Automatic Clustering Using Nature-Inspired Metaheuristics: A Survey

Adán José-García[a], Wilfrido Gómez-Flores[a,*]

[a] *Technology Information Laboratory, Center for Research and Advanced Studies of the National Polytechnic Institute, Ciudad Victoria, Tamaulipas, Mexico*

Abstract

In cluster analysis, a fundamental problem is to determine the best estimate of the number of clusters; this is known as the automatic clustering problem. Because of lack of prior domain knowledge, it is difficult to choose an appropriate number of clusters, especially when the data have many dimensions, when clusters differ widely in shape, size, and density, and when overlapping exists among groups. In the late 1990s, the automatic clustering problem gave rise to a new era in cluster analysis with the application of nature-inspired metaheuristics. Since then, researchers have developed several new algorithms in this field. This paper presents an up-to-date review of all major nature-inspired metaheuristic algorithms used thus far for automatic clustering. Also, the main components involved during the formulation of metaheuristics for automatic clustering are presented, such as encoding schemes, validity indices, and proximity measures. A total of 65 automatic clustering approaches are reviewed, which are based on single-solution, single-objective, and multiobjective metaheuristics, whose usage percentages are 3 %, 69 %, and 28 %, respectively. Single-objective clustering algorithms are adequate to efficiently group linearly separable clusters. However, a strong tendency in using multiobjective algorithms is found nowadays to address non-linearly separable problems. Finally, a discussion and some emerging research directions are presented.

Keywords: Cluster analysis, automatic clustering, nature-inspired metaheuristics, single-objective and multiobjective metaheuristics

1. Introduction

Cluster analysis is an unsupervised learning technique aimed at discovering the natural grouping of objects according to the similarity of measured intrinsic characteristics [1]. The two fundamental problems in automatic clustering are determining the optimal number of clusters and identifying all data groups correctly. In this sense, the number of combinations in assigning N objects into K clusters is:[1]

$$S\left(N,K\right)=\frac{1}{K!}\sum_{i=0}^{K}\left(-1\right)^{K-i}\binom{K}{i}i^{N}. \tag{1}$$

On the other hand, the search space size in finding the optimal number of clusters is:[2]

$$B\left(N\right)=\sum_{K=1}^{N}S(N,K). \tag{2}$$

Besides, the clustering (or grouping) problem of finding an optimal solution is NP-hard when $K > 3$ [2]; hence, even for moderate-sized problems, the clustering task could be computationally prohibitive [3].

To limit the search space size, many clustering methods described in the literature assume a fixed number of clusters, which is unknown *a priori* in many clustering practices. To overcome this inconvenience, automatic clustering approaches aimed at finding the adequate number of clusters within the range $[K_{\min}, K_{\max}]$ have been developed.

The principal clustering techniques developed in the last 50 years were reviewed by Jain [4], who presented the evolution and trends in data clustering. Also, Xu and Wunsch [5] focused on algorithms for grouping data sets that are used in statistics, computer science, and machine learning. In the last decade, developments in automatic clustering have been strengthened [6, 7, 8]. In particular, nature-inspired metaheuristics have been applied to obtain satisfactory suboptimal solutions to the automatic clustering problem in an acceptable timescale [9]. These kinds of metaheuristics model the behavior of natural phenomena, which exhibit an ability to learn or adapt to new situations to solve problems in complex and changing environments [10].

Some review articles on clustering analysis that use nature-inspired metaheuristics have been published by Handl and Meyer [11], Sheikh et al. [12], Hruschka et al. [9], Rana et al. [13], Bong and Rajeswari [6], Nanda and Panda [14], and Alam et al. [15]. A review of ant-based and swarm-based clustering techniques was presented by Handl and Meyer [11]. A survey on genetic algorithms applied to clustering was summarized by Sheikh et al. [12]. Hr-

[*] Corresponding Author: Wilfrido Gómez-Flores, Technology Information Laboratory, CINVESTAV-IPN, Science and Technology Park TECNOTAM, Km. 5.5 Ciudad Victoria-Soto La Marina Road; Email, wgomez@tamps.cinvestav.mx; Phone, +52 (834) 107-0220.

