination CH+DB+SIL (refer to Section II-C1). To generate the consensus partitions, two recently proposed ensemble techniques

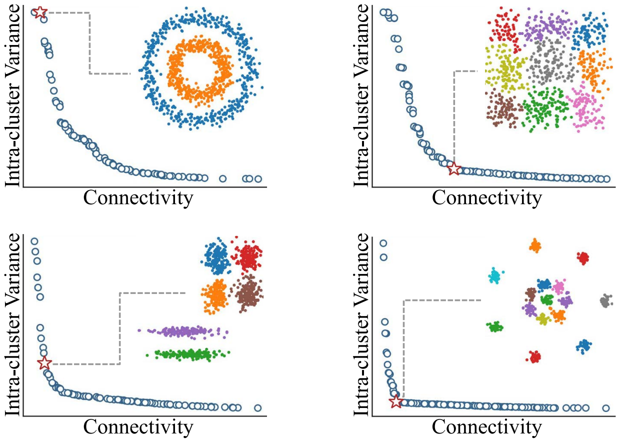


FIGURE 2. **PFAs produced by algorithm Δ-*MOCK* [26] for a set of four example problems. The clustering solution exhibiting the highest similarity with respect to the real partition of the data is illustrated, and the location of this solution in the corresponding PFA is indicated with a star marker.**

are adopted [44]: *LWEA* (*locally weighted evidence accumulation*) and *LWGP* (*locally weighted graph partitioning*).

4) LIMITATIONS OF CURRENT DECISION-MAKING APPROACHES

Despite their simplicity, methods that use an additional clustering criterion present an inherent limitation: they implicitly assume that the criterion chosen will be compatible with the properties of the data. This opposes the motivations for the use of a multiobjective formulation of clustering. It is unlikely that a single criterion can properly capture all aspects of a partition and induce an effective discrimination between candidate solutions in every scenario. Each criterion introduces a specific bias and its effectiveness will depend on the characteristics of the particular problem at hand (usually unknown in advance). To a certain extent, this may be offset by the use of more elaborate criteria, attempting to evaluate multiple aspects simultaneously, or by the use of criteria combinations (e.g., CH+DB+SIL [9]). However, finding the right weighting between various aspects or criteria may not be straightforward; it may certainly be problem-dependent.

Methods in the second category are supported by common assumptions in multiobjective optimization regarding the regions of the Pareto front which should be prioritized (in the absence of explicit decision-maker's preferences). Whilst favoring knee regions would probably be a wise choice in the general case [14], in the specific setting of EMC the best trade-off in the PFA may not always correspond to the correct answer. It has been argued, for example, that the cluster structure of the data is reflected in the shape of the Pareto front [10]. Evidently, this depends on the particular optimization criteria used but, more importantly, on how compatible these criteria are with the target data. If all criteria are compatible and contribute equally to solving the problem, then we would expect that the best trade-offs between these criteria would provide us with the most reasonable answers. Otherwise, if one of the criteria is much more helpful than the others, we would expect better choices to be located closer to the corresponding extreme region of the Pareto front. These behaviors are clearly illustrated in Figure 2.

Finally, consensus-based approaches do not necessarily select one of the solutions in the PFA; instead, they construct a new solution from them. These strategies operate under the premise that all partitions in the PFA contain useful information that can be integrated and exploited to construct a higher-quality final solution. This assumes that PFA members are all equally reliable, which implies that the optimized clustering criteria are all compatible with the characteristics of the data. This may not be the case, as discussed before. When one of the objectives contributes significantly more than others in solving the problem, and the best choices are therefore located at one of the extreme regions of the PFA (as seen in Figure 2), including the information of solutions from other regions of the PFA may be equivalent to introducing noise and can negatively affect the generated consensus.

III. DECISION MAKING BASED ON MACHINE LEARNING

The above discussion highlights the challenging nature of decision making, and the fact that current techniques operate under assumptions that do not necessarily apply given the peculiarities of EMC. This stresses the need for alternative, more effective and robust strategies to accomplish this task.

This section introduces a novel approach as our attempt to meet this need: *machine learning-based decision making* (MLDM). Our MLDM strategy treats decision making as a supervised learning problem. The goal is to learn by example, i.e., to learn from the decision-making process in example settings, and build a model which can capture the available knowledge for subsequent exploitation in unknown settings. MLDM consists of two main stages: the *learning stage* and the *decision-making stage*. These stages are separately described in Sections III-A and III-B. Then, Sections III-C and III-D respectively discuss some design choices adopted in this study and the characterization process of PFAs, which is an essential component of the proposed methodology.

A. LEARNING STAGE: MODEL CONSTRUCTION

The learning stage of strategy MLDM, depicted in Figure 3, is responsible for the construction of a regression model. This model is used later, at the decision-making stage (see Section III-B), to predict the quality of the solutions in a given PFA and enable the identification of the best alternative.

This stage starts with the formation of a repository of PFAs for the purposes of training the predictive model. These PFAs are produced through multiple independent executions of a chosen EMC algorithm (process A in Figure 3), over a collection of sample clustering problems. Each training PFA obtained consists of a set of nondominated partitions that the EMC method offers as potential solutions to the given problem. By sample clustering problems, we mean datasets for which the correct cluster structure (i.e., the ground truth) is known, so that learning can occur in a supervised manner.

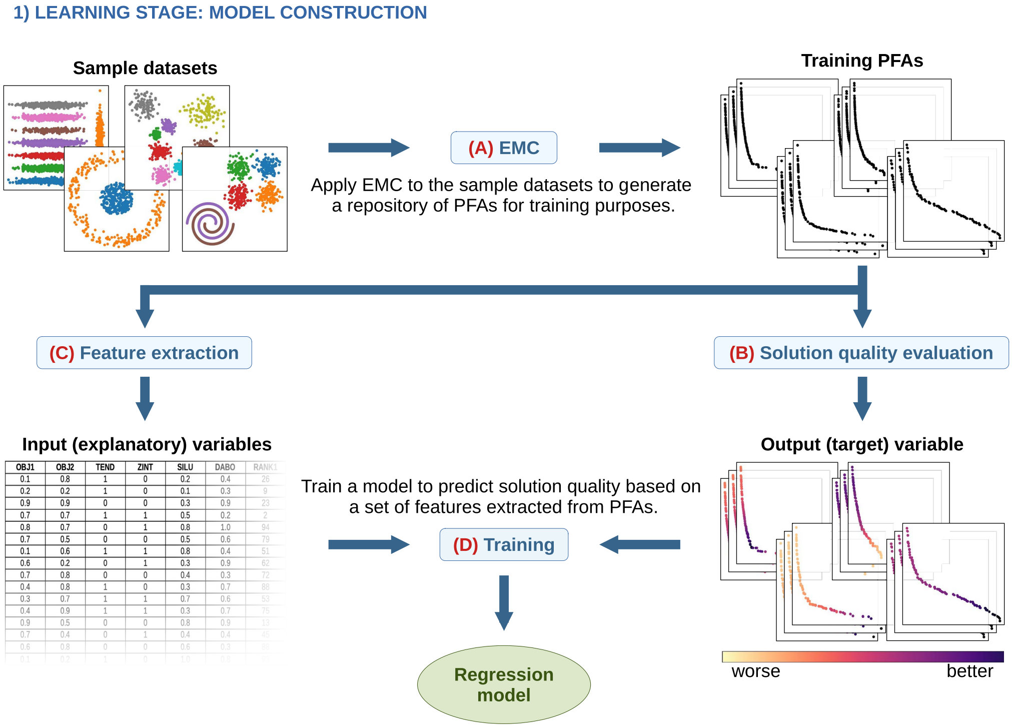

1) LEARNING STAGE: MODEL CONSTRUCTION
Sample datasets
(A) EMC
Apply EMC to the sample datasets to generate a repository of PFAs for training purposes.
Training PFAs
(C) Feature extraction
(B) Solution quality evaluation
Input (explanatory) variables
OBJ1 OBJ2 TEND ZINT SILU DABO RANK1
Train a model to predict solution quality based on a set of features extracted from PFAs.
(D) Training
Output (target) variable
worse
better
Regression model

FIGURE 3. **Learning stage of MLDM. Construction of a predictive (regression) model from a collection of training PFAs for sample problems. This process relies on the characterization of PFAs and their solutions (feature extraction), and on the use of a measure of partition quality (cluster validity index) as a target variable.**

Since the correct clustering solution is known for these (sample) problems, a direct comparison using this solution as a reference provides us with an objective measurement of the quality of every member of the training PFAs (process B in Figure 3). Such a comparison is made by means of an external cluster validity index (refer to Section III-C for details), and the resulting quality value is used as the target (response) variable to be predicted by the regression model.

At the core of the proposed decision-making methodology is the characterization of the PFAs (process C in Figure 3), i.e., the extraction of a set of features (explanatory variables) that the model will later learn to associate with the target variable at the training step (process D in Figure 3). These features are extracted for every member of the PFA (just as quality measurements are computed independently for each of these members, in process B of the figure). The feature set encompasses individual aspects of the PFA members and the partitions they represent, as well as global aspects of the PFA and the particular clustering problem being solved. Section III-D elaborates further on the characterization process.

B. DECISION-MAKING STAGE: MODEL APPLICATION

In the decision-making stage of MLDM, the knowledge acquired during the learning stage is exploited to guide the selection of a final solution in a real (unsupervised) setting.

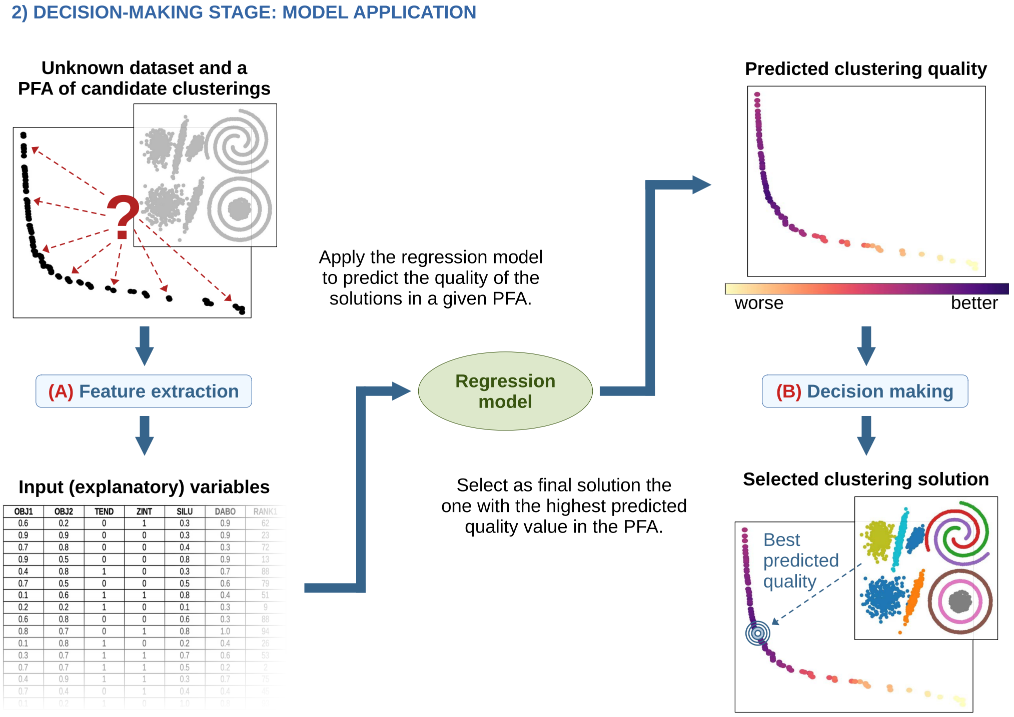
As illustrated in Figure 4, the input to this stage is a PFA generated by the chosen EMC method, which comprises a range of candidate solution alternatives for an unknown clustering problem (a problem for which information about the correct partitioning is unavailable, as it generally occurs in practice). Given that all these solution alternatives are nondominated, i.e., equally good in the Pareto-optimality sense, the goal is to employ the regression model constructed in advance to estimate their quality, enable discrimination (breaking ties), and hence identify a promising final choice.

More specifically, once the input PFA is characterized (process A in Figure 4, which is explained later in Section III-D), the regression model is applied to the feature vectors extracted for all PFA members to get their corresponding quality estimates. The candidate PFA member with the highest estimated quality value is chosen and presented as the final solution recommendation (process B in Figure 4).

C. DESIGN CHOICES AND CONSIDERATIONS

The methodology proposed, as described above, is independent from the specific definition of its main design components: (i) the EMC algorithm used to generate the PFAs; (ii) the set of features used to characterize the PFAs; (iii) the external cluster validity index used as the target variable; and (iv) the machine learning technique used to build the predictive model. Despite this flexibility, evaluating the impact of varying these components is beyond the scope of this study. We adopted specific choices for our experimental analysis:

- Algorithm Δ-*MOCK* [26] is chosen as the EMC method. As discussed before, Δ-*MOCK* is the successor of


2) DECISION-MAKING STAGE: MODEL APPLICATION
Unknown dataset and a PFA of candidate clusterings
?
(A) Feature extraction
Input (explanatory) variables
Apply the regression model to predict the quality of the solutions in a given PFA.
Regression model
Predicted clustering quality
worse
better
(B) Decision making
Select as final solution the one with the highest predicted quality value in the PFA.
Selected clustering solution
Best predicted quality

OBJ1	OBJ2	TEND	ZINT	SILU	DABO	RANK1
0.6	0.2	0	1	0.3	0.9	62
0.9	0.9	0	0	0.3	0.9	23
0.7	0.8	0	0	0.4	0.3	72
0.9	0.5	0	0	0.8	0.9	13
0.4	0.8	1	0	0.3	0.7	88
0.7	0.5	0	0	0.5	0.6	79
0.1	0.6	1	1	0.8	0.4	51
0.2	0.2	1	0	0.1	0.3	9
0.6	0.8	0	0	0.6	0.3	88
0.8	0.7	0	1	0.8	1.0	94
0.1	0.8	1	0	0.2	0.4	26
0.3	0.7	1	1	0.7	0.6	53
0.7	0.7	1	1	0.5	0.2	2
0.4	0.9	1	1	0.3	0.7	75
0.7	0.4	0	1	0.4	0.4	45
0.1	0.2	1	0	1.0	0.8	93

FIGURE 4. Decision-making stage of MLDM. The regression model obtained during the learning stage is employed to estimate the quality of the candidate solutions in a given PFA for an unknown problem. This predicted quality information is then exploited to identify the most promising solution alternative.

