Evolutionary Multi-objective Clustering Over Multiple Conflicting Data Views

Mario Garza-Fabre, Julia Handl, Adán José-García

Evolutionary Multiobjective Clustering Over Multiple Conflicting Data Views

Mario Garza-Fabre, Julia Handl, and Adán José-García

***Abstract*—Multiview data analysis provides an effective means to integrate the distinct information sources which are inherent to many applications. Data clustering in a multiview setting specifically aims to identify the most appropriate grouping for a collection of entities, where those entities (or their relationships) can be described from multiple perspectives. Leveraging recent advances in multiobjective clustering, we propose a new evolutionary method to tackle this challenge. Designed around a flexible and unbiased solution representation, together with strategies based on the minimum spanning tree and neighborhood relations, our algorithm optimizes multiple objectives simultaneously to effectively explore the space of candidate tradeoffs between the data views. Through a series of experiments, we investigate the suitability of our proposal in the context of a bioinformatics application, clustering of plausible protein structures, and a diverse set of synthetic problems. The specific case of two data views is considered in this article. The evaluation with respect to a variety of reference approaches demonstrates the effectiveness of our method in discovering high-quality partitions in a multiview setting. Robustness against unreliable data sources and the ability to automatically determine the number of clusters are additional advantages evidenced by the results obtained.**

***Index Terms*—Clustering methods, multiobjective clustering, multiview learning, representation, unsupervised learning.**

I. Introduction

HOW CAN we balance the bias introduced by model assumptions and the efficiency of the subsequent search over model parameters? The definition of a flexible, yet efficient representation is arguably a key requirement for the design of a versatile algorithm for multiobjective clustering. Previously, this has been achieved through the adoption of a graph-based encoding, and the use of the minimum spanning tree (MST) and nearest neighbor relations to direct the search toward the most plausible regions of the solution space [1], [2]. The resulting approach assumes the preservation of local neighborhood relationships only, without introducing bias at the level of the individual clusters (e.g., regarding shape) or intercluster relations. Consequently, it is sufficiently flexible to be useful in the context of vastly different clustering criteria.

Can such a framework be extended to multiview clustering (MVC)? MVC encompasses scenarios in which multiple data sources exist (e.g., separate feature spaces or different measures to determine relational information), and where the potential complementarity of these sources should be exploited to produce higher quality partitions. Previous multiobjective clustering approaches [1], [2] reach their limits in such a setting, as the computation of the MST and the determination of neighbor relations is currently rooted in a single, specific dissimilarity space. This potentially prevents the fair and full consideration of multiple views during the search process, hindering the direct application of the approach to MVC.

This article focuses on addressing this limitation. Our aim is to propose a new MVC method that can benefit from the flexibility of a graph-based representation but also supports the effective integration of multiple data views. We explore two different strategies for the use of MST information, effectively redefining the search space of candidate partitions from the perspective of multiple views. Our method capitalizes on these strategies by framing MVC as a multiobjective problem and simultaneously optimizing separate objectives to account for individual data views. The resulting approach has the ability to identify the number of clusters automatically, but can also exploit this domain knowledge, when available.

We analyze the suitability of our proposal in the context of a bioinformatics application: the clustering of candidate protein structures. In addition, we consider a collection of synthetic datasets of varying characteristics. Our analysis involves comparisons against a set of single-view and multiview (including multiobjective) reference approaches. It is important to highlight that the design of our algorithm, as described in this article, assumes the use of a reduced number of data views (two or three), and that our experimental evaluation includes test scenarios with two data views only. Although the implications of an increased number of views and potential adaptations are discussed, the evaluation of our proposal under this setting is beyond the scope of this study.

The organization of this article is as follows. First, Section II provides background concepts and discusses related works. Our proposed approach is introduced in Section III. Section IV describes our experimental setup. Section V presents the results and analyzes our findings. Finally, Section VI concludes.

Manuscript received 31 May 2022; revised 23 September 2022; accepted 25 October 2022. Date of publication 7 November 2022; date of current version 1 August 2023. This work was supported in part by the Engineering and Physical Sciences Research Council, U.K., under Grant EP/M013766/1. *(Corresponding author: Mario Garza-Fabre.)*

Mario Garza-Fabre is with the Center for Research and Advanced Studies, Cinvestav Campus Tamaulipas, 87130 Ciudad Victoria, Mexico (e-mail: mario.garza@cinvestav.mx).

Julia Handl is with the Management Sciences Group, University of Manchester, M15 6PB Manchester, U.K. (e-mail: julia.handl@manchester.ac.uk).

Adán José-García is with CNRS, Centrale Lille, Universite de Lille, 59000 Lille, France (e-mail: adan.josegarcia@univ-lille.fr).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TEVC.2022.3220187.

Digital Object Identifier 10.1109/TEVC.2022.3220187

II. Background and Related Work

In this section, the necessary background is presented together with a discussion of the most relevant literature.

A. Multiobjective Clustering

We are interested in a type of machine learning problem that is prevalent in unsupervised learning settings and regularly appears across a wide range of application domains. Cluster analysis involves the identification of distinct clusters (groups) from a given collection of N data entities, $X = \{x_1, \ldots, x_N\}$, such that these clusters reflect the similarities and differences between those entities. More formally, the goal is to find the best possible partition of X into k disjoint clusters, $C^* = \{c_1, \ldots c_k\}$, such that $f(C^*) = \min\{f(C)|C \in \Omega\}$, without loss of generality. Here, Ω denotes the set of all possible partitions, i.e., the search space, and $f : \Omega \to \mathbb{R}$ is a clustering criterion (also known as cluster validity index).

From the above definition, the important role of criterion f is evident: it is responsible for enabling an effective differentiation between candidate solutions and, therefore, needs to correctly evaluate the properties that determine partition quality. Although no consensus exists regarding those properties, most definitions agree that the fundamental characteristics of clusters are homogeneity (compactness) within and heterogeneity (separation) across groups. Many clustering criteria have been proposed [3], [4], each evaluating, in particular ways, one of these properties or a combination of them. It is unlikely, however, that a single solution can simultaneously optimize all the desirable, but usually conflicting, characteristics [5].

To the best of our knowledge, Delattre and Hansen [6] were the first to recognize the multicriterion nature of clustering, proposing an algorithm which relies on two specific criteria. More generally, posed as a multiobjective optimization problem, clustering becomes the problem of optimizing a vector function $\mathbf{f}(C) = [f_1(C), \ldots, f_m(C)]^T$, where $f_i : \Omega \to \mathbb{R}$ denotes the ith clustering criterion to be considered. Under this formulation, and given the potential conflict between the m optimization criteria, the goal now becomes to identify the best possible tradeoff solutions, namely, to find the *Pareto-optimal set* $P^* = \{C^* \in \Omega \mid \nexists C \in \Omega : C \prec C^*\}$, whose image in the objective function space is the so-called *Pareto front*.[1] The simultaneous optimization of multiple clustering criteria affords a more comprehensive description of cluster quality and often leads to the discovery of higher quality partitions, which can be difficult to obtain by optimizing a single criterion.

Multiobjective approaches for clustering started to attract increasing attention when *multiobjective evolutionary algorithms* (MOEAs) were proposed to tackle this problem [1], [7]. MOEAs (and other population-based metaheuristics) are well suited to the multiobjective formulation, as they can approximate P^* in a single execution and exhibit flexibility with respect to the specific optimization criteria used [8].

[1] $\prec$ denotes the *Pareto-dominance* relation: C *dominates* C' ($C \prec C'$) if and only if $\forall i : f_i(C) \leq f_i(C') \wedge \exists j : f_j(C) < f_j(C'), i, j \in \{1, \ldots, m\}$. All solutions in P^* are said to be *nondominated* with respect to each other.

B. Multiview Clustering

MVC extends the definition of clustering to leverage scenarios where multiple *data views* are available [9]. Each data view can be seen as a different representation (or modality) of the same data. The idea of MVC is to exploit the complementary perspectives that multiple views can provide, in order to discover higher quality partitions.

Scenarios with multiple data views present themselves in different forms and arise in diverse application domains. On the one hand, data views may correspond to multiple feature spaces characterizing the same entities. For example, José-García et al. [10] considered an application to breast tumor classification where five separate feature spaces describe different aspects of ultrasound images. Analogously, Devagiri et al. [11] used independent subsets of features to define views describing the operation, performance, and context of the heating and tap-water subsystems in the domain of smart buildings. Zeng et al. [12] reported an application of MVC to the identification of groups of closely related genes in a genome. To this end, the authors extract features from two heterogeneous sources, gene expression data and text literature data, and consider each as a separate data view. He et al. [13] used gene expression data as one of the views, in combination with a second view based on DNA methylation expression data, in an effort to identify groups of patients with specific cancer subtypes.

On the other hand, multiple distance or proximity measures can be exploited to obtain a more complete picture of the similarity between entities [14]. Multiple dissimilarity matrices can be determined through the application of either a single proximity measure to separate feature spaces, or different measures within a single, fixed feature space [15]. Liu et al. [16], for example, use both the Euclidean distance and the *path distance* [17] as distinct data views to increase robustness when dealing with a range of data properties (e.g., spherically or irregularly shaped clusters). Based on a similar rationale, José-García et al. [10] explored the use of the Euclidean, cosine, and *maximum edge* [18] distances as separate data views.

The use of multiple similarity measures as data views is particularly relevant in scenarios where defining feature spaces is not straightforward [10], [19], and there is significant ambiguity in how best to define similarity. For this reason, MVC has seen significant use in the analysis of Web-search results. Saha et al. [20] and Saini et al. [21] employed problem-specific similarity measures to define separate, semantic, and syntactic views. In contrast, Mitra et al. [22] combined semantic and syntactic information into a single view, the textual view, but use a separate similarity measure to generate an additional view from the images contained in Web documents. In this article, we explore another such application, the clustering of candidate protein structures. A meaningful feature space is difficult to define in this domain, and even pairwise comparisons

between structures are not straightforward. Exploiting multiple structural similarity measures may therefore provide a significant advantage in this particular context.

C. Multiobjective Approach to Multiview Clustering

The success of Pareto-based optimization for clustering has motivated recent interest in extending this approach to the MVC setting. Although complementary clustering criteria have sometimes been employed in an MVC setting [23], [24], the main focus has been on optimizing a single criterion, evaluated separately for each data view. The multiobjective approach to MVC is thus expected to facilitate the discovery of tradeoff partitions that reflect the consensus (or conflict) that may exist across the multiple data views available.

To the best of our knowledge, Brusco and Stahl report one of the earliest works explicitly formulating MVC as a multiobjective problem [25]. The authors consider multiple dissimilarity matrices computed independently from different feature sets, and then define separate objectives based on these matrices. However, the advantages of the multiobjective formulation are not fully exploited, as scalarization is applied to transform the problem back into a single-objective one.