[1] $S\left(N,K\right)$ is known as the Stirling number of second kind.

[2] $B(N)$ is known as the Bell numbers.

uschka et al. [9] presented a brief summary of evolutionary algorithms and reviewed the initialization procedure, encoding scheme, crossover, mutation, and fitness evaluation for single and multiobjective cases. Bong and Rajeswari [6] investigated multiobjective nature-inspired clustering techniques applied to image segmentation. Recently, Nanda and Panda [14] surveyed some nature-inspired metaheuristics focused on the partitional clustering paradigm. Reviews on particle swarm optimization algorithms and their applications to data clustering were presented by Rana et al. [13] and Alam et al. [15].

Despite the relevance of these review articles, to the best of our knowledge, no review paper about nature-inspired metaheuristics for automatic clustering has been published. Therefore, we present an in-depth review of nature-inspired metaheuristics for automatic clustering that have been reported in the last two decades. This paper contributes in the following two main aspects: (*i*) it presents an up-to-date overview on single-solution, single-objective, and multiobjective metaheuristics applied to automatic clustering, and (*ii*) it provides a review of important aspects, such as encoding schemes, validity indices, data sets, and applications.

The outline of this paper is as follows. Section 2 describes the basic terms and concepts related to automatic clustering analysis. Section 3 presents single-solution metaheuristics that use a single agent or solution, which moves through the search space in a piecewise style. Section 4 reviews single-objective metaheuristics, in which a population of potential solutions cooperate to optimize a unique cost function. Section 5 presents multiobjective metaheuristics, which optimize distinct cost functions and consider a trade-off among them. Finally, future tendencies and conclusions are given in Sections 7 and 8, respectively.

2. Basic Preliminaries

2.1. Definitions

The following terms and notation are used throughout this paper:

- A *pattern* (or object) is a single data item represented by a vector of measurements $\mathbf{x} = \{x_1, x_2, \dots, x_D\}^T$, where $x_i \in \mathbb{R}$ is a *feature* (or attribute), and D denotes the *dimensionality.*

- A *data set* is denoted as $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} \in \mathbb{R}^D$, where N is the total number of patterns in the data set.

- A *cluster* (or group) can be defined as high-density regions separated by low-density regions within the feature space.

- *Clustering*, denoted as $\mathbf{C} = \{\mathbf{c}_k \,|k = 1, \dots, K\}$, refers to the set of mutually disjoint clusters that partitions $\mathbf{X}$ into K groups.

- The *number of objects* in cluster $\mathbf{c}_k$ is denoted by $n_k = |\mathbf{c}_k|$.

- The *centroid of cluster* (or prototype) $\mathbf{c}_k$ is expressed as $\bar{\mathbf{c}}_k = {}^1\!/_{n_k} \sum_{\mathbf{x}_i \in \mathbf{c}_k} \mathbf{x}_i$, whereas the *centroid of data set* $\mathbf{X}$ is $\bar{\mathbf{X}} = {}^1\!/_N \sum_{\mathbf{x}_i \in \mathbf{X}} \mathbf{x}_i$.

- A *distance measure* is a metric (or quasi-metric) used to quantify the proximity between patterns.

- A *cluster validity index* uses a distance measure to quantitatively evaluate the obtained clustering.

2.2. Clustering techniques

The specialized literature on cluster analysis commonly classifies clustering techniques into partitional and hierarchical [1, 4, 14], which are detailed in the following subsections.

2.2.1. Partitional Clustering

Partitional clustering can be performed in two different modes: hard (or crisp) and fuzzy. Hard clustering assumes that the membership between patterns and clusters is binary; thus, each pattern belongs to exactly one cluster. On the other hand, fuzzy clustering assigns different degrees of membership to the patterns for each cluster to build a non-binary relationship between them.

Hard clustering divides a data set directly into a prespecified number of clusters without a hierarchical structure [16], where a data set $\mathbf{X}$ is partitioned into K nonoverlapping groups $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K\}$, such that the following three conditions should be satisfied:

- $\mathbf{c}_i \neq \emptyset$, $i = 1, \dots, K$;
- $\bigcup_{i=1}^{K} \mathbf{c}_i = \mathbf{X}$;
- $\mathbf{c}_i \cap \mathbf{c}_j = \emptyset$, $i, j = 1 \dots, K$ and $i \neq j$.