MOCK [10], a prominent algorithm from the EMC literature. Furthermore, the PFAs produced by Δ-*MOCK* present characteristics which are representative of the PFAs produced by several other EMC methods that rely on the same (or otherwise equivalent) optimization criteria [1], [9], [10], [11], [16], [45], [46], [47].

- To our knowledge, this is the first work that explores a methodology like the one proposed, including the need for defining a set of features (explanatory variables) to characterize the solutions in the PFAs produced by EMC methods. As such, the engineering of these features is seen as one of the contributions of this paper, thus receiving a separate, detailed treatment in Section III-D.
- Our supervised learning approach (building a model from sample problems with known solution), allows us to employ an external cluster validity index as an indicator of partition quality. This indicator is exploited as the target (output) variable, i.e., as the solution quality measure that the model will learn to estimate on the basis of the extracted features of PFA members. The *adjusted Rand index* (ARI) is the particular measure we adopted [48]. ARI evaluates the pairwise co-assignment of elements to clusters between two given partitions (in this case, a candidate partition in the PFA and the correct clustering for the sample problem). This measure is defined in the range [∼0, 1], where a value of 1 indicates a perfect agreement between the two partitions.
- Finally, given that the model is intended to predict solution quality, i.e., the (continuous) ARI value for PFA members, we approach decision making as a regression (rather than classification) task. The *random forest* technique has been adopted to construct such a regression model [49]. This technique has shown robustness to deal with mixed-type and high-dimensional features sets (as in our case, see Section III-D), obtaining promising results in different application domains [50], [51], [52].

D. CHARACTERIZATION OF APPROXIMATION SETS

The characterization process of the PFAs is a critical component that enables the application of the above-described decision-making methodology. This process involves extracting a set of features that allow us to describe each of the candidate partitions in the PFA, so that these features can later be linked to the adopted measure of solution quality. In other words, these features will assume the role of explanatory (input) variables during the supervised construction and subsequent utilization of our method's predictive model.

Although separate feature vectors are to be extracted for the different PFA members, these features need to capture aspects of these members both at the individual level and at the global, context level. That is, the decision on a final solution should not be made by only analyzing individual candidates in isolation, but by also taking into account the

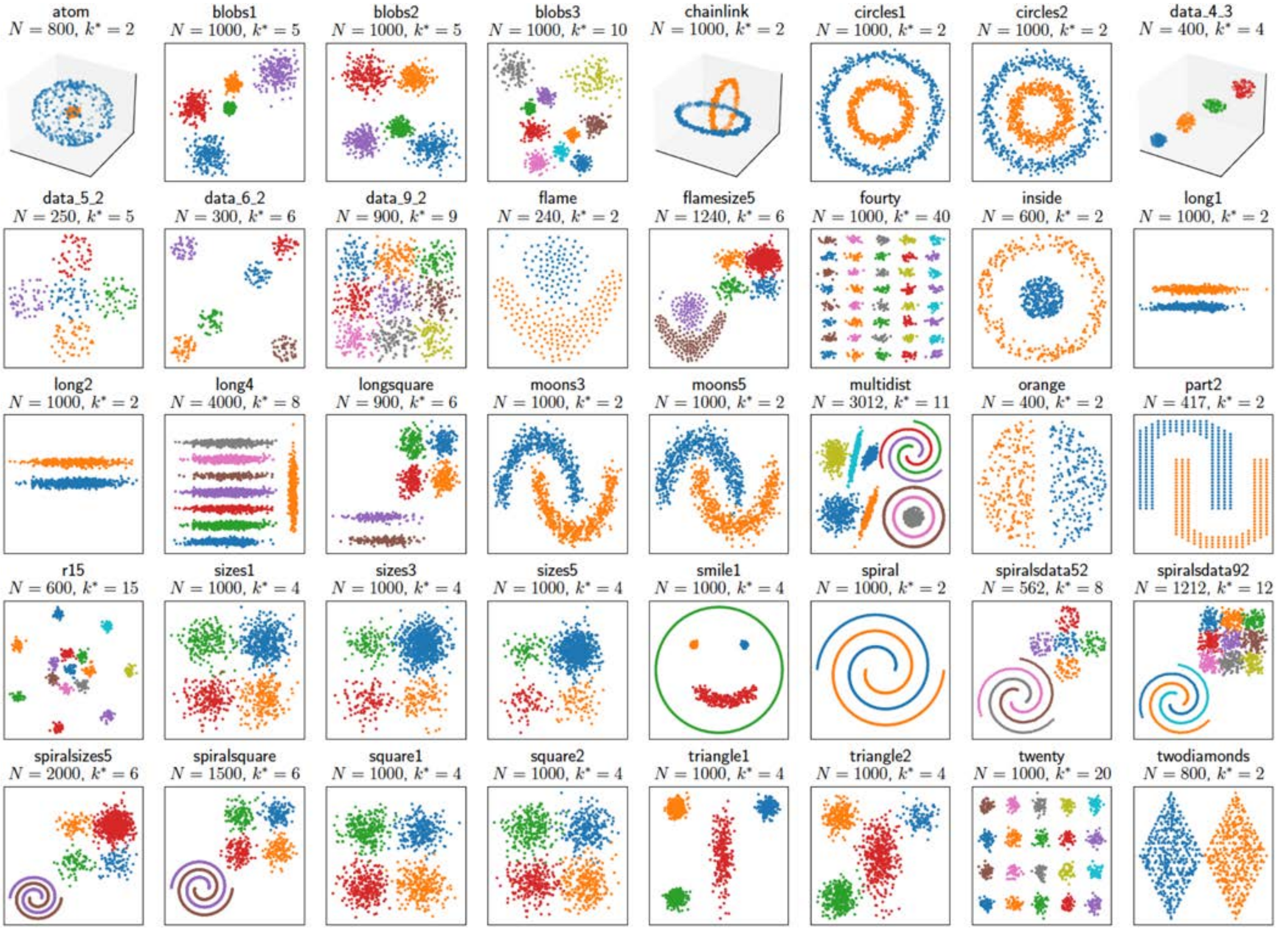


FIGURE 5. Illustration of the 40 synthetic datasets considered in this study. The size (N) and number of clusters (k^*) are specified for each of these datasets.

relationships between these candidates in the PFA. Looking at such relationships makes possible the evaluation of properties related to the geometry of the PFA as well as those referring to the entire set as a whole. These properties, together with information regarding the particular clustering problem being addressed, represent the context against which these alternative solution choices are presented for consideration.

In this study, a total of 55 features are defined to evaluate different aspects of PFA members and their context. The full description of these features, as well as of the five categories in which they are organized, is provided in Appendix A.

IV. EXPERIMENTAL SETUP

The following subsections describe the main settings of our experimental study, including the clustering problems used for testing, the decision-making approaches adopted as references, and the performance assessment measures considered.

A. CLUSTERING PROBLEMS

A total of 50 clustering problems are considered in our experiments, out of which 40 are synthetic and 10 are real-world datasets. The 40 synthetic problems, as illustrated in Figure 5, vary in size and present a diversity of characteristics regarding the shape, overlap/separation, and density of the clusters. The motivation for using these synthetic, low-dimensional problems, is to be able to associate the performance of the methods evaluated with the observable features of the data.

The 10 real-world problems are included to evaluate our proposal and reference methods under conditions which can be more representative of those encountered in practice. Table 1 specifies the size, dimensionality, and correct number of clusters in these real-world datasets: Banknote authentication (Banknote); Breast cancer Wisconsin diagnostic (Breast), Optical recognition of handwritten digits (Digits); Ecoli; Iris; Statlog landsat satellite (Landsat); Palmer archipelago penguin data (Palmer); Seeds; Thyroid disease (Thyroid); and Wine. All these datasets except one are available from the UCI machine learning repository.[2] The remaining problem, namely, Palmer, is provided by its authors via GitHub.[3]

[2] https://archive.ics.uci.edu

[3] https://allisonhorst.github.io/palmerpenguins

TABLE 1. The 10 real-world datasets considered in this study.

Dataset	Size (N)	Dimensionality	Number of clusters (k^*)
Banknote	1372	4	2
Breast	683	9	2
Digits	5620	64	10
Ecoli	336	7	8
Iris	150	4	3
Landsat	6435	36	6
Palmer	333	5	3
Seeds	210	7	3
Thyroid	215	5	3
Wine	178	13	3

B. REFERENCE APPROACHES

Our proposed MLDM method is evaluated with respect to a diverse set of decision-making approaches that have been proposed in the EMC literature. We consider representatives from the three categories described in Section II-C as well as some additional references, as detailed below.

1) REFERENCE METHODS BASED ON ADDITIONAL CLUSTERING CRITERIA

We include comparisons against the use of three separate clustering criteria: SIL [30], DB [32], and DUNN [37]. These indices, as discussed in Section II-C1, are commonly used for decision-making purposes. In addition, we consider the recently proposed index combination CH+DB+SIL [9].

2) REFERENCE METHODS BASED ON THE SHAPE OF THE PFA

Our analysis includes two methods based on the selection of a final solution from the knee of the PFA: *MOCK*'s strategy based on the computation of control fronts [10], and Shirakawa and Nagao's approach of selecting the solution that minimizes the sum of objective values (SUMO) [12]. These approaches have previously been discussed in Section II-C2.

3) REFERENCE METHODS BASED ON ENSEMBLE CLUSTERING

The two ensemble-based approaches by Zhu et al. [9], to be referred to as LWEA and LWGP, are included in our comparison. These approaches generate a set of consensus partitions using the techniques proposed by Huang et al. [44], and then apply index combination CH+DB+SIL to select a final solution (refer to Section II-C3 for additional details).

4) ADDITIONAL BASELINE REFERENCES

Finally, the following baselines are considered: the best and worst solutions in the PFA (BEST and WORST), representing the upper and lower bounds on the achievable performance; the extreme points of the PFA (EXT1 and EXT2), referring to the naive approach of always favoring one objective function over the other (intra-cluster variance and connectivity, respectively); and a random selection (RAND), which any reasonable decision-making strategy should outperform.

C. PERFORMANCE ASSESSMENT

The results of our experiments are evaluated considering three different aspects of performance. First, we evaluate prediction performance, referring to the capacity of our proposed method, and more specifically of the regression model employed, to accurately estimate the quality of the candidate solutions in the PFA. As indicated in Section III-C, we adopted ARI [48] as the measure of solution quality to be predicted by the model. Therefore, we evaluate prediction performance in terms of the *root-mean-square error* (RMSE) between the predicted and actual (measured) ARI values of candidate solutions. Lower RMSE values are always preferred, with 0 being the best possible value for this measure.

The second aspect evaluated is decision-making performance. Since the goal of decision making is to ultimately select a good solution from the PFA, this aspect refers precisely to the quality of the solutions chosen, which is given by their actual ARI values (as computed with respect to the correct solution of each problem). As stated in Section III-C, ARI is defined in the range $[\sim 0, 1]$ and is to be maximized.

The last considered aspect of performance is the effectiveness of the methods at determining the correct number of clusters, k. As discussed in Section II-C, selecting a final solution also implies deciding on the value of k, given the diversity of choices available in the PFA. Thus, we complement our evaluation of decision-making approaches by analyzing the absolute differences between the correct value k^* and the value of k of the solutions selected, $|k^* - k|$ (the lower the difference, the better the performance of the method is).

Finally, given the stochastic nature of some of the decision-making methods evaluated, we consider 20 independent repetitions of each experiment. In the specific case of MLDM, each repetition consists of the full process of training the regression model and testing; during training, the main hyperparameters of the random forest technique are adjusted by means of exhaustive (grid) search and 5-fold cross-validation. Statistical significance analyses are conducted for our main results using the (nonparametric) *Mann-Whitney U test*, considering in all the cases a significance level of $\alpha = 0.05$ and the *Holm-Bonferroni* correction procedure.

V. EXPERIMENTS AND RESULTS

This section presents the results of a series of experiments conducted to evaluate the suitability of the MLDM method proposed in this paper. First, the experiments of Sections V-A and V-B focus on synthetic clustering problems, each considering a particular decision-making scenario with different difficulty. Then, Section V-C extends this evaluation, analyzing the performance of our proposal on real-world datasets.

A. EXPERIMENT 1: KNOWN DATASETS

In the scenario considered in this experiment, we aim to select a final solution from an unknown PFA, generated for a dataset

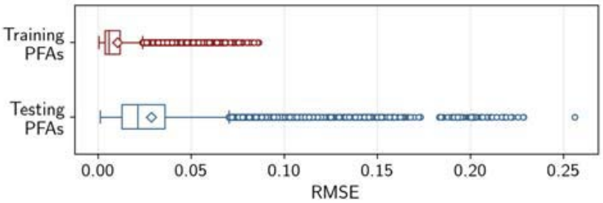


FIGURE 6. Experiment 1 - Prediction performance. The RMSE obtained by MLDM is shown for both training and testing PFAs. Summary of the results for all 40 synthetic problems and 20 independent repetitions of each experiment.

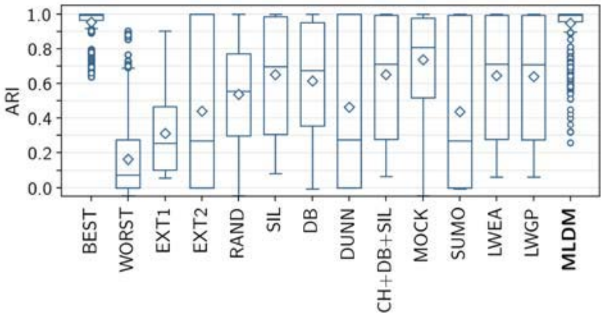


FIGURE 7. Experiment 1 - Decision-making performance. ARI values of the solutions selected by MLDM and reference approaches. Results are summarized for the 40 synthetic problems and the 20 repetitions performed.