Caballero et al. [23] also evaluated a single clustering criterion independently for multiple dissimilarity matrices (a scenario where different criteria are computed separately for multiple dissimilarity matrices is also considered). In contrast to the above work by Brusco and Stahl, Caballero et al. explicitly treat MVC as a multiobjective problem so as to approximate the Pareto front in a single execution of their scatter tabu search algorithm. In proposing a multiobjective spectral clustering approach to MVC, Wang et al. [26] emphasized the relevance of handling data views through individual optimization objectives: the multiobjective approach avoids assumptions of compatibility across data views (an inherent assumption of methods based on the direct aggregation of views), and removes the need to define weights that reflect the importance (or reliability) of these different sources.

Perhaps, the first application of MOEAs in the context of MVC is the one reported by Wahid et al. [27], [28]. In fact, multiple views are used only during the initial stage to seed the proposed clustering ensemble method with a diverse set of partitions. The subsequent, evolutionary-based stage optimizes the combination of clusters from these initial partitions. Another early application of MOEAs to MVC is reported by Saeidi et al. [24] to address the problem of software systems modularization. The authors find that treating MVC explicitly as a multiobjective problem, using a Pareto-based approach, produces better results in comparison to using a linear combination of the objective functions. More recently, Jiang et al. [29] evaluated the performance of five well-known MOEAs at addressing this task. This study shows that a multiobjective approach frequently obtains higher quality partitions than three single-objective MVC methods from the literature.

Liu et al. [16] designed MOEAs to tackle an MVC scenario consisting of two dissimilarity matrices, each computed using a different distance function. The first MOEA optimizes a measure of cluster compactness simultaneously for both matrices, and is intended for situations where k is known in advance. The second proposal modifies the objective functions to incorporate information on cluster separation, which facilitates the automatic discovery of k. In the works of Saha et al. [20] and Mitra et al. [22], each data view is handled as a separate objective function, but an additional objective is included to evaluate the agreement between the partitions encoded across the different views. Solutions to the resulting problem are obtained by using a simulated annealing-based multiobjective optimizer. A similar approach is later reported by Saini et al. [21], where alternative data views and a search engine based on differential evolution are considered.

Even though some of the above studies argue that the methods proposed can scale to any number of views and the corresponding number of objective functions [20], they primarily focus on problems with two or three views (our work also centers on problems with such a reduced number of views). A notable exception is the work of Jiang et al. [29], where problems involving four and six data views are considered. Recently, José-García et al. [10] proposed a *many-view* clustering approach with the aim of addressing this limitation and improving the applicability of MVC to scenarios where a larger number of views are available. This study evidences that having the opportunity to increase the number of views, and exploit the complementary information they provide, can translate into meaningful increases in clustering performance.

D. Discussion of the Related Works

Here, we further discuss the existing multiobjective approaches to MVC (Section II-C) from the perspective of two important design aspects: the solution representation used and the ability to automatically determine the number of clusters, k. With a few exceptions [21], [27], [28], most metaheuristic methods adopt either a label-based or a prototype-based representation. On the one hand, the label-based approach directly encodes cluster memberships [16], [24], specifying which cluster each of the N data entities belongs to. Despite offering a straightforward encoding, this representation scales poorly with respect to the problem size N and has been shown to be highly nonsynonymously redundant [30], [31].

On the other hand, the prototype-based representation encodes partitions most commonly by means of cluster centroids [20], [22], [29]. Although this approach presents better scalability properties, it does not offer a direct mapping to partition space under an MVC setting. That is, a solution is interpreted as a partition by assigning data entities to the cluster represented by the closest centroid; however, this is likely to result in multiple distinct partitions, as the notion of closeness changes with the different subsets of features or dissimilarity measures associated with each particular view. Moreover, this representation is only applicable if the views are available in the form of feature spaces, which is not always the case, as discussed at the end of Section II-B. An alternative to cope with this last issue is the use of cluster medoids rather than centroids [10], [23]. Also, it has been shown that the availability of scalarizing vectors, within a *many-objective*

optimization approach [32], can be exploited to address the assignment issue [10]. Nevertheless, prototype-based representations are known to be inherently biased toward spherically shaped clusters, which remains a limitation.

The multiobjective MVC methods discussed in Section II-C either require the value of k to be fixed in advance [10], [23], [29], or they keep this parameter variable and try to automatically determine its value during optimization [16], [20], [21]. The former approach is not always realistic, as the correct value of k can be unknown in practice. The latter is more generally applicable, but completely disregards any insight or domain expertise that might be available.

The multiobjective approach described in Section III uses a graph-based encoding which: 1) captures solutions with any number of clusters of arbitrary shapes; 2) ensures that every candidate solution is interpreted as the same partition across all data views; and 3) is explicitly leveraged as a mechanism for data view integration. Our method is able to identify k automatically but can also exploit the availability of this domain knowledge in order to improve clustering performance.

III. Proposed Algorithm

This section introduces the evolutionary multiobjective algorithm that we propose to address the challenge of MVC. Our method is referred to as ΔMV, since it builds upon the Δ-MOCK algorithm recently proposed in the context of multiobjective clustering [2]. Note, however, that ΔMV involves significant adaptations to handle the new MVC setting.

A key aspect of any MVC method is the specific mechanisms that enable the integration of multiple data views. In ΔMV, a graph-based solution encoding and strategies based on the MST allow us to define a search space of candidate partitions that represent different tradeoffs between the data views considered. Then, the simultaneous optimization of multiple objective functions, each accounting for a separate data view, allows ΔMV to search for the optimal tradeoff partitions within this set. Our proposal is able to automatically determine the number of clusters, k, but is also able to exploit any insight available regarding the value of this parameter. These are the main features distinguishing ΔMV from other multiobjective MVC approaches reported in the literature.

The overall optimization framework, main design components, and source code availability of our implementation of algorithm ΔMV, are discussed in the following sections.

A. Multiobjective Optimization Framework

The overall functioning of our ΔMV algorithm is outlined in Algorithm 1. In the first stage, problem granularity is defined (intended for scalability purposes, as described in Section III-B) and a population of candidate individuals (clustering solutions) is initialized. Then, in the optimization stage, the evolutionary cycle of renewing the population by means of mating, the genetic operators (recombination and mutation), and survivor selection, is repeated for a given number of generations ($G_{\max}$). The final stage produces a Pareto-front approximation and selects a promising nondominated individual in order to provide the user with a final solution recommendation. Note that when a value of k is provided as input, external archives are used to store the Pareto-front approximation, as explained in Section III-E. These archives are updated after initialization and every time a new set of individuals is created by means of the genetic operators.

Algorithm 1 Algorithm ΔMV

Require: Problem granularity level (δ), Population size (P), Generations ($G_{\max}$) [, Number of clusters (k)]
Ensure: Pareto front approximation ($\mathcal{P}^*$[, $\mathcal{P}_k^*$]), Final solution recommendation ($\mathbf{x}^*$)

STAGE 1: Problem preparation and initialization
set_problem_granularity(δ)
$\mathcal{P} \leftarrow$ *initialization*(P)
if ⟨ k provided as input ⟩ **then**
  $\mathcal{A}, \mathcal{A}_k \leftarrow$ *update_external_archives*($\mathcal{P}, P$)
STAGE 2: Evolutionary-based optimization
for *generation* $\leftarrow 1, \ldots, G_{\max}$ **do**
  $\hat{\mathcal{P}} \leftarrow$ *mating_selection*($\mathcal{P}$)
  $\mathcal{P}' \leftarrow$ *genetic_operators*($\hat{\mathcal{P}}$)
  if ⟨ k provided as input ⟩ **then**
    $\mathcal{A}, \mathcal{A}_k \leftarrow$ *update_external_archives*($\mathcal{P}', P$)
  $\mathcal{P} \leftarrow$ *survival_selection*($\mathcal{P} \cup \mathcal{P}'$)
STAGE 3: Model selection and output generation
if ⟨ k provided as input ⟩ **then**
  $\mathcal{P}^*, \mathcal{P}_k^* \leftarrow \mathcal{A}, \mathcal{A}_k$
  $\mathbf{x}^* \leftarrow$ *select_final_solution*($\mathcal{P}_k^*$)
else
  $\mathcal{P}^* \leftarrow$ *Pareto_nondominated*($\mathcal{P}$)
  $\mathbf{x}^* \leftarrow$ *select_final_solution*($\mathcal{P}^*$)

B. Graph-Based Representation of Variable Granularity

Choosing a suitable representation is critical, as this component effectively determines the space of clustering solutions that can be reached by the evolutionary algorithm. As such, we leverage our method's representation so that each potential solution defines a tradeoff concerning multiple data views. We provide a general description of our representation below, but details on how this approach is further adapted and exploited to cater for the MVC setting are discussed later in Section III-C.

The adopted graph-based representation, illustrated in Fig. 1, is based on the original proposal by Park and Song [33] and a coarse-grained version devised more recently [2], [34]. Rather than using a separate node to represent each of the N data entities, as done in the original representation, we preprocess the dataset in advance to identify groups of entities that can be handled jointly in a coarse-grained manner. This strategy can reduce solution length, therefore significantly shrinking the search space, which improves our ability to solve large problem instances reliably. Note, however, that our representation still preserves the advantages of the original proposal: it can encode partitions with varying numbers of clusters without the introduction of bias regarding their shape.

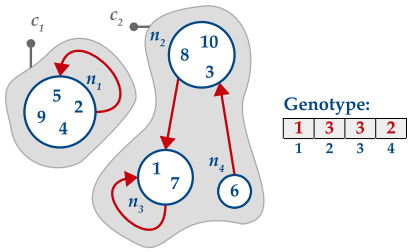

Fig. 1. Graph-based representation of variable granularity. A node of the graph can represent a set of entities and is accounted for by a separate gene of the genotype. The ith gene, when assuming an allele value of j, encodes a link connecting node n_i to node n_j; for example, the second gene of the genotype shown has a value of 3, which defines a link from node n_2 to node n_3. Here, four nodes represent $N = 10$ entities, and a partition with $k = 2$ clusters is defined by the connected components formed by the links encoded.

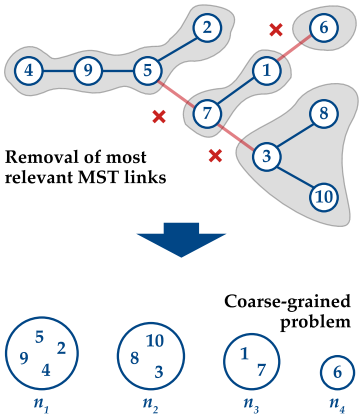

Fig. 2. Removal of relevant links splits the MST into a set of connected components, which serve as the building blocks of candidate partitions when using the coarse-grained encoding.

Defining our representation involves two steps: 1) setting the granularity level by identifying the groups of entities that will be represented as nodes of the graph and 2) deciding on the possible alleles for each gene of the genotype, which determines the links that can be established between the nodes.