Perhaps the most fundamental algorithm related to hard clustering is the k-means algorithm, which attempts to minimize the sum-of-squared-error criterion [17, 18]. Fifty years after its formulation, k-means is still popular and widely used because of its simplicity and low computational complexity [4]. However, a predefined number of clusters is required at the beginning of the algorithm, which is unknown in several real-world clustering applications. Hence, the k-means algorithm has been extended to automatically find the number of clusters; some of these extended approaches include the X-means [19] and the G-means algorithm [20].

Fuzzy clustering is an alternative definition given in terms of fuzzy sets, in which each pattern belongs to more than one cluster simultaneously, with a certain degree of membership $u_j \in [0, 1]$. The membership value of the ith pattern in the jth cluster should satisfy the following two conditions:

- $\sum_{j=1}^{K} u_j(\mathbf{x}_i) = 1$, $i = 1, \dots, N$;

- $\sum_{i=1}^{N} u_j(\mathbf{x}_i) < N$, $j = 1, \ldots, K$.

The most well-known fuzzy algorithm is fuzzy c-means [21], which is essentially a fuzzy extension of the k-means method.

2.2.2. Hierarchical Clustering

Hierarchical clustering algorithms produce a hierarchy of clustering called a dendrogram (or tree structure), which represents the nested grouping of the objects in a data set. The procedure builds N successive clustering levels, in which the current clustering is based on the solution obtained at the previous level. Therefore, hierarchical clustering does not require *a priori* knowledge about the number of clusters; however, the obtained groups are static because the objects assigned to a given cluster cannot move to another one.

Agglomerative and divisive approaches are the two main categories of hierarchical clustering, of which single-link and complete-link [4] algorithms are the most well-known.

2.3. Proximity Measures

Clustering algorithms measure the proximity between objects to form groups [1]. The selection of the appropriate proximity measure is important because memberships are defined for every object in data set $\mathbf{X}$. Depending on the kind of proximity measure, different groupings can be created [22]. A proximity measure can be either a distance (dissimilarity) or a similarity between a pair of objects, between an object and a prototype, or between a pair of prototypes. The most common proximity measures used in the automatic clustering techniques described herein are detailed below.

- The Minkowski metric [16], or L_p-norm, is a dissimilarity measure defined as

$$\mathrm{d}_p(\mathbf{x},\mathbf{y}) = \left(\sum_{i=1}^{D} |x_i - y_i|^p\right)^{1/p}, \tag{3}$$

  where $\mathbf{x}$ and $\mathbf{y}$ are D-dimensional data vectors. Note that when $p = 2$, the Minkowski metric becomes the well-known Euclidean distance (or L_2-norm), which is denoted as $\mathrm{d_e}(\mathbf{x},\mathbf{y})$. Two other common special cases of the Minkowski metric are the Manhattan distance (or L_1-norm), when $p = 1$, and the Chebyshev distance (or L_∞-norm), when $p \to \infty$, which is computed as

$$\mathrm{d}_\infty(\mathbf{x},\mathbf{y}) = \max_{1\leq i\leq D} |x_i - y_i|\,. \tag{4}$$

- The similarity between two vectors $\mathbf{x}$ and $\mathbf{y}$ can be measured by the cosine of the angle between them:

$$\cos(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|\,\|\mathbf{y}\|}, \tag{5}$$

  where $\|\cdot\|$ denotes the L_2-norm. There is a relationship between cosine similarity and Euclidean distance, such that the cosine similarity can be transformed into a dissimilarity measure as [23]

$$\mathrm{d}_{\cos}(\mathbf{x},\mathbf{y}) = 1 - \cos(\mathbf{x},\mathbf{y}) = \tfrac{1}{2}\mathrm{d}_\mathrm{e}^2(\mathbf{x},\mathbf{y}). \tag{6}$$

