Chapter 6
Evolutionary Clustering and Community Detection

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

Abstract This chapter provides a formal definition of the problem of cluster analysis, and the related problem of community detection in graphs. Building on the mathematical definition of these problems, we motivate the use of evolutionary computation in this setting. We then review previous work on this topic, highlighting key approaches regarding the choice of representation and objective functions, as well as regarding the final process of model selection. Finally, we discuss successful applications of evolutionary clustering and the steps we consider necessary to encourage the uptake of these techniques in mainstream machine learning.

6.1 Introduction

Unsupervised learning is concerned with the identification of patterns in data in scenarios where information on the outcome of interest is not available directly. In other words, while supervised learning is concerned with the mapping from an input space X to an output (target) space Y, unsupervised learning is strictly limited to the analysis of X and the discovery of patterns inherent to that space.

The most commonly encountered question in unsupervised learning relates to the presence of natural groups within the data, i.e., subsets of entities that are inherently similar or related to each other and, at the same time, inherently dissimilar or unrelated to other entities within a data set. Where data is available in the form

J. Handl (✉)
University of Manchester, Manchester, UK
e-mail: julia.handl@manchester.ac.uk

M. Garza-Fabre
Cinvestav, Campus Tamaulipas, Km. 5.5 carretera Cd. Victoria-Soto La Marina, Cd. Victoria 87130, Tamaulipas, Mexico
e-mail: mario.garza@cinvestav.mx

A. José-García
Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
e-mail: adan.josegarcia@univ-lille.fr

W. Banzhaf et al. (eds.), *Handbook of Evolutionary Machine Learning*,
Genetic and Evolutionary Computation,
https://doi.org/10.1007/978-981-99-3814-8_6

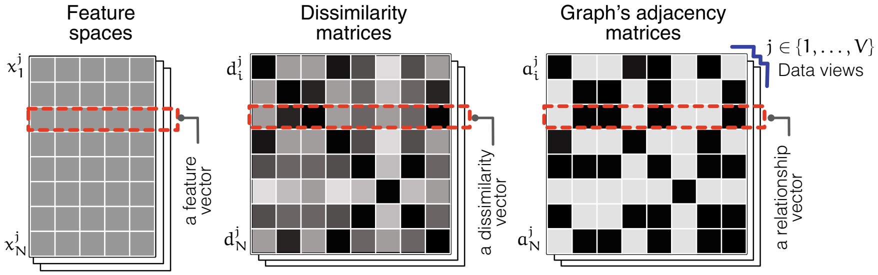


Fig. 6.1 Unsupervised learning scenarios relevant to the problems of cluster analysis and community detection. The different ways in which entities are characterized are highlighted in red: $\mathbf{x}_i^j$ is a feature vector, $\mathbf{d}_i^j$ is a dissimilarity vector, and $\mathbf{a}_i^j$ is a row of the adjacency matrix, such that $j \in \{1, \ldots, V\}$, where V represents the number of distinct data views available in the form of feature spaces, dissimilarity matrices, or relational spaces

of features, directly characterizing each entity, or in the form of (dis)similarities, capturing the pairwise relationships between all samples, the above problem is referred to as *clustering* or *cluster analysis* [33]. Otherwise, when the data is characterized primarily through a partial set of relations between the entities, information which is commonly represented as a graph, the resulting problem is known as *community detection* [12]. Figure 6.1 provides a side-by-side comparison of these three key scenarios, which may overlap in practice. Due to their close relationship, this chapter aims to provide a holistic overview of evolutionary approaches designed for all three problem settings.

6.2 Unsupervised Learning Scenarios

We are concerned with unsupervised learning on a set of entities X. Here, X is made up of individual entities $x_i \in X$, with $i \in \{1, \ldots, N\}$ and N being the cardinality of X (i.e., N is the total number of entities in the data set).

Depending on the particular scenario of unsupervised learning, further information about entities may be available as follows:

1. In many instances of unsupervised learning, each entity x_i is directly represented through a feature vector:

$$\mathbf{x}_i^j = (x_{i1}^j, \ldots, x_{iD^j}^j) \ .$$

Here, D^j represents the dimensionality of the j-th feature space, F^j. Some applications are characterized by the availability of multiple feature spaces (data views), so $j \in \{1, \ldots, V\}$, where V represents the number of distinct feature spaces available.

2. In certain applications of unsupervised learning, a feature representation of individual entities is not appropriate or available. Instead, entities may be characterized indirectly, through their dissimilarity or similarity relationships to all other entities within the data set, X. Concretely, in such scenarios, each entity x_i is represented through a dissimilarity vector:

   $$\mathbf{d}_i^j = (d_{i1}^j, \dots, d_{iN}^j) \ .$$

   In vector $\mathbf{d}_i^j$, d_{it}^j captures the dissimilarity between entity i and entity t, whereas N corresponds to the total number of entities within the data set. As above, some applications are characterized by the availability of multiple dissimilarity values, so $j \in \{1, \dots, V\}$, where V represents the number of distinct dissimilarity matrices (data views) available. Without loss of generality, this definition focuses on dissimilarities only, as relational information presented as a similarity matrix can be mapped to a dissimilarity matrix through a suitable mathematical transformation.
3. Finally, unsupervised learning scenarios can be characterized by the availability of relational information between subsets of entities only. This information can be naturally captured in the form of a graph. Concretely, in such scenarios, each entity x_i is modeled as a node of a graph, and the available relational information, for that node, is given by a single row of the graph's adjacency matrix:

   $$\mathbf{a}_i^j = (a_{i1}^j, \dots, a_{iN}^j) \ .$$

   In this case, a_{it}^j captures the strength of the relationship between the i-th and t-th data entities, with a value of $a_{it}^j = 0$ indicating the absence of relational information (and, therefore, absence of an edge between the associated nodes). As before, N refers to the total number of entities within the data set. In the simplest case, the entries of $\mathbf{a}_i^j$ are binary, i.e., $a_{it}^j = 0$ or $a_{it}^j = 1$, representing the absence or presence of a relation, respectively. Where present, the relation is represented as an unweighted edge in the graph. Alternatively, edges within the graph may be weighted and/or directed, reflecting the strength, and potential directionality, of the relationship between connected entities. Again, some applications may be characterized by the availability of multiple sets of relations, so $j \in \{1, \dots, V\}$, where V represents the number of distinct relational spaces (data views) available.