The first step is crucial, as the initial groups of data entities serve as the basis of all candidate partitions generated during optimization. Building upon previous work [2], [34], these groups are defined by the connected components obtained when removing the most relevant subset of links from the MST, see Fig. 2. The relevance of an MST link is given by its *degree of interestingness* (DI) [2], whose computation details are provided in the supplementary material accompanying this article. Parameter δ is used to control problem granularity, denoting the percentage of the $N-1$ total links that will be retained ($0 \leq \delta < 100$). More specifically, $\lfloor(\delta/100)(N-1)\rfloor$ less relevant (lowest DI) MST links will be preserved, and only the remaining $\lceil([(100-\delta)]/100)(N-1)\rceil$ most prominent (highest DI) MST links will be removed to produce the initial set of cluster building blocks. Note that using $\delta = 0$ is equivalent to adopting the original (fine-grained) representation.

The second step enables further control of the regions of the search space on which ΔMV will focus. We restrict the set of possible allele values for the ith gene of the genotype to only those values that create links within the following three categories, which represent meaningful changes at the phenotype level (i.e., they either result in distinct nodes being separated or merged): 1) a link connecting node n_i to itself; 2) a link connecting n_i to a node n_j which contains one of the L nearest neighbors of a member of n_i; and 3) a link from n_i to a node n_j enclosing an entity to which a member of n_i was directly connected in the MST. A similar approach was considered previously [2], [34]. However, a key difference in this previous work is that links were defined at the data entity level, rather than at the node level, as done here (where each node may represent multiple entities). Consequently, although defining links between distinct pairs of entities, multiple alleles could ultimately translate into connections between the same pair of nodes, introducing redundancy. Our new approach removes this redundancy and ensures that possible changes at the phenotype level are sampled with equal probability. We adopt a setting of $L = 10$, as in previous studies [2], [34].

C. Adaptations to Integrate Multiple Data Views

We investigate two distinct approaches to integrate multiple data views, which are realizations of the strategies described in Section III-B for defining the initial set of cluster building blocks and the possible allele values of our method's representation. A subscript is used to distinguish between these variants of our ΔMV method, as specified below.

1) *Aggregation of Data Views (ΔMV_{avg}):* The dissimilarity matrices corresponding to all data views are first (independently) normalized to the range [0, 1]. Then, these matrices are averaged, resulting in a new dissimilarity matrix that is used during the construction of the MST and the generation of the nearest neighbor information (which is used to determine the relevance of MST links and possible allele values for each gene of the genotype).
2) *Separation of Data Views (ΔMV_{sep}):* MSTs are constructed independently for each data view and then individually split into multiple connected components. This results in different coarse-grained versions of the problem, which are later merged into a single final version. As further explained and graphically illustrated in the supplementary material, the nodes of the final coarse-grained problem are given by the (nonempty) intersections computed for all combinations of nodes from the individual problem versions. Given the potential lack of agreement between the data views, this merging process is likely to result in a final coarse-grained problem larger than expected in a single-view setting for the same value of parameter δ. Since separate MSTs are constructed, and the nearest neighbor information changes from one view to another, the definition of the possible allele values also needs to be adapted. The set of alleles for a given gene of the genotype will comprise those alleles that define links satisfying the conditions of the three categories described in Section III-B, for at least one of the data views.

Approach $\Delta\text{MV}_{\text{avg}}$ can be expected to be more successful in exploiting the existing consensus between the data views, as it biases the search toward those partitions. We hypothesize, however, that $\Delta\text{MV}_{\text{sep}}$ can be more suitable when the

views are conflicting with each other, or when one of them is significantly more reliable (e.g., less affected by noise) than the other; the separation of views enforced by $\Delta\text{MV}_{\text{sep}}$ gives it a better ability to cater for these potential situations.

An important aspect is the scalability of these approaches in terms of the number of data views. In principle, both the computation of an average dissimilarity matrix in $\Delta\text{MV}_{\text{avg}}$ and the merging of coarse-grained problem versions in $\Delta\text{MV}_{\text{sep}}$ can generalize to an arbitrary number of views. Nevertheless, increasing the number of views is expected to accentuate the above-discussed tendency of $\Delta\text{MV}_{\text{sep}}$ to increase the size of the resulting coarse-grained problem and the number of allele choices (leading to a larger search space), which diminishes the benefits of the variable-granularity representation.

D. Optimization Criteria and Delta-Evaluation

We formulate MVC as a multiobjective optimization problem. A single clustering criterion, the *average silhouette width* (ASW) [35], is computed independently for each data view, resulting in a set of objective functions. ASW evaluates aspects of both intracluster homogeneity and intercluster separation, showing a promising performance when compared with respect to other well-known clustering criteria [3].

The *silhouette width* estimates the extent to which an entity is correctly assigned to its cluster. This is evaluated by measuring the dissimilarity of the entity to all other entities within the same cluster, and contrasting this measurement to that computed with respect to the entities of the closest neighboring cluster. Criterion ASW is thus defined as the average of the silhouette width estimates across all entities

$$\text{ASW}(C) = \frac{1}{N}\sum_{i=1}^{N}\frac{b(i)-a(i)}{\max\{a(i),b(i)\}} \tag{1}$$

where C is a candidate partition; $a(i)$ is the average dissimilarity between data entity i, belonging to cluster $c \in C$, and all other entities $j \neq i$ such that $j \in c$; and $b(i)$ is the minimum of the average dissimilarities between i and all entities $l \in c'$, for any $c' \in C$ such that $c' \neq c$. ASW takes values in the range $[-1, 1]$ and is to be maximized. The supplementary material includes an illustrative example of the computation of ASW.

The use of a variable-granularity representation enables the delta-evaluation of candidate solutions. As explained in Section III-B, groups of data entities are initially identified, which represent a partial clustering upon which all other solutions will be constructed (i.e., by merging these initial groups). Hence, our method leverages this strategy and evaluates solutions incrementally: it first precomputes ASW in advance for such a partial solution, and then updates this criterion to account for the peculiarities of each final clustering evaluated during the search. We applied a similar strategy before [2], where the delta-evaluation of the *intracluster variance* and *connectivity* criteria was found to be a major contributor to the computational efficiency of algorithm Δ-MOCK.

E. Determination of the Number of Clusters

As discussed earlier, the genetic representation adopted enables the generation of solutions with variable k. Leveraging this flexibility, our method is able to determine k automatically, but it can also receive a recommendation as input to bias the search and ensure the delivery of solutions with specific k. In the former case, the optimization criteria are responsible for guiding the search toward partitions that fit the natural grouping of the data without introducing any particular bias on the value of k. Hence, due to the potential conflict between the data views (each accounted for by a separate objective function) and the fact that each view could favor partitions with different k, the Pareto-front approximation generated by ΔMV will likely contain solutions exhibiting a range of values for k. This approximation solution set, $\mathcal{P}^*$, together with a final solution recommendation, $\mathbf{x}^* \in \mathcal{P}^*$, is delivered by ΔMV as output, as shown in Algorithm 1.

In the latter case, ΔMV exploits available domain knowledge regarding the value of k. However, rather than enforcing this as a constraint, we acknowledge the fact that this information can be inaccurate in practice (it is unavailable in most cases), and treat it as a recommendation to bias the search process. This is achieved by using an additional (helper) objective function defined as the absolute difference between the number of clusters in the candidate partition and the target k value. In this way, solutions with the desired k are favored, but the simultaneous optimization of the original criteria still leads to solutions with different k whenever they better fit the real grouping of the data according to the views considered.

As indicated in Algorithm 1, when a value of k is given as input, ΔMV's output includes two distinct approximations to the Pareto front of the original multiobjective problem (excluding the additional objective described above), namely, $\mathcal{P}^*$ and $\mathcal{P}^*_k$ (it also recommends a final solution, $\mathbf{x}^* \in \mathcal{P}^*_k$). Both $\mathcal{P}^*$ and $\mathcal{P}^*_k$ consist of approximation sets constructed and maintained throughout the search process using corresponding external archives, $\mathcal{A}$ and $\mathcal{A}_k$. Whereas $\mathcal{A}$ stores nondominated solutions irrespective of their value of k, $\mathcal{A}_k$ filters solutions and only stores those satisfying the target k value. These archives are updated every time new candidate solutions are generated, both by the initialization routine and by the genetic operators. A maximum size of $|\mathcal{A}| = |\mathcal{A}_k| = P$ has been adopted (where P is the size of the population), and discrimination among nondominated solutions is carried out by means of the crowding distance measure [36].

F. Initialization and Genetic Operators

Our initialization routine creates an initial population exhibiting diversity, but at the same time high-quality individuals. First, the full MST-based solution ($k = 1$) is added to the population, see Fig. 3. This is followed by the inclusion of promising MST-based solutions for which k is uniformly chosen, without replacement, from the set $\{2, 3, \ldots, k_{\max}\}$, where $k_{\max}$ is a user-defined parameter. Solutions are included one by one until the population is full or the possible k values are exhausted. Note that this process needs to be repeated multiple times when using approach $\Delta\text{MV}_{\text{sep}}$, which operates on MSTs computed independently for each data view. If the possible k values are exhausted, the remaining population slots are filled with randomly generated solutions.

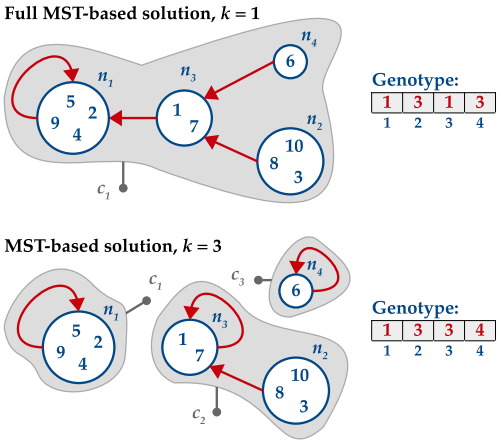

Fig. 3. MST-based solution is created by restoring the links originally removed from the MST to define the coarse-grained version of the problem (see Fig. 2 and Section III-B). To create a solution with k clusters, all links are restored except for the $k-1$ most relevant ones (according to measure DI). In these examples, restoring all links results in the full MST-based solution with $k=1$, and restoring all but the two most relevant links results in a solution with $k=3$ clusters.

The mating strategy adopted is based on (deterministic) binary tournament selection. Uniform crossover and uniform mutation are chosen as the genetic operators (with probability parameters p_c and p_m, respectively). Finally, the survivor selection scheme uses nondominated sorting and crowding distance, as in the *Nondominated Sorting Genetic Algorithm-II* (NSGA-II) [36]. Note that this Pareto-based population replacement strategy imposes limitations on the number of data views, and corresponding objective functions, that ΔMV can effectively handle. This is due to the inability of the Pareto-dominance relation to induce an effective discrimination in *many-objective* scenarios (involving four or more objectives), which significantly decreases selection pressure and thus the effectiveness of methods relying on this concept. An increase in the number of data views would therefore imply the adaptation of ΔMV's optimization engine, adopting strategies specifically designed for many-objective optimization [32].