- The point symmetry distance between the pattern $\mathbf{x}_i$ and the cluster $\mathbf{c}_k$ is defined as [24]

$$\mathrm{d_{ps}}\left(\mathbf{x}_i,\mathbf{c}_k\right) = \min_{\mathbf{x}_j\in\mathbf{X}\setminus\mathbf{x}_i}\left\{\frac{\mathrm{d_e}(\mathbf{x}_i-\bar{\mathbf{c}}_k,\mathbf{x}_j-\bar{\mathbf{c}}_k)}{\mathrm{d_e}(\mathbf{x}_i,\bar{\mathbf{c}}_k)+\mathrm{d_e}(\mathbf{x}_j,\bar{\mathbf{c}}_k)}\right\}. \tag{7}$$

- The modified point symmetry distance between the object $\mathbf{x}_i$ and the cluster $\mathbf{c}_k$ is expressed as [25]

$$\mathrm{d}^*_{\mathrm{ps}}\left(\mathbf{x}_i,\mathbf{c}_k\right) = \frac{1}{2}\sum\min_{\mathbf{x}_j\in\mathbf{c}_k}(2)\left\{\mathrm{d_e}\left(2\bar{\mathbf{c}}_k-\mathbf{x}_i,\mathbf{x}_j\right)\right\}. \tag{8}$$

  The point $2\bar{\mathbf{c}}_k - \mathbf{x}_i$ is called the symmetrical (reflected) point of $\mathbf{x}_i$ with respect to the prototype $\bar{\mathbf{c}}_k$. The function $\sum\min(n)$ computes the sum of the n lowest arguments.

2.4. Validity Indices

A cluster validity index (CVI) evaluates the goodness of a particular clustering structure determined by a clustering algorithm by using only information inherent in the data [1]. A CVI should satisfy the following requirements: (*i*) it should have intuitive meaning, (*ii*) it should be easy to compute, and (*iii*) it should be mathematically justifiable. Also, the CVI should be capable of imposing an ordering of the clusters in terms of its goodness [26]. Regarding the kind of clustering algorithm, the CVI could be either hard or fuzzy.

Commonly, a CVI defines a relationship between cluster cohesion (within-group scatter) and cluster separation (between-group scatter) to estimate the quality of a clustering solution [27]. This characteristic has been exploited by clustering algorithms based on metaheuristics, in which CVIs are used as objective functions to be optimized.

The most common CVIs used by the automatic clustering algorithms reviewed in this study are depicted below. The following notations are used: an abbreviation of the CVI name followed by two subscript symbols, a letter to indicate if the CVI is hard ($\mathcal{H}$) or fuzzy ($\mathcal{F}$) and an arrow to denote if the CVI is maximized (↑) or minimized (↓) by the metaheuristic.

- Within-group scatter ($\mathrm{WGS}_{\mathcal{H}\downarrow}$) [18]. The within-cluster dispersion by the sum of the squared distances between points in a cluster to the centroid is measured as

$$\mathrm{WGS}(\mathbf{C}) = \sum_{\mathbf{c}_k\in\mathbf{C}}\sum_{\mathbf{x}_i\in\mathbf{c}_k}\mathrm{d}_\mathrm{e}^2\left(\mathbf{x}_i,\bar{\mathbf{c}}_k\right). \tag{9}$$

- Between-group scatter ($\mathrm{BGS}_{\mathcal{H}\uparrow}$) [1]. The separation between a pair of cluster prototypes is defined as

$$\mathrm{BGS}(\mathbf{C}) = \sum_{\mathbf{c}_k,\mathbf{c}_r\in\mathbf{C}} \mathrm{d_e}\left(\bar{\mathbf{c}}_k,\bar{\mathbf{c}}_r\right). \tag{10}$$

- Connectivity index ($\mathrm{Conn}_{\mathcal{H}\downarrow}$) [7]. This reflects the cluster connectedness by evaluating the degree to which neighboring objects have been placed in the same cluster. It is computed as

$$\mathrm{Conn}(\mathbf{C}) = \sum_{i=1}^{N}\left(\sum_{j=1}^{L} x_{i,n_{ij}}\right), \tag{11}$$

where

$$x_{r,s} = \begin{cases} \frac{1}{j}, & \text{if } \neg\exists\mathbf{c}_k : r\in\mathbf{c}_k \wedge s\in\mathbf{c}_k; \\ 0, & \text{otherwise.}\end{cases}$$

The variable n_{ij} is the jth nearest neighbor of the ith object. The parameter L determines the number of neighbors that contribute to the connectivity measure.