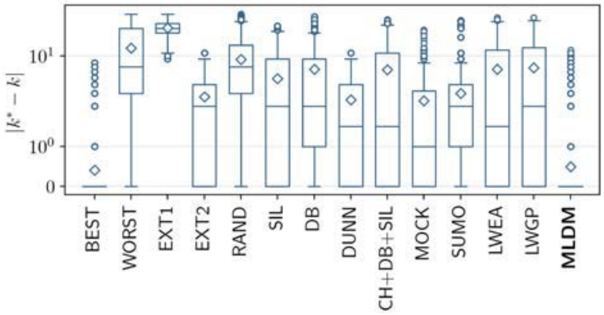


FIGURE 8. Experiment 1 - Performance at determining k. Absolute differences between k^* and the value of k of the solutions selected. Results are summarized for the 40 synthetic problems and 20 runs performed. Note that the y-axis of this plot is in logarithmic scale.

which is already known to MLDM. That is, the specific PFAs used for testing are completely unknown to MLDM, but other sample PFAs, obtained for the same dataset, were used during the training of MLDM's regression model. Despite being generated independently, the testing PFAs may share some similarities with the ones included in the training set. Hence, this particular scenario is evidently less challenging for MLDM, in comparison to the one analyzed later in Section V-B. Note, however, that this scenario is representative of situations where the same clustering problem (or a similar one) needs to be solved repeatedly (with certain frequency). In market segmentation, for example, the goal is to identify groups of customers so that differentiated, more effective strategies can be devised. We may expect the characteristics of the problem (and those of the PFAs produced for it) to remain comparable if the analysis always centers on the same type of information (e.g., demographics). Thus, it should be possible to learn from the outcome of previous decision-making processes, as evidenced by historical data or even by data from other business branches.

This experiment considers the 40 synthetic problems described in Section IV-A. For each problem, a total of 40 PFAs were generated through independent runs of Δ-*MOCK*. Out of these 40 PFAs, 20 were included in the training set and the remaining 20 were reserved for testing purposes. Considering that the average carnality of the PFAs is 100, the training and testing sets used in this experiment each contains approximately 80,000 solution samples. As indicated in Section IV-C, we performed 20 independent repetitions of the full process of training (including cross-validated hyperparameter tuning) and testing. The results are summarized in Figures 6, 7, and 8, which cover the three performance aspects discussed in Section IV-C and include comparisons against the references described in Section IV-B. Detailed results on decision-making performance, focusing on individual problems and including the findings of the statistical significance analysis, are provided in Table 2 (Appendix B).

As can be seen from Figure 6, MLDM reports promising results in terms of prediction performance, scoring relatively low RMSE values in most cases when applied to the unknown, testing PFAs (note, however, that these RMSE values are higher in comparison to those observed for the training data, which is an expected behavior in supervised learning). Low RMSE values confirm that MLDM's regression model has reasonably succeeded in estimating partition quality. This translates into a highly competitive decision-making performance, as shown in Figure 7 and Table 2. The solutions selected by MLDM yield high ARI values, significantly surpassing those selected by the reference methods and competing closely with the best solutions available in the PFAs (baseline BEST). MLDM was able to identify high-quality solutions for most problems (indeed, it chose the best solution alternative in many cases), in spite of the wide range of qualities observed across PFA members (see the large differences between baselines BEST and WORST). These observations are also consistent with the results of Figure 8, where MLDM is found to be the best performer in terms of the correct determination of the number of clusters.

The strongest contender among the references considered is MOCK's strategy, based on the identification of the knee of the PFA. This is followed by five approaches that seem to provide comparable (average) performances: SIL, DB, and CH+DB+SIL, which are based on the use of additional clustering criteria, and LWEA and LWGP, which are based on ensemble clustering. It is interesting to observe that the use of strategies DUNN and SUMO leads to the selection of

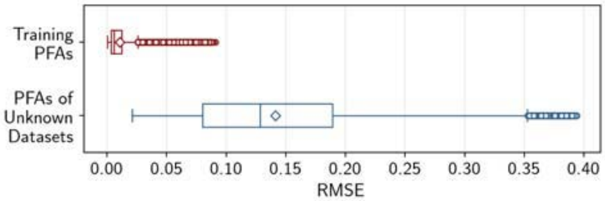


FIGURE 9. Experiment 2 - Prediction performance. The RMSE obtained by MLDM is shown for both training PFAs (included as a reference) and testing PFAs generated for unknown problems. Summary of the results for all 40 synthetic datasets and 20 independent repetitions of each experiment.

solutions which are, at least in average, poorer in quality than those selected at random (baseline RAND).

B. EXPERIMENT 2: UNKNOWN DATASETS

The scenario of our second experiment focuses on the selection of a final solution from a PFA generated for a dataset which is unknown to MLDM. That is, contrasting with the experiment presented earlier in Section V-A, in the more challenging setting considered herein the training set completely excludes PFAs produced for the same dataset being used for testing. Therefore, this experiment is intended to investigate the ability of our proposal to learn from the knowledge available for example problems and exploit it to guide decision making in the context of new applications.

For each of the 40 synthetic problems, we generated 20 PFAs by means of independent runs of algorithm Δ-*MOCK*. In this case, however, we only used 39 problems at a time for training, leaving the remaining problem out for testing. Consequently, 40 configurations of this experiment were required, each allowing a different problem to be excluded from training and reserved for testing. These configurations consider training and testing sets with roughly 78,000 and 2,000 solution samples, respectively (given that the 20 PFAs of every problem contain about 100 solutions). Furthermore, for each experiment configuration 20 independent repetitions of the training and testing processes were performed, as indicated in Section IV-C. Below, the performance of MLDM is compared against several reference methods (Section IV-B) and analyzed from multiple perspectives (Section IV-C).

Our results make evident the more challenging conditions of this new scenario. From Figure 9, it is possible to observe a decrease in prediction performance when MLDM is applied to PFAs of unknown problems, with significantly higher RMSE values than those reported for the previous experiment (see Figure 6). Such a decrease in prediction performance is also reflected in reduced decision-making and k-determination capabilities, as can be seen from Figures 10 and 11 (and by contrasting these results with those shown previously in Figures 7 and 8). It is noteworthy that the increased difficulty of this scenario is only relevant to MLDM; reference methods thus maintain the same behaviors as observed and discussed at the end of Section V-A.

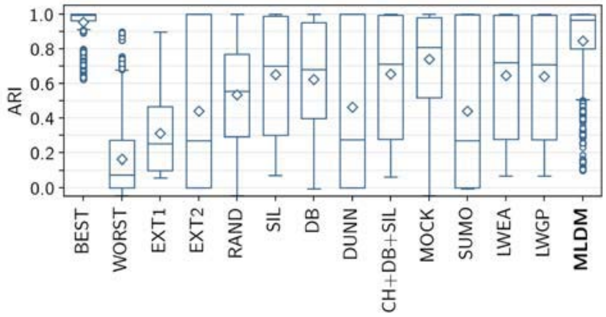


FIGURE 10. Experiment 2 - Decision-making performance. ARI values of the solutions selected by MLDM and reference approaches. Results are summarized for the 40 synthetic problems and the 20 repetitions performed.

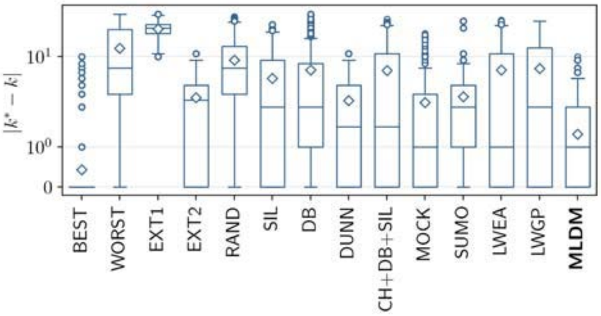


FIGURE 11. Experiment 2 - Performance at determining k. Absolute differences between k^* and the value of k of the solutions selected. Results are summarized for the 40 synthetic problems and 20 runs performed. Note that the y-axis of this plot is in logarithmic scale.

Despite the aforementioned performance drops, MLDM's results in terms of decision-making and k-determination are clearly competitive. From the overall results of Figures 10 and 11, our method is seen to outperform all reference approaches evaluated. This suggests that the estimations of partition quality produced by MLDM's regression model, although not as accurate as those observed in Section V-A, are still sufficiently informative so as to induce an effective discrimination among the competing solutions in the PFAs. At this point, it is worth considering the question of how strong the correlation between prediction performance and decision-making performance is, which we investigate by analyzing Figure 12. The figure confirms that the two performance aspects are certainly correlated, where lower RMSE values tend to be associated with a higher quality of the solutions chosen. More interestingly, though, the figure also reveals that there is an important number of cases indicating the selection of high-quality solutions in spite of relatively high prediction errors. From this, we can stress that even inaccurate predictions can provide useful information to guide the identification of promising solution alternatives, which is the ultimate goal of decision making. The accuracy of the regression model's predictions is therefore a sufficient, but not necessary condition for the effectiveness of MLDM.

Finally, some interesting behaviors can be derived from the analysis of problem-specific results of MLDM and reference

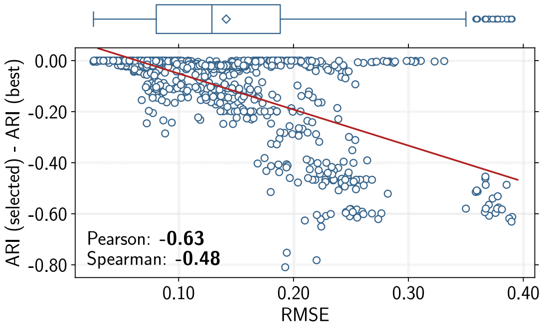


FIGURE 12. Experiment 2 - Correlation between prediction and decision-making performance. The horizontal axis refers to the RMSE values reported by MLDM. The vertical axis refers to the difference between the ARI of the solution selected by MLDM and the ARI of the best solution available in the PFA; a value of 0 indicates that MLDM selected the best solution, whereas a negative result indicates that a solution with lower ARI was selected. Pearson's and Spearman's correlation coefficients are shown as a reference.

methods. Figure 13 presents individual results for a sample of 12 problems, but detailed results for the full set of 40 problems can be found in Table 3 (Appendix B). On the one hand, we would like to emphasize that all contestant methods have stood out, showing a good performance for specific subsets of problems. However, they lack robustness and fail when problem properties change. For example, DB, DUNN, and SUMO are among the best performers for problems that present non-overlapping and non-linearly separable clusters (these properties can be visualized in Figure 5), such as: atom, chainlink, circles1, inside, orange, part2, and smile1. Note that the remaining references, namely, SIL, CHz+DB+SIL, MOCK, LWEA, and LWGP, in most cases scored a poor performance for this particular subset of problems. The completely opposite situation occurs when we consider datasets with linearly separable clusters and varying degrees of overlap, such as: blobs1, blobs2, data_5_2, data_9_2, r15, sizes1, sizes5, square2, triangle2, and twodiamonds. In these problems, methods SIL, CH+DB+SIL, MOCK, LWEA, and LWGP report high ARI values, whereas methods DB, DUNN, and SUMO show a low performance.

On the other hand, it is possible to highlight the increased robustness that MLDM has shown across the diversity of characteristics covered by our dataset collection. Our proposal competes with some of the best results for most of the above-mentioned problems. Moreover, there are other problems for which our proposal is clearly the best performer (in fact, for some problems MLDM is the only method providing a reasonable result): flamesize5, longsquare, moons3, moons5, spiralsizes5; with the exception of longsquare, these problems seem to combine characteristics of non-linearly separable and overlapping clusters. Finally, we can also identify three datasets, namely, circles2, data_9_2, and spiralsdata92, for which MLDM exhibits a notably low performance. These are difficult problems, as can be judged from the results of all baselines and strategies evaluated. It might also be the case, however, that the particular properties of these problems are not well represented in the training data, which would certainly explain the low performance observed (as a supervised learning method, MLDM's success depends on the availability of representative training samples).

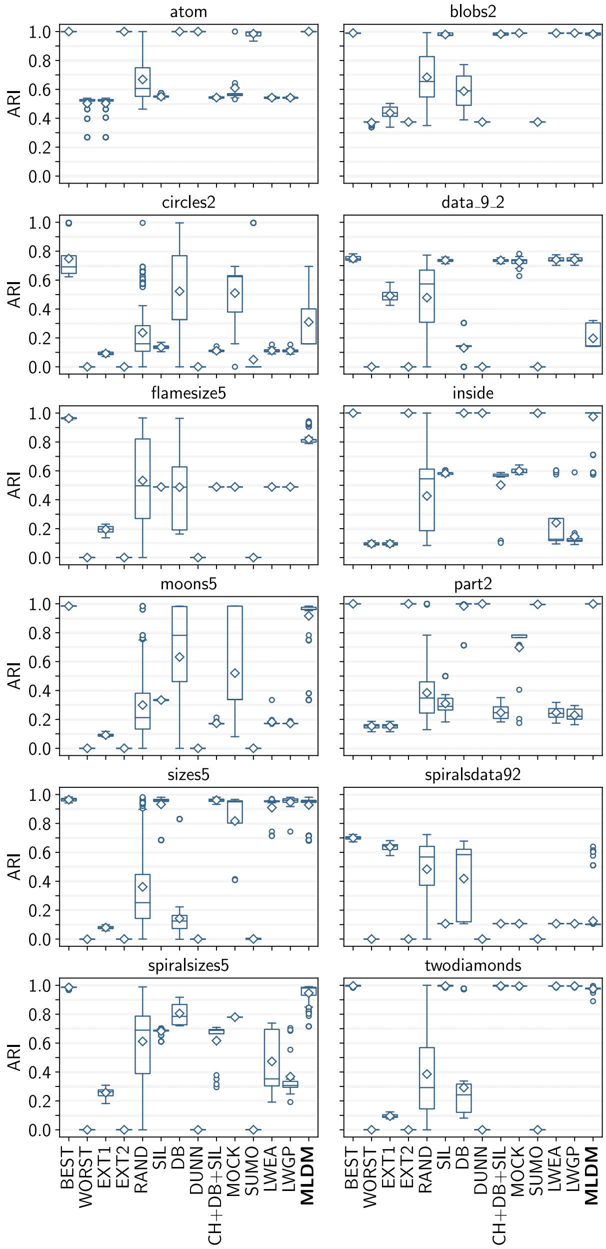


FIGURE 13. Experiment 2 - Decision-making performance. The ARI of the solutions selected by MLDM and reference approaches is shown. Individual results for a subset of 12 problems (summary of the 20 repetitions performed).