G. Automated Model Selection

The selection of a single solution from the Pareto-front approximation is a final step in multiobjective clustering and often represents the ultimate goal in practical contexts. In the more general area of multiobjective optimization, the final solution is commonly taken from the *knee* regions, as they offer the most promising tradeoffs and usually correspond to the preferences of decision makers [37]. Such a strategy has been adopted in multiobjective clustering as well, for instance, in the original version of the MOCK algorithm [1].

However, the assumptions motivating the selection of the best tradeoffs do not necessarily apply in the specific setting of MVC. As we analyze in Section V-B, we may deal with data views presenting varying levels of reliability. We observe that, when one view is significantly more reliable (less noisy) than the others, the extreme region of the Pareto front for the corresponding objective function tends to coincide with the location of the optimal partition. Also, the more reliable the data view, the better defined the cluster structure tends to be, yielding better values for the optimization criteria.

We therefore define a simple, yet effective selection strategy based on the above observations. Our strategy selects the solution which leads to the highest sum of (unnormalized) objective values. Given that the objectives refer to the same clustering criterion but are computed independently for each data view (see Section III-D), and that higher objective values are obtained for more reliable views, our strategy involves an implicit weighting of the views based on their reliability.

H. Source Code Availability

The source code of ΔMV (written in C++) and the datasets used in our experiments are made available to the research community through the following repositories: https://github.com/garzafabre/DeltaMV, https://evoclustering.github.io.

IV. Experimental Setup

The following sections summarize the clustering algorithms, test datasets, settings, and performance indicators considered in the experiments presented in Section V.

A. Clustering Methods Evaluated

Four variants of our ΔMV algorithm are evaluated in order to investigate the impact of certain design choices: $\Delta\text{MV}_{\text{avg}}$ and $\Delta\text{MV}_{\text{sep}}$, which refer to our two strategies to handle multiple data views (see Section III-C), and the corresponding versions of these approaches which receive a specific value of k as input, denoted by $\Delta\text{MV}^k_{\text{avg}}$ and $\Delta\text{MV}^k_{\text{sep}}$. We adopt the following parameter settings during the evaluation of our proposal: population size, $P=100$; number of generations, $G_{\max}=300$; recombination probability, $p_c=0.7$; mutation probability, $p_m=1/n$, where n is the genotype length; maximum number of clusters for initialization, $k_{\max}=50$; and neighborhood parameter, $L=10$. This has been decided based on preliminary testing, ensuring that exactly the same settings are considered for all ΔMV variants (and the baseline evolutionary methods described below). This favors a fair comparison and enables a more focused assessment of the specific strategies implemented by each approach. A proper sensitivity analysis (not considered here) might reveal better choices to further boost the performance of individual approaches.

Given that our primary case study, namely, the bioinformatics application described in Section IV-B, involves datasets which are only available in the form of dissimilarity matrices (rather than feature spaces), our selection of reference methods is limited to approaches that can readily operate in this setting (methods based on cluster centroids are therefore excluded). Note that some of the references considered are single-view clustering methods (PAM, SL, AL, and SVEA, as described below). Throughout our experiments, a subscript will be added to the acronyms of these approaches when evaluating their performance from the perspective of the first (v_1) or second data view (v_2) of the clustering problems adopted. Three categories of reference approaches are considered:

1) Multiview Clustering Methods From the Literature: Our comparative analysis includes the *multiview multiobjective clustering* (MVMC) algorithm by José-García et al. [10]. MVMC is a recently proposed multiobjective approach to MVC that has shown promising results when evaluated with respect to previous multiobjective approaches [22], [29]. We also compare against the *Multiview Spectral Clustering Algorithm* (SC) proposed by Kanaan-Izquierdo et al. [38], which obtained highly competitive results when compared against seven other multiview algorithms from the literature.

2) Well-Known Single-View Clustering Methods: We include the *Partitioning Around Medoids* (PAM) algorithm, which uses data entities as cluster representatives (medoids), and two variants of *Agglomerative Hierarchical Clustering*, namely, *Single-Linkage* (SL) and *Average-Linkage* (AL).

3) Baseline Single-View and Multiview Evolutionary Approaches: Additional baselines allow us to verify whether the performance of ΔMV is explained by the use of multiple data views or its specific optimization framework. SVEA^k and MVEA^k are, respectively, single-view and multiview evolutionary approaches, implemented within the same framework and based on the same components as ΔMV. Whereas SVEA^k focuses on the individual data views, MVEA^k focuses on the aggregation of views and is therefore applied to the average dissimilarity matrix (the same matrix used by $\Delta\text{MV}^k_{\text{avg}}$). Both SVEA^k and MVEA^k receive k as input, which they handle through an additional objective function, as done by our method (see Section III-E). Because of this, both SVEA^k and MVEA^k are indeed multiobjective methods; however, MVEA^k optimizes a single objective to account for the (aggregated) views, in contrast to $\Delta\text{MV}^k_{\text{avg}}$ which uses a separate objective for each view (hence, MVEA^k can be seen as a single-objective version of our strategy $\Delta\text{MV}^k_{\text{avg}}$).

B. Multiview Clustering Problems

A total of 420 MVC problems are used in the experiments of this article. Out of these problems, 400 correspond to our primary case study, a real bioinformatics application. We additionally include 20 2-D, synthetic datasets with the aim of ensuring that our evaluation is comprehensive and considers clustering problems of varying characteristics regarding cluster shapes, overlap, and separability. All the problems considered involve two data views, which will be referred to as v_1 and v_2 throughout this study.

1) Bioinformatics Application: Clustering is an important task in structural bioinformatics. For example, it has recently been used for the structure-based classification of proteins of SARS-CoV-2 and other coronaviruses, with the aim of aiding the rational design of drugs and vaccines [39]. Another example is the application of clustering to analyze the diversity of (experimentally determined) structures in the protein data bank (PDB) [40] for various purposes, including the identification of unique protein-ligand complexes [41] and the definition of structural classes for specific regions (e.g., loops) of protein chains [42]. The clustering of protein structures is also relevant to the study of protein function; given that a protein's function is strongly dependent on its shape, clustering can facilitate functional inference from cluster members, which are in close proximity in structure space [43]. Finally, clustering is a common task within protein structure prediction (PSP) pipelines. PSP involves determining a protein's structure from its amino acid sequence, a challenge that has eluded a definitive solution for decades (although substantial progress has recently been reported [44], [45]). In PSP, clustering is frequently used during model (decoy) selection, step at which promising structures are chosen for further refinement from large collections of candidates initially produced at a lower resolution; the clustering-based approach consists in identifying groups within those collections and selecting representative structures from the most populated ones [46], [47], [48].

An important decision when applying clustering in these contexts concerns the adoption of a measure to determine the similarity between candidate structures. A variety of such measures exists, but there is no consensus as to which of them should be used. In this study, we acknowledge the intrinsic multiview nature of this task: multiple measures can be exploited simultaneously as data views, to assess the similarity of candidate structures more comprehensively.

A new collection of 400 MVC problems was created, using five different measures of structural similarity as alternative data views: 1) *Hamming distance between contact maps* (CMP); 2) *global distance test—total score* (GTS); 3) *global distance test - high accuracy* (GHA); 4) *distance between extreme points of secondary structure elements* (SSX); and 5) *distance in torsion (dihedral) angle space* (TOR). Multiview scenarios thereby correspond to the $\binom{5}{2} = 10$ possible ways to choose v_1 and v_2 from these measures. Further details on these measures are provided in the supplementary material, where we also describe the distribution of the 400 problems based on the ten possible multiview scenarios, their number of clusters, $k^* \in \{5, 10, 15\}$, and their size, $N \in \{500, 1000, 1500\}$.

Problems were defined following a two-stage process, which we briefly summarize below (for a detailed description, the reader is referred to the supplementary material). First, a diverse set of candidate structures for a given protein was identified. Then, each of these candidates was used to seed a series of iterative perturbation processes aimed at producing new structures around the initial configurations. In this way, a cluster of candidate structures was populated for every seed structure used. Varying the distance (radius) within which structures were generated from the seed, allowed us to control problem difficulty by affecting cluster homogeneity and the extent to which clusters can overlap with each other, as illustrated in Fig. 4. In all the cases, the initial cluster seeds were sampled from a collection of structures produced by independent runs of the state-of-the-art method *Rosetta* [49]. Moreover, all cluster members obtained from such seeds were generated using *Rosetta*'s structural perturbation operators (namely, fragment insertions). Thus, all the datasets constructed consist of plausible structures that can be reached during the search process of actual prediction protocols.

2) Synthetic Problems: For the 20 synthetic problems, data view v_1 is given by the *Euclidean distance*, whereas v_2 is given by the maximum edge distance (MED) [18]. Integrating

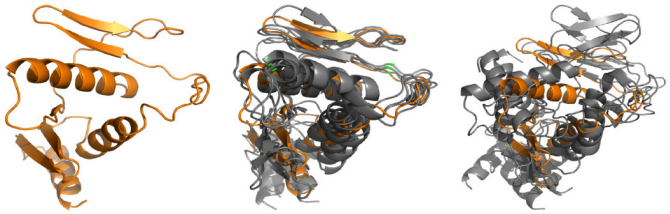

Fig. 4. Generation of clusters of protein structures. Clusters of different homogeneity are obtained by varying the radius r (root-mean-square deviation, measured in Å) within which cluster members (gray) are created starting from a seed structure (orange). Four cluster members are considered in these example clusters; their best possible alignment with respect to the seed structure is shown to illustrate cluster homogeneity. (a) Seed structure. (b) Cluster, $r = 5$ Å. (c) Cluster, $r = 10$ Å.

these particular dissimilarity measures in a multiview setting has been shown to increase robustness to arbitrarily shaped clusters [10]. Out of the 20 problems considered, 7 are explored for the first time in this study: *blobs2*, *blobs3*, *circles1*, *circles2*, *long4*, *moons3*, and *moons5*. The remaining 13 problems have been taken from [1]: *fourty*, *long1*, *longsquare*, *sizes1*, *sizes5*, *smile1*, *spiral*, *spiralsquare*, *square1*, *square4*, *triangle1*, *triangle2*, and *twenty*. As detailed and graphically illustrated in the supplementary material, these 20 problems vary in size, $N \in \{900, 1000, 1500, 4000\}$, number of clusters, $k^* \in \{2, 4, 5, 6, 8, 10, 20, 40\}$, and overall data characteristics (e.g., shape, overlap, and separability of the clusters).

C. Performance Assessment

Section V reports statistics computed from 31 independent executions performed for each clustering method studied, for every problem considered. The ability of an algorithm to produce high-quality partitions is assessed using the *adjusted rand index* (ARI) [50], which analyzes the pairwise co-assignment of data entities between the partitions obtained and the correct partition (ground truth, which is known for all the problems considered). The ARI measure is defined in the range $[\sim 0, 1]$, and higher values indicate a better clustering performance. An important aspect of our algorithm is its ability to determine the number of clusters, k, automatically. For this reason, the difference between the k values obtained and the correct number of clusters k^* is also analyzed.