Considering the above definitions, the touching points between the three scenarios are clear. First, given the choice of a suitable distance function, problem instances consistent with Scenario 1 can always be transformed into instances consistent with Scenario 2, by mapping each feature space to a dissimilarity matrix. Second, Scenario 3 presents a generalization of Scenario 2. Specifically, it relaxes two assumptions usually made in cluster analysis: (i) the assumption that relational information is available (or can be derived) for all pairs of data entities; and (ii) the assumption of symmetry made by standard, metric distance functions. In other words, unsupervised learning problems arising in the form of Scenarios 1 and 2 can equivalently be modeled as learning problems on complete (i.e., fully connected), weighted, and undirected graphs, with edge weights capturing the similarity between entities.

It is also evident that we may encounter instances that combine aspects of different problem scenarios. In providing the above definitions, we have ensured to cater for *multi-view* settings, where data entities can be characterized from a number of incommensurable perspectives. In a multi-view setting, however, each individual data view may arise in a form consistent with a different scenario, so the full problem instance may in fact involve multiple scenarios. For example, a learning problem may involve two views, one available as a feature space and one available in the form of a dissimilarity matrix. Similarly, information provided in the form of a graph may be accompanied by node features, i.e., by a feature vector associated with each of the nodes (data entities) described by the graph. In this case, the learning problem can be thought of as a multi-view problem involving two views: a set of relational information captured by the graph's edges, and a feature space directly characterizing each entity.

6.3 Cluster Analysis and Community Detection

The previous section has defined different scenarios we may encounter in unsupervised learning, with the aim of highlighting the commonalities and potential interplay between those settings. We will now set out to provide a formal definition of the problems of cluster analysis and community detection. While these learning tasks have typically been studied in separate threads in the academic literature, below we will focus on a joint definition. The motivation behind this is the close relationship between them, as highlighted in the scenario definitions above.

The problems of cluster analysis and community detection aim to partition a given set of entities X into sub-groups. A generic definition of these partitioning problems is as follows:

$$\text{argmin}_{K,\Pi}(f^h(\Pi(X))) \ .$$

Here, K is the decision variable indicating the number of groups (clusters or communities) in a partition. Π defines a partition of X into subsets $\{X_1, \dots, X_K\}$, such that $\forall x_i \in X : \exists X_k : x_i \in X_k$. Commonly, definitions of these problems further assume Π to induce a *crisp partition*, i.e., $\forall l \neq m : X_l \cap X_m = \emptyset$.

In the above definition, f^h represents an objective function that captures the quality of a partition and, without loss of generality, is to be minimized. Note that $h \in \{1, \dots, H\}$ and, with $H > 1$, i.e., where multiple objective functions are to be used, this formulation results in a *multi-objective optimization problem*. The use of multiple objective functions can arise for a variety of reasons. It can help capture multiple different aspects of the quality of a partition, such as relationships within groups and across groups, or it can capture the quality of a partition with regard to multiple views that may be available for the entities considered.

In terms of the problem definition, key differences between cluster analysis and community detection relate only to the specific choices of objective functions, and the information these functions are based on. Fundamentally, this is where clustering algorithms rely on a direct feature representation of each entity, or the full matrix of pairwise dissimilarities between entities. In contrast, objective functions for community detection are expressed in terms of a graph's edges, and the presence or level of relationship (similarity) they convey.

In practice, specialized algorithms for both problems exist, which are not always transferable across problem boundaries. In particular, some of the best-known clustering algorithms assume the presence of a feature space, and do therefore not directly generalize to community detection settings.

6.4 Practical Challenges and Opportunities for Evolutionary Computation

There is a long history of research on cluster analysis and community detection, and this is evident from the large variety of algorithms and objective functions available for both problems [2, 51, 56]. There is a clear trade-off between the flexibility of a given algorithm and its scalability to large problems. Some of the most established algorithms are those that introduce strong assumptions about the data and/or rely on a greedy or local search heuristic. A prominent example of this is the k-means algorithm [39]: it relies on the external definition of the number of clusters, and lends itself to the (local) optimization of a specific objective function (within-cluster variance) only. In return, it achieves a run time complexity that is linear in terms of the number of entities in a data set, and remains one of the most commonly used algorithms in practice [31, 33]. Given steadily increasing demands regarding the scale of data and the speed with which it requires processing, the identification of fast, specialized heuristics (and associated objective functions) remains an important active area of research.

On the other hand, there are applications in which the narrow assumptions made by such heuristics are constraining. Their use may prevent problem formulations that are sufficiently comprehensive to cover all relevant problem aspects, including the full range of data views available. Meta-heuristics such as evolutionary algorithms can provide a powerful alternative in this setting. One of the specific advantages of meta-heuristic algorithms is their flexibility to adapt to different problem formulations, i.e., the opportunity they provide to exchange or experiment with additional objective functions, constraints, and data views, without a complete overhaul of the underlying optimization engine. The ability to perform a wider, global exploration of the search space and escape more easily from local optima is another fundamental advantage of meta-heuristic methods [53, 59].

In the following, we highlight some of the key developments made by evolutionary computation researchers in designing approaches to clustering and community detection. Our discussion focuses on the crucial design challenges associated with the adoption of existing meta-heuristic optimizers for the unsupervised learning domain, including issues around problem representation and operator design, the interaction of these choices with problem formulation (in terms of objective functions), and the strategies for addressing final model selection. It is our experience that the largest performance differences in unsupervised learning arise from decisions with respect to the problem formulation, rather than the specifics (or parameterization) of the optimizer used. Hence, our discussion will focus on the choice of meta-heuristic optimizer only where this choice is essential to support aspects of the problem formulation.