C. EXPERIMENT 3: REAL-WORLD DATASETS

So far we have considered low-dimensional, synthetic clustering problems. This has allowed us to ensure that our collection of test scenarios spans a diversity of characteristics and, more importantly, has enabled us to relate such characteristics

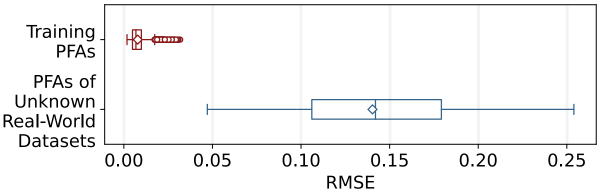


FIGURE 14. **Experiment 3 - Prediction performance. The RMSE obtained by MLDM is shown for both training PFAs (included as a reference) and testing PFAs generated for unknown problems. Summary of the results for all 10 real-world datasets and 20 independent repetitions of each experiment.**

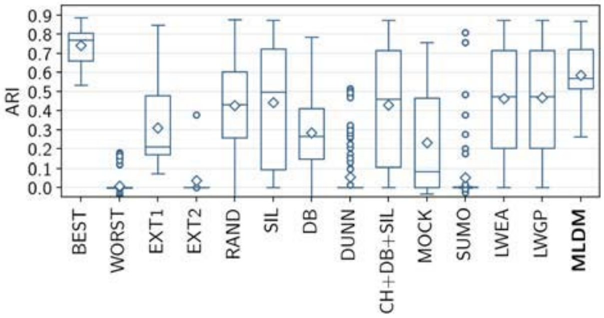


FIGURE 15. **Experiment 3 - Decision-making performance. ARI values of the solutions selected by MLDM and reference approaches. Results are summarized for the 10 real-world datasets and the 20 repetitions performed.**

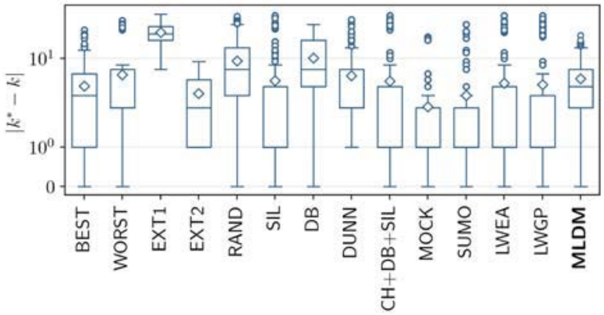


FIGURE 16. **Experiment 3 - Performance at determining k. Absolute differences between k^* and the value of k of the solutions selected. Results are summarized for the 10 real-world datasets and 20 runs performed. Note that the y-axis of this plot is in logarithmic scale.**

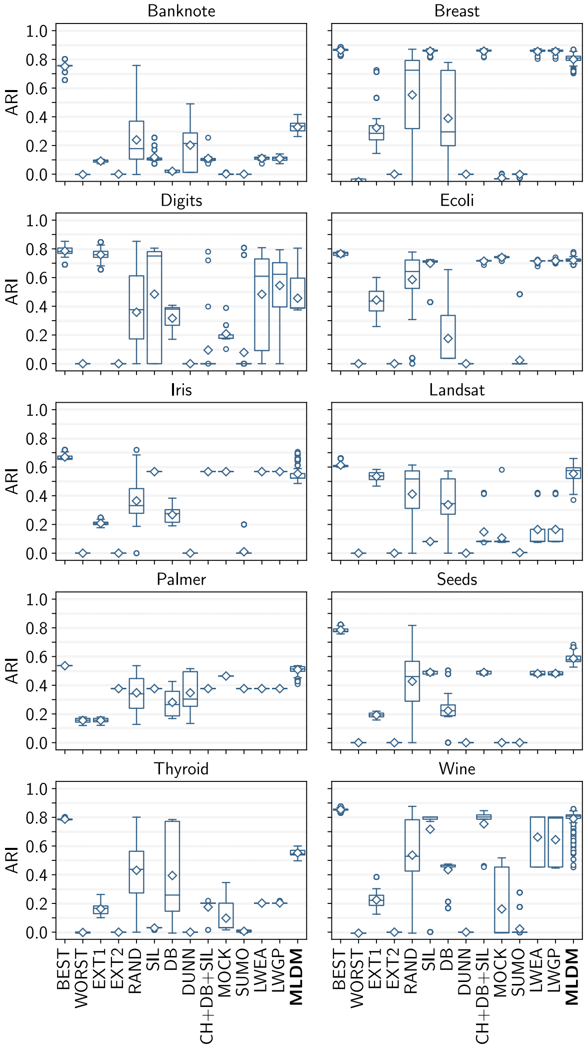
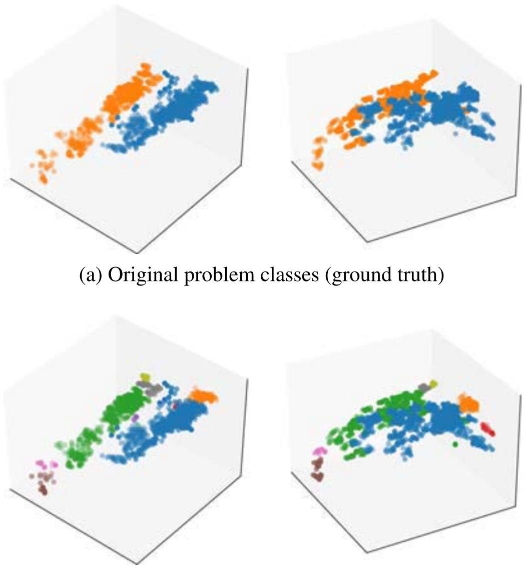


FIGURE 17. **Experiment 3 - Decision-making performance. The ARI of the solutions chosen by MLDM and reference methods is shown. Individual results for the 10 real-world datasets (summary of the 20 repetitions performed).**

with the performance of MLDM and the reference methods evaluated. Nevertheless, it is essential to validate our findings and the suitability of these approaches under more realistic conditions. We thereby replicate the experiment presented previously in Section V-B, focusing now on the set of 10 real-world datasets described in Section IV-A.

As explained in Section V-B, the experiment has been designed to investigate the ability of MLDM to model available knowledge from example settings and exploit it to accomplish decision making in the context of a completely new (previously unseen) problem. Distinct configurations of the experiment are considered so that each of the 10 real-world problems is used exactly once for testing, ensuring that no sample PFAs for the same problem are included in the training set (the training set involves only the remaining 9 problems). This results in training and testing sets with 18,000 and 2,000 solution samples, respectively (we generated 20 PFAs for each dataset, each of which containing about 100 solution samples). As before, we ran every configuration of this experiment multiple times independently, presenting summaries of the results obtained from the perspective of different performance indicators in Figures 14, 15, and 16. Additionally, Figure 17 and Table 4 (Appendix B) provide separate results for the 10 problems considered.

The results obtained on the real-world problems bear some resemblance to those observed for the synthetic data in Section V-B. Despite the relatively high RMSE values reported in Figure 14, Figure 15 indicates that MLDM selected final solutions which in average present a better



(a) Original problem classes (ground truth)

(b) Best clustering found according to the ARI measure

FIGURE 18. Multidimensional scaling projection of the Banknote dataset. Colors illustrate the two original problem classes (a) as well as the $k = 9$ clusters in the best solution observed according to the ARI measure (b). Left and right sides show two distinct rotations of the three-dimensional plots.

quality in comparison to those selected by all the reference methods and baselines (with the evident exception of BEST), therefore being the best overall performer in this test. Analyzing decision-making performance from the perspective of individual problems, the results of Figure 17 are consistent with previous observations regarding the increased robustness that our proposal exhibits across test scenarios. MLDM ranks among the best performers for all the 10 problems considered, unlike the reference methods which only stand out in particular cases. Specifically, the most robust references, SIL, LWEA, and LWGP, only performed well in half of the problems: Breast, Digits, Ecoli, Iris, and Wine; CH+DB+SIL performed well in all these problems, except Digits; MOCK appears as one of the best performers only for three problems: Ecoli, Iris, and Palmer; DB competes with some of the best results only for problems Landsat and Thyroid; and, finally, approaches DUNN and SUMO scored a poor performance in all cases.

It is interesting to note, however, that the real-world datasets certainly posed some challenges for all the methods evaluated. In general, we can see from Figure 15 that all the methods scored ARI values which are consistently lower than those reported for the previous experiment (Figure 10). Approaches DB, DUNN, MOCK, and SUMO have performed even worse than baseline RAND (which selects a solution at random). Furthermore, it is possible to observe that baseline BEST (upper bound on achievable performance) shows an average ARI of about 0.75 (in contrast to the value of 0.95 that it reports in Figure 10). This indicates an unavailability of high-quality solutions in most of the PFAs considered (at least from the perspective of the ARI indicator), which can indeed explain the lower performance of all methods.

An alternative explanation for the lower ARI values observed consistently in this experiment, is the assumption that the class assignments (labels) specified for these real-world problems reflect the correct partition of the data, which we use as the reference (ground truth) for the computation of this measure. Although these class assignments effectively group the samples and are undoubtedly relevant to the particular application domains of these datasets, they do not necessarily match exactly the inherent cluster structure of the data. To illustrate this, consider the *multidimensional scaling* projection of dataset Banknote, shown in Figure 18. As can be seen from the figure, the two classes defined for this problem result in a clear separation of the data samples; however, from the clustering perspective it makes more sense to split these classes further into multiple clusters. This finding also offers an explanation to the large errors that all the methods report in terms of k-determination, as shown in Figure 16.

VI. CONCLUSION

Limitations of existing decision-making methods challenge the applicability of EMC algorithms, as the delivery of a single final solution is a necessary step for them to be fully useful in practice. The underlying assumptions of current proposals do not always hold under the peculiarities of the data and the application domains, stressing the need for alternative approaches to cope with the complex nature of this task. In view of this, we explored a novel approach to decision making, demonstrating the viability of addressing it through machine learning. The key concept of our supervised learning-based proposal involves: (i) learning in advance the association between partition quality and a set of features extracted from nondominated solutions; and (ii) exploiting the knowledge gained to drive decision making, enabling the identification of a promising final solution. Our main finding is that, by following the proposed methodology, it is possible to generalize prior learning to completely new scenarios.

Our proposal was evaluated and compared to eight representative decision-making approaches from the literature and some additional baselines. This evaluation was conducted over a diverse collection of synthetic and real-world datasets, under different experimental conditions. Our method consistently reported the best overall performance throughout our experiments. Moreover, it showed an increased versatility with respect to the changing problem characteristics. In general, our results underline the suitability of this proposal and its superiority with respect to existing techniques. On the other hand, the proposed method also presents a limitation that is inherent to supervised learning settings: it relies on the availability of training samples which are representative of the scenarios seen in practice. Although our proposal has shown some robustness, providing competitive results for problems which were completely excluded from the training process, special attention needs to be given to the training set compilation process in order to overcome this limitation.

The outcomes of this study are encouraging, confirming that the development of alternative decision-making approaches is a valuable direction which merits additional research. Preliminary analyses have revealed potential opportunities to further improve the effectiveness of our proposal through an in-depth inspection of our feature set. This should lead to the removal of irrelevant or redundant features, to achieve meaningful dimensionality reductions, as well as to the engineering of new features that can better capture the complexities of the decision-making task. Despite being proposed as a generic framework, to delimit the scope of this study the evaluation of our proposal centered around a specific EMC algorithm, Δ-*MOCK* [26], and the particular optimization criteria used by it (similar design choices had to be made regarding other components, such as the machine learning method used to construct the regression model and the partition quality criterion used as the response variable). Hence, extending this study to other different conditions, including the use of more than two optimization criteria, would certainly support the generality of our conclusions. Finally, our method exploits characteristics of the decision-making task which apply only to the specific EMC context. An interesting research direction concerns exploring the applicability of a methodology like the one proposed in this paper with the aim to assist decision making in the more general context of multiobjective optimization.

APPENDIX A
DEFINITION OF FEATURES

As discussed in Section III-D, a set of 55 features are used in this study for the characterization of candidate solutions in the PFAs. These features have been assigned specific acronyms and are organized into five distinct categories, which are separately defined in the following subsections.

A. CATEGORY 1: FEATURES DESCRIBING PFA MEMBERS INDIVIDUALLY

The first category involves features which are defined to describe aspects of the PFA members at the individual level. A total of 11 features are included in this category:

- **Objective values (OBJ1, OBJ2).** These features correspond to the (raw) objective values of the candidate partition (its specific coordinates in objective space, see Figure 19). In the particular case of algorithm Δ-*MOCK*, OBJ1 and OBJ2 refer to the values scored for the intra-cluster-variance and connectivity criteria.
- **Normalized objective values (NOBJ1, NOBJ2).** As illustrated in Figure 20, NOBJ1 and NOBJ2 are versions of features OBJ1 and OBJ2, considering the independent normalization of each dimension of the PFA to the range $[0, 1]$. Min-max normalization is applied.

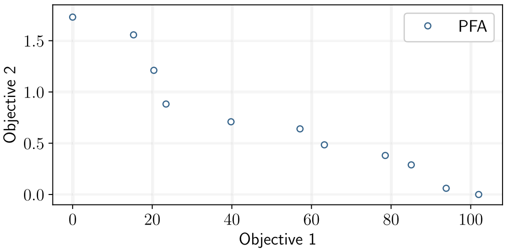


FIGURE 19. Features OBJ1 and OBJ2 refer to the objective values produced by Δ-*MOCK*, i.e, to the coordinates of PFA members in objective space.

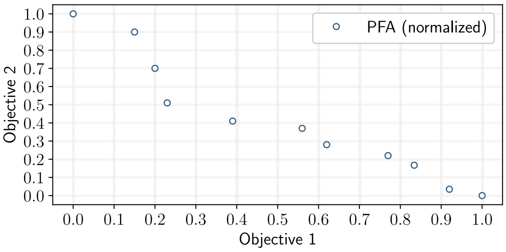


FIGURE 20. Features NOBJ1 and NOBJ2 correspond to the coordinates of PFA members, after normalizing each dimension independently to range [0, 1].