To further understand the performance and behavior of our ΔMV method, we analyze the characteristics of the Pareto-front approximations produced. This analysis relies on the visualization of the differences between the (first-order) *empirical attainment functions* (EAFs) of different ΔMV variants [51]. The plots for such visualizations, generated using the tools provided by López-Ibáñez et al. [52], allow us to identify whether and in which particular regions of the objective space a variant performs better than others. Every plot contrasts two approaches and shows the differences which favor each of them. In all the cases, the x-axis and y-axis correspond to the objective function values (silhouette width, to be maximized) computed individually for data views v_1 and v_2, respectively. Solid lines in these plots represent the grand best (upper line) and grand worst (lower line) attainment surfaces, while the dashed line denotes the median attainment surface. In addition, some plots include specific markers to illustrate the location of the optimum and examples of the solutions selected automatically by our method.

The (nonparametric) *Mann–Whitney U test* is used to investigate the statistical significance of the performance differences observed between the approaches compared. In all the cases, a significance level of $\alpha = 0.05$ is considered. Only specific, relevant pairs of approaches are analyzed, and *Bonferroni correction* is applied to account for multiple testing issues.

V. Results

This section discusses the results of a series of experiments conducted to investigate the performance of our MVC algorithm, ΔMV. First, Section V-A presents an overall evaluation of different variants of ΔMV and comparisons with respect to a set of reference methods. The ability of our method to automatically determine the number of clusters is also analyzed in that section. Next, Section V-B explores the robustness of our algorithm (and reference approaches) when dealing with noisy data views. Then, the effectiveness of our automated model selection strategy is studied in Section V-C. Finally, Section V-D analyzes the effects of varying problem granularity through our graph-based genetic representation.

A. Overall Results and Comparisons

This section analyzes the results obtained by our ΔMV algorithm, and compares its performance with respect to a set of single-view and multiview reference methods (see Section IV-A). The results of this comparison are summarized in Fig. 5. Tables with specific results for separate problem categories (protein datasets) and individual problems (synthetic datasets), and more detailed results of the statistical significance analysis, are included in the supplementary material.

As shown in Fig. 5, our method ΔMV is the best overall performer. In particular, $\Delta\text{MV}^k_{\text{avg}}$ and $\Delta\text{MV}^k_{\text{sep}}$, variants that receive k as input (similar conditions to all of the reference methods considered), score better (higher) ARI values than the contestant approaches, with statistically significant differences in most cases [the only exception is MVMC, for which no significant difference is observed with respect to $\Delta\text{MV}^k_{\text{avg}}$ in the synthetic problems, see Fig. 5(b)]. The analysis of individual instances (supplementary material) reveals that $\Delta\text{MV}^k_{\text{avg}}$ significantly outperforms $\Delta\text{MV}^k_{\text{sep}}$ in 176 cases (out of 420). Although this suggests that the multiview strategy adopted by $\Delta\text{MV}^k_{\text{avg}}$ is more effective, $\Delta\text{MV}^k_{\text{sep}}$ performs significantly better in 80 problems, and no significant differences are observed between these approaches in the remaining 164 cases; the overall results of Fig. 5 also indicate that the differences between $\Delta\text{MV}^k_{\text{avg}}$ and $\Delta\text{MV}^k_{\text{sep}}$ are not statistically significant. Subsequent analyses presented in Sections V-B and V-C allow us to further discuss the key differences between these specific variants and to identify scenarios where one of them may be expected to perform better than the other.

Variants of ΔMV that automatically determine k, namely, $\Delta\text{MV}_{\text{avg}}$ and $\Delta\text{MV}_{\text{sep}}$, also exhibit a highly competitive

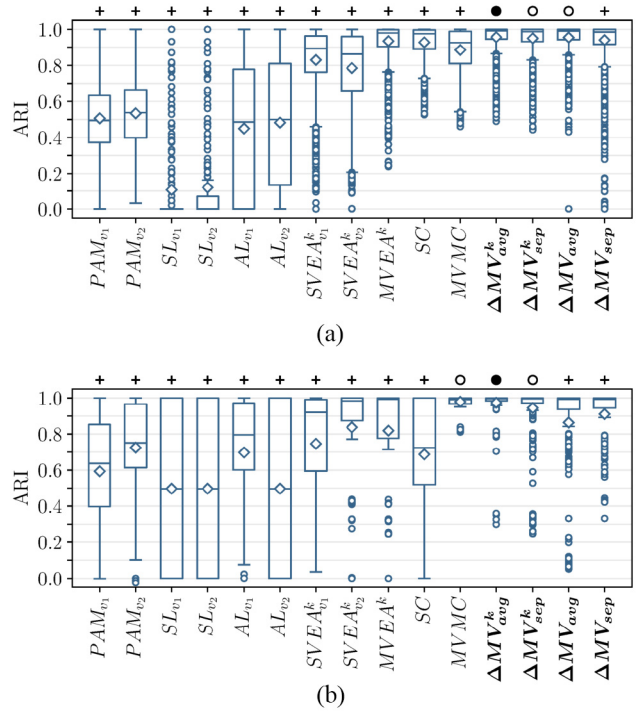

Fig. 5. Clustering performance (ARI) scored by all methods on the (a) protein and (b) synthetic datasets. Marker • at the top of the plots indicates the method with the highest median ARI value; marker + indicates a statistically significant difference with respect to this best performing method, whereas marker ∘ indicates that no significant difference is observed.

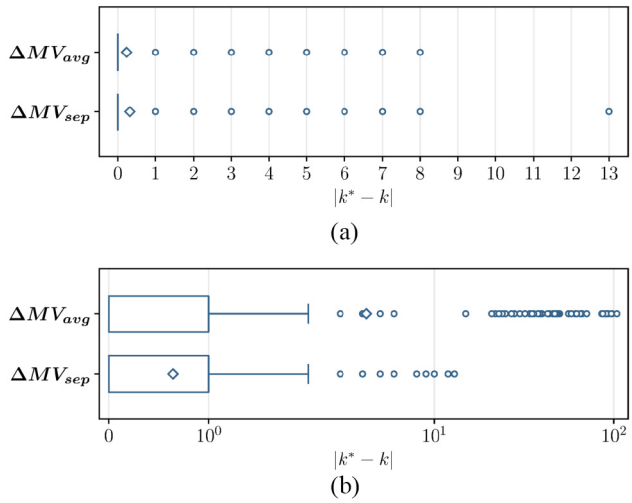

Fig. 6. Ability of approaches ΔMV_{avg} and $\Delta\text{MV}_{\text{sep}}$ to correctly determine k. The absolute differences between the k obtained and the correct value k^* are shown for (a) protein structure and (b) synthetic datasets (in logarithmic scale).

performance. In fact, our analysis reveals that in 339 and 329 problems, the performance differences between these variants and their corresponding counterparts having access to k ($\Delta\text{MV}^k_{\text{avg}}$ and $\Delta\text{MV}^k_{\text{sep}}$) are not statistically significant. Fig. 6 confirms that both $\Delta\text{MV}_{\text{avg}}$ and $\Delta\text{MV}_{\text{sep}}$ succeed in correctly identifying (or closely approximating) the value of k in the majority of the cases. All the above findings confirm the suitability of algorithm ΔMV for MVC settings, and its capacity to find high-quality partitions even in the absence of information regarding the correct value for parameter k.

The multiview spectral clustering approach, SC, shows outstanding results for the protein structure datasets. However, its performance decreases for the synthetic problems, especially when dealing with overlapping clusters. Conversely, the multiobjective approach MVMC is effective across the diverse characteristics in the synthetic datasets, scoring results comparable to those of ΔMV. The prototype-based representation offers MVMC some robustness to overlapping clusters, and the availability of a data view based on the MED distance allows the algorithm to deal with challenging cluster shapes, in spite of its representation. Despite these remarkable results, the performance of MVMC drops for the protein structure datasets; we hypothesize that this performance drop relates to the lack of a mechanism to help MVMC cope with irregular clusters, when the MED distance is not used as an additional data view (as is the case in this particular experiment).

The fact that all the single-view methods (PAM, SL, AL, and SVEA^k) are clearly outperformed by the multiview approaches supports the relevance of MVC. In particular, the results for the protein structure datasets [Fig. 5(a)] provide conclusive evidence of the benefits that MVC can bring to this domain. Single-view methods are unable to perform reliably well across all scenarios. While a given view can favor competitive results in some cases, the same view can lead to a poor performance when it becomes incompatible with the characteristics of the data. This is evidenced by the results for specific synthetic datasets (supplementary material): using the Euclidean distance (v_1), SVEA^k yields good results in problems with spherical clusters (e.g., blobs3, sizes5, and square4) but completely fails in cases of irregular, nonlinearly separable clusters (e.g., circles1 and spiral). On the contrary, problems with nonlinearly separable clusters are easily solved when *SVEA*^{k} uses the MED distance (v_2) instead. Using only the MED distance, however, SVEA^k fails when presented with overlapping clusters (e.g., sizes5 and square4).

As described in Section IV-A, SVEA^k and MVEA^k can be seen as the single-view and single-objective multiview counterparts of $\Delta\text{MV}^k_{\text{avg}}$, respectively. On the one hand, MVEA^k performs significantly better than SVEA^k (as well as all other single-view references) in the majority of the cases, highlighting the advantages of MVC. On the other hand, $\Delta\text{MV}^k_{\text{avg}}$ scores significantly better results than MVEA^k in most cases, confirming that the adoption of a multiobjective approach is promising in tackling the MVC challenge.

B. Robustness in the Presence of Unreliable Data Views

Most studies on MVC evaluate clustering methods under the implicit assumption of equal reliability across the data views. It can be the case, however, that some views are more reliable (e.g., less noisy) than others. Rather than contributing, would the inclusion of an unreliable data view impact negatively on performance? How robust MVC methods are for handling views of different reliability? We conduct an experiment to investigate the extent to which varying noise levels in either one or the two data views may affect the performance of our proposed ΔMV method and some reference approaches.

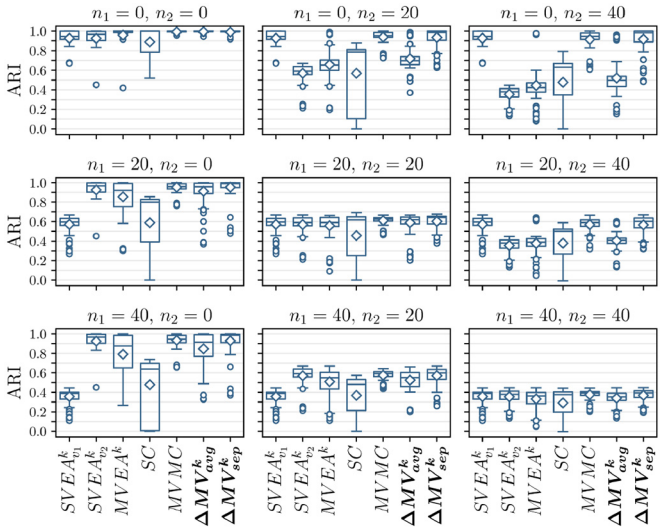

Fig. 7. Robustness to noise introduced independently to data views v_1 and v_2. Each plot corresponds to a different configuration of noise levels, for $n_1, n_2 \in \{0, 20, 40\}$. This figure integrates the results for both protein and synthetic datasets.

