An Evolutionary Many-objective Approach to Multiview Clustering Using Feature and Relational Data

Adán José-García[a,*], Julia Handl[a], Wilfrido Gómez-Flores[b], Mario Garza-Fabre[b]

[a]*Decision and Cognitive Sciences Research Centre, Alliance Manchester Business School, The University of Manchester, Manchester, M15 6PB, United Kingdom*
[b]*Centro de Investigación y de Estudios Avanzados del IPN, Unidad Tamaulipas, 87130, Cd. Victoria, Tamaulipas, México*

Abstract

Many application domains involve the consideration of multiple data sources. Typically, each of these *data views* provides a different perspective of a given set of entities. Inspired by early work on multiview (supervised) learning, multiview algorithms for data clustering offer the opportunity to consider and integrate all this information in an unsupervised setting. In practice, some complex real-world problems may give rise to a handful or more data views, each with different reliability levels. However, existing algorithms are often limited to the consideration of two views only, or they assume that all the views have the same level of importance. Here, we describe the design of an evolutionary algorithm for the problem of multiview cluster analysis, exploiting recent advances in the field of evolutionary optimization to address settings with a larger number of views. The method is capable of considering views that are represented in the form of distinct feature sets, or distinct dissimilarity matrices, or a combination of the two. Our experimental results on standard (including real-world) benchmark datasets confirm that the adoption of a many-objective evolutionary algorithm addresses limitations of previous work, and can easily scale to settings with four or more data views. The final highlight of our paper is an illustration of the potential of the approach in an application to breast lesion classification.

Keywords: Data Clustering, Multiview Clustering, Evolutionary Clustering, Evolutionary Multiobjective Clustering

1. Introduction

Data clustering is an unsupervised learning technique aimed at discovering homogeneous groups of unlabeled data objects according to measured intrinsic characteristics [1]. It presents a prevalent approach to data analysis in different scientific fields, such as computer vision, bioinformatics, and marketing [2, 3, 4], and the array of available methods range from statistical approaches over deep learning approaches and meta-heuristics to various hybrid approaches [2].

Many application areas, e.g., in bioinformatics or information retrieval, require the grouping of data characterized by multiple feature sets and/or multiple relational descriptions, resulting from the application of different dissimilarity functions [5, 6]. In the first case, the final clustering is obtained from the consensus of different feature spaces (i.e., feature data) using a fixed dissimilarity function. For example, in breast ultrasound image analysis, different sets of quantitative features can be extracted to describe the shape, orientation, margin, echo pattern, and posterior features of masses to perform the lesion classification [7]. On the other hand, in the second case, clustering is obtained from different proximity functions that enhance dissimilarity relationships (i.e., relational data) [8]. In this case, multiple dissimilarity matrices can be derived using conceptually different proximity measures such as the Euclidean distance, the maximum edge distance (MED) [9], and the Cosine distance. Finally, there are scenarios in which definitions of feature spaces are not straightforward, or only relation information about entities is available (e.g., in protein or document comparison), and will take the form of multiple dissimilarity matrices. In all the above scenarios, it is beneficial to integrate the available multiple information sources to generate more accurate and robust clustering results.

In a strict sense, the term "multiview clustering" refers to algorithms that can utilize multiple feature spaces, which describe distinct points of view of a phenomenon. However, the term "multiview" can be extended to account for the collaborative role of different dissimilarity matrices that map a single feature space to multiple views characterized by different relational descriptions [6]. Therefore, we consider a multiview clustering algorithm to be any algorithm that is sufficiently versatile to cope with different representations of the same data, including multiple feature sets and multiple relational descriptions, and can integrate these pieces of information in order to find clusters that are consistent across the different views. Furthermore, it should make no assumptions about the importance of different views. This is particularly relevant in situations in which individual views are incommensurable, i.e., where merging all views to form a single dataset is inconvenient due to their unique properties, or where views might present varying levels of reliability [10, 11]. Finally, a robust multiview clustering approach needs to be able to

*Corresponding author
Email addresses: adan.jose@cinvestav.mx (Adán José-García), julia.handl@manchester.ac.uk (Julia Handl), wgomez@cinvestav.mx (Wilfrido Gómez-Flores), mario.garza@cinvestav.mx (Mario Garza-Fabre)

reliably scale to more than two views (a property we call "many-view", borrowing from the term "many-objective" first introduced in the multiobjective optimization field [12]).

This paper describes a novel multiview data clustering approach, called MVMC, based on multiobjective evolutionary optimization, where the multiview property refers to the availability of multiple feature sets and/or multiple relational descriptions. The approach takes advantage of many-objective optimization concepts [12] to explore a range of (Pareto optimal) trade-offs, while scaling to settings with three or more data views, thus overcoming two of the most frequent limitations of existing multiview clustering methods. The suitability and performance of MVMC are investigated for a range of traditional benchmark datasets, involving experiments with multiple feature sets and multiple relational matrices, as well as for the problem of classifying breast lesions on ultrasound images. The major contributions of this paper are:

- A many-objective approach to multiview data clustering is introduced. This approach exploits the benefits of complementary information sources, taken from multiple feature sets or multiple relationships, to maintain or increase clustering performance as the number of data views increases.

- Our algorithm uses an encoding based on cluster prototypes, which provides good scalability to large datasets. Furthermore, we propose an innovative decoding strategy that directly exploits mechanisms of the underlying optimizer (scalarizing vectors) in mapping a set of cluster prototypes to an actual partition. The use of the reference vector allows us to ensure that no bias with regards to any particular view is introduced at the decoding stage.

- An unsupervised clustering selection method is proposed to choose the most suitable clustering solution from the Pareto front approximations produced by MVMC. This method is based on an established internal validation technique but utilizes the information contained in individual reference vectors to weigh all data views appropriately.

- The proposed approach is successfully applied to a breast lesions classification problem, which confirms that MVMC provides meaningful advantages in practice.

The remainder of this paper is organized as follows. First, Section 2 introduces the necessary background and related work. Section 3 describes in detail the proposed MVMC algorithm. Section 4 describes the settings, reference methods, and performance metrics used in our experiments. The experimental results on synthetic and real-world datasets are presented in Section 5. Section 6 presents the results obtained on multiview datasets. Section 7 presents the results obtained on the breast lesions classification problem. Finally, Section 8 presents the discussion, and Section 9 provides the conclusions and potential directions for future work.

2. Background and Related Work

This section introduces basic concepts that are essential for understanding this work and discusses relevant related work.

2.1. Multiview Data Clustering

A crisp clustering is the partitioning of N data objects into K mutually disjoint subsets [1]. Formally, let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a set of N objects to be partitioned into K clusters $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$, such that the following three conditions are satisfied: $\mathbf{c}_i \neq \emptyset$; $\mathbf{c}_1 \cup \ldots \cup \mathbf{c}_K = \mathbf{X}$; and $\mathbf{c}_i \cap \mathbf{c}_j = \emptyset$ for $i, j = 1, \ldots, K$ and $i \neq j$. The quality of a given partition can be assessed by defining an objective function over the objects' features or over relational information between the objects, e.g., cluster analysis often aims to identify partitions that minimize within-cluster variation.

The multiview clustering (MvC) solutions considered in this paper follow the same definition. However, the quality assessment for a given partition involves the consideration of multiple sources of information or data views. Since the individual views stem from various types of measurements, they may have different statistical properties and support different partitions [13]. Thus, MvC provides a principled approach for integrating multiple views to generate high-quality partitions that optimally trade-off the support provided from these different data sources.

Recent research has reported some first steps towards exploiting the intrinsic multi-criterion nature of MvC [14, 11, 15, 16, 17, 8]. In MvC, data views are available either in the form of *multiple feature sets* or as *multiple dissimilarity matrices* [5, 6, 18]. Representative approaches in the first category are described below. Wang et al. [14] proposed a multiobjective spectral clustering formulation to MvC, which requires the computation of the kernel construction and eigenvector decomposition. Although this method exhibited good performance in problems with two views, it requires high computational resources, and the extension to more than two data views was not analyzed. Jiang et al. [11] used multiobjective evolutionary optimization to approximate the set of optimal trade-off solutions, seeing each view as an independent objective and describing the solutions in the form of cluster centroids. However, a suitable mechanism for a 1-1 mapping each of these centroids to a candidate partition, whilst preserving the multi-criterion nature of the problem, was not described. During the search, Jiang et al. [11] separately perform the mapping (and subsequent evaluation) along each view, which is equivalent to a 1-m mapping, where m is the number of views. The resulting m objective values are then aggregated in a single vector, reflecting an ideal set of values rather than a trade-off achievable by a single partition. This overestimation is likely to impact on the algorithm's ability to approximate the true Pareto front. More recently, Saha et al. [15] proposed a multi-criterion approach considering $m + 1$ optimization criteria: m criteria to evaluate clustering quality on the m individual views, and an additional criterion to measure the agreement among the data views.

While the methods mentioned above are limited to applications where feature sets are readily available, MvC approaches

based on the use of multiple relational descriptions or dissimilarity matrices (usually derived by applying different distance functions) have also been proposed [5, 16, 19, 18]. The use of dissimilarity matrices as data views is of particular importance in scenarios where definitions of feature spaces are not available/straightforward, or a range of complementary measures can be derived. Even when well-defined feature spaces are available, the choice of a dissimilarity measure can be crucial to identifying particular types of cluster structures, impacting the clustering performance [20]. For instance, it is known that the Euclidean distance is more suitable for spherically-shaped clusters. Similarly, the maximum edge distance (MED) [9] excels at identifying irregular-shaped components [21, 8], and the Cosine distance is more suitable when orientation between patterns is more relevant than their magnitude [22].

Clustering algorithms usually require selecting a single dissimilarity measure such as the Euclidean, MED, or Cosine distance. The task of selecting the best dissimilarity measure for a given dataset or combining multiple available measures, is typically addressed early on in the data analysis pipeline and can represent a significant challenge. One approach is to assign different weights to different measures [19, 5], but the appropriate weights are difficult to determine without prior knowledge of the types of structures present in the data and the reliability of the information provided by these measures. For this reason, Liu et al. [16] presented a multiobjective evolutionary algorithm (based on NSGA-II) that simultaneously considered two different distance functions. Each individual is represented using a label-based encoding of size N (number of data points), and it is evaluated using the intra-cluster variance concerning both distance measures. Recently, Liu et al. [17] extended this work by proposing a fuzzy clustering approach based on a multiobjective differential evolution algorithm. In this approach, a centroid-based codification is used to represent the clustering solutions. However, the evaluation of these approaches has been limited to two-view data, and scalability to more than three views (and therefore objectives) will be limited for those approaches relying on Pareto dominance [23].

2.2. Multi- and Many-objective Optimization

Without loss of generality, a multiobjective optimization problem can be stated as follows [12]:

$$\begin{array}{ll} \text{minimize} & \mathbf{F}(\mathbf{z}) = (f_1(\mathbf{z}), \ldots, f_m(\mathbf{z}))^T \\ \text{subject to} & \mathbf{z} \in \Omega \end{array}, \qquad (1)$$

where Ω is the feasible decision space, $\mathbf{z} = (z_1, \ldots, z_l)^T$ is a candidate solution, and $\mathbf{F} : \Omega \rightarrow \mathbb{R}^m$ consists of m objective functions. For two solutions $\mathbf{z}_1, \mathbf{z}_2 \in \Omega$, $\mathbf{z}_1$ is said to dominate $\mathbf{z}_2$ (denoted as $\mathbf{z}_1 \prec \mathbf{z}_2$), if and only if $f_i(\mathbf{z}_1) \leq f_i(\mathbf{z}_2)$ for every $i \in \{1, \ldots, m\}$ and $f_j(\mathbf{z}_1) < f_j(\mathbf{z}_2)$ for at least one index $j \in \{1, \ldots, m\}$. A solution $\mathbf{z}^*$ is Pareto optimal to (1) if there is no other solution $\mathbf{z} \in \Omega$ such that $\mathbf{z} \prec \mathbf{z}^*$. $\mathbf{F}(\mathbf{z}^*)$ is called a Pareto optimal objective vector. The set of all Pareto-optimal solutions is called the Pareto optimal set (PS) and the set of all Pareto-optimal objective vectors is called the Pareto front (PF).

Multiobjective evolutionary algorithms (MOEAs) usually perform well on two- and three-objective problems. However, the performance of many existing algorithms severely degrades when the number of objectives is more than three [12, 23]. Optimization problems with more than three objectives are often referred to as many-objective problems (MaOPs). In recent years, developments in the evolutionary computation community have focused on the design of specialized MOEAs for such many-objective settings [12, 23], including indicator-based, decomposition-based, and dominance- and decomposition-based approaches.

In the following, we build on these developments in many-objective optimization to propose a multiview clustering method capable of scaling to an arbitrary number of views, overcoming the limitation of previous approaches.