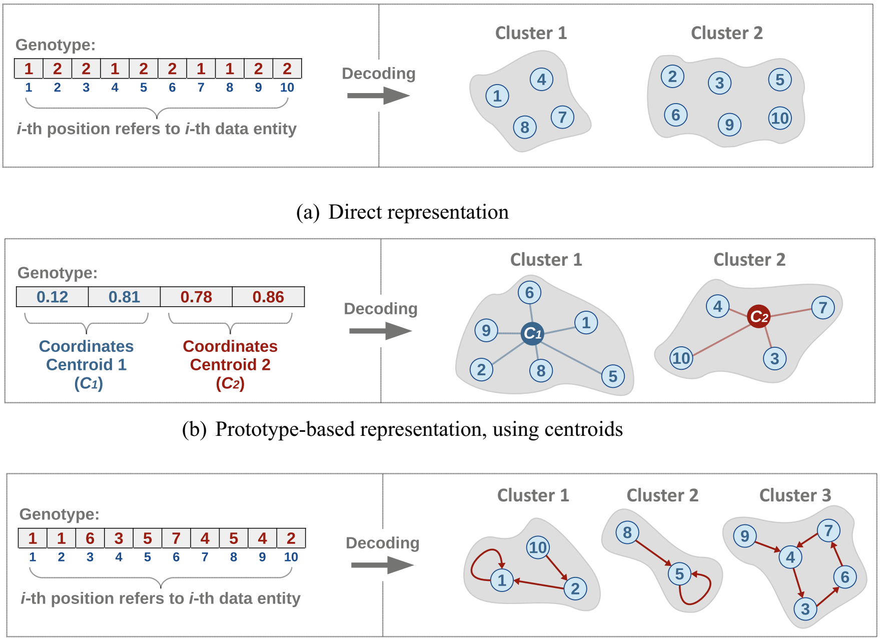
6.4.1 Solution Representation

Representation plays a crucial role in the adaption of meta-heuristic optimizers to a given problem [53, 59]. Jointly with the variation operators, they define the phenotypic neighborhoods that are accessible from a given candidate solution during optimization, thereby impacting on problem difficulty and the effectiveness of the overall search process. In cluster analysis, this aspect is further amplified by the fact that the choice of representation may predetermine the type of partition that can be induced, so it directly affects the formulation of the optimization problem and the set of phenotypes that can be reached [34, 45].

6.4.1.1 Direct Representations

The simplest representation of a partition relies on the use of a separate decision variable to indicate the cluster or community assignment for every data entity. A direct problem representation can thus be designed as a string $\mathbf{r} = (r_1, \ldots, r_N)$, where $r_i \in \{1, \ldots, K\}$ represents the cluster or community entity $x_i \in X$ is assigned to, as shown in Fig. 6.2a. Alternatively, a binary representation of these integer values could be adopted, resulting in a binary string of length $N \times \lceil \log_2(K) \rceil$.

The advantage of these direct representations is their generality. Other than a maximum number of clusters $K_{\max}$, they introduce no assumptions regarding the properties of partitions, and they can therefore be deployed in optimizing partitions with respect to any objective function or constraint. Nevertheless, both representations share the same core issue, which is rooted in the lack of a direct interpretation of the specific cluster or community labels: the labels only serve to induce the co-assignment (and separation) of entities, but the final phenotype (partition) is agnostic to the specific label used. Consider, for example, genotype $(1, 2, 2, 1, 2, 2, 1, 1, 2, 2)$ illustrated in Fig. 6.2a (for a problem with $N = 10$ and $K = 2$). If we exchange cluster labels, so that entities originally assigned to Cluster 1 are now assigned to Cluster 2 and vice versa, we will obtain a completely different new genotype, namely $(2, 1, 1, 2, 1, 1, 2, 2, 1, 1)$. Note, however, that both of these genotypes induce the same two sets of data entities. When it comes to evolutionary algorithms, in particu-



(a) Direct representation

(b) Prototype-based representation, using centroids

(c) Graph-based representation

Fig. 6.2 Illustration of the (**a**) direct, (**b**) prototype-based, and (**c**) graph-based representations used in evolutionary clustering (equivalent choices are available for community detection). An example problem instance with $N = 10$ data entities is considered in all cases

lar, the resulting redundancy makes it difficult to derive effective crossover operators for this representation. Adjustments designed to address this issue for grouping problems [11] have shown limited success, especially for the problem scales typically considered during cluster analysis [24]. Consequently, the use of direct representations in the clustering and community detection literature is sparse.

6.4.1.2 Prototype-Based Representations

In meta-heuristic approaches to cluster analysis, prototype-based representations have been extensively used to overcome issues with scalability. Rather than specifying cluster assignments directly, prototype-based approaches specify a set of representatives from which a partition can be induced: each entity is assigned to the cluster associated with the representative it is closest to—in the following we refer to this as the *cluster assignment step*. These representatives may adopt the form of cluster centroids or cluster medoids, depending on the clustering scenario at hand and, specifically, on the availability of a numerical feature space, which is a prerequisite for the calculation of cluster centroids. Hence, the representation can take one of the following forms:

1. A string of $K \times D$ real values, $\mathbf{r} = (r_{11}, \ldots, r_{1D}, \ldots, r_{K1}, \ldots, r_{KD})$, where $(r_{k1}, \ldots, r_{kD})$ represents the centroid vector of the k-th cluster. Figure 6.2b illustrates this representation based on centroids, for $K = 2$ and $D = 2$.
2. A string of K integer values, $\mathbf{r} = (r_1, \ldots, r_K)$, where $r_k \in \{1, \ldots, N\}$ identifies the specific entity $x_{r_k} \in X$ serving as the cluster medoid for cluster X_k. Alternatively, a binary representation of each medoid could be used.

Either representation can induce a partition on X by assigning each entity to its closest cluster representative, using a designated distance function (in the case of cluster centroids) or dissimilarity matrix (in the case of cluster medoids). Standard prototype-based approaches assume that the number of groups K in the partitions is known in advance, as this defines the representation length. This assumption can be relaxed to assume knowledge of the maximum number of groups $K_{\max}$ only, e.g., through the introduction of placeholder values [4]. The definition of a centroid or medoid does not extend directly to a graph-based scenario, and therefore community detection. However, alternative definitions of node representativeness, within graphs, have been derived and used in this context [64], allowing the adoption of a representation equivalent to the medoid-based approach described above.

A clear advantage of the prototype-based approach is its scalability: it eliminates the linear increase of representation length with the number of entities, N. Instead, representation length becomes a function of the number of desired groups, K, and dimensionality, D (for centroid-based representations). For medoid-based representations, the number of available allele values for each encoding position increases with N, so there remains a dependency of the search space on the number of entities. This becomes explicit in a binary representation: where this is used to represent medoids, the representation length is $K \times \lceil \log_2(N) \rceil$. A separate advantage of prototype-based representations is the fact that, for those assuming a fixed K, routine (generic) variation operators can potentially be used, in the case of real, integer, and binary encodings.