- Dunn index ($\mathrm{DI}_{\mathcal{H}\uparrow}$) [28]. This is a ratio-type index in which the cohesion is estimated by the nearest neighbor distance and the separation by the maximum cluster diameter. It is defined as

$$\mathrm{DI}(\mathbf{C}) = \frac{\min_{\mathbf{c}_k\in\mathbf{C}}\left\{\min_{\mathbf{c}_r\in\mathbf{C}\setminus\mathbf{c}_k}\left\{\delta\left(\mathbf{c}_k,\mathbf{c}_r\right)\right\}\right\}}{\max_{\mathbf{c}_k\in\mathbf{C}}\left\{\Delta\left(\mathbf{c}_k\right)\right\}}, \tag{12}$$

where

$$\begin{aligned}\delta\left(\mathbf{c}_k,\mathbf{c}_r\right) &= \min_{\mathbf{x}_i\in\mathbf{c}_k}\min_{\mathbf{x}_j\in\mathbf{c}_r}\left\{\mathrm{d_e}\left(\mathbf{x}_i,\mathbf{x}_j\right)\right\}, \\ \Delta\left(\mathbf{c}_k\right) &= \max_{\mathbf{x}_i,\mathbf{x}_j\in\mathbf{c}_k}\left\{\mathrm{d_e}\left(\mathbf{x}_i,\mathbf{x}_j\right)\right\}.\end{aligned}$$

Besides, there are distinct variants of DI, called generalized Dunn indices [29], which involve different measures of the criteria cohesion δ and separation Δ.

- Calinski-Harabasz index ($\mathrm{CH}_{\mathcal{H}\uparrow}$) [30]. This is a ratio-type index in which the cohesion is estimated by the sum of the distances of the patterns to their respective centroid and the separation is measured by the sum of the distances from each centroid to the global prototype. It is defined as

$$\mathrm{CH}(\mathbf{C}) = \frac{N-K}{K-1}\times\frac{\sum_{\mathbf{c}_k\in\mathbf{C}} n_k \mathrm{d_e}\left(\bar{\mathbf{c}}_k,\bar{\mathbf{X}}\right)}{\sum_{\mathbf{c}_k\in\mathbf{C}}\sum_{\mathbf{x}_i\in\mathbf{c}_k}\mathrm{d_e}\left(\mathbf{x}_i,\bar{\mathbf{c}}_k\right)}. \tag{13}$$

- Davies-Bouldin index ($\mathrm{DB}_{\mathcal{H}\downarrow}$) [31]. In this index, the cohesion is estimated by the mean distance of the objects to their respective centroid and the separation quantifies the distance between centroids. It is expressed as

$$\mathrm{S}(\mathbf{C}) = \frac{1}{K}\sum_{\mathbf{c}_k\in\mathbf{C}}\max_{\mathbf{c}_r\in\mathbf{C}\setminus\mathbf{c}_k}\left\{\frac{\mathrm{S}(\mathbf{c}_k)+\mathrm{S}(\mathbf{c}_r)}{\mathrm{d_e}\left(\bar{\mathbf{c}}_k,\bar{\mathbf{c}}_r\right)}\right\}, \tag{14}$$

where

$$\mathrm{S}(\mathbf{c}_k) = \frac{1}{n_k}\sum_{\mathbf{x}_i\in\mathbf{c}_k}\mathrm{d_e}\left(\mathbf{x}_i,\bar{\mathbf{c}}_k\right).$$

- FCM index ($\mathrm{FCM}_{\mathcal{F}\downarrow}$) [21]. This is the cost function originally used in fuzzy c-means algorithm, which minimizes the within-group scatter measure defined as

$$\mathrm{FCM}(\mathbf{C}) = \sum_{\mathbf{c}_k\in\mathbf{C}}\sum_{\mathbf{x}_i\in\mathbf{X}}\mathrm{u}_{ki}^{m}\mathrm{d}_{\mathrm{e}}^{2}\left(\mathbf{x}_i,\bar{\mathbf{c}}_k\right), \tag{15}$$

where $\mathrm{U}(\mathbf{X}) = \{\mathrm{u}_{ki}\}$ is the fuzzy membership matrix, and m denotes the *fuzzifier* parameter.