A new set of problems was created by independently adding noise to the data views of our original problems. Let n_i be the level of noise for data view v_i, $i \in \{1, 2\}$. Noise is added to v_i by swapping the columns and rows associated with random pairs of data entities, within the dissimilarity matrix describing that view. The value of n_i represents the percentage of entities affected. Effectively, this strategy perturbs the original data views, ensuring that they become increasingly independent of each other. Given that problem difficulty may depend on the specific pairs of entities affected, the noise was added in an incremental, controlled fashion; that is, if $a < b$, then the pairs of entities affected when $n_i = a$ are a subset of the pairs affected when $n_i = b$. This ensures that problems become more challenging as the level of noise increases.

This experiment focuses on subsets of ten protein problems and ten synthetic problems for which a good performance is observed across the algorithms evaluated; this suggests that the original data views are sufficiently accurate and can be used as the starting point for noise addition in a controlled manner. We use $n_1, n_2 \in \{0, 10, 20, 30, 40\}$ and explore the 25 configurations resulting from the possible combinations of these noise levels. To account for randomness, five problem samples were independently generated for each of the 25 noise configurations, for each of the 20 problems considered. This results in a total of 2500 multiview problems, for which we ran every algorithm 21 times independently. Fig. 7 summarizes the results of this experiment. Due to space limitations, the figure covers only nine of the 25 noise configurations, using $n_1, n_2 \in \{0, 20, 40\}$. Results for the full range of noise levels, presented separately for protein problems and synthetic problems, can be found in the supplementary material.

On the one hand, Fig. 7 allows us to confirm certain expected behaviors. The performance of the single-view baselines $\text{SVEA}^k_{v_1}$ and $\text{SVEA}^k_{v_2}$ consistently drops as noise levels n_1 and n_2 are independently increased, indicating that our mechanism for noise addition directly and effectively controls problem difficulty. Furthermore, the performance of all methods considered (both single-view and multiview) is equally affected and gradually decreases when n_1 and n_2 increase together, $n_1 = n_2$ (main diagonal of plots in Fig. 7).

On the other hand, more interesting behaviors are observed for the multiview methods in scenarios where $n_1 \neq n_2$. In particular, increasing the noise in one view, while keeping it constant in the other, reveals that the reference algorithm MVMC and our method's variant $\Delta\text{MV}^k_{\text{sep}}$ are the only multiview approaches that can manage situations where one of the views is reliable but the other is not. The multiobjective strategy adopted by MVMC is based on the use of scalarizing vectors, which effectively weigh the objective functions accounting for the individual data views. Given the use of a diverse set of scalarizing vectors, with some of them practically ignoring (assigning a low importance to) the noisy view, MVMC is able to succeed in this type of scenario.

The robustness shown by $\Delta\text{MV}^k_{\text{sep}}$ can be explained by the separate handling of data views that is enforced in the definition of the search space (possible genotypes) and throughout the entire clustering process. Such a separation gives $\Delta\text{MV}^k_{\text{sep}}$ the ability to exploit a reasonably accurate data view in spite of the low quality of the other. This can be difficult to achieve in approaches like $\Delta\text{MV}^k_{\text{avg}}$ and MVEA^k, which rely on the aggregation of data views (by optimizing separate objective functions to account for the different data views, $\Delta\text{MV}^k_{\text{avg}}$ still presents an advantage over MVEA^k). Similarly, the significant conflict introduced between data views poses a challenge for the effective computation of the common eigenvectors on which SC operates, explaining the low performance of this method during this experiment.

To further investigate the performance differences between $\Delta\text{MV}^k_{\text{avg}}$ and $\Delta\text{MV}^k_{\text{sep}}$, Fig. 8 exemplifies how the noise conditions may impact on the location of the optimum, relative to the Pareto-front approximations produced. When the two views present a similar reliability ($n_1 = n_2$), the optimum tends to be located around central regions, where $\Delta\text{MV}^k_{\text{avg}}$ seems to offer a better convergence (this may explain why $\Delta\text{MV}^k_{\text{avg}}$ is identified as the best overall performer in Section V-A). Contrarily, when one view is significantly more reliable than the other, the optimum is located closer to the corresponding extreme region of the Pareto front, where $\Delta\text{MV}^k_{\text{sep}}$ provides a clearly superior performance. The separate handling of data views is what grants $\Delta\text{MV}^k_{\text{sep}}$ access to such regions of the objective space (regions that may even be rendered unattainable for $\Delta\text{MV}^k_{\text{avg}}$ due to the aggregation of views).

C. Pareto-Front Approximations and Model Selection

With the aim of better understanding the behavior and performance differences between variants $\Delta\text{MV}^k_{\text{avg}}$ and $\Delta\text{MV}^k_{\text{sep}}$ of our proposed method, we contrast the characteristics of the Pareto-front approximations they produce. Furthermore, we analyze the effectiveness of our strategy to automatically select a final partition from these approximation solution sets.

First, from the EAFs of $\Delta\text{MV}^k_{\text{avg}}$ and $\Delta\text{MV}^k_{\text{sep}}$, as exemplified in Fig. 9, we can see that $\Delta\text{MV}^k_{\text{avg}}$ tends to achieve a

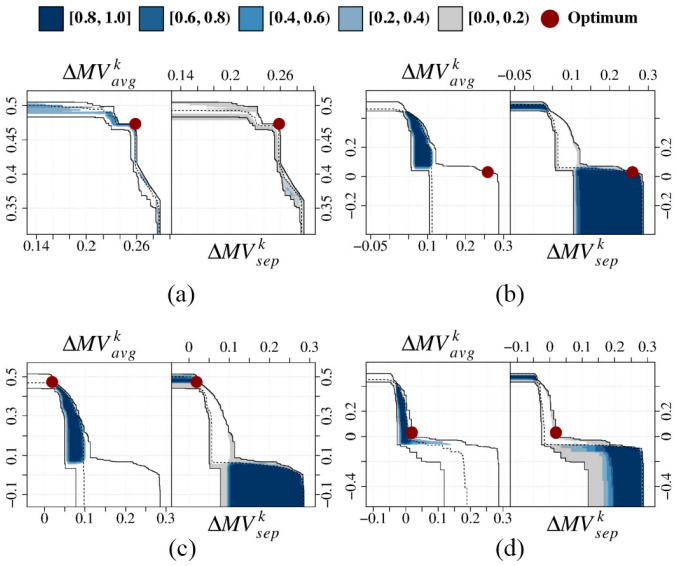

Fig. 8. EAFs of approaches ΔMV^k_{avg} and $\Delta\text{MV}^k_{\text{sep}}$. The same example protein problem and original data views are considered in all the cases, but distinct levels of noise n_1 and n_2 are introduced, as specified for each individual plot. In the plots, the x-axis and y-axis refer to data views v_1 and v_2, respectively. Differences in the point attainment probabilities are encoded by different colors, and a red marker shows the relative location of the optimal solution (see legend at the top). (a) $n_1 = 0$ and $n_2 = 0$. (b) $n_1 = 0$ and $n_2 = 40$. (c) $n_1 = 40$ and $n_2 = 0$. (d) $n_1 = 40$ and $n_2 = 40$.

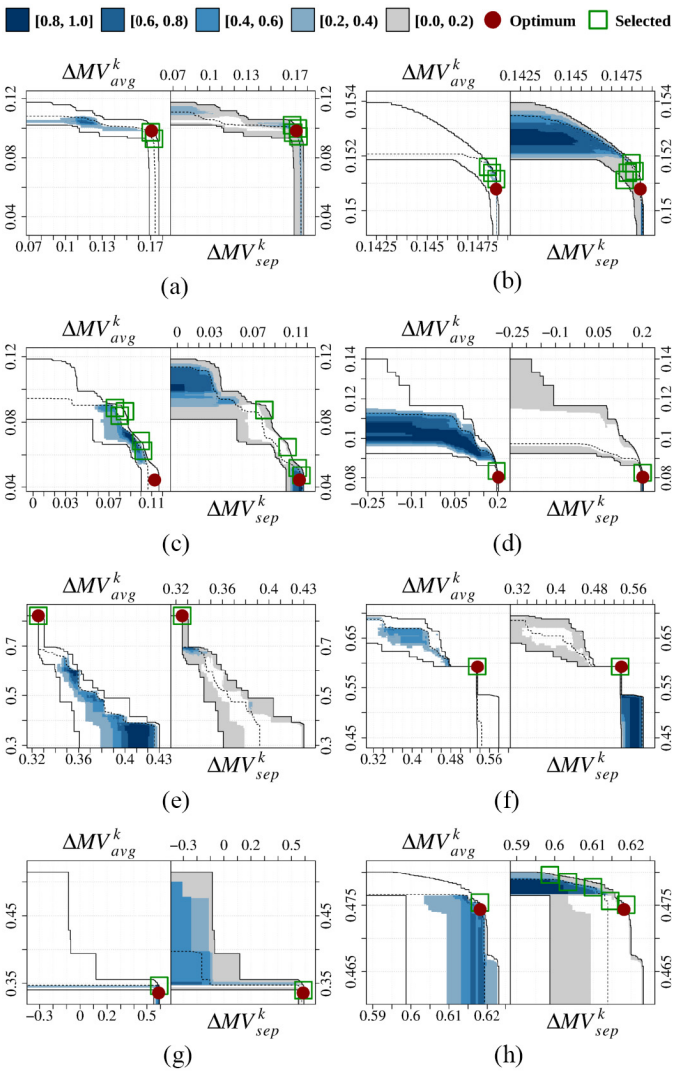

Fig. 9. Differences between the EAFs of approaches $\Delta\text{MV}^k_{\text{avg}}$ and $\Delta\text{MV}^k_{\text{sep}}$. (a)–(d) Results for a random sample of protein structure problems, indicating the specific pairs of data views used. (e)–(h) Results for specific synthetic problems, as indicated. In the plots, the x-axis and y-axis refer to data views v_1 and v_2, respectively. Differences in the point attainment probabilities are encoded using different colors, and specific markers illustrate the location of the optimum as well as examples of solutions selected automatically by our method (see legend at the top).

better convergence toward central regions of the Pareto front, as opposed to $\Delta\text{MV}^k_{\text{sep}}$, which performs better at the extreme regions. This behavior is evidenced by the results for several problems [e.g., Fig. 9(a)–(d)], confirming previous observations of Section V-B. Central regions correspond to a balanced tradeoff between the data views. Due to its direct aggregation of views, $\Delta\text{MV}^k_{\text{avg}}$ is able to concentrate its efforts on these central regions. In scenarios where data views are both reliable, this strategy may give $\Delta\text{MV}^k_{\text{avg}}$ an advantage over $\Delta\text{MV}^k_{\text{sep}}$. Contrarily, we would expect better solutions to be closer to an extreme point of the front if one view is more reliable than the other. This is the case for the problem considered in Fig. 9(c), where v_1 (GTS, on the x-axis) is clearly more reliable than v_2 (TOR, on the y-axis). Accordingly, the optimum is found near the extreme region corresponding to v_1, an area that is more readily accessible to the approach based on the separation of views, $\Delta\text{MV}^k_{\text{sep}}$.