3. The Proposed Approach

In the following we describe our many-view clustering method MVMC. As discussed above, MVMC aims to provide a robust solution approach to a broad range of multiview clustering scenarios, specifically, multiple feature sets and/or multiple sources of relational information. However, we believe that its most important methodological contribution is the decoding step, which exploits a specific synergy between the demands of clustering and the many-objective optimization field. Our algorithm embraces the use of medoids to ensure applicability to the full range of multiview clustering scenarios. We demonstrate that the direct integration of information about prototypes (used to represent clustering solutions), with information about the weight vectors (used within the underlying optimizer), can achieve the design of a decoding strategy that is unbiased towards any of the views considered. This strategy is different from all previous related work, which has consistently relied on decoding mechanisms that inadvertently introduce a view-specific bias into the optimization. This important contribution, and its positioning within the overall scope of our work, is further illustrated in Figure 1. Specifically, the second stage chart ("Proposed algorithm") highlights how the same pair of prototypes can lead to different clustering solutions, due to differences in the underlying weight vectors alone.

The general framework of MVMC is outlined in Algorithm 1, which is based on the decomposition-based algorithm (MOEA/D) [24]. It decomposes a multiobjective optimization problem (MOP) into many single-objective subproblems (corresponding to different scalarizations), which are simultaneously optimized, and the optimal solutions of all subproblems constitute the Pareto optimal set. MOEA/D is a well-know many-objective evolutionary algorithm that has demonstrated to be more efficient than other multiobjective methods regarding the quality of solutions and the convergence rate [24, 25]. For these reasons, MOEA/D is considered as the underlying optimizer in our proposed multiview data clustering approach. MVMC requires as its input: the number of subproblems (NP), a uniform spread of NP weight vectors ($\mathbf{W}$)[1], the m dissim-

[1]The weight vectors are generated by using Das and Dennis's method [26].

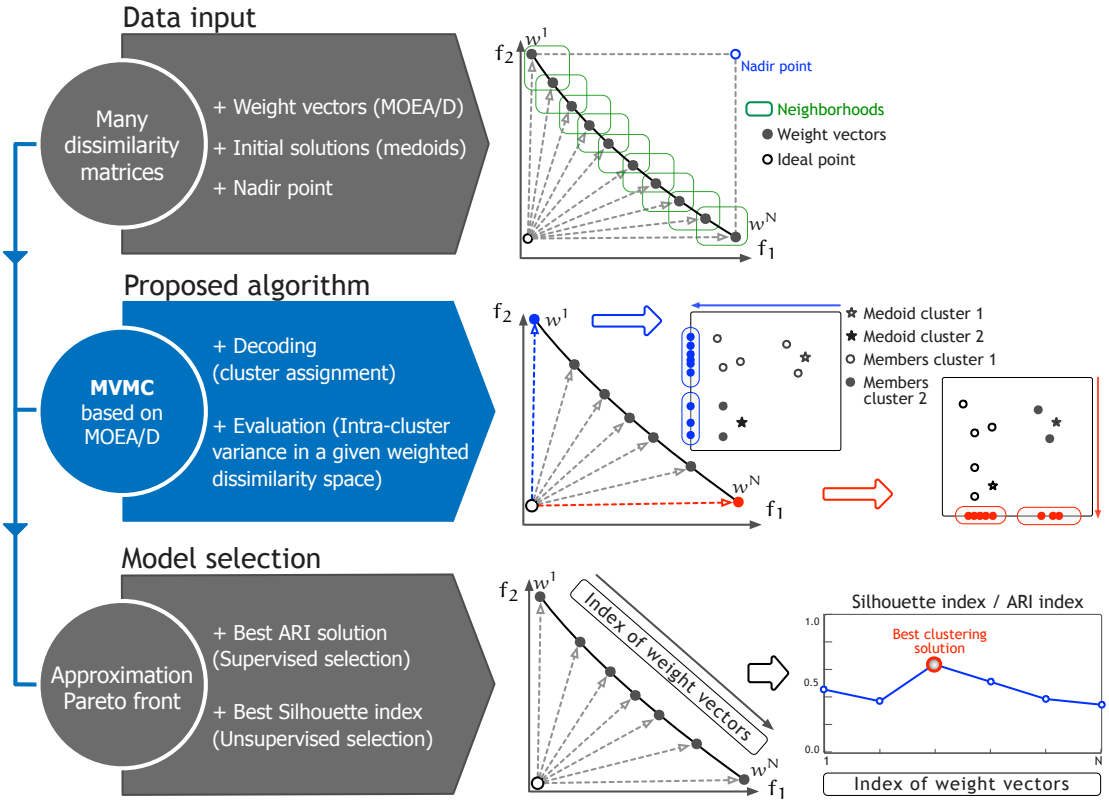

Figure 1: Main stages and components of the proposed MVMC approach. The multiple dissimilarity matrices can be obtained from considering: (i) a single feature set and distinct dissimilarity functions; and (ii) multiple feature sets and a single dissimilarity function.

ilarity matrices $\mathbb{D} = \{\mathbf{D}_1, \ldots, \mathbf{D}_m\}$, and the termination criterion, which in this case is the maximum number of generations ($G_{\max}$). The implementation details of each component of MVMC is described below.

3.1. Decomposition-based Many-objective Optimization

We use a decomposition-based many-objective optimizer as the underlying search engine for our clustering approach. Following common practice, we adopt the Tchebycheff approach [27] to decompose the multiobjective clustering problem into a scalar optimization subproblems set. Let $\mathbf{z}$ be a solution concerning a weight vector $\mathbf{w}^i$. In the Tchebycheff approach, the objective function of the i-th subproblem is expressed as follows:

$$g^{\text{te}}\left(\mathbf{z} \mid \mathbf{w}_i, \mathbf{z}^{\text{ref}}\right) = \max_{1 \le i \le m} \left\{ w_j^i \left| \frac{f_j(\mathbf{z}) - z_j^{\text{ref}}}{z_j^{\text{nad}} - z_j^{\text{ref}}} \right| \right\} , \qquad (2)$$

where $f_j : \Omega \to \mathbb{R}$ is the j-th clustering criterion $j \in \{1, \ldots, m\}$, which is to be minimized (without loss of generality), $\mathbf{w}^i = \left(w_1^i, \ldots, w_m^i\right)^T$ is the i-th weight vector, $\mathbf{z}^{\text{ref}} = \left(z_1^{\text{ref}}, \ldots, z_m^{\text{ref}}\right)$ is the reference point, and $\mathbf{z}^{\text{nad}} = \left(z_1^{\text{nad}}, \ldots, z_m^{\text{nad}}\right)$ is the nadir point.

3.2. MVMC's Representation and Initialization

We will use the terms encoding and representation synonymously to refer to the way a candidate solution is represented within an evolutionary algorithm. MVMC employs a medoid-based representation of candidate partitions, widely and successfully used in evolutionary approaches to data clustering [28, 29]. This representation can cope with large problem instances (with respect to the number of data points or the number of features) and impact computation efficiency during the clustering process. More importantly, this representation is more general than centroids, as it can be used both for problems defined in terms of feature spaces or dissimilarity matrices. In a medoid-based representation, each vector $\mathbf{z}$ comprises K genes $z_1, \ldots, z_K$, where K is the number of clusters, and each gene z_i can take values in the range $\{1, \ldots, N\}$, where N is the number of items in the data views. This set of medoids can then be interpreted as an actual partition by assigning all data points to their closest (least dissimilar) medoid, and we refer to this as the decoding step. As the concept of dissimilarity varies with respect to each view, a unique challenge in a multiview setting is the design of this step in a manner that facilitates the identification of clusters supported by different data views. In previous works [14, 11, 16, 17], the decoding of a set of candidate prototypes (medoids or centroids) has typically relied on using a single view (or a fixed weighting between views) to determine cluster assignments, potentially introducing a bias into the algorithm. Here, we demonstrate how the underlying principles of the decomposition approach can be exploited to define a decoding mechanism that introduces no additional assumptions on the relative importance of different views (see description in Section 3.3).

The initialization procedure of MVMC (line 1 of Algo-

Algorithm 1: General framework of MVMC algorithm.

Input: NP, $\mathbf{W}$, $\mathbb{D}$, $G_{\max}$
Output: Population $\mathbf{P}$, Pareto front approximation $\mathbb{S}$

1 $\left[\mathbf{P}, \mathbf{B}, \mathbf{z}^{\text{nad}}\right] \leftarrow$ INITIALIZATION($\mathbf{W}$)
2 **for** $g \leftarrow 1$ **to** $G_{\max}$ **do**
3 **for** $i \leftarrow 1$ **to** NP **do**
4 $\mathbf{u}_i \leftarrow$ REPRODUCTION($\mathbf{P}$,$\mathbf{B}(i)$)
5 $\mathbf{C}_i \leftarrow$ DECODIFICATION($\mathbf{u}_i$, $\mathbf{w}^i$, $\mathbb{D}$)
6 $\mathbf{f}_i \leftarrow$ EVALUATION($\mathbf{C}_i$, $\mathbb{D}$)
7 Update $\mathbf{z}^{\text{ref}}$ /* reference point */
  /* Update the neighboring solutions */
8 **foreach** $j \in \mathbf{B}(i)$ **do**
9 **if** $g^{te}\left(\mathbf{u}_i \mid \mathbf{w}^j, \mathbf{z}^{ref}\right) \leq g^{te}\left(\mathbf{z}_j \mid \mathbf{w}^j, \mathbf{z}^{ref}\right)$ **then**
10 $\mathbf{P}(j) = \mathbf{u}_i$
11 $\mathbf{Fit}(j) = \mathbf{f}_i$
12 $\mathbb{S}(j) = (\mathbf{C}_i, \mathbf{w}_i)$
13 **end**
14 **end**
15 **end**
16 **end**

rithm 1) consists of three main steps:

- Initialization of the parent population $\mathbf{P}$: Generate an initial population $\mathbf{P} = \{\mathbf{z}_1, \ldots, \mathbf{z}_{NP}\}$ randomly from Ω, where the i-th individual is a vector of size K denoted by $\mathbf{z}_i = [z_{i,1}, \ldots, z_{i,K}]$ representing K cluster medoids, $z_1, \ldots, z_K$.
- Assignment of neighborhood $\mathbf{B}$: For each weight vector $\mathbf{w}^i$, $i \in \{1, \ldots, NP\}$, $\mathbf{B}(i)$ consists of the indices of the T closest weight vectors of $\mathbf{w}^i$ (based on Euclidean distance).
- Initialization of the nadir point $\mathbf{z}^{\text{nad}} = \left(z_1^{\text{nad}}, \ldots, z_m^{\text{nad}}\right)$. It is computed as $z_j^{\text{nad}} = f_j(\mathbf{C}^1)$, $j \in \{1, \ldots, m\}$, $\mathbf{C}^1$ is the solution where all data points are clustered together (i.e., $K = 1$). Thus, z_j^{nad} defines the lower bound of the objective function when using the j-th dissimilarity matrix.

3.3. Decoding of Solutions

In multiview clustering settings that employ a prototype-based representation, the decoding of a consensus partition is an important step, as it is not straightforward to uniquely decode a set of prototypes in the context of multiple data views. This is because the decoding step requires the assignment of all data points to their closest prototype, but no general agreed notion of closeness exists in a multiview setting. Here, we set out our decoding mechanism, which assumes knowledge of the scalarizing vector associated with each candidate solution, to address this challenge. Let $\mathbf{z}_i$ and $\mathbf{w}^i$ be the medoid and weight vectors, respectively, corresponding to the i-th subproblem. Also, let $\{\mathbf{D}_1, \ldots, \mathbf{D}_m\}$ denote the m dissimilarity matrices[2], which represent the different data views. Thus, for the i-th subproblem, the cluster assignment for data point $s \in \{1, \ldots, N\}$ is obtained as:

$$\mathbf{C}_i(s) = \text{argmin}_{r \in \mathbf{z}_i} \{D_{\text{ws}}(r, s)\} \ , \tag{3}$$

where

$$D_{\text{ws}}^{r,s} = w_1^i D_1^{r,s} + \ldots + w_m^i D_m^{r,s} \ . \tag{4}$$

Here, $D_{\text{ws}}^{r,s}$ represents the weighted-sum distance matrix associated with the scalarizing vector $\mathbf{w}^i$. Thus, the partition $\mathbf{C}_i$ is derived, from the set of medoids, using the definition of closeness that is relevant to the particular sub-problem considered. Note that this decoding strategy requires knowledge of the scalarizing vector for each candidate solution. Therefore, this strategy cannot be used in Pareto-based optimization approaches for multiview clustering, as this information is not available.