- Silhouette index ($\mathrm{SI}_{\mathcal{H}\uparrow}$) [32]. This is a normalized summation-type index in which the cohesion is measured by the sum of the distances between all the points in the same cluster and the separation is based on the nearest neighbor distance between points in different groups. It is defined as

$$\mathrm{SI}(\mathbf{C}) = \frac{1}{N}\sum_{\mathbf{c}_k\in\mathbf{C}}\sum_{\mathbf{x}_i\in\mathbf{c}_k}\frac{\mathrm{b}\left(\mathbf{x}_i,\mathbf{c}_k\right)-\mathrm{a}\left(\mathbf{x}_i,\mathbf{c}_k\right)}{\max\left\{\mathrm{b}\left(\mathbf{x}_i,\mathbf{c}_k\right),\mathrm{a}\left(\mathbf{x}_i,\mathbf{c}_k\right)\right\}}, \tag{16}$$

where

$$\begin{aligned}\mathrm{a}\left(\mathbf{x}_i,\mathbf{c}_k\right) &= \frac{1}{n_k}\sum_{\mathbf{x}_j\in\mathbf{c}_k}\mathrm{d_e}\left(\mathbf{x}_i,\mathbf{x}_j\right), \\ \mathrm{b}\left(\mathbf{x}_i,\mathbf{c}_k\right) &= \min_{\mathbf{c}_r\in\mathbf{C}\setminus\mathbf{c}_k}\left\{\frac{1}{n_r}\sum_{\mathbf{x}_j\in\mathbf{c}_r}\mathrm{d_e}\left(\mathbf{x}_i,\mathbf{x}_j\right)\right\}.\end{aligned}$$

- Xie-Beni index ($\mathrm{XB}_{\mathcal{F}\downarrow}$) [26]. This is the ratio of the total variation to the minimum separation of the clusters defined as

$$\mathrm{XB}(\mathbf{C}) = \frac{\sum_{\mathbf{c}_k\in\mathbf{C}}\sum_{\mathbf{x}_i\in\mathbf{X}}\mathrm{u}_{ki}^{2}\mathrm{d}_{\mathrm{e}}^{2}\left(\mathbf{x}_i,\bar{\mathbf{c}}_k\right)}{N\min_{\mathbf{c}_k,\mathbf{c}_r\in\mathbf{C}}\left\{\mathrm{d}_{\mathrm{e}}^{2}\left(\bar{\mathbf{c}}_k,\bar{\mathbf{c}}_r\right)\right\}}. \tag{17}$$

Note that the numerator is the FCM index with $m = 2$.

- Turi index ($\mathrm{TI}_{\mathcal{H}\downarrow}$) [33]. In this ratio-type index, the inter-cluster separation is estimated by the minimum distance between centroids and the intra-cluster dispersion is computed by the average distance between each object and its respective centroid:

$$\mathrm{TI}(\mathbf{C}) = (c \times \mathcal{N}(\mu, \sigma) + 1) \times \frac{\mathrm{intra}\,(\mathbf{C})}{\mathrm{inter}\,(\mathbf{C})}, \tag{18}$$

where c is a user-specified parameter, $\mathcal{N}(\mu, \sigma)$ is a normal distribution with mean μ and standard deviation σ, and the terms $\mathrm{intra}\,(\mathbf{C})$ and $\mathrm{inter}\,(\mathbf{C})$ are defined as

$$\mathrm{intra}\,(\mathbf{C}) = \frac{1}{N} \sum_{\mathbf{c}_k \in \mathbf{C}} \sum_{\mathbf{x}_i \in \mathbf{c}_k} \mathrm{d_e}\left(\mathbf{x}_i, \bar{\mathbf{c}}_k\right),$$
$$\mathrm{inter}\,(\mathbf{C}) = \min_{\mathbf{c}_r \in \mathbf{C} \setminus \mathbf{c}_k} \left\{\mathrm{d_e}\left(\bar{\mathbf{c}}_k, \bar{\mathbf{c}}_r\right)\right\}.$$