Despite these distinct strengths, key limitations of the prototype-based representations result from the cluster assignment step. The first of these limitations relates to the cluster shapes that can be obtained. In particular, when the standard Euclidean distance is used to support cluster assignment, this restricts the search to clusters with a hyper-spherical shape (i.e., clusters that are compact). This restriction at the representational level will override decisions made during the choice of the objective functions. Consequently, the use of a prototype-based representation can significantly limit an algorithm's ability to discover data partitions that violate assumptions of compactness but provide promising characteristics from other points of view, e.g., clusters that are elongated but spatially well-separated. Recent work has highlighted potential avenues for reducing this limitation [36]. As illustrated in Fig. 6.3, transformations of the original dissimilarity space (choices of more complex distance functions) can potentially address this issue, enabling the use of prototype-based representations for the identification of non-compact clusters.

A second difficulty with prototype-based representations arises in clustering settings involving multiple data views. As the cluster assignment step requires the calcu-

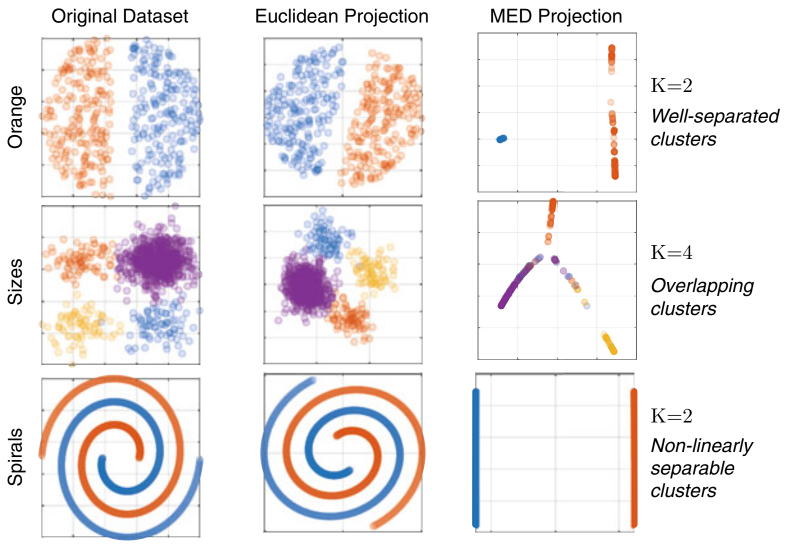


Fig. 6.3 The impact of the distance function on how an algorithm “perceives” the relationships between a given set of entities, which will directly determine cluster assignments in a prototype-based representation: original data (left); embedding of the associated Euclidean distances (center); and embedding of the associated MED distances [6] (right). This shows that the MED distance favors the clear spatial separation of elongated cluster structures (e.g., on the `Spirals` data) but is less capable than the Euclidean distance in separating overlapping clusters (e.g., on the `Sizes` data). This underlines that the choice of distance function can play a significant role in determining the types of clusters that can be identified, irrespective of the choice of objective function

lation or use of dissimilarity information, it becomes a potential source of bias when multiple incommensurable sources of dissimilarity information exist. In particular, reliance on a single source, or on a fixed weighting between the available sources, will override decisions made during the choice of the objective functions. Recent work has highlighted how the introduction of such bias can be avoided in the context of a many-objective optimizer [36]. Specifically, the use of a decomposition-based approach [32, 63] has been exploited as a mechanism to directly align reference vectors deployed within the optimizer with the weights used during cluster assignment (see Fig. 6.4), avoiding the introduction of unintended bias during the search process.

6.4.1.3 Graph-Based Representations

Finally, graph-based representations have enjoyed notable success in identifying partitions, for both clustering and the problem of community detection [30, 51]. In community detection, the rationale for a graph-based representation is straightforward, as it aligns very closely with the learning problem at hand. In cluster analysis, this particular choice of representation is less intuitive. However, in the context

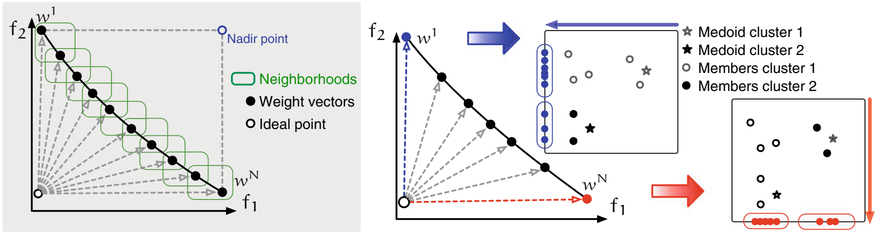


Fig. 6.4 Example of a many-objective optimizer considering many views at the cluster assignment step: This is possible due to explicit knowledge of the weights associated with each reference vector, which are exploited during the decoding (and cluster assignment) as well as the evaluation step of the algorithm

of multi-objective clustering in particular, graph-based representations have been shown to provide an effective mechanism to explore a diverse range of candidate partitions [24].

The most straightforward version of the graph-based representation captures the set of all edges within a graph (or all pairwise relations within a data set), and introduces a binary decision variable to allow for the inclusion or removal of each of them. In other words, the representation of a partition is given as a string of E binary values, $\mathbf{r} = (r_1, \ldots, r_E)$, where E denotes the number of edges in the original graph and $r_e \in \{0, 1\}$ indicates the absence or presence of a given link in the candidate graph. To interpret each candidate graph as a partition, an additional decoding step determines the set of connected components defined by the links; each such component is interpreted as a separate cluster or community. This representation can be suitable for small problem instances, and introduces no additional bias, but lacks scalability in the case of cluster analysis and densely connected graphs, as the number of edges will then grow as $\mathcal{O}(N^2)$.