3.4. Objective Functions

The within-cluster scatter (WCS) has been selected as the optimization criterion. Let $\mathbf{C}_i$ be a clustering decoded from candidate solution $\mathbf{z}_i$, and let $\mathbf{D}_j$, $j \in \{1, \ldots, m\}$, be a specific dissimilarity matrix. Then, the WCS for the j-th objective of the i-th subproblem is computed as:

$$f_j(\mathbf{C}_i) = \sum_{\mathbf{c}_k \in \mathbf{C}} \sum_{\mathbf{a}, \mathbf{b} \in \mathbf{c}_k} d_j(\mathbf{a}, \mathbf{b}) \ , \tag{5}$$

where $d_j(\mathbf{a}, \mathbf{b})$ is the dissimilarity between the data objects $\mathbf{a}$ and $\mathbf{b}$ as defined in $\mathbf{D}_j$. There are no additional constraints. After the evaluation step, the ideal point $\mathbf{z}^{\text{ref}}$ is updated as follows (line 7 of Algorithm 1):

$$z_j^{\text{ref}} = \begin{cases} f_{i,j} \ , & \text{if } f_j(\mathbf{C}_i) < z_j^{\text{ref}} \ , \\ z_j^{\text{ref}} \ , & \text{otherwise} \ . \end{cases} \tag{6}$$

3.5. Reproduction Operators

The reproduction procedure (line 4 of Algorithm 1) generates offspring solutions to update the parent population. First, two indices k and l are randomly selected from $\mathbf{B}(i)$. Then a new solution $\mathbf{u}$ is generated from $\mathbf{z}_k$ and $\mathbf{z}_l$. Since $\mathbf{z}_k$ and $\mathbf{z}_l$ are the current best solutions to neighbors of the i-th subproblem, their offspring $\mathbf{y}$ should be a good solution and very likely to have a better fitness value to the neighbors of the i-th subproblem.

Our MVMC algorithm uses a medoid-based representation, integer values denoting data-point positions, which are the decision variables to be optimized (see Section 3.2). Hence, we use the position-based crossover (PBC) and position-based mutation (PBM) [30] operators, which are convenient for integer-based representations where unique allele values need to be created.

3.6. Selection of Clustering Solutions

Multiobjective clustering approaches return a set of solutions, representing different trade-offs between the objectives. In many real-world applications of multiobjective optimization, the decision-maker conducts the selection of the best solution(s)

[2] Dissimilarities are normalized using unity-based minmax normalization.

according to particular preferences. However, in some scenarios, automatic methods may be desirable to select a single best solution from the Pareto front approximations (PFAs). Therefore, in this section, we introduce an automated method for assessing the quality of individual clustering solutions.

Algorithm 2: Clustering selection procedure

Input: $\mathbb{D}$, $\mathbb{S}$
Output: The best solution $\mathbf{C}'$ in $\mathbb{S}$
foreach $(\mathbf{C}_i, \mathbf{w}_i) \in \mathbb{S}$ **do**
  /* Step 1: Computation of the distance matrix */
  Compute D_{ws} considering $\mathbf{w}_i$ and $\mathbb{D}$ using Eq. (4)
  /* Step 2: Computation of the Silhouette index */
  Compute the Silhouette value of $\mathbf{C}_i$ using Eq. (7)
end
/* Step 3: Report the best clustering solution */
Select the solution $\mathbf{C}'$ having the best Silhouette value

The proposed method to select the clustering solution uses the Silhouette cluster validity index to evaluate the quality of each solution in the final approximation front: it is further outlined in Algorithm 2 and Figure 1. The Silhouette index is a popular and general-purpose validation technique commonly used in cluster analysis to determine the number of clusters. This index defines a relation between the intra-cluster cohesion and the inter-cluster separation to estimate the quality of a clustering solution. Here, the Silhouette Width is used as a *posteriori* method to reduce the PFAs obtained by the MVMC algorithm to a single clustering solution. To make this index a more effective technique for our particular purposes, the Silhouette considers a weighted-sum distance to measure the similarity between data points belonging to the same clusters. Let $(\mathbf{C}, \mathbf{w})$ be a clustering solution and its corresponding weight vector in the set of trade-off solutions $\mathbb{S}$ (output of Algorithm 1). Then the Silhouette Width is computed as:

$$\text{Sil}(\mathbf{C}) = \frac{1}{N} \sum_{\mathbf{c}_k \in \mathbf{C}} \sum_{\mathbf{x}_i \in \mathbf{c}_k} \frac{\text{b}(\mathbf{x}_i, \mathbf{c}_k) - \text{a}(\mathbf{x}_i, \mathbf{c}_k)}{\max\{\text{b}(\mathbf{x}_i, \mathbf{c}_k), \text{a}(\mathbf{x}_i, \mathbf{c}_k)\}}, \tag{7}$$

where

$$\text{a}(\mathbf{x}_i, \mathbf{c}_k) = \frac{1}{|\mathbf{c}_k| - 1} \sum_{\mathbf{x}_j \in \mathbf{c}_k} d_{\text{ws}}\left(\mathbf{x}_i, \mathbf{x}_j\right), \tag{8}$$

$$\text{b}(\mathbf{x}_i, \mathbf{c}_k) = \min_{\mathbf{c}_r \in \mathbf{C} \setminus \mathbf{c}_k} \left\{ \frac{1}{|\mathbf{c}_r|} \sum_{\mathbf{x}_j \in \mathbf{c}_r} d_{\text{ws}}\left(\mathbf{x}_i, \mathbf{x}_j\right) \right\}.$$

In (7), $\text{a}(\cdot)$ denotes the average distance between $\mathbf{x}_i$ and all data points in the same cluster, whereas $\text{b}(\cdot)$ denotes the smallest average distance of $\mathbf{x}_i \in \mathbf{c}_k$ to all data points in any other cluster different from $\mathbf{c}_k$. The Silhouette index returns values in the interval $[-1, +1]$, where the higher its value, the better the match of the data points to the underlying cluster structure.

Notice that the particular dissimilarity matrix $D_{\text{ws}} = [d_{\text{ws}}(i, j)]$ used to measure the quality of $\mathbf{C}$ in $\mathbb{S}$, provides information about the importance of the data views (distance matrices). The use of the weighted-sum distance in (8) helps to select the best clustering solution by considering the information obtained in the multiobjective clustering phase.

3.7. MVMC Source Code

The source code of implementing the MVMC algorithm proposed in this paper and the collection of datasets considered in our experiments are available through the following repository: `https://github.com/adanjoga/mvmc`.

4. Experimental Setup

This section describes the reference methods, the performance assessment measures, and the settings adopted for this study. The parameter settings adopted by our proposed approach MVMC are summarized in Table 1. Parameter settings are kept constant across our different experiments, except for the population size and the number of generations. Following the general suggestions for MOEA/D [24], these are incremented as the number of objectives (here data views) increases.

Table 1: Summary of the main parameter settings used in this study.

Parameters	**2-views**	**3-views**	**4-views**	**5-views**
Population size (NP)	100	150	175	210
Number of generations (G_{max})	100	200	300	400
Recombination probability (Pr)	0.5	0.5	0.5	0.5
Mutation probability (Pm)	0.03	0.03	0.03	0.03
Neighborhood size (T)	20	20	20	20

4.1. Reference Methods

Our experiments aim to show that the conceptual advantages of multiobjective multiview clustering translate into an improvement in the clustering performance compared to traditional clustering algorithms. We compare the proposed MVMC against several well-known and conceptually different clustering algorithms: a partitional clustering approach, *k*-means [31]; two hierarchical clustering methods, Single-Linkage (SL) and WARD; a genetic clustering algorithm (GCA), which is based on the same representation, genetic operators, and settings as described above for MVMC in the case of two-objective instances.

Furthermore, three multiview clustering approaches are considered when evaluating multiview data problems. A multiview spectral clustering (MVSC) algorithm [32], which derives multiple similarity matrices from the input data and allows using an arbitrary number of views. A multiview approach using multiobjective optimization [33], which is referred to as Mitra's approach. The third multiview clustering approach proposed by Jiang et al. [11] encompasses a set of multi-objective algorithms[3]: SPEA2, NSGA-II, NSGA-III, and

[3]The source code is available at `https://sites.google.com/site/bojiangzjut/`.

MOEA/D. During the evolutionary search of this approach, a centroid-based solution is mapped to multiple partitions (one per view). Then each partition is evaluated using the objective function associated with the corresponding view. At the end of the search, a single consensus partition per candidate solution is obtained by decoding the final centroids within the full feature space (i.e., by concatenating all views).

The proposed approach MVMC generates a set of non-dominated clustering solutions, but a single solution is usually required. Thus, three different strategies for selecting the most suitable solution from the Pareto front approximations (PFAs) are explored: MVMC_{SIL}, MVMC_{AUC}, and MVMC_{ACC}. MVMC_{SIL} implements the proposed unsupervised selection method described in Section 3.6. MVMC_{AUC} and MVMC_{ACC} select the best solution based on the maximum accuracy value and the maximum AUC values, respectively, so they are supervised approaches.

4.2. Performance Assessment

The Adjusted Rand Index (ARI) measure is used to evaluate clustering performance [34]. ARI works by counting the number of pairwise co-assignments of data points between two given partitions L and T. ARI is defined in the range $[\backsim 0, 1]$. Values of ARI closer to unity are preferred as they indicate a better correspondence between L and T. The ARI measure serves as an indicator of the clustering method's performance at solving a particular problem as it compares the partition generated by a clustering method (L) and the correct partition of the data (T). The clustering performance of the multiview clustering algorithms is also evaluated by using the Normalized Mutual Information (NMI) metric [35]. NMI is defined in the range $[0, 1]$, where values closer to unity indicate a perfect correlation between the partitions.

The results obtained for the breast tumor classification problem are also evaluated in terms of accuracy (ACC), the area under the ROC curve (AUC), sensitivity (SEN), and specificity (ESP). These metrics are widely used in medicine as statistical measures of a binary classification test. All these indices lie in the range $[0, 1]$, and values tending towards unity indicate a better classification.

For all the stochastic clustering methods analyzed and compared in this study, a total of 31 independent executions for each dataset were performed. In all the cases, statistical significance is evaluated using the Kruskal–Wallis test, considering a significance level of $\alpha = 0.05$ and Bonferroni correction.

5. Results on Multiview Clustering using Multiple Relational Data

5.1. Synthetic Datasets

This section investigates the ability of MVMC to generate high-quality clustering solutions on synthetic datasets with varying sizes, dimensionalities, degrees of overlapping, and cluster shapes. These synthetic datasets have been organized into four categories regarding their type of underlying data distribution: well-separated clusters ($\mathcal{G}1$), overlapping clusters ($\mathcal{G}2$), nonlinearly separable clusters ($\mathcal{G}3$), and mixtures of different data distributions ($\mathcal{G}4$). Figure A.11 in Appendix A illustrates the diversity of properties covered by our test datasets.

In this study, different data views are derived from a single feature set by using two conceptually different dissimilarity measures: the Euclidean distance (denoted by ▲) and the MED distance based on Euclidean (MED_{euc}, denoted by ▼). The single-view clustering algorithms k-means, SL, WARD, and GCA perform the clustering task using one dissimilarity measure, whereas MVSC and MVMC simultaneously optimize two data views, each accounting for a different dissimilarity measure. The results of this analysis are summarized in Figure 2. For more detailed results and the corresponding statistical significance analysis, refer to Table A.5 (Appendix).

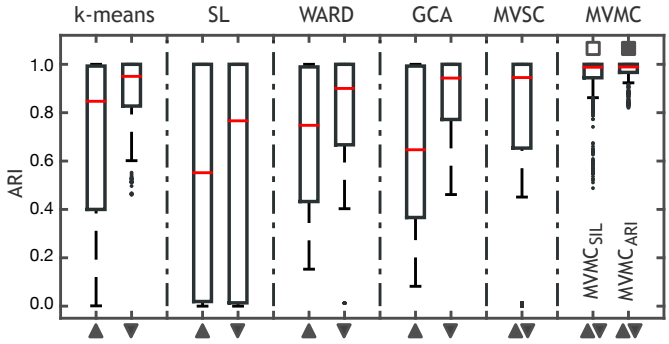

Figure 2: Clustering performance (in terms of ARI) on synthetic datasets obtained by the MVMC configurations MVMC_{ARI} and MVMC_{SIL}, and the other clustering algorithm k-means, SL, WARD, GCA, and MVSC. The data views are derived using two different dissimilarity measures: Euclidean (▲) and MED_{euc} (▼). The square symbol, □, indicates no statistically significant difference between groups compared to the best one, ■.

Given the single-view clustering algorithms being evaluated, the chosen dissimilarity measures, and the particular properties of our test datasets, we can anticipate certain behaviors as a result of our experiments. First, a good performance can be expected when using the Euclidean distance on datasets in categories $\mathcal{G}1$ and $\mathcal{G}2$, as these contain spherical, Gaussian clusters that are favored by this measure. In contrast, we can expect a poor performance of this distance on datasets with nonlinearly separable clusters, particularly for categories $\mathcal{G}3$ and $\mathcal{G}4$. Second, a good performance is expected when using the MED distance on $\mathcal{G}1$ and $\mathcal{G}3$, as these datasets contain well-separated clusters regardless of their linear separability, which is the assumption made by the MED distance (i.e., clusters of arbitrary shape can be detected) [9]. However, we can anticipate a poor performance of the MED distance for datasets having overlapping clusters, including the synthetic datasets in $\mathcal{G}2$ and $\mathcal{G}4$.