- PBM index ($\mathrm{PBM}_{\mathcal{F}\uparrow}$) [34]. This index is defined as the product of cohesion and separation measures. The former is calculated by the sum of all pattern distances in a cluster to its respective centroid, whereas the later is estimated by the maximum distance between centroids. The PBM index is expressed as

$$\mathrm{PBM}(\mathbf{C}) = \left[\frac{1}{K} \times \frac{\mathrm{E}\,(\mathbf{C})}{\mathrm{FCM}\,(\mathbf{C})} \times \max_{\mathbf{c}_k, \mathbf{c}_r \in \mathbf{C}} \left\{\mathrm{d_e}\left(\bar{\mathbf{c}}_k, \bar{\mathbf{c}}_r\right)\right\}\right]^2, \tag{19}$$

where the denominator in the second factor involves the FCM index with $m = 1$, and the numerator is computed as

$$\mathrm{E}\,(\mathbf{C}) = \sum_{\mathbf{x}_i \in \mathbf{X}} \mathrm{d_e}\left(\mathbf{x}_i, \bar{\mathbf{X}}\right).$$

- CS index ($\mathrm{CS}_{\mathcal{H}\downarrow}$) [35]. This is a ratio-type index that estimates the cohesion by using the cluster diameters and the separation measurement is based on the nearest neighbor distance between prototypes. It is computed as

$$\mathrm{CS}(\mathbf{C}) = \frac{\sum_{\mathbf{c}_k \in \mathbf{C}} \Delta\left(\mathbf{c}_k\right)}{\sum_{\mathbf{c}_k \in \mathbf{C}} \min_{\mathbf{c}_r \in \mathbf{C} \setminus \mathbf{c}_k} \left\{\mathrm{d_e}\left(\bar{\mathbf{c}}_k, \bar{\mathbf{c}}_r\right)\right\}}, \tag{20}$$

where

$$\Delta\left(\mathbf{c}_k\right) = \frac{1}{n_k} \sum_{\mathbf{x}_i \in \mathbf{c}_k} \max_{\mathbf{x}_j \in \mathbf{C}} \left\{\mathrm{d_e}\left(\mathbf{x}_i, \mathbf{x}_j\right)\right\}.$$

- Sym-index ($\mathrm{Sym}_{\mathcal{H}\uparrow}$) [36]. This ratio-type index is an adaptation of the PBM index [34]; the cohesion is estimated by the sum of the point symmetry distances in the same cluster and the separation is quantified by the maximum Euclidean distance between centroids. It is defined as

$$\mathrm{Sym}(\mathbf{C}) = \frac{\max_{\mathbf{c}_k, \mathbf{c}_r \in \mathbf{C}} \left\{\mathrm{d_e}\left(\bar{\mathbf{c}}_k, \bar{\mathbf{c}}_r\right)\right\}}{K \sum_{\mathbf{c}_k \in \mathbf{C}} \sum_{\mathbf{x}_i \in \mathbf{c}_k} \mathrm{d_{ps}}\left(\mathbf{x}_i, \mathbf{c}_k\right)}. \tag{21}$$

- Weight of Evidence Information ($\mathrm{WIM}_{\mathcal{H}\uparrow}$) [37]. This is an uncertainty measure based on mutual information, which estimates the amount of evidence information between an object and a cluster at a given confidence level. The value of WIM for an object $\mathbf{x}_i$ belonging to cluster $\mathbf{c}_k$ is given by

$$\mathrm{WIM}(\mathbf{C}) = I\left(\mathbf{c}_k : \mathbf{x}_i\right) - I\left(\neq \mathbf{c}_k : \mathbf{x}_i\right), \tag{22}$$

where the mutual information is defined in terms of the joint probability distribution of $\mathbf{x}_i$ given $\mathbf{c}_k$ and the marginal probability of $\mathbf{c}_k$ as

$$I\left(\mathbf{c}_k : \mathbf{x}_i\right) = \log\left(\frac{P(\mathbf{c}_k \mid \mathbf{x}_i)}{P(\mathbf{c}_k)}\right).$$

2.5. Clustering Optimization Problem

The clustering problem can be formulated as an optimization problem that could be solved by single-objective and multiobjective metaheuristics [38].