The varying reliability of the data views, and the fact that the characteristics of the Pareto front are highly problem-dependent, create challenges for model selection. Fig. 9 suggests that the implicit weighting of data views in our strategy offers some robustness, allowing our method to frequently identify solutions near the optimal partition. A clear example is shown in Fig. 9(e). Given the elongated clusters in the synthetic problem *long4*, the second view (MED distance, on the y-axis) is the one which contributes the most, causing significantly higher objective values in comparison to the first view (Euclidean distance, on the x-axis). These higher objective values dominate the ranking (based on the sum of objective values) and ensure that our method favors solutions near the optimum (at the extreme end of the front).

To further investigate our strategy for model selection, Fig. 10 contrasts the quality of the solutions selected with respect to the following baselines: 1) the best and worst solutions available, representing upper and lower bounds on the attainable ARI values; 2) the extreme points of the Pareto-front approximation, $\text{Ext}(v_1)$ and $\text{Ext}(v_2)$, which refer to a naive strategy always prioritizing one of the data views; and 3) a randomly selected solution, an approach that should be beaten by any reasonably informed selection mechanism. The large discrepancy seen between the best and worst attainable ARI values highlights the diversity of solution qualities that the approximation sets present. Our strategy is able to correctly identify the best solution (or one of similar quality) in many cases, outperforming the baselines. More specifically, our analysis of statistical significance indicates that the solutions selected are comparable in quality to the best available

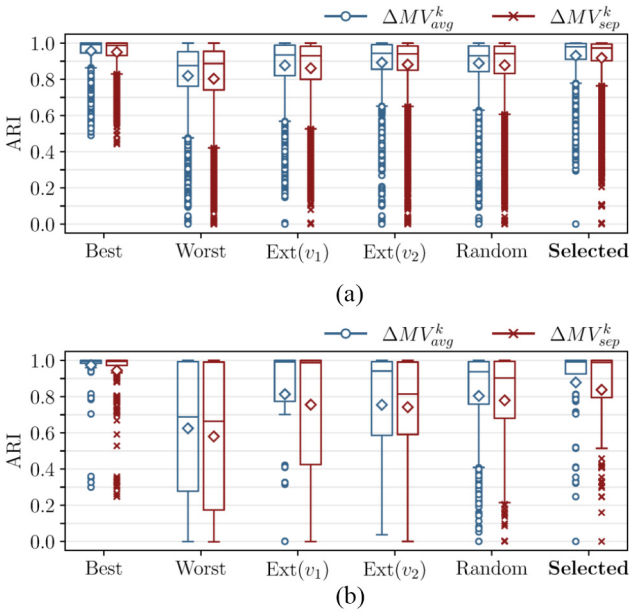

Fig. 10. Effectiveness of our automatic model selection strategy when applied to the Pareto-front approximations produced by approaches $\Delta\text{MV}^k_{\text{avg}}$ and $\Delta\text{MV}^k_{\text{sep}}$. Results are presented separately for (a) protein datasets and (b) synthetic datasets.

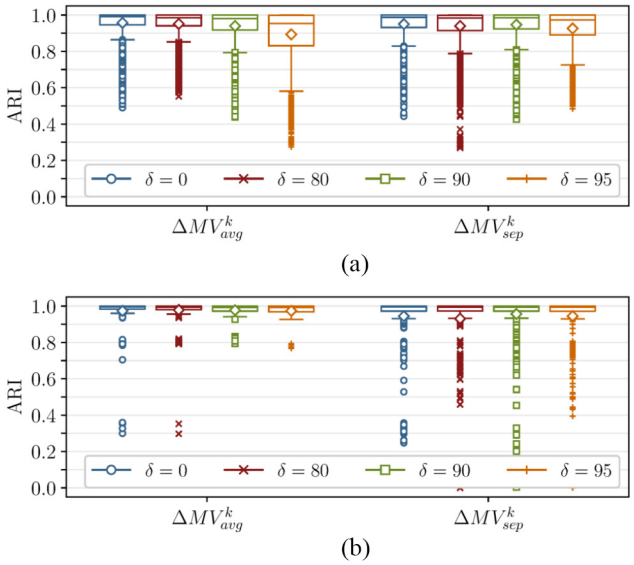

Fig. 11. Impact of parameter δ on the clustering performance of approaches $\Delta\text{MV}^k_{\text{avg}}$ and $\Delta\text{MV}^k_{\text{sep}}$. Four different settings are considered, $\delta \in \{0, 80, 90, 95\}$, and results are shown separately for (a) protein problems and (b) synthetic problems.

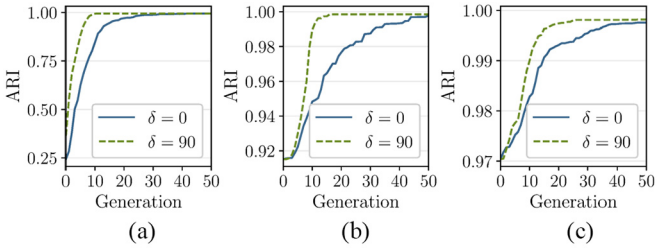

Fig. 12. Convergence of approach $\Delta\text{MV}^k_{\text{avg}}$ when using $\delta \in \{0, 90\}$. The average ARI reached is shown for the first 50 generations. Results are presented for (a) and (b) two sample protein problems and (c) sample synthetic problem.

solution in 152 (∼36%) and 166 (∼40%) of the 420 test scenarios, according to the results for $\Delta\text{MV}^k_{\text{avg}}$ and $\Delta\text{MV}^k_{\text{sep}}$ (see the supplementary material). However, a significant drop in quality occurs in the remaining majority of the cases, evidencing the challenging nature of model selection and motivating future research to improve this aspect of our proposal.

D. Impact of Representation Granularity

As described in Section III-B, our variable-granularity representation aims to address the main criticism of the original graph-based encoding: its limited scalability. Although the focus of this article is on the ability of our method to handle multiple data views, rather than on its potential to scale to large datasets, it is important to consider this aspect which is a consequence of our chosen genetic representation.

Our method partly overcomes scalability issues inherent to the graph-based representation by narrowing the set of alternative allele values for each decision variable. The reduction of genotype length, achieved by using a value of $\delta > 0$, contributes further to this end and directly translates into a reduction of the search space. Whilst this may not necessarily translate into an improvement in final clustering performance (ARI value reached at the end of the search), as seen in Fig. 11, it can have a major impact on convergence speed, as illustrated in Fig. 12. This means that we can potentially produce higher quality partitions using a lower number of generations.

Computational efficiency also benefits from the delta-evaluation of candidate solutions, which is enabled by our flexible representation as discussed in Section III-D. As shown in Fig. 13, we can achieve meaningful reductions in problem size, and the incremental evaluation of solutions can lead to significant savings in execution time. From the figure, it is possible to observe that this is particularly true for strategy $\Delta\text{MV}^k_{\text{avg}}$. Strategy $\Delta\text{MV}^k_{\text{sep}}$, on the other hand, will not always achieve the desired reduction in problem size due to the potential conflict between data views. Consequently, we can see that the savings in execution time vary from problem to problem and correlate strongly with the reduction obtained in representation length. As discussed in Section III-C, an increase in the number of data views would certainly aggravate this behavior, preventing $\Delta\text{MV}^k_{\text{sep}}$ from fully exploiting the advantages of our variable-granularity representation.

VI. Conclusion

This article highlights mechanisms to efficiently adapt graph-based representations, as used in algorithms MOCK [1] and Δ-MOCK [2] for cluster analysis, to a multiview learning setting. The resulting method, ΔMV, exhibits properties that we generally deem favorable in multiobjective clustering: robustness toward nonspherical clusters as well as toward changes in the reliability of different objectives. Experiments

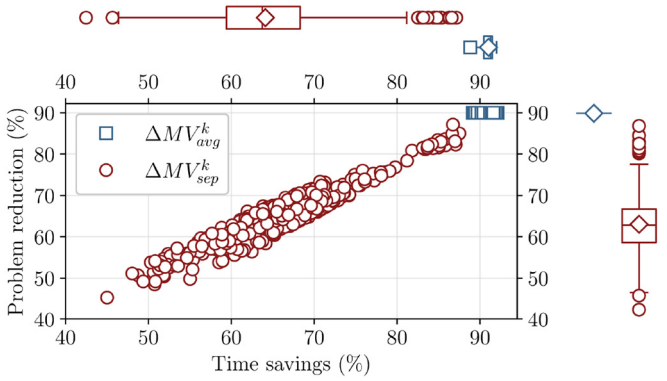

Fig. 13. Effects of problem granularity on computational efficiency. Using $\delta = 90$, the plot contrasts the problem reductions achieved against the resulting savings in execution time, quantified relatively to the original problem sizes. Results are shown for the full set of 420 clustering problems.

with different design choices emphasize the important trade-offs between bias, flexibility, and search performance that we would expect to see in any difficult optimization problem. These underline the relevance of our proposals and help highlight the boundaries for graph-based representations in multiview settings. On the whole, our algorithm reports a strongly competitive performance with respect to existing multiview methods. More generally, our work identifies a route forward for the transfer of evolutionary graph-based clustering approaches to MVC scenarios.

The evaluation of our proposal considers the problem of clustering plausible protein structures, an important application from the field of bioinformatics, as the primary case study. Our results support the inherent multiview nature of this challenge which, to the best of our knowledge, had not been previously considered from this particular perspective. This finding confirms that multiview learning and the clustering methods proposed in this article can find a significant impact in practical contexts.

Our analysis is limited to the case of two views and a single clustering criterion (the Silhouette width). Future work will consider the performance of our approach for more than two views, as well as experimentation in settings where objectives may involve a combination of different views and clustering criteria. In principle, the proposed method can cater for both settings. However, the expected difficulties include further increases in the size of the search space and approximation front, and these aspects will warrant additional attention.

Multiview settings may sometimes involve partial views, i.e., feature or dissimilarity measurements for a subset of entities only. Our methodology is currently silent on the integration of such views into the process of MST and neighborhood construction. Extending its applicability to such scenarios will require additional investigation and development.

REFERENCES

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

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

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

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

[5] J. Kleinberg, "An impossibility theorem for clustering," in *Advances in Neural Information Processing Systems 15 (NIPS)*, vol. 15, S. Becker, S. Thrun, and K. Obermayer, Eds. Camnridge, MA, USA: MIT Press, 2002, pp. 463–470.

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

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

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

[9] G. Chao, S. Sun, and J. Bi, "A survey on multiview clustering," *IEEE Trans. Artif. Intell.*, vol. 2, no. 2, pp. 146–168, Apr. 2021.