Alternatively, and more commonly, a candidate partition can be represented by explicitly defining connections between data entities or graph nodes [49, 50]. Here, the representation of a partition is given as $\mathbf{r} = (r_1, \ldots, r_N)$, a string of N integer values, where N denotes the number of data entities or nodes in the graph. In a clustering context, the associated interpretation is for $r_i \in \{1, \ldots, N\}$ to represent the index of one other data entity which x_i is connected to, as illustrated in Fig. 6.2c. In a community detection setting (or a clustering setting with additional assumptions), a more compressed form of this representation becomes available by exploiting the concept of an adjacency list: here $r_i \in \{1, \ldots, L_i\}$ refers to an index into the original graph's adjacency list for node x_i, of length L_i. Although an integer encoding is used in both cases, in the compressed form the possible alleles (range of integer values) may vary across genotype positions.

As before, an additional decoding step is required to interpret such an integer string as a partition. First, a candidate graph is constructed, containing the set of N nodes and edges between entities x_i and x_j, iff $r_i = x_j$ or, in the compressed encoding, if x_j corresponds to the r_i-th entry in the adjacency list of x_i (or vice versa). Subsequently, the connected components of this graph are determined, and each of them is interpreted as a separate cluster or community. Similar to the binary representation discussed above, this approach can be highly effective in small problem instances, but can suffer from poor scalability in the case of cluster analysis and densely connected graphs. While the integer string is restricted to length N, the number of the allele choices can still grow as a function of N, leading to an exponential increase in the size of the solution space, which is highly redundant.

In cluster analysis, the scalability issues of the locus-based adjacency representation have been addressed in different ways [17, 18, 24, 65]. Firstly, by limiting the adjacency list for each node to a reduced number of possibilities, e.g., representing the nearest neighbors of each data point. Secondly, by designing specialized initialization schemes, exploiting minimum-spanning trees, in order to bias the search process toward the most promising edges of the graph. Lastly, by pre-processing the data set in advance, so that only the most relevant decisions become the focus of the optimization process [17]. In community detection, node and edge centrality may be used as additional sources of heuristic bias, during the initialization and variation stages of the search [27].

6.4.2 Objective Functions

Assuming the use of any competent optimizer (as well as the absence of harmful bias at the representation level), the choice of objective function is arguably the most important decision determining the outcome of an unsupervised learning task. Building on the general problem formulation derived in Sect. 6.3, this section aims to highlight the different types of criteria we may choose to integrate as objectives for a given application, and the rationale behind this.

One of the most challenging aspects of cluster analysis and community detection is the lack of a clearly defined quality criterion. As a consequence, the literature contains a wide variety of objective functions, each prioritizing different (and sometimes combinations of) quality aspects of a partition [2, 35]. Furthermore, while there are conceptual similarities between the objective functions used in cluster analysis and community detection, the two fields generally use different approaches. For example, both fields feature measures that consider the strength of inter-group relations, but clustering objectives typically do so by assessing the maximum or average dissimilarity within a cluster (e.g., diameter [28] or within-cluster variance [38]), while the field of community detection relies on variants of modularity, which assess within-cluster edge densities relatively to a statistical null model [46].

The diversity of potential objectives arises due to the range of complementary, yet conflicting aspects that we attribute to a good partition of a graph or data set, and the difficulty in defining such aspects mathematically. Generally, there is an agreement that partitions are derived with the aim of identifying groups that are homogeneous and mutually distinct. However, there are very different ways in which homogeneity and distinctiveness can be described mathematically. For example, within-group homogeneity may be translated into a requirement to resemble all other entities in the same group, or it may suffice to resemble a cluster representative, or even just another subset of entities in the group; such different assumptions will result in drastically different groups in many data sets.

A direct implication of this ambiguity is that the available objective functions typically capture some but not all desirable aspects of a clustering or community detection problem. Thus, there is a need to carefully align objectives with the modeling intentions in a given application. Furthermore, when desirable partition attributes are not clear in advance, the simultaneous consideration of multiple objectives can provide a more comprehensive problem formulation that avoids premature decisions on the importance of different quality facets.

Meta-heuristic algorithms are well-suited to assist with the above challenges. Unlike purpose-built heuristics, they are sufficiently flexible to allow for experimentation with different objective functions. Moreover, due to their population-based approach, multi-objective evolutionary algorithms (MOEAs) provide a particularly convenient approach to the simultaneous optimization of multiple objectives, and the exploration of trade-offs between them. Cluster analysis and community detection are often applied for exploratory data analysis, i.e., to assist with the understanding of a novel data set. MOEAs facilitate the exploration of a select range of candidate solutions, and the learning associated with this can add distinct value to the process.

Multiple objectives in clustering and community detection do not solely derive from the definition of multiple quality criteria. A further complicating factor in the definition of a partition is the decision on the number of groups to create, K. Algorithms like k-means [39] side-step this issue by making K a user-defined parameter. However, where algorithms are required to determine K as part of the optimization process, integration of objectives with opposing biases (with respect to K) has been found to be particularly useful [17, 24]. Elsewhere, the value of K has been explicitly employed as an additional optimization criterion [37, 48, 62].

As highlighted in Sect. 6.3, multiple objectives can also play the role of assessing partition quality with respect to multiple views of a data set. If a set of entities is characterized by multiple feature or dissimilarity spaces, we will usually wish to identify partitions that take into account, and are consistent with, all available information. Where the relative importance or reliability of these views are unknown, capturing them through individual objectives, and optimizing these using a Pareto optimization approach, can help with exploring a range of trade-offs and understanding both conflict and alignment between them [7, 18, 36, 55].

Finally, unsupervised learning settings may involve additional constraints or objectives that need to be considered in characterizing optimal partitions. These may include, for example, considerations related to group sizes [40], prior domain knowledge [9], or aspects of fairness [8]. Multi-objective approaches provide a convenient mechanism to integrate any of these into the search process.

6.4.3 Model Selection

The previous section has highlighted the potential benefits of problem formulations integrating a number of optimization criteria, either to capture desirable properties of a partition more comprehensively or to account for the availability of multiple views, and explore the trade-offs between them. However, one of the challenges this presents is the need for final model selection: typically, an unsupervised learning scenario requires the ultimate selection of a single solution.