From the experimental results summarized in Figure 2, we confirm the above expectations. On the one hand, we observe that the individual single-view algorithms obtain good results on datasets that comply with the assumptions made by the particular dissimilarity measure employed. Accordingly, these methods do not show consistent performance across the range

of different cluster properties. On the other hand, the multiview algorithm results confirm that superior performance can be achieved through the simultaneous consideration of multiple dissimilarity measures. Specifically, the proposed MVMC strategies MVMC_{ARI} and MVMC_{SIL} consistently exhibited a highly competitive performance in all test scenarios. According to Figure 2, the MVMC strategy MVMC_{ARI} scored the best overall performance (0.97 ± 0.04). Although this strategy reports a consistently good performance across the entire range of data properties considered, no statistically significant differences have been found concerning MVMC_{SIL} (0.94 ± 0.11).

Consideration of the associated Pareto front approximations (PFAs) highlights a further advantage of our approach in exploratory data analysis. As apparent from Figure 3, the shape and span of the PFA provide additional insight regarding the strength of the signal and the type of data structures in a given dataset. There is no conflict between the two data views for datasets containing spherical, spatially separated clusters, and the PFA collapses to a single point. In contrast, for datasets such as `Inside` and `Sizes5`, the Pareto front highlights the conflict between views and helps confirm the presence (or absence) of structurally pronounced trade-off solutions associated with distinct knees in the Pareto front [36].

5.2. *Real-world Datasets*

This section investigates the scalability of MVMC when increasing the number of data views (dissimilarity measures). Six real-world datasets from the UCI repository [37] are considered: `Iris`, `Wine`, `Breast`, `Thyroid`, `Glass`, and `Ecoli`. Multiple data views are derived from a single feature set (dataset) by using four different dissimilarity measures: Euclidean distance (▲), MED based on Euclidean (MED_{euc}, denoted by ▼), Cosine distance (△) and MED based on Cosine (MED_{cos}, denoted by ▽). A total of ten combinations of data views are studied: six bi-objective configurations, three three-objective configurations, and one four-objective configuration. The single-view algorithms *k*-means, WARD, and GCA perform the clustering task using one of the four individual dissimilarity measures. In contrast, MVSC and MVMC simultaneously optimize one of the ten combinations which involve two or more data views. The results are summarized in Figures 4 and 5. For more detailed results and the corresponding statistical significance analysis, refer to Table A.6 (Appendix).

Figure 4 compares the performance of the six basic bi-objective configurations of MVMC against the three single-view clustering algorithms. We can see that configuration $\text{MVMC}_{\blacktriangle\triangledown}$ shows the best performance, with an average ARI of 0.81 ± 0.12. Additionally, no statistically significant differences are observed between this strategy and configurations $\text{MVMC}_{\blacktriangle\triangle}$ (0.75 ± 0.20) and $\text{MVMC}_{\triangle\blacktriangledown}$ (0.77 ± 0.15). These results suggest that the consideration of two dissimilarity measures allows MVMC to produce better clustering solutions than the algorithms based on a single dissimilarity measure.

We also compare the MVMC strategies MVMC_{ARI} and MVMC_{SIL} with the MVSC algorithm when increasing the number of views. Figure 5 illustrates the clustering performance obtained by these algorithms for all the ten different combinations of data views. It can be seen that the use of more than two dissimilarity measures allows MVMC to produce even better clustering solutions than the bi-objective configurations. We observe that strategy MVMC_{ARI} obtained the best performance when using all four data views, with an average ARI of 0.846 ± 0.08. Additionally, no statistically significant differences have been found between this strategy and all the three-objective configurations of MVMC_{ARI}. The best overall performance for strategy MVMC_{SIL} was for instances {▲▽} and {▲▼ △ ▽}, whereas for MVSC, the best performance was attained for the bi-objective instance {▲ △}.

In general, we observe an improvement in the clustering performance for the MVMC configurations when considering three and four objectives, concerning the bi-objective configurations of MVMC. For strategy MVMC_{ARI}, it is evident that the increase in the number of views systematically translates into an increased clustering performance. Real-world datasets typically exhibit overlapping clusters or noisy data, resulting in low performance of the contestant multiview approach MVSC

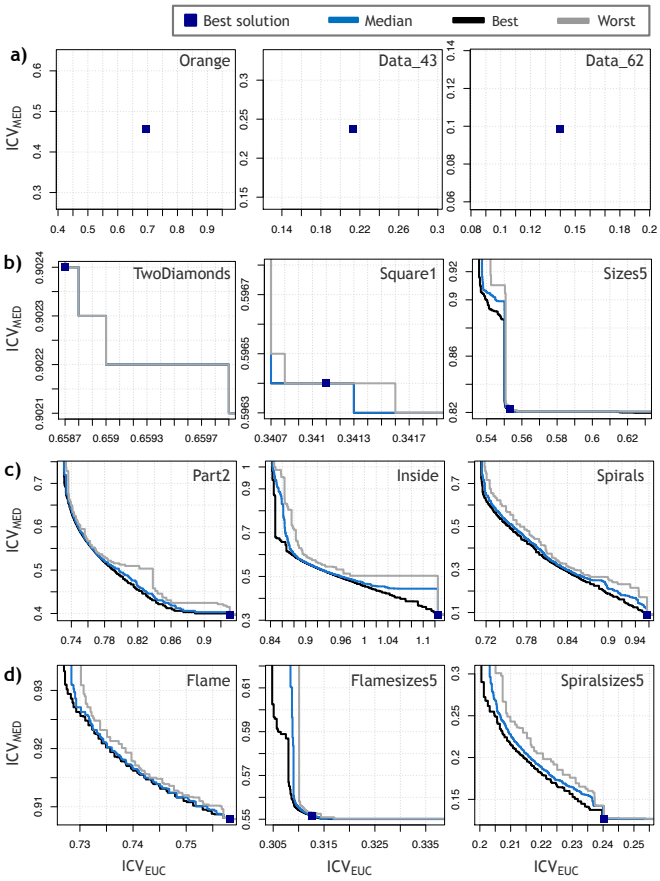

Figure 3: PFAs obtained by the proposed approach MVMC on some synthetic datasets when minimizing the within-cluster scatter (WCS) using the Euclidean distance (x-axis) and MED distance (y-axis). For each subfigure, the best, average, and the worst PFAs derived from 31 runs are plotted using black, blue, and gray lines, respectively. Additionally, each subfigure includes a square to illustrate the best solution (maximum ARI value).

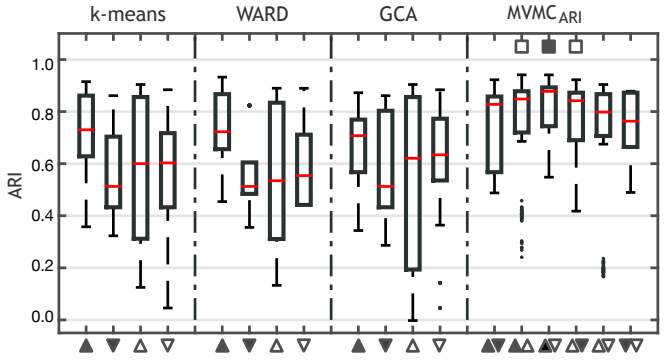

Figure 4: Clustering performance (in terms of ARI) on real-world datasets obtained by MVMC_{ARI}, six bi-objective configurations, and the other single-view algorithms *k*-means, WARD, and GCA. The square symbol, □, indicates no statistically significant difference between groups compared to the best one, ■.

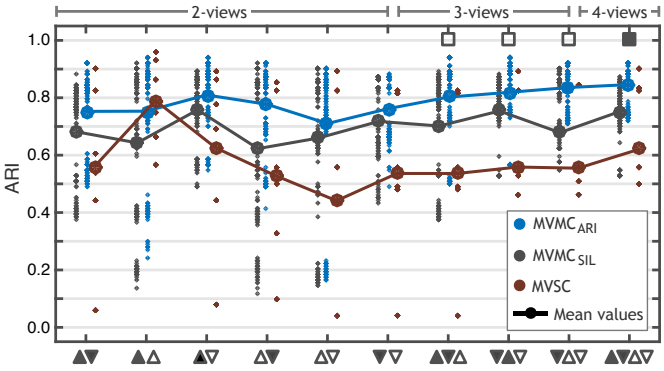

Figure 5: Clustering performance (in terms of ARI) obtained by the MVMC strategies MVMC_{ARI} and MVMC_{SIL} for all ten configurations of data views. Results for approach MVSC have also been included as a reference. The square symbol, □, indicates no statistically significant difference between groups compared to the best one, ■.

and our MVMC_{SIL} method, which is the proposed unsupervised model selection strategy. The selection of the best solution, even from a high-quality PFA, is a challenging problem. Although the strategy employed by our method represents one step in this direction, further research is required to develop effective, fully unsupervised clustering selection techniques.

We investigate the characteristics of the Pareto front approximations (PFAs) generated by our proposed approach (MVMC). For the UCI datasets, all PFAs have several data points, confirming the challenging nature of these datasets and the sensitivity of the algorithm performance to the type of distance function used. An analysis of clustering performance along the PFA suggests that, for the UCI real-world datasets, the best solutions tend to correspond to regions in the middle of the front, i.e., are best identified by considering the trade-off between multiple views. Some of these solutions (e.g., `Ecoli`) are structurally pronounced, and therefore reflected by a knee in the associated PFA. For other datasets such as `Glass`, the PFA provides little evidence of a pronounced cluster structure. Figure 6 exemplifies the contrasting characteristics found in the PFAs generated by MVMC.

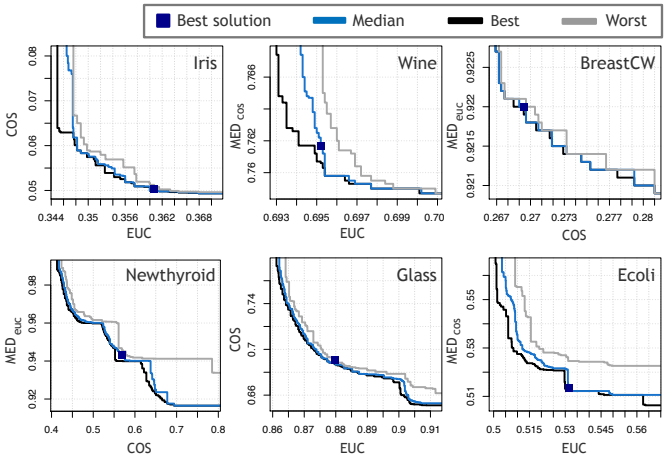

Figure 6: Examples of PFAs obtained by MVMC on real-world datasets when minimizing the within-cluster scatter (WCS) using for different dissimilarity measures: Euclidean, MED_{euc}, Cosine, and MED_{cos}. For each subfigure, the best, median, and the worst PFAs derived from 31 runs are plotted using black, blue, and gray lines, respectively. Each subfigure includes a blue square to illustrate the best clustering solution (maximum ARI value). Note that the best ARI solution is not always located on the approximation front, indicating the limited power of both views in recovering the ground truth structure.

6. Results on Multiview Clustering Using Multiple Feature Data

This section investigates the performance of MVMC on a collection of multiview datasets proposed in the specialized literature. Table 2 presents details of the studied multiview datasets: The Euclidean distance was used to compute the dissimilarity matrix for each view in MVMC algorithm. These datasets with multiple data views are available from the following sources:

- Image segmentation (Image) [11]: This dataset contains 2310 objects from a seven outdoor image database. This dataset is divided into two views: shape view and RGB view.
- Water Treatment Plant (WTP) [11]: This dataset contains daily measures of sensors in an urban WTP. The dataset is divided into four data views: input view, output view, performance input view, and global performance view.
- Amsterdam Library of Object Image (ALOI)[4]: It is a collection of images of different objects recorded under various conditions. These images are represented by four feature sets: RGB color histograms, HSB color histograms, color similarity, and Haralick features. In our experiments, we randomly selected three-class subsets (i.e., 300 images) from the original dataset.

[4]ALOI dataset: `http://elki.dbs.ifi.lmu.de/wiki/DataSets/MultiView`

- Corel image dataset (Corel)[5]: It is a collection of images having different properties such as different colors, lighting, and angle. In our experiments, we randomly selected five classes (i.e., 500 images) with six views from the original dataset. The views are color histogram, moment, coarseness, Tamura texture, wavelet, and MARSAR texture.

- US Postal Service (USPS)[6]: The dataset is obtained from a collection of Dutch utility maps. The following two data views are considered: Fourier coefficients of the character shapes and profile correlations.