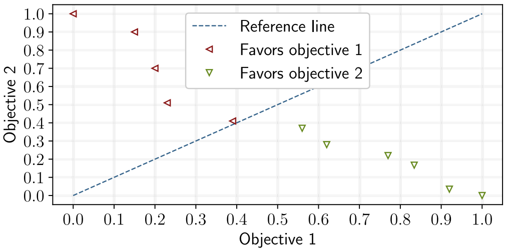


FIGURE 21. Feature TEND captures the tendency that a PFA member can present towards favoring one objective over the other, which depends on its location with respect to a line connecting reference points (0, 0) and (1, 1).

- **Average of normalized objectives (ANOBJ).** This feature is computed as the arithmetic mean of the normalized objective values (NOBJ1 and NOBJ2 features).
- **Tendency towards a particular objective (TEND).** Each solution in the PFA exhibits a particular trade-off, which can favor one of the optimized criteria more than the other. Accordingly, we assign three possible values to the TEND feature, $\{0, 1, 2\}$. As shown in Figure 21, this depends on the location of the PFA member with respect to a reference line connecting points $(0, 0)$ and $(1, 1)$ of the normalized objective space. If the PFA member appears in the upper half, then it shows a tendency towards the first criterion and we assign a value of 0. If it locates in the opposite half, then it favors the second criterion and we assign a value of 2. Otherwise, no tendency is observed and we assign a value of 1.

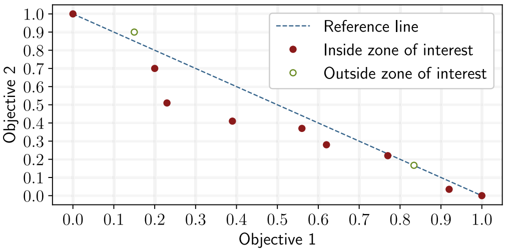


FIGURE 22. Feature ZINT indicates whether a PFA member is inside the interest zone, delimited by a line connecting the extreme points of the PFA.

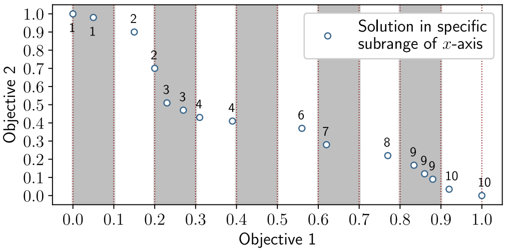


FIGURE 23. Feature SUBX refers to the specific number of subrange of the horizontal, x-axis where the candidate PFA solution locates.

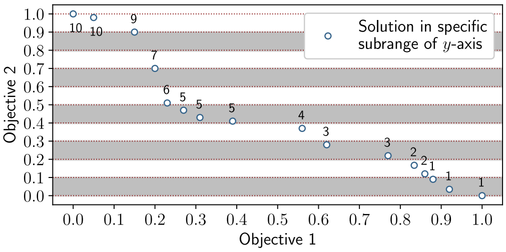


FIGURE 24. Feature SUBY refers to the specific number of subrange of the vertical, y-axis where the candidate PFA solution locates.

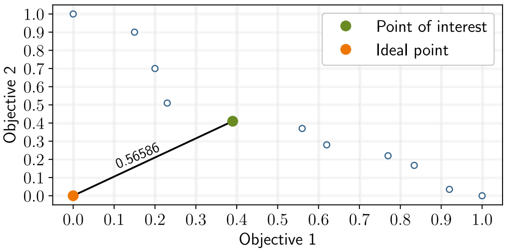


FIGURE 25. IDEAL denotes the distance from the PFA member to the ideal point approximation at coordinates (0, 0) of the normalized objective space.

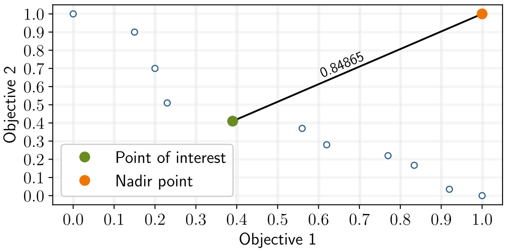


FIGURE 26. NADIR refers to the distance from the PFA member to the nadir point approximation at coordinates (1, 1) of the normalized objective space.

- **Zone of interest (ZINT).** The computation of this binary feature requires tracing a reference line connecting the extreme points of the PFA. As illustrated in Figure 22, these points are located at the (0, 1) and (1, 0) corners of the normalized objective space. PFA members appearing in the upper half (above the reference line) are considered to be outside the zone of interest and assigned a value of 0 for the ZINT feature. Such solutions exhibit less interesting trade-offs in comparison to PFA members located at the opposite half, the interest zone, which are assigned a value of 1.
- **Subrange of x-axis and y-axis (SUBX, SUBY).** The x-axis and y-axis are discretized into a number of subranges, see Figures 23 and 24. Features SUBX and SUBY indicate the specific number of subrange (x-axis and y-axis, respectively) where the PFA member resides. In this study, the x-axis and y-axis were both split into 10 subranges after being independently normalized to range [0, 1]. Thus, features SUBX and SUBY are assigned a value of 1 if the PFA member is within subrange [0.0, 0.1] of the corresponding axis, a value of 2 if it lies at subrange (0.1, 0.2], and so on.
- **Distance to the ideal point approximation (IDEAL).** In multiobjective optimization, the vector given by the best (feasible) value that can be reached for every objective function constitutes the so-called *ideal point*. Considering the normalization of the PFA to the range [0, 1] in both dimensions, point (0, 0) can thus be seen as our current approximation to the ideal point. We compute feature IDEAL as the distance from the PFA member to such an approximated ideal point, see Figure 25.
- **Distance to the nadir point approximation (NADIR).** Contrary to the ideal point, the *nadir point* is given by the worst value for each objective function in the entire Pareto-optimal set. Therefore, point (1, 1) of the normalized objective space represents an approximation to the nadir point from the perspective of the PFA under consideration. Feature NADIR, as illustrated in Figure 26, is calculated as the distance of the PFA member with respect to such an approximation.

B. CATEGORY 2: FEATURES DESCRIBING THE PARTITION THAT THE PFA MEMBERS REPRESENT

Each member of the PFA represents a candidate solution to the particular clustering problem under consideration. The second category is thus concerned with features which refer to properties and measures of quality of these candidate partitions. Four different features belong to this category:

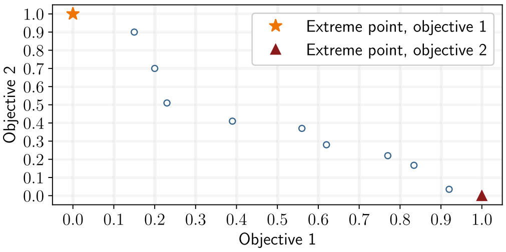


FIGURE 27. The extreme points of the PFA correspond to the candidate solutions that exhibit the best obtained values for the objective functions.

- **Number of clusters (KCLU).** As initially exemplified through Figure 1, the PFA may involve solutions showing a diversity of numbers of clusters, k. Feature KCLU refers to the specific value of k of the partition represented by a given PFA member.
- **Internal cluster validity indices (SIL, DB, DUNN).** These are unsupervised measures that assess solution quality by analyzing specific aspects of the clusters in the partitions they define. We include in our feature set three indices which are popular choices and have also been exploited for decision-making purposes, as seen in Section II-C1: SIL [30], DB [32], and DUNN [37].

C. CATEGORY 3: FEATURES DESCRIBING THE PFA MEMBER IN RELATION TO OTHER PFA MEMBERS

Unlike previous categories, the third category of features considers properties of the PFA member whose evaluation depends on other candidate members of the PFA (such as neighboring solutions or the extreme points of the PFA). This category includes the following 16 features:

- **Ranking for individual objectives (RANK1, RANK2).** Features RANK1 and RANK2 are computed as the rank positions of the candidate solution after sorting the full list of PFA members according to their values for the first and second objective functions, respectively. Given that all PFA members are nondominated with respect to each other, a total order is obtained when considering the two objective functions independently. Thus, no PFA member is ranked equal to any other, and every member receives a distinct value for features RANK1 and RANK2. The solution with the best objective value, i.e., the extreme point of the PFA, is assigned rank 1 for the corresponding objective, whereas the solution with the worst objective value (at the opposite extreme) is assigned a rank that equals the cardinality of the PFA.
- **Extreme points of the PFA (EXT1, EXT2).** Binary features indicating whether or not the PFA member is the extreme point (best objective value) for the first and second objective functions, respectively (see Figure 27).
- **Membership to convex hull (INCVX).** The convex hull (or convex closure) is given by the smallest subset of points which define a convex polygon enclosing all points in a set. After discarding solutions outside the zone of interest (see Figure 22 and description of feature ZINT), the convex hull is computed for the remaining PFA members, as shown in Figure 28. Feature INCVX is assigned a value of 1 whenever the PFA member is a vertex of the resulting convex hull, and 0 otherwise.

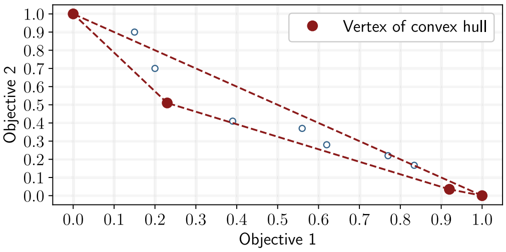


FIGURE 28. Feature INCVX indicates whether a PFA member is a vertex of the convex hull (computed for points in the zone of interest, see Figure 22).

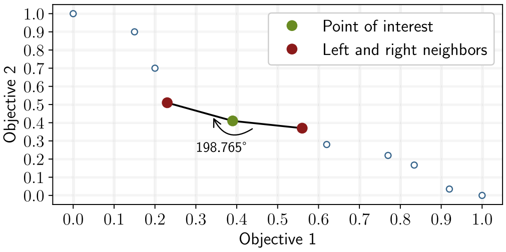


FIGURE 29. Feature ANGNE is defined as the angle between the lines connecting the PFA member characterized to its left and right neighbors.

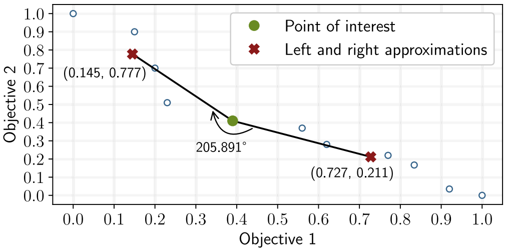


FIGURE 30. Feature ANGAP is computed as the angle between the lines connecting the PFA member with its left and right approximated neighbors.

- **Angle between closest neighbors (ANGNE).** As illustrated in Figure 29, this feature is computed as the angle between the lines joining the solution being characterized with its left and right closest neighbors in the PFA.
- **Angle between left and right approximations (ANGAP).** ANGAP is proposed as a "smooth" version of feature ANGNE defined above. Rather than considering the left and right neighbors, ANGAP uses hypothetical points calculated as the average of the coordinates of all PFA members at the left (respectively right) side of the point being characterized, see Figure 30.
- **Contribution to hypervolume (CHVOL).** The hypervolume, as discussed further in Appendix A-D (definition of feature HVOL), is a well-known performance indicator in evolutionary multiobjective optimization, evaluating the quality of a PFA as a whole [53].

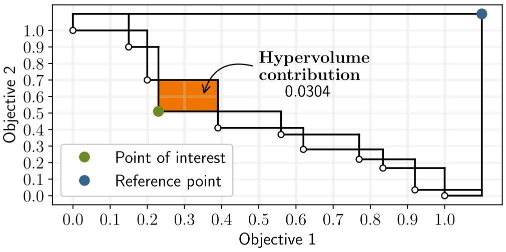


FIGURE 31. Feature CHVOL describes a PFA member in terms of its contribution to the value of the hypervolume indicator.

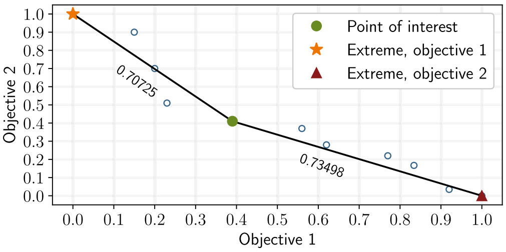


FIGURE 32. Features DEXT1 and DEXT2 refer to the distance from the PFA member to the extreme points of the first and second objectives, respectively.

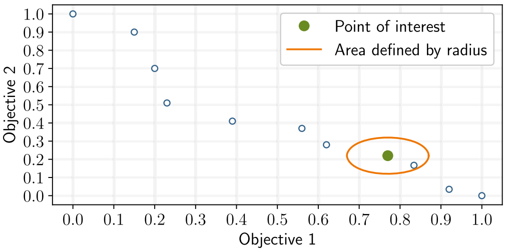


FIGURE 33. Feature RADIUS computes the number of neighboring PFA members within a certain radius from the solution being characterized.

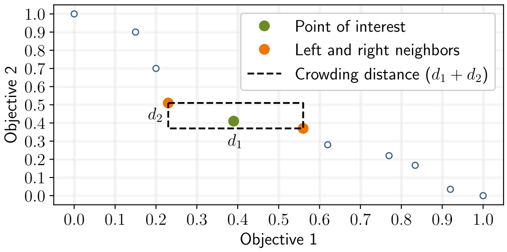


FIGURE 34. Feature CROWD refers to the crowding distance of the PFA member, a measure introduced as part of algorithm NSGA-II [55].

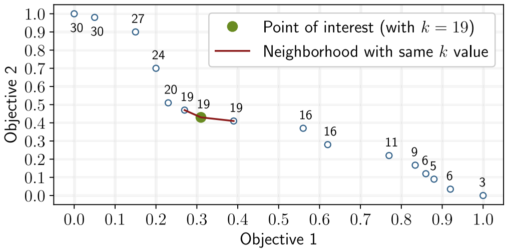


FIGURE 35. Feature NEIGK counts the total number of consecutive neighbor solutions around the PFA member having the same value for *k*.