Problems of model selection equally arise in traditional approaches to cluster analysis and community detection. For example, the use of k-means commonly involves running this algorithm for a range of possible cluster numbers, and using an additional cluster validity index [3, 35] to support the selection of a final solution from the set of alternatives obtained. Similar approaches can and have been suggested in an evolutionary clustering context, including scenarios involving multiple criteria [5, 15] and multiple data views [36]. A cluster validity index frequently used for model-selection purposes is the *Silhouette Width* [54], due to its prominence in the field of clustering. In general, measures for model selection may be at their most effective if they do not fully coincide with the objectives already considered during the optimization stage of the search process [26], i.e., when they provide complementary guidance.

More innovative approaches focus on analyzing the shape of the approximation set returned by multi-objective optimizers, using this to pinpoint the most promising solution alternatives. For example, one approach to solution selection focuses on the identification of *knee* solutions [23, 24, 42, 57], i.e., solutions that present particularly promising performance trade-offs between the objective functions considered. This builds on ongoing work in the evolutionary multi-objective optimization literature [10, 29, 58], but also shares similarities with existing approaches from the clustering literature, such as the Gap statistic [61] and the elbow method [60].

Some authors have explored the interpretation of the final approximation set as a clustering ensemble [21]. Drawing on previous work in ensemble learning [13], the aim is to derive a single consensus solution that best represents the information captured in the approximation front [25, 43, 44, 52, 65]. The rationale behind this approach is that all solutions in the approximation set contain useful information regarding the correct partition. However, potential disadvantages include: the focus on a majority consensus, which means that information from the extremes of the approximation front, in particular, may be insufficiently represented; and the assumption that all nondominated partitions are equally reliable, which is not necessarily the case and can affect the resulting consensus.

Model selection is a challenging task and remains an active area of research. As analyzed in [19], all of the above-discussed strategies have shown some success during empirical evaluations, but their limitations are rooted in assumptions that may not hold in every scenario. As an alternative, the potential of machine learning to capture the complexities of the task is showcased [19] reporting promising results. A supervised learning approach is explored, relying on the initial construction of

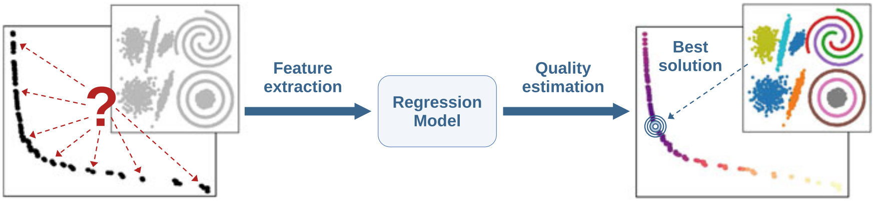


Fig. 6.5 Supporting model selection in multi-objective clustering through machine learning. Each individual solution in the approximation front is characterized (feature extraction). Then, the quality of solutions is estimated by a (previously trained) regression model. The solution with the highest estimated quality is identified and selected as the final result [19]

a regression model that is later exploited to estimate the quality of solutions in a given approximation front (see Fig. 6.5). Model construction depends on: (i) a collection of sample problems with known solutions; (ii) a set of training fronts for each sample problem, generated by some algorithm; (iii) the extraction of features for all solutions in the training fronts; (iv) the quality assessment of these solutions by direct comparison with the known correct solution using an *external* validity index[1]; and (v) the use of a machine learning method to build the regression model, using the extracted features as explanatory variables and the solution-quality measurements as the response variable. The current approach to feature extraction seeks to characterize nondominated points based on individual properties of the partitions they encode, their relation to other solutions in the approximation front, as well as global aspects of the entire front and the particular clustering problem being solved. Feature extraction is a key issue in this methodology and deserves further investigation.

6.4.4 Choice of Optimizer

A variety of evolutionary algorithms and alternative meta-heuristics have been applied to the tasks of clustering and community detection [1, 20, 34, 47]. As discussed earlier, the choice of representation (with associated variation operators) and the choice of optimization criteria are arguably the most important design aspects in determining the performance of these approaches. Furthermore, these aspects ultimately predetermine the general class of algorithms that should be used.

For example, the continuous encoding implicit to standard prototype-based representations allows for the use of optimization methods designed for optimization over continuous spaces. In contrast, the use of a graph-based representation necessitates the use of integer or binary optimization approaches. Similarly, choices about the number and nature of objectives have driven decisions toward the adoption of multi-objective and many-objective optimization approaches. Given identical choices of

[1] Cluster validity indices can be external or internal, depending on whether or not they depend on knowledge of the correct partition (ground truth) to determine solution quality.

representation, variation operators, and objective(s), there are bound to remain performance differences between distinct choices of optimizers. There has been limited investigation of this aspect in the literature [41, 45], largely (we speculate) due to prioritization: in terms of recovery of the ground truth, the associated performance differences are likely to be outweighed by those reflecting fundamental changes in problem formulation. Nevertheless, one way of addressing this shortfall, going forward, may be the generation and inclusion of benchmarks derived from clustering problems into benchmark suites routinely used for the development of general-purpose meta-heuristics.

6.5 Final Perspectives

As highlighted above, evolutionary approaches have an important role to play in helping to explore novel, and potentially more comprehensive, formulations of clustering and community detection. In this chapter, we have aimed to highlight those design issues that we find to be of particular relevance in defining the capabilities of such approaches. However, in addition to the points highlighted so far, there are other areas requiring further development.

Whether it refers to the size of the problem (data set) and its dimensionality, or to the number of optimization criteria, scalability remains a key challenge for the use of evolutionary algorithms (and other meta-heuristics) in unsupervised learning applications. Regarding problem size, recent work has highlighted some progress and core mechanisms that can be exploited, including the use of stratification [14] and changes to problem resolution which impact on the granularity of the search [17]. Increases in the number of optimization criteria, either to handle multiple performance aspects or data views, can benefit from current developments in the area of many-objective optimization [36].