- Columbia Consumer Video (CCV)[7]: This dataset involves the two data views as follows: Scale-invariant feature transform and space-time interest points. For simplicity, the multi-labeled and unlabeled samples are removed from the original database.

In addition to the algorithms considered in the previous sections, we include a comparison with Mitra's approach[8] [33], and Jiang et al.'s [11] approaches to multiview clustering: SPEA2, NSGA-II, NSGA-III and MOED/D. Table 3 indicates the best clustering solution in terms of the ARI and NMI metrics. In comparison to the contestant techniques, MVMC and Mitra's algorithm clearly benefit from the flexibility of a multiobjective approach. MVMC's scalability to many objectives should give it an advantage over Mitra's approach in a many-view setting (i.e., with three or more views), but the unavailability of the code prevents us from investigating Mitra's approach in this particular scenario.

Also, from the results in Table 3, it is notable that MVMC outperformed Jiang et al.'s approaches to multiview clustering [11]. This superior performance of MVMC is mainly due to its solution decoding scheme. In Jiang et al.'s work, a different partition is derived for each view during the decoding stage. This decoding strategy maps a single candidate solution to several possible partitions, one per view, and evaluates each of these partitions using the objective function associated with that view. Such approach presents the following limitation: as a given candidate solution is mapped to multiple, view-specific partitions, the objective vector eventually assigned to the candidate solution is an aggregation of the objective values associated with these partitions. Therefore, this decoding strategy introduces a bias into the algorithm which, as observed in Table 3, significantly impacts on clustering performance. Figure 10 illustrates the impact of the decoding steps used in Jiang's approach and the one proposed in MVMC.

[5]Corel dataset: `https://archive.ics.uci.edu/ml/datasets/corel+image+features`

[6]USPS dataset: `https://archive.ics.uci.edu/ml/datasets/Multiple+Features`

[7]CCV dataset: `http://www.ee.columbia.edu/ln/dvmm/CCV/`

[8]Mitra's results are taken directly from the paper as no implementation of the algorithm is available.

7. Application of MVMC to Breast Tumor Classification

Finally, we investigate the capabilities of MVMC on a challenging multiview dataset. The problem under study is associated with computer-aided diagnosis (CAD) systems for breast ultrasound (BUS), where the objective is to generate a tumor classification providing a second opinion and avoiding inter-observer variation. Generally, the CAD system pipeline involves four stages: image preprocessing, lesion segmentation, feature extraction, and lesion classification. Specifically, in the lesion classification step, machine learning techniques are used to distinguish between benign and malignant tumors.

Towards this end, a quality assurance tool has recently been designed which standardizes mammographic reports and is known as the breast imaging-reporting and data system (BI-RADS). The last edition of the BI-RADS lexicon for ultrasound considers five qualitative terms to describe the shape, orientation, margin, echo pattern, and posterior features of the masses. Therefore, a common approach when designing CAD systems based on BI-RADS is to nominally depict each qualitative term of the BI-RADS lexicon for masses using quantitative features. Then, the collection of quantitative characteristics forms a single feature vector that constitutes the input of a classifier.

Here, we address the classification of breast lesions as an unsupervised machine learning problem (i.e., as a data clustering problem). In this regard, instead of collecting all the quantitative features to form a single feature set, each feature set derived from the BI-RAD lexicons is considered as a distinct view by our MVMC algorithm. The advantage of this exploratory approach is a distinct insight into the level of natural cluster structure in each of the separate views, and the extent to which different views provide complementary information.

7.1. Description of the Breast Ultrasound Dataset

The dataset consists of 2054 BUS images acquired at the National Cancer Institute of Rio de Janeiro, Brazil. All the images were obtained from patients with posterior indication of biopsy, from which 1351 images presented benign lesions, and 703 had malignant tumors. A senior radiologist manually outlined each breast lesion with the help of specialized software. Then, to cover the five terms of the BI-RADS lexicon, morphological and texture features were extracted for each BUS image from the contour outlined by the radiologist. The properties of the data views used in this study are summarized in Table 4 and illustrated in Figure 7. In total, 139 quantitative features were computed to generate five feature spaces, one for each term of the BI-RADS lexicon [7].

For the experiments of this study, a total of 26 data-view configurations were obtained from the five feature sets: ten two-objective, ten three-objective, five four-objective, and one five-objective configuration corresponding to combinations of two, three, four, and five data views, respectively. A specific problem configuration will be referred to as V*m-seq* throughout this study, where $m = \{2, 3, 4, 5\}$ is the number of data views, and *seq* denotes a sequence of m different letters referring

Table 2: Characteristics of the multiview datasets considered in this study. "Multiview dataset" refers to the dataset acronym, "N" denotes the number of data points, "V" is the number of data views, "D" is the feature space dimensionality for each view, and "K" is the actual number of clusters.

Multiview dataset	**N**	**V**	**D**	**K**
Image	2100	2	[9, 10]	7
WTP	230	4	[22, 7, 5, 4]	5
Aloi	333	4	[8, 27, 77, 13]	3
Corel	500	6	[64, 9, 10, 8, 104, 15]	5
USPS	2000	2	[76, 216]	10
CCV	6773	2	[5000, 5000]	20

Table 3: Detailed results in terms of ARI and NMI metrics on all multiview datasets (mean values from 31 runs). The performance of MVMC is compared with respect to six multiview clustering approaches and the *k*-means algorithm. The results for Mitra's approach were taken from the original paper [33]. The best ARI and NMI values scored for each dataset has been shaded and highlighted in bold and, additionally, the statistically best ($\alpha = 0.05$) results are highlighted in boldface.

Multiview dataset	**Mitra's**	***k*-means**		**MVSC**		**SPEA2**		**NSGA-II**		**NSGA-III**		**MOEA/D**		**MVMC**	
	NMI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI	NMI	ARI
Image	–	**0.537**	0.372	**0.519**	0.362	0.393	0.259	0.401	0.257	0.411	0.243	0.434	0.287	**0.631**	**0.552**
WTP	–	0.074	0.025	0.038	0.009	0.101	0.031	0.103	0.043	0.080	0.006	0.100	0.022	**0.192**	**0.151**
Aloi	–	0.624	0.438	0.644	0.587	0.700	0.561	0.699	0.561	0.694	0.560	0.681	0.559	**0.999**	**0.998**
Corel	–	**0.535**	**0.456**	0.445	0.405	0.439	0.393	0.432	0.381	0.327	0.273	0.365	0.295	**0.508**	**0.422**
USPS	**0.781**	0.622	0.489	**0.753**	**0.703**	0.420	0.280	0.449	0.307	0.464	0.308	0.423	0.276	**0.768**	**0.725**
CCV	**0.234**	0.081	0.079	**0.241**	**0.216**	0.217	0.203	0.193	0.149	0.181	0.133	0.211	0.198	**0.297**	**0.259**

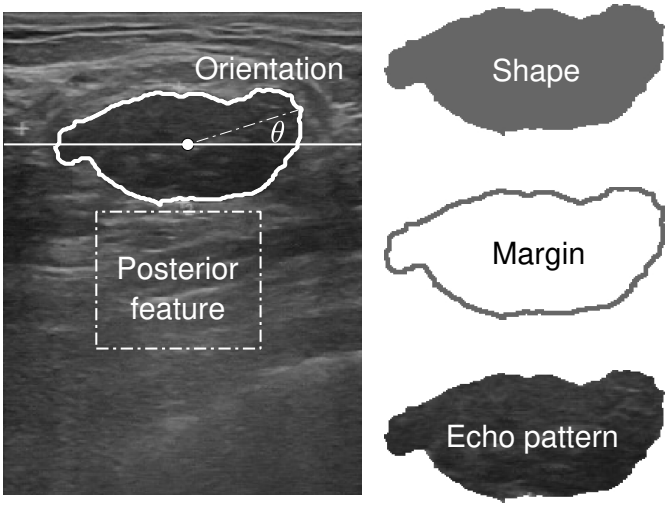

Figure 7: Illustration of the different data views derived from an ultrasound image for each BI-RADS lexicon.

to the data views used in the configuration as specified in Table 4. Finally, the Euclidean distance was used to compute the multiple dissimilarity matrices from the five input feature sets. For the implementation of *k*-means, the information from multiple data views was merged, assuming commensurability (and therefore equal weighting) between the individual feature spaces.

Table 4: Description of the different data views derived from the ultrasound images for each BI-RADS lexicon. The size of these data views is $N = 2054$ data points, and the number of clusters is $K = 2$.

Data views	*D*	*k*-means		
		AUC	**SEN**	**ESP**
Shape (S)	29	0.81	0.73	0.90
Margin (M)	14	0.82	0.81	0.83
Orientation (O)	2	0.67	0.42	0.92
Echo pattern (E)	90	0.65	0.74	0.55
Posterior features (P)	4	0.61	0.70	0.52

7.2. Clustering Performance

The results of MVMC and its variants for model selection are summarized in Figure 8, with more detailed results and the corresponding statistical significance analysis presented in Table A.7 (Appendix). Overall, clustering seems capable of obtaining good results for many problem configurations, suggesting the existence of natural cluster structures in the data views considered.

Our strategies MVMC_{AUC} and MVMC_{ACC} consistently produced better results than the reference algorithms *k*-means and MVSC. Notably, the strategy MVMC_{AUC} scored high-average values across different indices and diverse configuration problems, having a distinct number of views. For instance, the configurations `V2-MP` (0.846, 0.846, 0.849, 0.884), `V3-MEP` (0.849, 0.849, 0.847, 0.850), `V4-MOEP` (0.851, 0.847, 0.833, 0.860), and `V5-SMOEP` (0.851, 0.847, 0.834, 0.859) obtained high-average values for all indices, which are presented as a

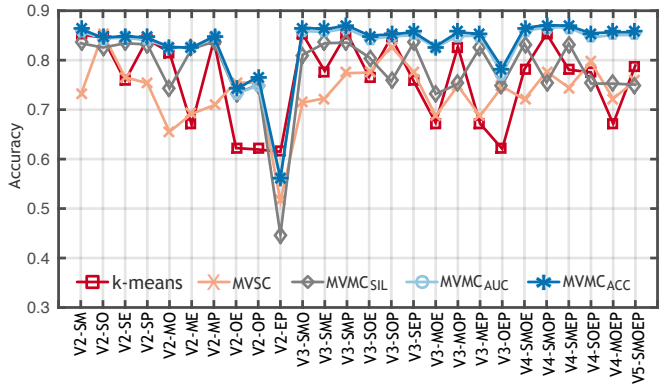

Figure 8: Summary of the clustering performance for the 26 problem configurations. Curves show the average accuracy scored by *k*-means, MVSC, MVMC$_{SIL}$, MVMC$_{AUC}$, and MVMC$_{ACC}$. Refer to Table A.7 for detailed results (Appendix).

tuple (ACC, AUC, SEN, SPE). Its performance across the different indices reflects the overall effectiveness of the proposed multiview approach, demonstrating that the simultaneous optimization of data views leads to better clustering performance and that this extends to the correct identification of both positive and negative cases.

Understanding these results in terms of the quality of the underlying feature spaces, it is worth noting that problem configurations involving the Shape (S) and Margin (M) views performed best, especially the Margin view. For instance, the configurations `V2-SM`, `V2-ME`, `V2-MP`, `V3-MEP`, `V3-SMP`, `V4-MOEP`, `V4-SMOP`, and `V5-SMOP` obtained high-average values for all the indices and across the different clustering algorithms. Interestingly, a poor clustering performance was observed for some problem configurations across all the algorithms studied. For instance, configurations, including the Echo (E) and Posterior (P) views, repeatedly resulted in low performance, indicating a poor predictive performance of these particular feature spaces. Due to the more flexible weighting approach implemented by MVMC$_{AUC}$ and MVMC$_{ACC}$ (using reference vectors), they are robust to the inclusion of such an individual, poor-quality feature space, outperforming *k*-means, and MVSC along with all performance indicators.

Multiview clustering's full potential derives from the presence of complementary, high-quality feature spaces whose joint consideration can improve performance. Our results (see Figure 8) indicate a general improvement in performance with the increase of the number of feature spaces considered, which provides evidence for the existence of complementary information in the data considered. Furthermore, a more detailed analysis (see Table A.7) shows distinct differences in the performance of single views, regarding their overall performance (accuracy) and their relative emphasis on sensitivity versus specificity. The latter highlights a particular opportunity for the fruitful combination of views: An example of this is the feature spaces related to Shape (S) and Margin (M). While the Shape feature set supports solution with high specificity and low sensitivity, the opposite is the case for the feature set Margin (M). As shown in Figure 9, the simultaneous optimization of both feature spaces results in the identification of solutions that strike a balance between the two views, corresponding to a trade-off between specificity and sensitivity, and resulting in a distinct improvement in classification accuracy.