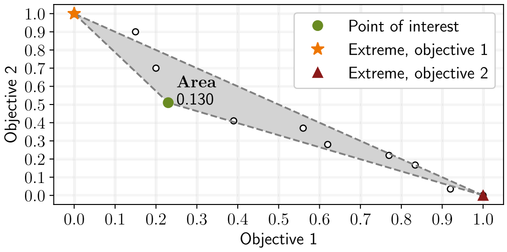


FIGURE 36. Feature TRIAR is given by the area of the triangle formed by the PFA member under consideration and the extreme points of the PFA.

The contribution of individual solutions to the value of this indicator (see Figure 31) has been used as a criterion to guide the search process [54]. This approach is adopted as one of our features to characterize PFA members.

- **Distance to the extreme points (DEXT1, DEXT2).** Features DEXT1 and DEXT2 are given by the Euclidean distance from the point being characterized to the extreme points of the PFA, as shown in Figure 32 (objective values are normalized to range [0, 1]).
- **Number of points within a certain radius (RADIUS).** This feature, exemplified in Figure 33, refers to the total number of solutions lying within a certain radius r from the PFA member under consideration. In this study, we adopted a fixed value of $r = 0.1$, which represents about 7% of the distance between the extreme points of the PFA (assuming the previous normalization of the PFA).
- **Crowding distance (CROWD).** The *crowding distance* is a measure implemented within the *nondominated sorting genetic algorithm 2* (NSGA-II) as a means to promote the diversity and distribution of solutions in the PFA [55]. Feature CROWD refers to the value of this measure, whose computation is illustrated in Figure 34.
- **Neighbors with the same *k* value (NEIGK).** As shown in Figure 35, feature NEIGK specifies the number of consecutive neighbors (at both sides) presenting the same value of k as the PFA member characterized.
- **Percentage of points with the same *k* value (PERCK).** PERCK is the percentage of the full set of solutions in the PFA that exhibit the same value of k as the specific PFA member being characterized.
- **Triangle area (TRIAR).** This feature is illustrated in Figure 36. As can be seen, it is defined as the area of the triangle which has the PFA member and the extreme points as its vertices. The area can be calculated from the length of the triangle's sides, using Heron's formula. If the point being characterized is outside the zone of interest (see description of feature ZINT, Appendix A-A), the negative of the computed area is used instead.
- **Triangle height (TRIHE).** In a similar manner to feature TRIAR, this feature considers the triangle defined

TABLE 2. Detailed results for Experiment 1, presented separately for the 40 synthetic clustering problems. The median ARI of the final solutions selected by the decision-making approaches analyzed is shown, highlighting the best performance scored for each problem. Results for all reference methods considered (except baselines BEST, WORST, EXT1, EXT2, and RAND) are marked ● to indicate that a statistically significant difference is observed with respect to MLDM.

Dataset	N	k	BEST	WORST	EXT1	EXT2	RAND	SIL	DB	DUNN	CH+DB+SIL	MOCK	SUMO	LWEA	LWGP	MLDM
atom	800	2	1.000	0.524	0.526	**1.000**	0.629	0.545 ●	**1.000**	**1.000**	0.543 ●	0.563 ●	0.995 ●	0.542 ●	0.543 ●	**1.000**
blobs1	1000	5	1.000	0.522	0.525	0.614	0.756	0.997	0.532 ●	0.614 ●	0.997	**1.000**	0.614 ●	**1.000**	**1.000**	0.999
blobs2	1000	5	0.990	0.374	0.444	0.374	0.704	0.975 ●	0.662 ●	0.374 ●	0.978 ●	**0.990**	0.374 ●	**0.990**	**0.990**	**0.990**
blobs3	1000	10	0.996	0.047	0.766	0.047	0.894	**0.996**	0.750 ●	0.808 ●	**0.996**	0.808 ●	0.047 ●	**0.996**	**0.996**	**0.996**
chainlink	1000	2	1.000	0.085	0.085	**1.000**	0.207	0.145 ●	0.687 ●	**1.000**	0.097 ●	0.253 ●	**1.000**	0.107 ●	0.106 ●	**1.000**
circles1	1000	2	1.000	0.070	0.070	**1.000**	0.196	0.189 ●	**1.000**	**1.000**	0.077 ●	0.671 ●	**1.000**	0.082 ●	0.078 ●	**1.000**
circles2	1000	2	0.695	0.000	0.093	0.000	0.186	0.145 ●	0.327 ●	0.000 ●	0.116 ●	0.623 ●	0.000 ●	0.117 ●	0.114 ●	**0.695**
data_4_3	400	4	1.000	0.255	0.255	**1.000**	0.612	**1.000**	0.580 ●	**1.000**	**1.000**	**1.000**	0.997 ●	**1.000**	**1.000**	**1.000**
data_5_2	250	5	0.914	0.000	0.272	0.000	0.481	**0.914**	0.204 ●	0.000 ●	**0.914**	**0.914**	0.000 ●	0.905	**0.914**	**0.914**
data_6_2	300	6	1.000	0.332	0.332	**1.000**	0.596	**1.000**	0.725 ●	**1.000**	**1.000**	0.983 ●	**1.000**	**1.000**	**1.000**	**1.000**
data_9_2	900	9	0.735	0.000	0.487	0.000	0.544	**0.731** ●	0.144 ●	0.000 ●	**0.731** ●	0.720	0.000 ●	0.729 ●	0.729 ●	0.714
flame	240	2	0.967	0.000	0.096	0.000	0.275	0.485 ●	0.950 ●	0.000 ●	0.485 ●	0.696 ●	0.000 ●	0.485 ●	0.485 ●	**0.967**
flamesize5	1240	6	0.963	0.000	0.187	0.000	0.518	0.489 ●	0.209 ●	0.000 ●	0.489 ●	0.489 ●	0.000 ●	0.489 ●	0.489 ●	**0.956**
fourty	1000	40	1.000	0.880	0.880	0.926	0.944	**1.000**	0.880 ●	**1.000**	**1.000**	0.926 ●	0.926 ●	**1.000**	**1.000**	**1.000**
inside	600	2	1.000	0.093	0.093	**1.000**	0.545	0.587 ●	**1.000**	**1.000**	0.566 ●	0.590 ●	**1.000**	0.153 ●	0.129 ●	**1.000**
long1	1000	2	1.000	0.107	0.107	**1.000**	0.273	0.375 ●	**1.000**	**1.000**	**1.000**	0.514 ●	0.998 ●	**1.000**	**1.000**	**1.000**
long2	1000	2	1.000	0.100	0.100	**1.000**	0.234	0.292 ●	**1.000**	**1.000**	**1.000**	0.687 ●	0.998 ●	**1.000**	**1.000**	**1.000**
long4	4000	8	0.960	0.000	0.407	0.000	0.572	0.589 ●	0.837 ●	0.000 ●	0.422 ●	0.485 ●	0.000 ●	0.418 ●	0.416 ●	**0.944**
longsquare	900	6	0.995	0.275	0.382	0.275	0.680	0.275 ●	0.567 ●	0.275 ●	0.275 ●	0.332 ●	0.275 ●	0.275 ●	0.275 ●	**0.992**
moons3	1000	2	0.996	0.000	0.086	0.000	0.211	0.191 ●	0.854 ●	0.000 ●	0.187 ●	0.303 ●	0.000 ●	0.190 ●	0.190 ●	**0.992**
moons5	1000	2	0.984	0.000	0.089	0.000	0.236	0.338 ●	**0.984**	0.000 ●	0.171 ●	0.340 ●	0.000 ●	0.172 ●	0.172 ●	**0.984**
multidist	3012	11	0.745	0.398	0.556	0.398	0.614	0.603 ●	0.398 ●	0.398 ●	0.607 ●	0.447 ●	0.398 ●	0.598 ●	0.592 ●	**0.713**
orange	400	2	1.000	0.079	0.079	**1.000**	0.249	**1.000**	**1.000**	**1.000**	0.543 ●	0.796 ●	0.995 ●	0.548 ●	0.548 ●	**1.000**
part2	417	2	1.000	0.168	0.168	**1.000**	0.375	0.296 ●	**1.000**	**1.000**	0.209 ●	0.783 ●	0.995 ●	0.226 ●	0.214 ●	**1.000**
r15	600	15	0.991	0.264	0.730	0.264	0.859	**0.991**	0.706 ●	0.264 ●	**0.991**	**0.991**	0.264 ●	**0.991**	**0.991**	0.989
sizes1	1000	4	0.958	0.000	0.230	0.000	0.451	**0.958**	0.239 ●	0.000 ●	**0.958**	**0.958**	0.001 ●	**0.958**	**0.958**	**0.958**
sizes3	1000	4	0.974	0.000	0.141	0.000	0.310	**0.974** ●	0.001 ●	0.000 ●	0.970 ●	**0.974**	0.000 ●	0.973	0.971	0.973
sizes5	1000	4	0.965	0.000	0.083	0.000	0.252	0.956	0.113 ●	0.000 ●	**0.960**	0.954	0.000 ●	0.955	0.954	0.954
smile1	1000	4	1.000	0.671	0.676	**1.000**	0.793	0.740 ●	**1.000**	**1.000**	0.681 ●	0.831 ●	0.999 ●	0.684 ●	0.681 ●	**1.000**
spiral	1000	2	1.000	0.105	0.106	**1.000**	0.524	0.106 ●	0.532 ●	**1.000**	0.106 ●	0.830 ●	0.998 ●	0.116 ●	0.107 ●	**1.000**
spiralsdata52	562	8	0.728	0.000	0.435	0.000	0.509	0.277 ●	0.649 ●	0.000 ●	0.277 ●	0.277 ●	0.000 ●	0.277 ●	0.277 ●	**0.704**
spiralsdata92	1212	12	0.694	0.000	0.626	0.000	0.597	0.106 ●	0.601 ●	0.000 ●	0.106 ●	0.106 ●	0.000 ●	0.106 ●	0.106 ●	**0.673**
spiralsizes5	2000	6	0.986	0.000	0.255	0.000	0.666	0.685 ●	0.780 ●	0.000 ●	0.671 ●	0.780 ●	0.000 ●	0.312 ●	0.308 ●	**0.985**
spiralsquare	1500	6	0.999	0.261	0.264	0.800	0.674	0.290 ●	0.641 ●	0.800 ●	0.290 ●	0.894 ●	0.800 ●	0.291 ●	0.290 ●	**0.998**
square1	1000	4	0.976	0.000	0.286	0.000	0.490	**0.976**	0.489 ●	0.000 ●	**0.976**	**0.976**	0.000 ●	**0.976**	**0.976**	**0.976**
square2	1000	4	0.906	0.000	0.271	0.000	0.527	0.901	0.462 ●	0.000 ●	0.901	0.901	0.000 ●	**0.902**	0.900	0.901
triangle1	1000	4	1.000	0.271	0.271	**1.000**	0.661	**1.000**	0.299 ●	**1.000**	**1.000**	0.982 ●	**1.000**	**1.000**	**1.000**	**1.000**
triangle2	1000	4	0.980	0.000	0.199	0.000	0.433	**0.980**	0.244 ●	0.000 ●	**0.980**	**0.980**	0.000 ●	**0.980**	**0.980**	**0.980**
twenty	1000	20	1.000	0.702	0.702	0.949	0.854	**1.000**	0.708 ●	**1.000**	**1.000**	**1.000**	0.949 ●	**1.000**	**1.000**	**1.000**
twodiamonds	800	2	0.998	0.000	0.096	0.000	0.307	**0.998**	0.250 ●	0.000 ●	**0.998**	0.995	0.000 ●	0.995	0.995	0.995
Overall			0.995	0.073	0.253	0.269	0.555	0.697 ●	0.674 ●	0.275 ●	0.710 ●	0.808 ●	0.269 ●	0.711 ●	0.707 ●	**0.993**

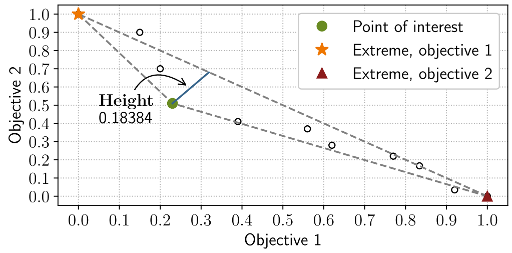


FIGURE 37. Feature TRIHE is given by the height of the triangle formed by the PFA member under consideration and the extreme points of the PFA.

by the extreme points of the PFA and the candidate solution being characterized. Feature TRIHE is given by the height of such a triangle (considering as the base of the triangle the line connecting the extreme points of the PFA, see Figure 37). If the solution considered is beyond the zone of interest (refer to the definition of feature ZINT in Appendix A-A), the negative of the computed triangle's height is used as the value of feature TRIHE.

D. CATEGORY 4: FEATURES DESCRIBING GLOBAL ASPECTS OF THE PFA

Features in the fourth category capture aspects of the PFA as a whole. That is, their computation involves the full set of solutions in the PFA. Therefore, it is worth noting that the value of these features is invariant across the feature vectors extracted for all solutions which are members of the same PFA (unlike features in the three previous categories). The following 22 features are defined within this category:

- **Cardinality of the PFA (CARD).** As the name of this feature suggests, it refers to the total number of candidate clusterings in the PFA. In the specific case of this study, the maximum cardinality is 100, which corresponds to the population size used by algorithm Δ-*MOCK* during PFA generation. However, given that the PFA consists of nondominated solutions only, it is possible that it contains fewer solutions in some cases.
- **Minimum objective values in the PFA (MIN1, MIN2).** Features MIN1 and MIN2 are given by the minimum

TABLE 3. Detailed results for Experiment 2, presented separately for the 40 synthetic clustering problems. The median ARI of the final solutions selected by the decision-making approaches analyzed is shown, highlighting the best performance scored for each problem. Results for all reference methods considered (except baselines BEST, WORST, EXT1, EXT2, and RAND) are marked • to indicate that a statistically significant difference is observed with respect to MLDM.