Finally, the field needs to address issues around the uptake of the algorithms by practitioners and the mainstream machine learning literature. While promising applications of evolutionary clustering approaches have been reported [1], and several different implementations are now available through GitHub, uptake of the techniques outside of the evolutionary computation community remains limited. Future work needs to prioritize the dissemination of these methods within the machine-learning and practitioner community, and the translation of algorithms into standard software packages, with a view of encouraging increased adoption of this research into practical applications. It is crucial that such messaging is transparent in terms of the relative strengths and weaknesses of current approaches. As set out in our chapter, evolutionary computation offers advantages when it comes to comprehensive, flexible problem formulations, and the literature has highlighted this, e.g., by demonstrating how changes in problem formulation can drive robustness toward a wider range of data properties than traditional algorithms [16, 22, 24]. However, it is clear that these benefits also come at significant computational cost, creating barriers for applications requiring fast throughput or handling of very large data sets. Helping practitioners understand the resulting trade-offs relies on routinely including comparisons to state-of-the-art non-evolutionary machine learning techniques, as well as considerations of time complexity [30], into published work. The best choice of algorithm for a given unsupervised learning setting will always depend on a range of factors, and we need to ensure that the place of evolutionary-based approaches is better understood.

References

1. Aljarah, I., Faris, H., Mirjalili, S.: Evolutionary data clustering: Algorithms and applications. Springer, Berlin (2021)
2. Aljarah, I., Habib, M., Nujoom, R., Faris, H., Mirjalili, S.: A comprehensive review of evaluation and fitness measures for evolutionary data clustering. In: Aljarah, I., Faris, H., Mirjalili, S. (eds.) Evolutionary Data Clustering: Algorithms and Applications, pp. 23–71. Springer Singapore, Singapore (2021)
3. Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J.M., Perona, I.: An extensive comparative study of cluster validity indices. Pattern Recogn. **46**(1), 243–256 (2013)
4. Bandyopadhyay, S., Maulik, U.: Genetic clustering for automatic evolution of clusters and application to image classification. Pattern Recogn. **35**(6), 1197–1208 (2002)
5. Bandyopadhyay, S., Mukhopadhyay, A., Maulik, U.: An improved algorithm for clustering gene expression data. Bioinformatics **23**(21), 2859 (2007)
6. Bayá, A.E., Granitto, P.M.: How many clusters: a validation index for arbitrary-shaped clusters. IEEE/ACM Trans. Comput. Biol. Bioinf. **10**(2), 401–414 (2013)
7. Caballero, R., Laguna, M., Martí, R., Molina, J.: Scatter Tabu search for multiobjective clustering problems. J. Oper. Res. Soc. **62**(11), 2034–2046 (2011)
8. Chhabra, A., Masalkovaitė, K., Mohapatra, P.: An overview of fairness in clustering. IEEE Access **9**, 130698–130720 (2021)
9. Davidson, I., Ravi, S.S.: Clustering with constraints: Feasibility issues and the k-means algorithm. In: Proceedings of the 2005 SIAM International Conference on Data Mining, pp. 138–149. SIAM (2005)
10. Deb, K., Gupta, S.: Understanding knee points in bicriteria problems and their implications as preferred solution principles. Eng. Optim. **43**(11), 1175–1204 (2011)
11. Falkenauer, E.: Genetic Algorithms and Grouping Problems. Wiley Ltd, Chichester (1998)
12. Fortunato, S.: Community detection in graphs. Phys. Rep. **486**(3), 75–174 (2010)
13. Fred, A.L.N., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Trans. Pattern Anal. Mach. Intell. **27**(6), 835–850 (2005)
14. Garcia-Piquer, A., Bacardit, J., Fornells, A., Golobardes, E.: Scaling-up multiobjective evolutionary clustering algorithms using stratification. Pattern Recogn. Lett. **93**, 69–77 (2017)
15. Garcia-Piquer, A., Sancho-Asensio, A., Fornells, A., Golobardes, E., Corral, G., Teixidó-Navarro, F.: Toward high performance solution retrieval in multiobjective clustering. Inf. Sci. **320**, 12–25 (2015)
16. Garza-Fabre, M., Handl, J., Knowles, J.: A new reduced-length genetic representation for evolutionary multiobjective clustering. In: Trautmann, H., Rudolph, G., Klamroth, K., Schütze, O., Wiecek, M., Jin, Y., Grimme, C. (eds.) Evolutionary Multi-Criterion Optimization, pp. 236–251. Springer International Publishing, Münster (2017)
17. Garza-Fabre, M., Handl, J., Knowles, J.: An improved and more scalable evolutionary approach to multiobjective clustering. IEEE Trans. Evol. Comput. **22**(4), 515–535 (2018)
18. Garza-Fabre, M., Handl, J., José-García, A.: Evolutionary multi-objective clustering over multiple conflicting data views. IEEE Trans. Evolut, Comput (2022)