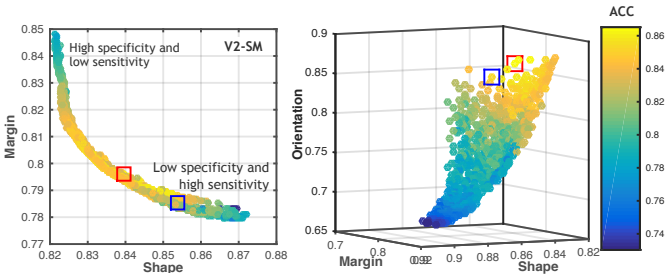

Figure 9: PFAs obtained from all independent executions of MVMC in terms of accuracy (ACC) for the problem configurations `V2-SM` (left) and `V3-SMO` (right). The clustering solutions are highlighted using different color intensities: the more tends to yellow, the better the performance in terms of ACC. The red and blue squares represent the best solutions selected by the strategies M$_{AUC}$ and M$_{ACC}$, respectively.

Figure 9 exemplifies this effect and further characteristics of the two- and three-dimensional PFAs generated by MVMC. It also illustrates the clustering solutions selected by strategies MVMC$_{AUC}$ and MVMC$_{ACC}$, indicating the position of the most accurate clustering solutions. By analyzing the shape of the PFAs for this dataset, it is clear that there is significant conflict between the multiple views, and that an optimal trade-off solution is difficult to identify from the structure present in the PFA alone. We observe that, for the feature sets considered here, the Pareto front provides us with different trade-offs regarding false positives and false negatives, akin to a ROC curve. Furthermore, consistently with our findings on the UCI datasets, we find that the regions of the objective space associated with the best clustering solutions tend to be those that correspond to a balanced trade-off between the data views; and that MVMC$_{AUC}$ and MVMC$_{ACC}$ consistently select these. These findings support the hypothesis that the most promising clustering solutions tend to integrate information from multiple views and that many-objective optimization techniques have an important role in supporting the generation of such candidate solutions.

8. Discussion

In this manuscript, we have introduced a new algorithm for many-view clustering and demonstrated its performance empirically. To better understand the empirical performance differences, it is useful to consider the theoretical underpinning of our work and, key differences to prior algorithms for multiview clustering. Our observations in this regard are as follows:

- The majority of existing work on multiview clustering has used algorithms based on Pareto optimization rather than approaches that employ a scalarizing function. This is simply reflective of the wider (and earlier) prevalence of these

algorithms in the field of evolutionary multi-objective optimization [38], rather than a conscious choice. As a consequence, though the known limitations of Pareto-based approaches (specifically poor scalability with regards to the number of objectives) have been transmitted to the field of multiview clustering, with the majority of existing evolutionary algorithms also based on Pareto dominance.

- Our decoding approach cannot be used in Pareto-based optimization approaches for multiview clustering. This is because the approach requires knowledge of the scalarizing vector for each candidate solution. In a Pareto-based optimization approach, this information is not available — in other words, given a single solution, we do not specifically know which scalarization vector it is optimal for; we only have generic information about its dominance relationships with other solutions.

- For Pareto-based approaches to multiview clustering, we are currently not aware of any method for decoding a medoid/centroid representation that does not rely on the choice of a single view or a specific weighting between views to support distance calculations in this step. The approaches commonly used in the literature are: (i) Repeated decoding of each solution along each view. This is used during the search in [11]. Here, the decoding step maps a single candidate solution to several possible partitions (one per view), and evaluates each using the objective function associated with that view. The limitation of doing this is as follows: as a given candidate solution is mapped to multiple, view-specific partitions, the objective vector eventually assigned to the candidate solution is actually an aggregation of the objective values associated with these partitions. In other words, we can only think of the resulting objective vector as an ideal point, and it does not indicate a trade-off between objectives that a single partition can necessarily achieve. (ii) Assumption of a fixed weighted sum of all views across all candidate solutions. This decoding is used implicitly at the end of Jiang et al.'s (2016) algorithm. In order to get a single partition per candidate solution, their algorithm decodes the final centroids within the full feature space (i.e., by concatenating all views). In doing so, it effectively weights each view by dimensionality and scale of the features within.

  In Figure 10, we include illustrative examples of the approximation fronts achieved by these different approaches to highlight the significant impact of this decoding stage and explain the specific "biases" introduced.

- For Pareto-based approaches to multiview clustering, the only known representations that do not require the choice of a single view or a specific weighting between views are those that directly encode the partition and therefore avoid the decoding step altogether. There are examples of such representations in the evolutionary clustering literature, but they are generally thought of as ineffective due to the size of the resulting representation (one decision variable is typically required per data point), the redundancy of the representation, and the difficulty of deriving effective variation operators that addresses this redundancy.

- There is no previous work on many-view clustering that exploits the scalarizing vector during the decoding and evaluation stage, and empirically validates the performance of this approach. Just like our paper, [11] uses MOEA/D, but derives a different partition for each view (see Figure 1 in the paper). For their work, this has the advantage that the same decoding scheme can consistently be used for the Pareto-based and scalarizing approaches compared in the paper. However, from a theoretical point of view, it suffers from the disadvantage highlighted above — the objective vectors observed during the search process are no longer reflective of the trade-off genuinely achievable by a single partition,which can impact the effectiveness of the search.

9. Conclusions

Multiview clustering is an important problem in many applications, due to the increasing availability of multiple data sources and representations describing the same entities. In our work, the term multiview clustering is used to refer to the consideration of different data aspects such as feature sets (i.e., quantitative measures) and relational information (i.e., dissimilarity relationships). Most existing approaches to multiview clustering are limited to the processing of two data views, or their clustering performance decreases as the number of views increases. In this paper, we have demonstrated how recent advances in many-objective optimization support the design of an evolutionary algorithm for data clustering capable of scaling the number of data views.

The proposed approach, called MVMC, has been evaluated across a range of data, including previous benchmarks for multiview clustering from the literature, and an application to breast tumor classification. We used synthetic and real-world data to evaluate the behavior of MVMC in situations where multiple relational data (i.e., different dissimilarity matrices) and multiple feature sets are available. Across our experiments, MVMC achieved highly competitive results when compared with respect to several traditional single-objective clustering techniques and two state-of-the-art approaches to multiview clustering.

Importantly, our findings generalize to more than two data views: the versions for three- and four-objectives scored significantly better results than the bi-objective versions in terms of clustering performance. In general, for a given dataset having different cluster properties, two possible cases can occur in MVMC when considering distinct data views: (i) a predominant data view can lead to useful solutions, or (ii) a trade-off between two or more views can lead to effective solutions. In the first case, both the MVMC and the single-objective clustering algorithms using the appropriate data view are successful. However, in the second case, only

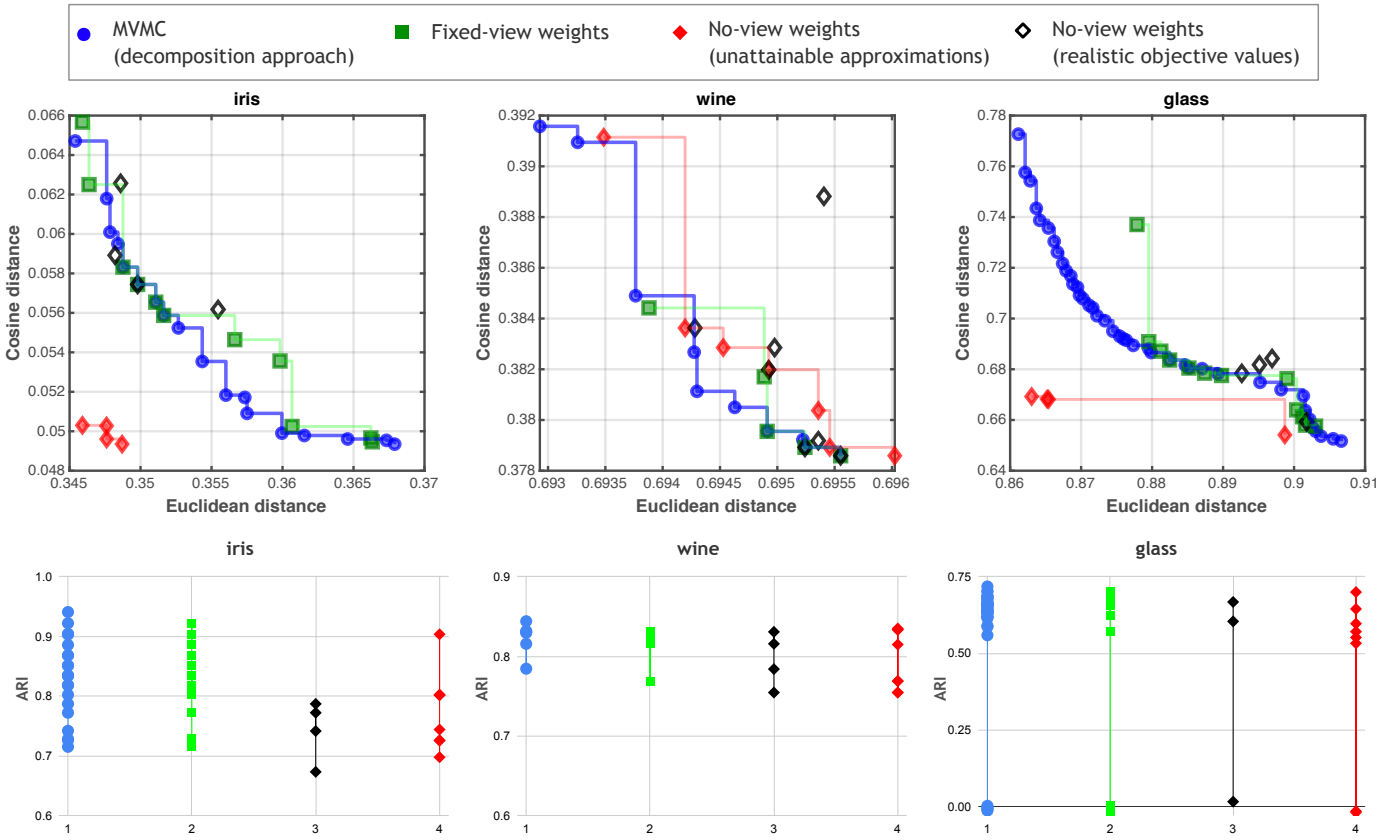

Figure 10: Illustration of the importance of the decoding step in moving from a medoid-based representation to an actual partition. Shown are the approximation fronts and ARI values obtained when running the same setup of MOEA/D, but varying just the decoding mechanism to (blue) rely on the scalarizing-vector (MVMC), (green) rely on a single fixed (equal) weighting of all views. In (red) and (black), we consider Jiang et al.'s (2016) decoding approach. In red, we can see the ARI values observed for each of the two possible partitions associated with a given candidate solution (set of medoids). In the approximation front, we can observe the resulting estimation of the trade-offs achievable during the search, which is likely to be overestimated as the vectors aggregate objective values across two different partitions. In (black), we illustrate the effect of decoding the same candidate solutions (sets of medoids) using a single fixed weight vector (i.e., adopting the approach deployed in [11] at the end of the search). In the ARI plot (bottom row), we can observe that this decoding results in a smaller number of solutions (only one partition per set of medoids) and a realistic assessment of the objective values of these solutions in the approximation fronts (top row). From these plots, the impact of the decoding step on the outcome of the search process, and the improved coverage afforded by the scalarizing approach, are evident. As one might expect, a fixed weight vector performs poorly in covering the extremes of the Pareto front. For Jiang et al.'s (2016) approach, there is some evidence that the use of potentially unachievable objective vectors during the search hinders appropriate convergence to the Pareto front.

MVMC is successful. Our results demonstrate that the latter scenario is more frequent, emphasizing the value of a multiview approach. MVMC achieved highly competitive results across our experiments compared with respect to several traditional single-objective clustering techniques and two state-of-the-art approaches to multiview clustering.

The key limitations of our work lie in the model selection from the Pareto front. As shown in our results, the use of an unsupervised selection approach results in a significant performance drop compared to the best solution available and reduces the performance advantage relative to the reference methods. Therefore, the immediate priority for our work are improvements to the model selection step, to allow unsupervised selection of the best solution from the Pareto front approximations. Furthermore, it would be interesting to extend the algorithm to determine the optimal number of clusters automatically. Additionally, to confirm that MVMC can provide meaningful advantages in practice, we are hoping to explore its application in large-scale multiview data applications, such as document clustering and medical data.

Acknowledgments

The first author acknowledges the support from CONACYT-Mexico through a postdoctoral fellowship.

Appendix A. Additional material

This appendix includes figures and tables complementing the results of the experiments presented in Sections 5, 6 and 7. Regarding the results on synthetic datasets, Figure A.11 illustrates the diversity of cluster structures covered by our test datasets, whereas Table A.5 presents the clustering results for the different algorithms studied for this experiment. Table A.6

shows the results obtained for real-world data in terms of average ARI values. Table A.7 details the results in terms of ACC, AUC, SEN, and SPE indices for the different methods considered in this study.