Dataset	N	k	BEST	WORST	EXT1	EXT2	RAND	SIL	DB	DUNN	CH+DB+SIL	MOCK	SUMO	LWEA	LWGP	MLDM
atom	800	2	1.000	0.528	0.528	**1.000**	0.606	0.548 •	**1.000**	**1.000**	0.543 •	0.563 •	0.995 •	0.545 •	0.542 •	**1.000**
blobs1	1000	5	1.000	0.503	0.506	0.614	0.779	0.997 •	0.526 •	0.614 •	0.997 •	**1.000 •**	0.614 •	**1.000 •**	**1.000 •**	0.963
blobs2	1000	5	0.990	0.374	0.435	0.374	0.654	0.975 •	0.590 •	0.374 •	0.981	**0.990 •**	0.374 •	**0.990 •**	**0.990 •**	0.985
blobs3	1000	10	0.996	0.047	0.790	0.047	0.895	**0.996 •**	0.787 •	0.808 •	**0.996 •**	0.808 •	0.047 •	**0.996 •**	**0.996 •**	0.984
chainlink	1000	2	1.000	0.086	0.086	**1.000**	0.197	0.144 •	0.687 •	**1.000**	0.098 •	0.256 •	**1.000**	0.102 •	0.102 •	**1.000**
circles1	1000	2	1.000	0.070	0.070	**1.000**	0.179	0.173 •	**1.000 •**	**1.000 •**	0.075 •	0.697	**1.000 •**	0.079 •	0.079 •	0.697
circles2	1000	2	0.692	0.000	0.091	0.000	0.160	0.136 •	0.327 •	0.000 •	0.109 •	**0.624 •**	0.000 •	0.109 •	0.109 •	0.160
data_4_3	400	4	1.000	0.246	0.251	**1.000**	0.633	**1.000**	0.507 •	**1.000**	**1.000**	**1.000**	0.997 •	**1.000**	**1.000**	**1.000**
data_5_2	250	5	0.914	0.000	0.266	0.000	0.471	**0.914 •**	0.204 •	0.000 •	**0.914 •**	**0.914 •**	0.000 •	**0.914 •**	**0.914 •**	0.447
data_6_2	300	6	1.000	0.320	0.320	**1.000**	0.616	**1.000 •**	0.750 •	**1.000 •**	**1.000 •**	**1.000 •**	**1.000 •**	**1.000 •**	**1.000 •**	0.967
data_9_2	900	9	0.747	0.000	0.489	0.000	0.573	0.737 •	0.144 •	0.000 •	0.737 •	0.735 •	0.000 •	0.736 •	**0.738 •**	0.145
flame	240	2	0.967	0.000	0.095	0.000	0.283	0.479	**0.950 •**	0.000 •	0.477	0.489	0.000 •	0.487	0.487	0.484
flamesize5	1240	6	0.963	0.000	0.197	0.000	0.497	0.489 •	0.487 •	0.000 •	0.489 •	0.489 •	0.000 •	0.489 •	0.489 •	**0.815**
fourty	1000	40	1.000	0.891	0.891	0.926	0.946	**1.000 •**	0.891 •	**1.000 •**	**1.000 •**	0.926 •	0.926 •	**1.000 •**	**1.000 •**	0.999
inside	600	2	1.000	0.094	0.094	**1.000**	0.545	0.577 •	**1.000**	**1.000**	0.565 •	0.589 •	**1.000**	0.127 •	0.119 •	**1.000**
long1	1000	2	1.000	0.104	0.104	**1.000**	0.260	0.375 •	**1.000**	**1.000**	**1.000**	0.516 •	0.998 •	**1.000**	**1.000**	**1.000**
long2	1000	2	1.000	0.099	0.100	**1.000**	0.256	0.291 •	**1.000**	**1.000**	**1.000**	0.686 •	0.998 •	**1.000**	**1.000**	**1.000**
long4	4000	8	0.953	0.000	0.396	0.000	0.589	0.589 •	**0.817**	0.000 •	0.413 •	0.485 •	0.000 •	0.420 •	0.419 •	**0.817**
longsquare	900	6	0.995	0.275	0.387	0.275	0.675	0.275 •	0.567 •	0.275 •	0.275 •	0.332 •	0.275 •	0.275 •	0.275 •	**0.870**
moons3	1000	2	0.996	0.000	0.092	0.000	0.234	0.209 •	0.854 •	0.000 •	0.187 •	**0.996 •**	0.000 •	0.190 •	0.188 •	0.988
moons5	1000	2	0.984	0.000	0.091	0.000	0.212	0.334 •	0.782 •	0.000 •	0.171 •	0.339 •	0.000 •	0.172 •	0.172 •	**0.960**
multidist	3012	11	0.715	0.398	0.558	0.398	0.609	0.600 •	0.398 •	0.398 •	0.600 •	0.447 •	0.398 •	0.605 •	0.605 •	**0.634**
orange	400	2	1.000	0.081	0.081	**1.000**	0.288	**1.000 •**	**1.000 •**	**1.000 •**	0.521 •	0.796 •	0.995	0.576 •	0.561 •	0.995
part2	417	2	1.000	0.155	0.155	**1.000**	0.348	0.290 •	**1.000**	**1.000**	0.247 •	0.783 •	0.996 •	0.242 •	0.222 •	**1.000**
r15	600	15	0.991	0.264	0.703	0.264	0.837	**0.991 •**	0.697 •	0.264 •	**0.991 •**	**0.991 •**	0.264 •	**0.991 •**	**0.991 •**	0.862
sizes1	1000	4	0.958	0.000	0.227	0.000	0.472	**0.958**	0.000 •	0.000 •	**0.958**	**0.958**	0.000 •	**0.958**	**0.958**	**0.958**
sizes3	1000	4	0.974	0.000	0.140	0.000	0.347	**0.974 •**	0.176 •	0.000 •	0.971	**0.974**	0.003 •	**0.974**	0.971	0.971
sizes5	1000	4	0.967	0.000	0.078	0.000	0.252	0.960 •	0.125 •	0.000 •	**0.962 •**	0.950	0.000 •	0.954	0.960	0.953
smile1	1000	4	1.000	0.677	0.677	**1.000**	0.797	0.739 •	**1.000 •**	**1.000 •**	0.680 •	0.812 •	0.999	0.683 •	0.681 •	0.999
spiral	1000	2	1.000	0.097	0.097	**1.000**	0.414	0.101 •	0.532 •	**1.000 •**	0.101 •	0.829 •	0.998	0.108 •	0.099 •	0.998
spiralsdata52	562	8	0.720	0.000	0.437	0.000	0.525	0.277 •	**0.649 •**	0.000 •	0.277 •	0.277 •	0.000 •	0.277 •	0.277 •	0.617
spiralsdata92	1212	12	0.701	0.000	**0.644**	0.000	0.568	0.106 •	0.584 •	0.000 •	0.106 •	0.106 •	0.000 •	0.106 •	0.106 •	0.101
spiralsizes5	2000	6	0.986	0.000	0.264	0.000	0.690	0.687 •	0.785 •	0.000 •	0.688 •	0.780 •	0.000 •	0.352 •	0.307 •	**0.981**
spiralsquare	1500	6	0.999	0.260	0.260	0.800	0.665	0.291 •	0.644 •	0.800	0.289 •	**0.894 •**	0.800	0.292 •	0.288 •	0.800
square1	1000	4	0.976	0.000	0.289	0.000	0.514	**0.976**	0.489 •	0.000 •	**0.976**	**0.976**	0.000 •	**0.976**	**0.976**	**0.976**
square2	1000	4	0.911	0.000	0.258	0.000	0.496	**0.910**	0.453 •	0.000 •	**0.910**	**0.910**	0.000 •	**0.910**	**0.910**	0.909
triangle1	1000	4	1.000	0.272	0.272	**1.000**	0.714	**1.000 •**	0.325 •	**1.000 •**	**1.000 •**	0.982 •	**1.000 •**	**1.000 •**	**1.000 •**	0.889
triangle2	1000	4	0.980	0.000	0.203	0.000	0.445	0.979 •	0.246 •	0.000 •	**0.980 •**	**0.980 •**	0.000 •	**0.980 •**	**0.980 •**	0.967
twenty	1000	20	1.000	0.706	0.712	0.949	0.862	**1.000 •**	0.716 •	**1.000 •**	**1.000 •**	**1.000 •**	0.949 •	**1.000 •**	**1.000 •**	0.987
twodiamonds	800	2	0.995	0.000	0.092	0.000	0.292	**0.995 •**	0.242 •	0.000 •	**0.995 •**	**0.995 •**	0.000 •	**0.995 •**	**0.995 •**	0.975
Overall			0.995	0.074	0.250	0.269	0.553	0.699 •	0.679 •	0.275 •	0.711 •	0.808 •	0.269 •	0.718 •	0.708 •	**0.967**

(best) value scored for the first and second objectives, respectively, considering all PFA members.

- **Maximum objective values in the PFA (MAX1, MAX2).** Contrary to features MIN1 and MIN2, features MAX1 and MAX2 are given by the maximum (worst) values yielded by any of the PFA members for the first and second objective functions, respectively.
- **Average objective values in the PFA (AVG1, AVG2).** These features are computed as the arithmetic mean of the values scored for the two objective functions, considering all candidate partitions in the PFA.
- **Average normalized objective values (NAVG1, NAVG2).** These features are defined equivalently to the AVG1 and AVG2 features described above. However, NAVG1 and NAVG2 are computed after normalizing the PFA to range [0, 1] independently in each dimension.
- **Hypervolume of the PFA (HVOL).** The hypervolume is one of the most widely used indicators to assess the performance of evolutionary multiobjective optimizers [53]. It is able to simultaneously evaluate both aspects of convergence and diversity of the PFAs produced by these algorithms. Briefly, this indicator is defined as the volume of the region of the objective space, delimited by a reference point, which is dominated by the solutions in the PFA (see Figure 38). Feature HVOL is given by the value for such an indicator, which in this study is computed for the normalized PFA and using always a fixed reference point, namely, (1.01, 1.01).

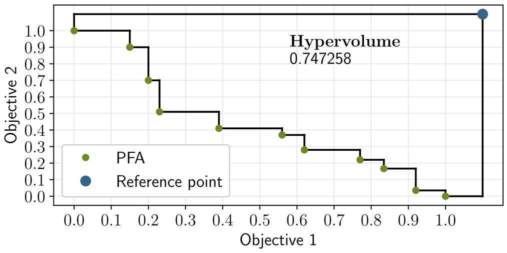


FIGURE 38. Feature HVOL is given by the hypervolume of the PFA.

- **Minimum and maximum hypervolume contribution (MINHV, MAXHV).** In Appendix A-C, we describe

TABLE 4. Detailed results for Experiment 3, presented separately for the 10 rea-world clustering problems. The median ARI of the final solutions selected by the decision-making approaches analyzed is shown, highlighting the best performance scored for each problem. Results for all reference methods considered (except baselines BEST, WORST, EXT1, EXT2, and RAND) are marked • to indicate that a statistically significant difference is observed with respect to MLDM.

Dataset	N	k	BEST	WORST	EXT1	EXT2	RAND	SIL	DB	DUNN	CH+DB+SIL	MOCK	SUMO	LWEA	LWGP	MLDM
Banknote	1372	2	0.755	0.000	0.094	0.000	0.178	0.104 •	0.028 •	0.214 •	0.102 •	0.000 •	0.000 •	0.115 •	0.109 •	**0.336**
Breast	683	2	0.863	0.000	0.284	0.000	0.724	**0.863** •	0.295 •	0.000 •	**0.863** •	0.000 •	0.000 •	**0.863** •	**0.863** •	0.809
Digits	5620	10	0.781	0.000	**0.759**	0.000	0.376	0.750 •	0.380 •	0.000 •	0.000 •	0.178 •	0.000 •	0.609	0.623 •	0.389
Ecoli	336	8	0.770	0.000	0.436	0.000	0.642	0.715 •	0.038 •	0.000 •	0.715 •	**0.741** •	0.000 •	0.715 •	0.715 •	0.722
Iris	150	3	0.660	0.000	0.207	0.000	0.331	**0.568** •	0.276 •	0.000 •	**0.568** •	**0.568** •	0.000 •	**0.568** •	**0.568** •	0.523
Landsat	6435	6	0.605	0.000	0.538	0.000	0.517	0.081 •	0.345 •	0.000 •	0.082 •	0.082 •	0.004 •	0.083 •	0.083 •	**0.575**
Palmer	333	3	0.536	0.157	0.157	0.377	0.340	0.377 •	0.266 •	0.304 •	0.377 •	0.464 •	0.377 •	0.377 •	0.377 •	**0.514**
Seeds	210	3	0.776	0.000	0.193	0.000	0.461	0.497 •	0.223 •	0.000 •	0.497 •	0.000 •	0.000 •	0.478 •	0.478 •	**0.577**
Thyroid	215	3	0.787	0.000	0.166	0.000	0.438	0.031 •	0.259 •	0.000 •	0.202 •	0.031 •	0.007 •	0.202 •	0.202 •	**0.544**
Wine	178	3	0.854	0.000	0.221	0.000	0.529	0.802 •	0.466 •	0.000 •	0.802	0.000 •	0.000 •	0.802 •	0.795 •	**0.808**
Overall			0.770	0.000	0.211	0.000	0.431	0.496 •	0.265 •	0.000 •	0.461 •	0.083 •	0.000 •	0.472 •	0.472 •	**0.569**

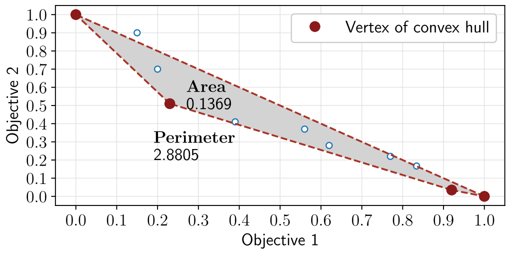


FIGURE 39. Features ACVX and PCVX are respectively defined as the area and perimeter of the convex hull of the PFA (computed only for points in the zone of interest, see description of feature ZINT and Figure 22).

feature CHVOL as the contribution of a specific PFA member to the hypervolume indicator. Features MINHV and MINHV respectively refer to the minimum and maximum individual contributions in the entire PFA.