19. Garza-Fabre, M., Sánchez-Martínez, A.L., Aldana-Bobadilla, E., Landa, R.: Decision making in evolutionary multiobjective clustering: a machine learning challenge. IEEE Access 1–22 (2022)
20. Gharehchopogh, F.S., Abdollahzadeh, B., Khodadadi, N., Mirjalili, S.: Chapter 20 - metaheuristics for clustering problems. In: Mirjalili, S., Gandomi, A.H. (eds.), Comprehensive Metaheuristics, pp. 379–392. Academic (2023)
21. Golalipour, K., Akbari, E., Hamidi, S.S., Lee, M., Enayatifar, R.: From clustering to clustering ensemble selection: a review. Eng. Appl. Artif. Intell. **104**, 104388 (2021)
22. Gong, C., Chen, H., He, W., Zhang, Z.: Improved multi-objective clustering algorithm using particle swarm optimization. PLoS ONE **12**(12), e0188815 (2017)
23. Gupta, A., Datta, S., Das, S.: Fuzzy clustering to identify clusters at different levels of fuzziness: an evolutionary multiobjective optimization approach. IEEE Trans. Cybern. **51**, 2601–2611 (2021)
24. Handl, J., Knowles, J.: An evolutionary approach to multiobjective clustering. IEEE Trans. Evol. Comput. **11**(1), 56–76 (2007)
25. Handl, J., Knowles, J.: Evidence accumulation in multiobjective data clustering. In: International Conference on Evolutionary Multi-Criterion Optimization, pp. 543–557. Springer (2013)
26. Handl, J., Knowles, J., Kell, D.B.: Computational cluster validation in post-genomic data analysis. Bioinformatics **21**(15), 3201–3212 (2005)
27. Handl, J., Ospina-Forero, L., Cann, T.: Multi-objective community detection for bipartite graphs. Under Submission (2022)
28. Hansen, P., Jaumard, B.: Minimum sum of diameters clustering. J. Classif. **4**(2), 215–226 (1987)
29. He, Z., Yen, G.G., Ding, J.: Knee-based decision making and visualization in many-objective optimization. IEEE Trans. Evol. Comput. **25**(2), 292–306 (2021)
30. Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A., Ponce, A.C., de Carvalho, L.F.: A survey of evolutionary algorithms for clustering. IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.) **39**(2), 133–155 (2009)
31. Ikotun, A.M., Ezugwu, A.E., Abualigah, L., Abuhaija, B., Heming, J.: K-means clustering algorithms: a comprehensive review, variants analysis, and advances in the era of big data. Inf. Sci. **622**, 178–210 (2023)
32. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: a short review. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 2419–2426. IEEE (2008)
33. Jain, A.K.: Data clustering: 50 years beyond K-means. Pattern Recogn. Lett. **31**(8), 651–666 (2010)
34. José-García, A., Gómez-Flores, W.: Automatic clustering using nature-inspired metaheuristics: a survey. Appl. Soft Comput. **41**, 192–213 (2016)
35. José-García, A., Gómez-Flores, W.: A survey of cluster validity indices for automatic data clustering using differential evolution. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '21, pp. 314–322. ACM, New York (2021)
36. José-García, A., Handl, J., Gómez-Flores, W., Garza-Fabre, M.: An evolutionary many-objective approach to multiview clustering using feature and relational data. Appl. Soft Comput. **108**, 107425 (2021)
37. Liu, Y., Özyer, T., Alhajj, R., Barker, K.: Integrating multi-objective genetic algorithm and validity analysis for locating and ranking alternative clustering. Informatica **29**, 33–40 (2005)
38. MacQueen, J.: Classification and analysis of multivariate observations. In: 5th Berkeley Symposium Mathematics Statistics Probability, pp. 281–297 (1967)
39. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. University of California Press (1967)
40. Malinen, M.I., Fränti, P.: Balanced k-means for clustering. In: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), pp. 32–41. Springer (2014)

41. Martínez-Peñaloza, M.-G., Mezura-Montes, E., Cruz-Ramírez, N., Acosta-Mesa, H.-G., Ríos-Figueroa, H.-V.: Improved multi-objective clustering with automatic determination of the number of clusters. Neural Comput. Appl. **28**, 2255–2275 (2017)
42. Matake, N., Hiroyasu, T., Miki, M., Senda, T.: Multiobjective clustering with automatic K-determination for large-scale data. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, GECCO '07, pp. 861–868. ACM, London (2007)
43. Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S.: Multiobjective genetic algorithm-based fuzzy clustering of categorical attributes. IEEE Trans. Evol. Comput. **13**(5), 991–1005 (2009)
44. Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S.: Multiobjective genetic clustering with ensemble among pareto front solutions: application to MRI brain image segmentation. In: 2009 Seventh International Conference on Advances in Pattern Recognition, pp. 236–239 (2009)
45. Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S.: A survey of multiobjective evolutionary clustering. ACM Comput. Surv. **47**(4) (2015)
46. Newman, M.E.J.: Modularity and community structure in networks. Proc. Natl. Acad. Sci. **103**(23), 8577–8582 (2006)
47. Osaba, E., Del Ser, J., Camacho, D., Bilbao, M.N., Yang, X.S.: Community detection in networks using bio-inspired optimization: latest developments, new results and perspectives with a selection of recent meta-heuristics. Appl. Soft Comput. **87**, 106010 (2020)
48. Özyer, T., Liu, Y., Alhajj, R., Barker, K.: Multi-objective genetic algorithm based clustering approach and its application to gene expression data. In: Yakhno, T. (ed.) Advances in Information Systems, pp. 451–461. Springer, Berlin (2005)
49. Park, Y.J., Song, M.S.: A genetic algorithm for clustering problems. In: Genetic Programming, pp. 568–575. Morgan Kaufmann, Madison (1998)
50. Pizzuti, C.: Ga-net: a genetic algorithm for community detection in social networks. In: International Conference on Parallel Problem Solving from Nature, pp. 1081–1090. Springer (2008)
51. Pizzuti, C.: Evolutionary computation for community detection in networks: a review. IEEE Trans. Evol. Comput. **22**(3), 464–483 (2018)
52. Qian, X., Zhang, X., Jiao, L., Ma, W.: Unsupervised texture image segmentation using multiobjective evolutionary clustering ensemble algorithm. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), pp. 3561–3567 (2008)
53. Rothlauf, F.: Representations for genetic and evolutionary algorithms, 2nd edn. Springer, Berlin (2006)
54. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
55. Saha, S., Mitra, S., Kramer, S.: Exploring multiobjective optimization for multiview clustering. ACM Trans. Knowl. Discov. Data **12**(4), 44:1–44:30 (2018)
56. Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O.P., Tiwari, A., Er, M.J., Ding, W., Lin, C.T.: A review of clustering techniques and developments. Neurocomputing **267**, 664–681 (2017)
57. Shirakawa, S., Nagao, T.: Evolutionary image segmentation based on multiobjective clustering. In: IEEE Congress on Evolutionary Computation, pp. 2466–2473 (2009)
58. Shukla, P.K., Braun, M.A., Schmeck, H.: Theory and algorithms for finding knees. In: International Conference on Evolutionary Multi-Criterion Optimization, pp. 156–170. Springer (2013)
59. Talbi, E.G.: Metaheuristics from design to implementation. Wiley (2009)
60. Thorndike, R.L.: Who belongs in the family. In: Psychometrika. Citeseer (1953)
61. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. J. R. Stat. Soc.: Ser. B (Stat. Methodol.) **63**(2), 411–423 (2001)
62. Wahid, A., Gao, X., Andreae, P.: Multi-view clustering of web documents using multi-objective genetic algorithm. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 2625–2632. IEEE, Beijing (2014)
63. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. **11**(6), 712–731 (2007)

64. Zhou, K., Martin, A., Pan, Q.: A similarity-based community detection method with multiple prototype representation. Phys. A **438**, 519–531 (2015)
65. Zhu, S., Lihong, X., Goodman, E.D.: Evolutionary multi-objective automatic clustering enhanced with quality metrics and ensemble strategy. Knowl.-Based Syst. **188**, 105018 (2020)