References

[1] Sergios Theodoridis and Konstantinos Koutrumbas. *Pattern Recognition*. Elsevier Inc., fourth edition, 2009.

[2] Andrzej Bielecki and Mateusz Wójcik. Hybrid System of ART and RBF Neural Networks for Online Clustering. *Applied Soft Computing*, 58:1–10, 2017.

[3] Patrick C. H. Ma, Keith C. C. Chan, Xin Yao, and David K. Y. Chiu. An Evolutionary Clustering Algorithm for Gene Expression Microarray Data Analysis. *IEEE Transactions on Evolutionary Computation*, 10(3):296–314, 2006.

[4] Chih Chin Lai and Chuan Yu Chang. A Hierarchical Evolutionary Algorithm for Automatic Medical Image Segmentation. *Expert Systems with Applications*, 36(1):248–259, 2009.

[5] Francisco de A.T. de Carvalho, Yves Lechevallier, and Filipe M. de Melo. Partitioning Hard Clustering Algorithms based on Multiple Dissimilarity Matrices. *Pattern Recognition*, 45(1):447–464, 2012.

[6] Francisco de A.T. de Carvalho, Yves Lechevallier, Thierry Despeyroux, and Filipe M. de Melo. Multi-view Clustering on Relational Data. In *Advances in Knowledge Discovery and Management*, pages 37–51. Springer, 2014.

[7] Arturo Rodríguez-Cristerna, Wilfrido Gómez-Flores, and Wagner Coelho de Albuquerque Pereira. A Computer-Aided Diagnosis System for Breast Ultrasound based on Weighted BI-RADS Classes. *Computer Methods and Programs in Biomedicine*, 153:33–40, 2018.

[8] Adán José-García, Julia Handl, Wilfrido Gómez-Flores, and Mario Garza-Fabre. Many-view Clustering: An Illustration using Multiple Dissimilarity Measures. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '19, pages 213–214. ACM, 2019.

[9] Ariel E Bayá and Pablo M Granitto. How Many Clusters: A Validation Index for Arbitrary-Shaped Clusters. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(2):401–14, 2013.

[10] Guoqing Chao, Shiliang Sun, and Jinbo Bi. A Survey on Multi-View Clustering. *arXiv e-prints*, page arXiv:1712.06246, 2018.

[11] Bo Jiang, Feiyue Qiu, Shipin Yang, and Liping Wang. Evolutionary Multi-Objective Optimization for Multi-View Clustering. In *IEEE Congress on Evolutionary Computation*, CEC 2016, pages 3308–3315. IEEE, 2016.

[12] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary Many-objective Optimization: A Short Review. In *IEEE Congress on Evolutionary Computation*, CEC 2008, pages 2419–2426. IEEE, 2008.

[13] Xiao Cai, Feiping Nie, and Heng Huang. Multi-View K-Means Clustering on Big Data. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2598–2604, Beijing, China, 2013. AAAI Publications.

[14] Xiang Wang, Buyue Qian, Jieping Ye, and Ian Davidson. Multi-Objective Multi-View Spectral Clustering via Pareto Optimization. In *Proceedings of the International Conference on Data Mining*, SIAM 2013, pages 234–242, Philadelphia, PA, 2013. Society for Industrial and Applied Mathematics.

[15] Sriparna Saha, Sayantan Mitra, and Stefan Kramer. Exploring Multiobjective Optimization for Multiview Clustering. *ACM Transactions on Knowledge Discovery from Data*, 20(2):1–30, 2018.

[16] Cong Liu, Jie Liu, Dunlu Peng, and Chunxue Wu. A General Multiobjective Clustering Approach Based on Multiple Distance Measures. *IEEE Access*, 6:41706–41719, 2018.

[17] Cong Liu, Qianqian Chen, Yingxia Chen, and Jie Liu. A Fast Multiobjective Fuzzy Clustering with Multimeasures Combination. *Mathematical Problems in Engineering*, 2019:1–21, 2019.

[18] Francisco de A.T. de Carvalho, Filipe M. de Melo, and Yves Lechevallier. A multi-view Relational Fuzzy c-medoid Vectors Clustering Algorithm. *Neurocomputing*, 163:115–123, 2015.

[19] J.Z. Huang, M.K. Ng, Hongqiang Rong, and Zichen Li. Automated Variable Weighting in k-means type Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):657–668, 2005.

[20] Taiyun Kim, Irene Rui Chen, Yingxin Lin, Andy Yi-Yang Wang, Jean Yee Hwa Yang, and Pengyi Yang. Impact of Similarity Metrics on Single-cell RNA-seq Data Clustering. *Briefings in Bioinformatics*, 2018.

[21] Adán José-García and Wilfrido Gómez-Flores. Evolutionary Clustering Using Multi-prototype Representation and Connectivity Criterion. In *Mexican Conference on Pattern Recognition*, MCPR 2017, pages 63–73. Springer, 2017.

[22] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting Similarities among Languages for Machine Translation. *arXiv e-prints*, page arXiv:1309.4168, 2013.

[23] Kalyanmoy Deb and Himanshu Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2013.

[24] Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

[25] Hui Li and Qingfu Zhang. Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, 2009.

[26] Indraneel Das and J. E. Dennis. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

[27] Kaisa Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, 1999.

[28] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. NP-Hardness of Euclidean Sum-of-Squares Clustering. *Machine Learning*, 75(2):245–248, 2009.

[29] Adán José-García and Wilfrido Gómez-Flores. Automatic Clustering Using Nature-Inspired Metaheuristics: A Survey. *Applied Soft Computing*, 41:192–213, 2016.

[30] Gilbert Syswerda. Shedule Optimization Using Genetic Algorithms. In Davis Lawrance, editor, *Handbook of Genetic Algorithms*, chapter 21, pages 332–349. Van Nostrand Reinhold, New York, NY, 1991.

[31] J. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, 1967.

[32] Samir Kanaan-Izquierdo, Andrey Ziyatdinov, and Alexandre Perera-Lluna. Multiview and Multifeature Spectral Clustering Using Common Eigenvectors. *Pattern Recognition Letters*, 102:30–36, 2018.

[33] Sayantan Mitra, Mohammed Hasanuzzaman, and Sriparna Saha. A Unified Multi-view Clustering Algorithm Using Multi-objective Optimization Coupled with Generative Model. *ACM Transactions on Knowledge Discovery from Data*, 14(1):1–31, 2020.

[34] Lawrence Hubert and Phipps Arabie. Comparing Partitions. *Journal of Classification*, 2(1):193–218, 1985.

[35] Liam Paninski. Estimation of Entropy and Mutual Information. *Neural Computation*, 15(6):1191–1253, 2003.

[36] Julia Handl and Joshua Knowles. An Evolutionary Approach to Multiobjective Clustering. *IEEE Transactions on Evolutionary Computation*, 11(1):56–76, 2007.

[37] M. Lichman. UCI Machine Learning Repository. `http://archive.ics.uci.edu/ml`, 2013.

[38] Wei Peng, Qingfu Zhang, and Hui Li. Comparison between MOEA/D and NSGA-II on the Multi-objective Travelling Salesman Problem. In *Multi-objective Memetic Algorithms*, pages 309–324. Springer, 2009.

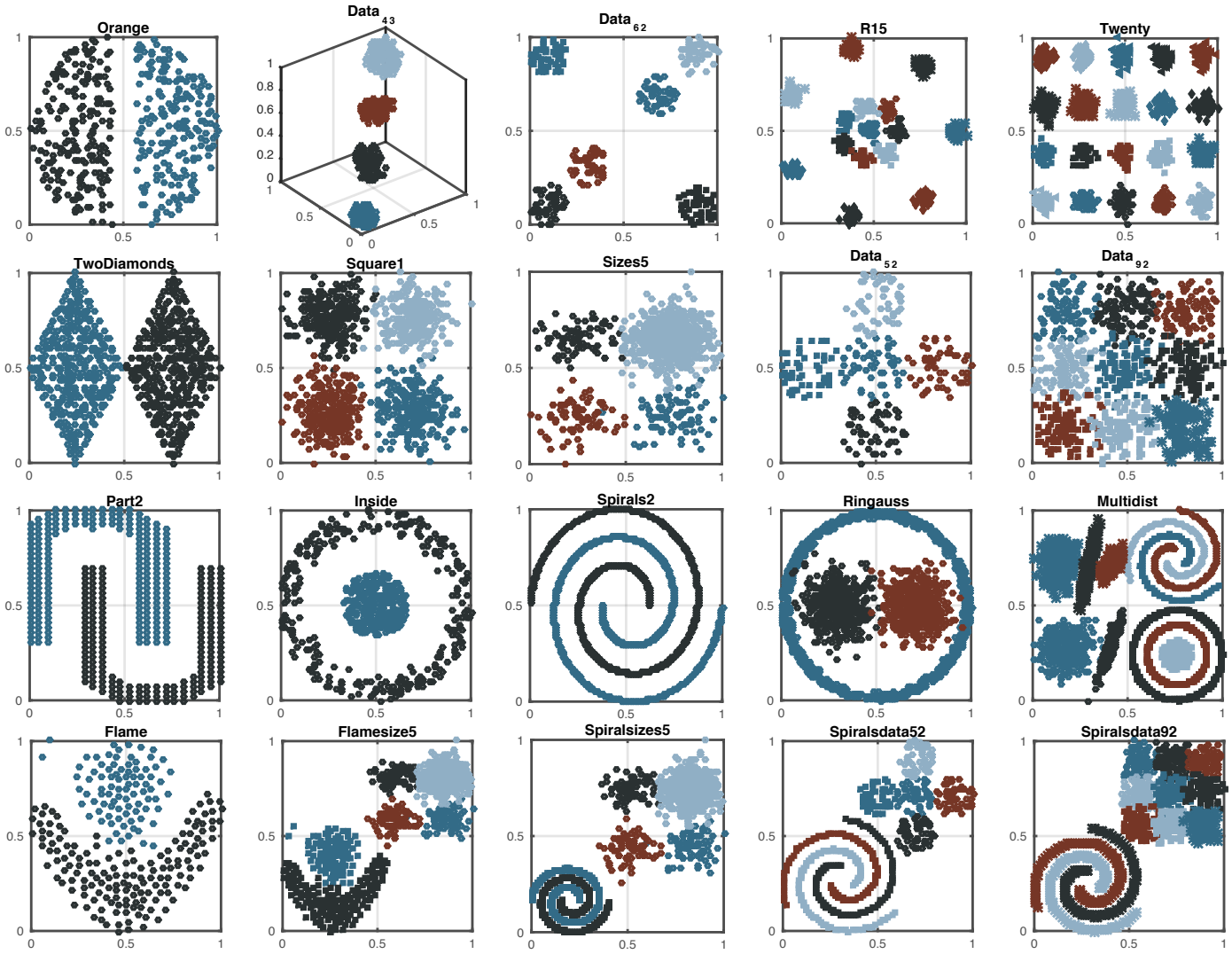

Figure A.11: This figure illustrates the diversity of properties covered by our collection of test datasets. The figures in the first row contain well-separated clusters. The figures in the second row correspond to overlapping clusters. The figures in the third row consist of nonlinearly separable clusters. Finally, the figures in the last row are datasets having mixtures of different data distributions.

Table A.5: ARI values on synthetic datasets obtained by the MVMC strategies MVMC_{SIL} and MVMC_{ARI}, and the clustering algorithm *k*-means, SL, WARD, GCA, and MVSC (mean values from 31 runs). The data views are derived using two different dissimilarity measures: Euclidean (▲) and MED_{euc} (▼). The best ARI value scored for each dataset has been shaded and highlighted in bold and, additionally, the statistically best (α = 0.05) results are highlighted in boldface.