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

[11] V. M. Devagiri, V. Boeva, S. Abghari, F. Basiri, and N. Lavesson, "Multi-view data analysis techniques for monitoring smart building systems," *Sensors*, vol. 21, no. 20, p. 6775, 2021.

[12] E. Zeng, C. Yang, T. Li, and G. Narasimhan, "Clustering genes using heterogeneous data sources," *Int. J. Knowl. Discov. Bioinform.*, vol. 1, no. 2, pp. 12–28, 2010.

[13] X. He, L. Li, D. Roqueiro, and K. Borgwardt, "Multi-view spectral clustering on conflicting views," in *Machine Learning and Knowledge Discovery in Databases*, M. Ceci, J. Hollmén, L. Todorovski, C. Vens, and S. Džeroski, Eds. Cham, Switzerland: Springer Int., 2017, pp. 826–842.

[14] S. Queiroz, F. D. A. T. de Carvalho, and Y. Lechevallier, "Nonlinear multicriteria clustering based on multiple dissimilarity matrices," *Pattern Recognit.*, vol. 46, no. 12, pp. 3383–3394, 2013.

[15] F. D. A. T. de Carvalho, Y. Lechevallier, and F. M. de Melo, "Partitioning hard clustering algorithms based on multiple dissimilarity matrices," *Pattern Recognit.*, vol. 45, no. 1, pp. 447–464, 2012.

[16] C. Liu, J. Liu, D. Peng, and C. Wu, "A general multiobjective clustering approach based on multiple distance measures," *IEEE Access*, vol. 6, pp. 41706–41719, 2018.

[17] B. Fischer and J. M. Buhmann, "Path-based clustering for grouping of smooth curves and texture segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 4, pp. 513–518, Apr. 2003.

[18] A. E. Bayá and P. M. Granitto, "How many clusters: A validation index for arbitrary-shaped clusters," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 10, no. 2, pp. 401–414, Mar./Apr. 2013.

[19] R. P. de Gusmão and F. D. A. T. de Carvalho, "Clustering of multiview relational data based on particle swarm optimization," *Expert Syst. Appl.*, vol. 123, pp. 34–53, Jun. 2019.

[20] S. Saha, S. Mitra, and S. Kramer, "Exploring multiobjective optimization for multiview clustering," *ACM Trans. Knowl. Discov. Data*, vol. 12, no. 4, pp. 1–30, Aug. 2018.

[21] N. Saini, D. Bansal, S. Saha, and P. Bhattacharyya, "Multi-objective multi-view based search result clustering using differential evolution framework," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114299.

[22] S. Mitra, M. Hasanuzzaman, and S. Saha, "A unified multi-view clustering algorithm using multi-objective optimization coupled with generative model," *ACM Trans. Knowl. Discov. Data*, vol. 14, no. 1, p. 2, Jan. 2020.

[23] R. Caballero, M. Laguna, R. Martí, and J. Molina, "Scatter tabu search for multiobjective clustering problems," *J. Oper. Res. Soc.*, vol. 62, no. 11, pp. 2034–2046, Nov. 2011.

[24] A. M. Saeidi, J. Hage, R. Khadka, and S. Jansen, "A search-based approach to multi-view clustering of software systems," in *Proc. 22nd IEEE Int. Conf. Softw. Anal. Evol. Reeng.*, Montreal, QC, Canada, Mar. 2015, pp. 429–438.

[25] M. J. Brusco and S. Stahl, "Multiobjective partitioning," in *Branch-and-Bound Applications in Combinatorial Data Analysis*, New York, NY, USA: Springer, 2005, pp. 77–87.

[26] X. Wang, B. Qian, J. Ye, and I. Davidson, "Multi-objective multi-view spectral clustering via pareto optimization," in *Proc. SIAM Int. Conf. Data Min.*, Austin, TX, USA, May 2013, pp. 234–242.
[27] A. Wahid, X. Gao, and P. Andreae, "Multi-view clustering of Web documents using multi-objective genetic algorithm," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, 2014, pp. 2625–2632.
[28] A. Wahid, X. Gao, and P. Andreae, "Multi-objective multi-view clustering ensemble based on evolutionary approach," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Sendai, Japan, May 2015, pp. 1696–1703.
[29] B. Jiang, F. Qiu, S. Yang, and L. Wang, "Evolutionary multi-objective optimization for multi-view clustering," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, Jul. 2016, pp. 3308–3315.
[30] F. Rothlauf and D. E. Goldberg, "Redundant representations in evolutionary computation," *Evol. Comput.*, vol. 11, no. 4, pp. 381–415, Dec. 2003.
[31] J. Handl and J. Knowles, "An investigation of representations and operators for evolutionary data clustering with a variable number of clusters," in *Parallel Problem Solving From Nature (PPSN IX)*, T. P. Runarsson, H.-G. Beyer, E. Burke, J. J. Merelo-Guervós, L. D. Whitley, and X. Yao, Eds. Berlin, Germany: Springer, 2006, pp. 839–849.
[32] K. Li, R. Wang, T. Zhang, and H. Ishibuchi, "Evolutionary many-objective optimization: A comparative study of the state-of-the-art," *IEEE Access*, vol. 6, pp. 26194–26214, 2018.
[33] Y. J. Park and M. S. Song, "A genetic algorithm for clustering problems," in *Genetic Programming*, Madison, WI, USA: Morgan Kaufmann, Jul. 1998, pp. 568–575.
[34] M. Garza-Fabre, J. Handl, and J. Knowles, "A new reduced-length genetic representation for evolutionary multiobjective clustering," in *Evolutionary Multi-Criterion Optimization*, H. Trautmann et al., Eds. Münster, Germany: Springer Int., Mar. 2017, pp. 236–251.
[35] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.
[36] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[37] K. Deb and S. Gupta, "Understanding knee points in bicriteria problems and their implications as preferred solution principles," *Eng. Optim.*, vol. 43, no. 11, pp. 1175–1204, 2011.
[38] S. Kanaan-Izquierdo, A. Ziyatdinov, and A. Perera-Lluna, "Multiview and multifeature spectral clustering using common Eigenvectors," *Pattern Recognit. Lett.*, vol. 102, pp. 30–36, Jan. 2018.
[39] R. Gowthaman, J. D. Guest, R. Yin, J. Adolf-Bryfogle, W. R. Schief, and B. G. Pierce, "CoV3D: A database of high resolution coronavirus protein structures," *Nucleic Acids Res.*, vol. 49, no. D1, pp. D282–D287, Sep. 2020.
[40] H. M. Berman et al., "The protein data bank," *Nucleic Acids Res.*, vol. 28, no. 1, pp. 235–242, Jan. 2000.
[41] N. K. Shinada, P. Schmidtke, and A. G. de Brevern, "Accurate representation of protein-ligand structural diversity in the protein data bank (PDB)," *Int. J. Mol. Sci.*, vol. 21, no. 6, p. 2243, 2020.
[42] M. Shapovalov, S. Vucetic, and R. L. Dunbrack Jr., "A new clustering and nomenclature for beta turns derived from high-resolution protein structures," *PLoS Comput. Biol.*, vol. 15, no. 3, pp. 1–32, Mar. 2019.
[43] J. Hou, S.-R. Jun, C. Zhang, and S.-H. Kim, "Global mapping of the protein structure space and application in structure-based inference of protein function," *Proc. Nat. Acad. Sci.*, vol. 102, no. 10, pp. 3651–3656, 2005.
[44] A. Kryshtafovych, T. Schwede, M. Topf, K. Fidelis, and J. Moult, "Critical assessment of methods of protein structure prediction (CASP)—Round XIV," *Proteins Struct. Funct. Bioinform.*, vol. 89, no. 12, pp. 1607–1617, 2021.
[45] J. Jumper and D. Hassabis, "Protein structure predictions to atomic accuracy with AlphaFold," *Nat. Methods*, vol. 19, no. 1, pp. 11–12, 2022.
[46] S. M. Kandathil, M. Garza-Fabre, J. Handl, and S. C. Lovell, "Improved fragment-based protein structure prediction by redesign of search heuristics," *Sci. Rep.*, vol. 8, no. 1, 2018, Art. no. 13694.
[47] N. Akhter et al., "Improved protein decoy selection via non-negative matrix factorization," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 19, no. 3, pp. 1670–1682, May/Jun. 2022.
[48] A. B. Zaman, P. Kamranfar, C. Domeniconi, and A. Shehu, "Reducing ensembles of protein tertiary structures generated de novo via clustering," *Molecules*, vol. 25, no. 9, p. 2228, 2020.
[49] C. A. Rohl, C. E. M. Strauss, K. M. S. Misura, and D. Baker, "Protein structure prediction using Rosetta," *Methods Enzymol.*, vol. 383, pp. 66–93, Jan. 2004.
[50] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, 1971.
[51] V. G. da Fonseca, C. M. Fonseca, and A. O. Hall, "Inferential performance assessment of stochastic optimisers and the attainment function," in *Evolutionary Multi-Criterion Optimization*. Zürich, Switzerland: Springer, 2001, pp. 213–225.
[52] M. López-Ibáñez, L. Paquete, and T. Stützle, "Exploratory analysis of stochastic local search algorithms in biobjective optimization," in *Experimental Methods for the Analysis of Optimization Algorithms*. Berlin, Germany: Springer, 2010, pp. 209–222.

Mario Garza-Fabre received the M.Sc. and Ph.D. degrees in computer science from the Center for Research and Advanced Studies (Cinvestav), Campus Tamaulipas, Ciudad Victoria, Mexico, in 2009 and 2014, respectively.

He joined Cinvestav, Campus Tamaulipas, in 2018, where he is currently an Associate Professor in Optimization and Computational Intelligence. His main research interests and experience involve the analysis and design of heuristic optimization techniques, as well as their application to problems from diverse areas, such as bioinformatics, data mining/machine learning, transportation, and communications.

Julia Handl received the B.Sc. degree (Hons.) in computer science from Monash University, Melbourne, VIC, Australia, in 2001, the M.Sc. degree in computer science from the University of Erlangen–Nuremberg, Erlangen, Germany, in 2003, and the Ph.D. degree in bioinformatics from the University of Manchester, Manchester, U.K., in 2006.

She is currently an Alan Turing Fellow and a Professor with the Management Sciences Group, University of Manchester. Her research interests include theoretical and empirical work related to the development and use of machine learning and optimization approaches in a variety of application areas.

Adán José-García received the M.Sc. and Ph.D. degrees in computer science from the Center for Research and Advanced Studies (Cinvestav), Ciudad Victoria, Mexico, in 2012 and 2017, respectively.

Since 2021, he has been working as a Research Fellow in Digital Health with CRIStAL Lab, University of Lille, Lille, France, and the Lille University Hospital, Lille. His research mainly involves creating and adapting cluster analysis methods and their applications to different research fields, such as biomedicine, labor market, and education. He currently focuses on developing integrative unsupervised learning approaches to address healthcare-related data problems and help to understand better disease complications and treatment goals.