- **Area and perimeter of convex hull (ACVX, PCVX).** These features, as illustrated in Figure 39, are computed as the area and the perimeter of the convex hull of the PFA, respectively. Note, however, that the computation of the convex hull disregards PFA members outside the zone of interest (see feature ZINT in Appendix A-A).
- **Cardinality of the convex hull (CCVX).** This feature is given by the total number of vertices in the convex hull. As discussed before, convex hull computation considers only PFA members inside the zone of interest (refer to the description of feature INCVX in Appendix A-C).
- **Minimum, maximum, and average value of k (MINK, MAXK, AVGK).** Considering that the PFA may contain candidate partitions with a range of values for k, features MINK, MAXK, and AVGK refer to the minimum, maximum, and average values of this parameter across the full set of PFA members, respectively.
- **Mode of the k values in the PFA (MODK).** Adhering to the definition of mode in statistics, feature MODK is computed as the k value that appears most frequently across the clustering solutions in the PFA.
- **Number of unique k values in the PFA (UNIK).** This feature simply reflects the total number of distinct values for parameter k in the PFA's candidate partitions.
- **Percentage of solutions favoring a particular objective (PTEND1, PTEND2).** As explained in Appendix A-A and Figure 21 for feature TEND, a PFA member may exhibit a tendency towards favoring one objective over the other depending on its location. Features PTEND1 and PTEND2 indicate the percentage of the PFA members showing a tendency towards the first and second objective functions, respectively.

E. CATEGORY 5: FEATURES DESCRIBING THE CLUSTERING PROBLEM BEING SOLVED

In this last category, we consider features that reflect aspects of the particular clustering problem at hand. Similar to the features in the fourth category (Section A-D), the value of these features is fixed for all members of the given PFA (it is, indeed, fixed for all solutions in all PFAs generated for the same problem). Two features are included in this category:

- **Size of the dataset (DATA).** This feature refers to the number of samples in the dataset under consideration.
- **Dimensionality of the dataset (DIME).** Feature DIME captures the total number of dimensions (i.e., features or variables) in the clustering problem.

APPENDIX B
DETAILED RESULT TABLES

This appendix includes detailed results for the main experiments of this study. Tables 2, 3, and 4 present the results for individual problems and the findings of the statistical significance analysis for Experiments 1, 2, and 3, respectively.

REFERENCES

[1] J. Handl and J. Knowles, "Evolutionary multiobjective clustering," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Cham, Switzerland: Springer, 2004, pp. 1081–1091.

[2] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "A survey of multiobjective evolutionary clustering," *ACM Comput. Surveys*, vol. 47, no. 4, pp. 61:1–61:46, 2015.

[3] R. A. Khurma and I. Aljarah, "A review of multiobjective evolutionary algorithms for data clustering problems," in *Evolutionary Data Clustering: Algorithms and Applications*. Singapore: Springer, 2021, pp. 177–199.

[4] J. Handl and J. Knowles, "Exploiting the trade-off—The benefits of multiple objectives in data clustering," in *Evolutionary Multi-Criterion Optimization*. Guanajuato, Mexico: Springer, 2005, pp. 547–560.

[5] V. Pareto, *Cours d'Economie Politique*. Geneva, Switzerland: Droz, 1896.

[6] S. Bandyopadhyay, A. Mukhopadhyay, and U. Maulik, "An improved algorithm for clustering gene expression data," *Bioinformatics*, vol. 23, no. 21, p. 2859, 2007.

[7] A. José-García, J. Handl, W. Gómez-Flores, and M. Garza-Fabre, "An evolutionary many-objective approach to multiview clustering using feature and relational data," *Appl. Soft Comput.*, vol. 108, Sep. 2021, Art. no. 107425.

[8] A. Garcia-Piquer, A. Sancho-Asensio, A. Fornells, E. Golobardes, G. Corral, and F. Teixidó-Navarro, "Toward high performance solution retrieval in multiobjective clustering," *Inf. Sci.*, vol. 320, pp. 12–25, Nov. 2015.

[9] S. Zhu, L. Xu, and E. D. Goodman, "Evolutionary multi-objective automatic clustering enhanced with quality metrics and ensemble strategy," *Knowl.-Based Syst.*, vol. 188, Jan. 2020, Art. no. 105018.

[10] J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 56–76, Feb. 2007.

[11] N. Matake, T. Hiroyasu, M. Miki, and T. Senda, "Multiobjective clustering with automatic k-determination for large-scale data," in *Proc. 9th Annu. Conf. Genetic Evol. Comput. (GECCO)*, London, U.K., 2007, pp. 861–868.

[12] S. Shirakawa and T. Nagao, "Evolutionary image segmentation based on multiobjective clustering," in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 2466–2473.

[13] A. Gupta, S. Datta, and S. Das, "Fuzzy clustering to identify clusters at different levels of fuzziness: An evolutionary multiobjective optimization approach," *IEEE Trans. Cybern.*, vol. 51, no. 5, pp. 2601–2611, May 2021.

[14] K. Deb and S. Gupta, "Understanding knee points in bicriteria problems and their implications as preferred solution principles," *Eng. Optim.*, vol. 43, no. 11, pp. 1175–1204, 2011.

[15] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "Multiobjective genetic clustering with ensemble among Pareto front solutions: Application to MRI brain image segmentation," in *Proc. 7th Int. Conf. Adv. Pattern Recognit.*, Feb. 2009, pp. 236–239.

[16] X. Qian, X. Zhang, L. Jiao, and W. Ma, "Unsupervised texture image segmentation using multiobjective evolutionary clustering ensemble algorithm," in *Proc. IEEE Congr. Evol. Comput., IEEE World Congr. Comput. Intell.*, Jun. 2008, pp. 3561–3567.

[17] A. Mukhopadhyay and U. Maulik, "Unsupervised pixel classification in satellite imagery using multiobjective fuzzy clustering combined with SVM classifier," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 4, pp. 1132–1138, Apr. 2009.

[18] A. Mukhopadhyay, U. Maulik, and S. Bandyopadhyay, "Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 991–1005, Oct. 2009.

[19] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *J. Intell. Inf. Syst.*, vol. 17, no. 2, pp. 107–145, Dec. 2001.

[20] A. José-García and W. Gómez-Flores, "Automatic clustering using nature-inspired metaheuristics: A survey," *Appl. Soft Comput.*, vol. 41, pp. 192–213, Apr. 2016.

[21] A. José-García and W. Gómez-Flores, "A survey of cluster validity indices for automatic data clustering using differential evolution," in *Proc. Genetic Evol. Comput. Conf.*, New York, NY, USA, Jun. 2021, pp. 314–322.

[22] I. Aljarah, M. Habib, R. Nujoom, H. Faris, and S. Mirjalili, "A comprehensive review of evaluation and fitness measures for evolutionary data clustering," in *Evolutionary Data Clustering: Algorithms and Applications*, I. Aljarah, H. Faris, and S. Mirjalili, Eds. Singapore: Springer, 2021, pp. 23–71.

[23] J. Kleinberg, "An impossibility theorem for clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 15, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, MA, USA: MIT Press, 2003, pp. 463–470.

[24] D. H. Wolper and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[25] A. D. McCarthy, T. Chen, and S. Ebner, "An exact no free lunch theorem for community detection," in *Complex Networks and Their Applications VIII*, H. Cherifi, S. Gaito, J. F. Mendes, E. Moro, and L. M. Rocha, Eds. Cham, Switzerland: Springer, 2020, pp. 176–187.

[26] M. Garza-Fabre, J. Handl, and J. Knowles, "An improved and more scalable evolutionary approach to multiobjective clustering," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 515–535, Aug. 2018.

[27] M. Delattre and P. Hansen, "Bicriterion cluster analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-2, no. 4, pp. 277–291, Jul. 1980.

[28] S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay, "Multiobjective genetic clustering for pixel classification in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 5, pp. 1506–1511, May 2007.

[29] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1650–1654, Dec. 2002.

[30] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, no. 1, pp. 53–65, Jan. 1987.

[31] A. Mukhopadhyay, S. Bandyopadhyay, and U. Maulik, "Analysis of microarray data using multiobjective variable string length genetic fuzzy clustering," in *Proc. IEEE Congr. Evol. Comput.*, May 2009, pp. 1313–1319.

[32] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.

[33] M. Halkidi, M. Vazirgiannis, and Y. Batistakis, "Quality scheme assessment in the clustering process," in *Principles of Data Mining and Knowledge Discovery*, D. A. Zighed, J. Komorowski, and J. Żytkow, Eds. Berlin, Germany: Springer, 2000, pp. 265–276.

[34] T. Özyer, Y. Liu, R. Alhajj, and K. Barker, "Multi-objective genetic algorithm based clustering approach and its application to gene expression data," in *Proc. Adv. Inf. Syst.*, T. Yakhno, Ed. Berlin, Germany: Springer, 2005, pp. 451–461.

[35] Y. Liu, T. Özyer, and K. Barker, "Integrating multi-objective genetic algorithm and validity analysis for locating and ranking alternative clustering," *Informatica*, vol. 29, pp. 33–40, Jan. 2005.

[36] L. Hubert and J. Schultz, "Quadratic assignment as a general data analysis strategy," *Brit. J. Math. Stat. Psychol.*, vol. 29, no. 2, pp. 190–241, Nov. 1976.

[37] J. C. Dunn, "Well-separated clusters and optimal fuzzy partitions," *J. Cybern.*, vol. 4, no. 1, pp. 95–104, 2008.

[38] M. Halkidi and M. Vazirgiannis, "Clustering validity assessment: Finding the optimal partitioning of a data set," in *Proc. IEEE Int. Conf. Data Mining*, Jun. 2001, pp. 187–194.

[39] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Statist.*, vol. 3, no. 1, pp. 1–27, Jan. 1974.

[40] G. Corral, A. Garcia-Piquer, A. Orriols-Puig, A. Fornells, and E. Golobardes, "Analysis of vulnerability assessment results based on CAOS," *Appl. Soft Comput.*, vol. 11, no. 7, pp. 4321–4331, Oct. 2011.

[41] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multiobjective optimization," in *Parallel Problem Solving from Nature—PPSN VIII*, X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervós, J. A. Bullinaria, J. E. Rowe, P. Tiňo, A. Kabán, and H.-P. Schwefel, Eds. Berlin, Germany: Springer, 2004, pp. 722–731.

[42] K. Golalipour, E. Akbari, S. S. Hamidi, M. Lee, and R. Enayatifar, "From clustering to clustering ensemble selection: A review," *Eng. Appl. Artif. Intell.*, vol. 104, Sep. 2021, Art. no. 104388.

[43] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Mar. 2003.

[44] D. Huang, C.-D. Wang, and J.-H. Lai, "Locally weighted ensemble clustering," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1460–1473, May 2018.

[45] E. Chen and F. Wang, "Dynamic clustering using multi-objective evolutionary algorithm," in *Computational Intelligence and Security*, Y. Hao, J. Liu, Y. Wang, Y.-M. Cheung, H. Yin, L. Jiao, J. Ma, and Y.-C. Jiao, Eds. Berlin, Germany: Springer, 2005, pp. 73–80.

[46] G. N. Demir, A. S. Uyar, and S. Gündüz-Ögüdücü, "Multiobjective evolutionary clustering of web user sessions: A case study in web page recommendation," *Soft Comput.*, vol. 14, no. 6, pp. 579–597, 2010.

[47] C.-W. Tsai, W.-L. Chen, and M.-C. Chiang, "A modified multiobjective EA-based clustering algorithm with automatic determination of the number of clusters," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2012, pp. 2833–2838.

[48] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Statist. Assoc.*, vol. 66, no. 336, pp. 846–850, 1971.

[49] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[50] B. Singh, P. Sihag, and K. Singh, "Modelling of impact of water quality on infiltration rate of soil by random forest regression," *Model. Earth Syst. Environ.*, vol. 3, no. 3, pp. 999–1004, Sep. 2017.

[51] Y. Li, C. F. Zou, M. Berecibar, E. E. Nanini, J. C. W. Chan, P. van den Bossche, J. Van Mierlo, and N. Omar, "Random forest regression for online capacity estimation of lithium-ion batteries," *Appl. Energy*, vol. 232, pp. 197–210, Dec. 2018.

[52] R. Couronné, P. Probst, and A.-L. Boulesteix, "Random forest versus logistic regression: A large-scale benchmark experiment," *BMC Bioinf.*, vol. 19, no. 1, p. 270, Dec. 2018.

[53] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. D. Fonseca, ''Performance assessment of multiobjective optimizers: An analysis and review,'' *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[54] K. Shang, H. Ishibuchi, L. He, and L. M. Pang, ''A survey on the hypervolume indicator in evolutionary multiobjective optimization,'' *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 1–20, Feb. 2021.

[55] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, ''A fast and elitist multiobjective genetic algorithm: NSGA-II,'' *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Jan. 2002.

MARIO GARZA-FABRE received the M.Sc. and Ph.D. degrees in computer science from Cinvestav, Mexico, in 2009 and 2014, respectively. From 2015 to 2018, he worked as a Research Associate at The University of Manchester and Liverpool John Moores University, U.K. He joining Cinvestav in 2018, where he is currently an Associate Professor. His research interests include the analysis, design, and application of (meta-)heuristic optimization techniques.

AARÓN L. SÁNCHEZ-MARTÍNEZ received the B.S. degree in information technologies engineering from the Polytechnic University of Victoria, Mexico, in 2019, and the M.Sc. degree in engineering and computational technologies from Cinvestav, Mexico, in 2021. He is currently working as a Software Engineer in the private sector. His research interests include evolutionary computing and its application to challenging optimization problems.

EDWIN ALDANA-BOBADILLA received the M.Sc. degree in computer engineering and the Ph.D. degree in computer science from the Universidad Nacional Autónoma de México (UNAM), Mexico, in 2009 and 2015, respectively. Since 2015, he has been commissioned by the National Council of Science and Technology of Mexico (CONACyT) and Cinvestav as a Researcher. His expertise spans software engineering, database systems, digital electronic, machine learning, natural language processing, stochastic processes, and optimization.

RICARDO LANDA received the M.Sc. and Ph.D. degrees in electrical engineering from Cinvestav, Mexico, in 2002 and 2007, respectively. He was a Research Fellow with the School of Computer Science, University of Birmingham, U.K., from 2008 to 2009. He has been a Professor at Cinvestav, since 2009. He has worked on algorithms for constraint handling and multiobjective optimization based on differential evolution, evolutionary programming, and cultural algorithms. His research interests include large scale optimization and both single and multiobjective.