					***k*-means**		**SL**		**WARD**		**GCA**		**MVSC**	**MVMC_{SIL}**	**MVMC_{ARI}**
Cat.	**Dataset**	**N**	**D**	**K**	**▲**	**▼**	**▲**	**▼**	**▲**	**▼**	**▲**	**▼**	**▲▼**	**▲▼**	**▲▼**
	Orange	400	2	2	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	Data_4_3	400	3	4	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
$\mathcal{G}1$	**Data_6_2**	300	2	6	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
	R15	600	2	15	**0.966**	0.975	0.548	0.751	**0.978**	0.939	**0.988**	0.975	**0.978**	**0.993**	**0.993**
	Twenty	1000	2	20	0.966	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.999**	**1.000**	**1.000**	**1.000**
	TwoDiamonds	800	2	2	**1.000**	0.895	0.000	0.000	0.748	0.684	**1.000**	0.888	**0.976**	**1.000**	**1.000**
	Square1	1000	2	4	**0.973**	0.906	0.000	0.000	**0.973**	0.777	0.971	0.902	0.649	**0.973**	**0.979**
$\mathcal{G}2$	**Size5**	1000	2	4	**0.920**	0.739	0.025	0.015	0.436	0.597	0.391	0.730	0.451	**0.944**	**0.962**
	Data_5_2	250	2	5	0.870	0.750	0.189	0.394	**0.895**	0.714	0.841	0.742	**0.915**	0.834	**0.930**
	Data_9_2	900	2	9	**0.831**	0.506	0.000	0.000	0.748	0.403	**0.823**	0.488	0.686	**0.825**	**0.838**
	Part2	417	2	2	0.265	**1.000**	**1.000**	**1.000**	0.433	**1.000**	0.237	**1.000**	**1.000**	**1.000**	**1.000**
	Inside	600	2	2	0.008	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.134	**1.000**	**1.000**	**1.000**	**1.000**
$\mathcal{G}3$	**Spirals**	1000	2	2	0.074	**1.000**	**1.000**	**1.000**	0.153	**1.000**	0.086	**1.000**	**1.000**	**1.000**	**1.000**
	Ringauss	2000	2	3	0.252	**0.963**	0.001	0.001	0.267	**0.972**	0.260	**0.943**	0.001	**0.971**	**0.972**
	Multidist	3012	2	11	0.532	**0.991**	0.805	0.805	0.584	**0.994**	0.604	**0.993**	0.878	**0.979**	**0.984**
	Flame	240	2	2	0.462	**0.910**	0.013	0.013	0.253	0.013	0.489	**0.854**	0.013	0.561	**0.967**
	Flamesize5	240	2	6	0.926	0.824	0.489	0.657	**0.932**	0.650	0.514	0.533	0.676	**0.948**	**0.976**
$\mathcal{G}4$	**Spiralsizes5**	2000	2	6	0.659	0.833	0.555	0.782	0.432	**0.861**	0.644	0.799	0.680	**0.974**	**0.980**
	Spiralsdata52	562	2	8	0.342	0.934	0.772	0.808	0.308	0.809	0.347	0.932	**0.978**	**0.948**	**0.960**
	Spiralsdata92	1212	2	12	0.610	0.623	0.128	0.130	0.610	0.538	**0.662**	0.647	0.560	**0.750**	**0.878**
	Mean ARI				0.688	0.892	0.526	0.568	0.688	0.798	0.650	0.871	0.772	**0.935**	**0.971**
	STD ARI				0.332	0.139	0.436	0.437	0.305	0.262	0.317	0.160	0.307	**0.114**	**0.044**

Table A.6: ARI values on real-world datasets obtained by MVMC_{ARI}, and the clustering algorithm *k*-means, SL, WARD, and GCA (mean values from 31 runs). The data views are derived using four different dissimilarity measures: Euclidean (▲), MED_{euc} (▼), Cosine (△), and MED_{cos} (▽). The best ARI value scored for each dataset has been shaded and highlighted in bold and, additionally, the statistically best (α = 0.05) results are highlighted in boldface.

Dataset	***k*-means**				**WARD**				**GCA**				**MVMC_{ARI}**					
	▲	**▼**	**△**	**▽**	**▲**	**▼**	**△**	**▽**	**▲**	**▼**	**△**	**▽**	**▲▼**	**▲△**	**▲▽**	**△▼**	**△▽**	**▼▽**
Iris	0.730	0.717	**0.904**	0.726	0.773	0.530	0.834	0.558	0.731	0.803	**0.902**	0.748	**0.922**	**0.927**	**0.928**	**0.922**	**0.903**	0.835
Wine	**0.915**	0.550	0.805	0.612	**0.932**	0.355	0.673	0.442	0.768	0.650	0.826	0.610	0.839	**0.850**	**0.864**	0.836	0.837	0.680
Breast	0.861	0.747	0.856	0.877	0.867	0.823	**0.890**	**0.890**	0.858	0.861	0.856	**0.884**	0.837	0.876	**0.881**	0.873	0.872	0.872
Thyroid	**0.718**	0.590	0.174	0.144	0.672	0.484	0.133	0.094	0.702	0.485	0.181	0.105	**0.834**	**0.730**	**0.878**	**0.837**	0.195	**0.869**
Glass	0.634	0.455	0.306	0.465	**0.656**	0.605	0.311	**0.711**	0.567	0.433	0.025	0.634	0.565	**0.729**	0.572	**0.694**	**0.748**	**0.664**
Ecoli	0.417	0.406	0.367	**0.574**	0.454	0.495	0.396	**0.550**	0.399	0.328	0.399	0.537	0.518	0.408	**0.736**	0.500	**0.706**	**0.650**
Mean ARI	0.713	0.577	0.569	0.566	0.726	0.549	0.539	0.509	0.671	0.593	0.532	0.586	0.753	**0.753**	**0.810**	**0.777**	0.710	0.762
STD ARI	0.163	0.152	0.294	0.235	0.171	0.157	0.305	0.334	0.150	0.205	0.349	0.246	0.155	**0.199**	**0.123**	**0.145**	0.241	0.101

Table A.7: Detailed results regarding the indices ACC, AUC, SEN, and SPE on all the multiview problem configurations (mean values from 31 runs). The best value scored for each index and problem configuration has been shaded and highlighted in bold and, additionally, the statistically best ($\alpha = 0.05$) results are highlighted in boldface. M_{SIL}, M_{AUC} and M_{ACC} denote different MVMC strategies for selecting the best clustering solution, whereas km denotes the results obtained by the k-means algorithm.

Conf.	**ACC**					**AUC**					**SEN**					**SPE**				
	k**m**	**MVS**	$\mathbf{M}_{SIL}$	$\mathbf{M}_{AUC}$	$\mathbf{M}_{ACC}$	k**m**	**MVS**	$\mathbf{M}_{SIL}$	$\mathbf{M}_{AUC}$	$\mathbf{M}_{ACC}$	k**m**	**MVS**	$\mathbf{M}_{SIL}$	$\mathbf{M}_{AUC}$	$\mathbf{M}_{ACC}$	k**m**	**MVS**	$\mathbf{M}_{SIL}$	$\mathbf{M}_{AUC}$	$\mathbf{M}_{ACC}$
V2-SM	0.852	0.730	0.834	**0.857**	**0.863**	**0.835**	0.738	0.786	**0.846**	**0.836**	**0.781**	0.764	0.632	**0.811**	0.751	0.889	0.712	**0.940**	0.881	**0.921**
V2-SO	0.846	**0.853**	0.826	0.844	0.845	**0.818**	**0.844**	0.773	0.812	0.810	**0.732**	**0.812**	0.608	0.709	0.696	0.905	0.875	**0.939**	**0.915**	**0.923**
V2-SE	0.758	0.765	**0.834**	**0.845**	**0.849**	0.777	0.781	0.786	**0.828**	**0.817**	**0.837**	**0.831**	0.632	0.775	0.716	0.717	0.731	**0.939**	0.881	**0.918**
V2-SP	**0.843**	0.754	0.832	**0.840**	**0.845**	**0.816**	0.759	0.783	**0.819**	**0.812**	0.733	**0.775**	0.627	**0.749**	0.709	0.900	0.742	**0.938**	0.888	**0.916**
V2-MO	**0.816**	0.655	0.743	**0.826**	**0.827**	**0.816**	0.700	0.717	**0.827**	**0.825**	0.817	**0.841**	0.634	**0.831**	**0.821**	**0.815**	0.559	0.800	**0.823**	**0.830**
V2-ME	0.669	0.691	**0.823**	**0.825**	**0.825**	0.712	0.719	**0.827**	**0.829**	**0.829**	**0.846**	0.808	**0.839**	**0.841**	**0.839**	0.577	0.631	**0.815**	**0.816**	**0.818**
V2-MP	0.840	0.709	0.837	**0.846**	**0.848**	0.836	0.671	**0.838**	**0.846**	**0.844**	0.824	0.552	**0.842**	**0.849**	**0.832**	**0.848**	0.791	**0.835**	**0.844**	**0.856**
V2-OE	0.622	**0.755**	0.733	0.732	**0.740**	0.657	**0.778**	0.703	**0.738**	0.725	**0.765**	**0.852**	0.608	**0.758**	0.677	0.548	0.705	**0.798**	0.718	**0.773**
V2-OP	0.619	**0.754**	0.750	0.749	**0.767**	0.646	**0.773**	0.679	**0.726**	0.715	**0.731**	**0.831**	0.457	0.652	0.549	0.561	0.714	**0.902**	0.799	**0.881**
V2-EP	0.615	0.515	0.447	**0.561**	**0.561**	**0.644**	0.475	0.407	**0.561**	**0.556**	**0.738**	0.350	0.280	**0.560**	**0.542**	**0.551**	**0.601**	0.535	**0.561**	**0.571**
V3-SMO	**0.852**	0.714	0.811	**0.859**	**0.866**	**0.836**	0.739	0.764	**0.845**	**0.841**	**0.784**	**0.819**	0.615	**0.799**	0.762	**0.888**	0.660	**0.913**	**0.890**	**0.920**
V3-SME	0.778	0.723	0.834	**0.857**	**0.862**	0.792	0.745	0.786	**0.848**	**0.839**	**0.836**	0.813	0.631	**0.817**	0.764	0.748	0.676	**0.940**	0.878	**0.914**
V3-SMP	0.855	0.774	0.835	**0.864**	**0.870**	0.833	0.768	0.786	**0.856**	**0.853**	0.765	0.748	0.630	**0.830**	**0.796**	0.902	0.787	**0.943**	0.882	**0.909**
V3-SOE	0.767	0.775	0.806	**0.843**	**0.850**	0.781	0.791	0.757	**0.826**	**0.820**	**0.828**	**0.841**	0.603	0.773	0.725	0.735	0.741	**0.911**	**0.879**	**0.915**
V3-SOP	**0.846**	0.828	0.757	**0.847**	**0.852**	0.818	**0.824**	0.684	**0.824**	**0.817**	0.727	**0.809**	0.454	**0.751**	0.707	**0.909**	0.838	**0.915**	0.898	**0.927**
V3-SEP	0.758	0.776	**0.835**	**0.851**	**0.857**	0.777	0.781	0.785	**0.839**	**0.830**	**0.838**	**0.798**	0.628	**0.800**	0.747	0.716	0.765	**0.942**	0.877	**0.914**
V3-MOE	0.674	0.687	**0.732**	**0.826**	**0.828**	0.716	**0.725**	0.704	**0.829**	**0.829**	**0.849**	**0.842**	0.615	**0.841**	0.830	0.582	0.607	**0.792**	**0.818**	**0.827**
V3-MOP	0.827	0.748	0.752	**0.849**	**0.856**	0.823	0.761	0.684	**0.847**	**0.840**	**0.809**	0.801	0.468	**0.841**	0.788	0.836	0.721	**0.899**	**0.853**	**0.892**
V3-MEP	0.672	0.687	0.826	**0.849**	**0.853**	0.714	0.697	**0.829**	**0.849**	**0.846**	**0.846**	0.730	**0.838**	**0.847**	**0.822**	0.581	0.665	0.820	**0.850**	**0.869**
V3-OEP	0.621	0.747	0.750	**0.772**	**0.784**	0.653	**0.765**	0.680	**0.768**	**0.755**	**0.755**	**0.822**	0.458	**0.756**	0.662	0.550	0.708	**0.903**	0.781	**0.848**
V4-SMOE	0.782	0.721	0.833	**0.860**	**0.865**	0.794	0.747	0.783	**0.849**	**0.844**	**0.835**	**0.832**	0.624	**0.818**	0.775	0.754	0.662	**0.942**	0.881	**0.912**
V4-SMOP	0.853	0.777	0.755	**0.866**	**0.871**	0.835	0.782	0.682	**0.855**	**0.849**	0.778	**0.797**	0.450	**0.819**	0.778	0.893	0.767	**0.914**	0.891	**0.919**
V4-SMEP	0.781	0.744	0.833	**0.866**	**0.869**	0.795	0.750	0.784	**0.856**	**0.851**	**0.838**	0.768	0.627	**0.822**	0.793	0.751	0.732	**0.941**	0.889	**0.909**
V4-SOEP	0.775	0.796	0.753	**0.846**	**0.854**	0.787	0.801	0.684	**0.832**	**0.828**	**0.825**	**0.818**	0.466	0.789	0.743	0.750	0.785	**0.902**	**0.876**	**0.912**
V4-MOEP	0.671	0.722	0.753	**0.851**	**0.857**	0.713	0.745	0.683	**0.847**	**0.839**	**0.847**	**0.818**	0.464	**0.833**	0.785	0.580	0.673	**0.903**	**0.860**	**0.894**
V5-SMOEP	0.789	0.759	0.751	**0.851**	**0.857**	0.800	0.770	0.680	**0.847**	**0.842**	**0.834**	0.807	0.456	**0.834**	0.796	0.766	0.734	**0.904**	**0.859**	**0.888**