Let $\Omega = \left\{\mathbf{C}^1, \mathbf{C}^2, \ldots, \mathbf{C}^{B(n)}\right\}$ be the set of all feasible clusterings, in which the elements are clustering solutions of a given data set $\mathbf{X}$, and f is a single criterion function, for instance, one of the validity indices listed in Section 2.4. Then, the goal of single-objective clustering problem (Ω, f) is to determine the clustering $\mathbf{C}^*$ for which

$$f\left(\mathbf{C}^*\right) = \min\left\{f(\mathbf{C}) \mid \mathbf{C} \in \Omega\right\}. \tag{23}$$

Note that $f(\cdot)$ is minimized without loss of generality. On the other hand, the goal of multiobjective clustering problem $(\Omega, f_1, f_2, \ldots, f_m)$ is to determine the clustering $\mathbf{C}^*$ for which

$$f(\mathbf{C}^*) = \min\left\{f_t(\mathbf{C}) \mid \mathbf{C} \in \Omega\right\}, \quad t = 1, 2, \ldots, m \tag{24}$$

where f_t, $t = 1, \ldots, m$, is a set of m (single) criterion functions. Usually, multiobjective problems have multiple optimal solutions, which are commonly identified by using the principle of Pareto dominance. For two clustering solutions $\mathbf{C}_1, \mathbf{C}_2 \in \Omega$, $\mathbf{C}_1$ is said to *dominate* $\mathbf{C}_2$ (denoted as $\mathbf{C}_1 \prec \mathbf{C}_2$), where the following two conditions should be satisfied:

$$f_t(\mathbf{C}_1) \leq f_t(\mathbf{C}_2), \quad \forall t \in 1, 2, \ldots, m \tag{25}$$

and

$$f_t(\mathbf{C}_1) < f_t(\mathbf{C}_2), \quad \exists t \in 1, 2, \ldots, m. \tag{26}$$

The set of all Pareto nondominated solutions is referred to as the Pareto-optimal set and its corresponding set of objective function values is referred to as the Pareto-optimal front.

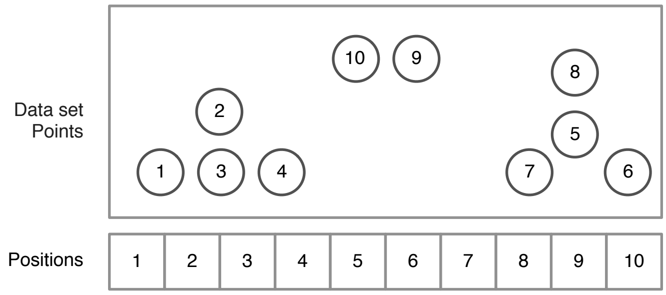


Figure 1: Didactic 10-point data set.

2.6. Encoding Schemes

Any iterative metaheuristic, wheather single-objective or multiobjective, requires a representation or an encoding of a solution, which is directly related to the objective function to be optimized [40]. The encoding schemes play a relevant role in the efficiency and effectiveness of any metaheuristic and constitute an essential step in its design. Besides, the efficiency of an encoding is also related to the search operators applied in this representation, such as mutation, recombination, neighborhood, etc. [41].

Encoding schemes can be categorized into binary, integer, and real, which have all been used in metaheuristics that attempted to solve the automatic clustering problem. Because the number of clusters is unknown *a priori*, the encoding scheme should be designed for varying the number of clusters within a predetermined range $[K_{\min}, K_{\max}]$, where $K_{\min} = 2$ and $K_{\max} = \text{round}\left(\sqrt{N}\right)$ [42]. Figure 1 illustrates the didactic 10-point data set used to exemplify binary and real encodings when varying the number of clusters in the representation.

2.6.1. Binary Encoding

Each clustering solution is represented as a binary string of length equal to N; that is, each position in the binary string corresponds to a particular object in the data set. The value of the ith position is “1” if the ith object is considered as a cluster prototype, and “0” otherwise. Figure 2 illustrates the clustering solutions of the data set in Figure 1 when $K = 3$ and $K = 5$.

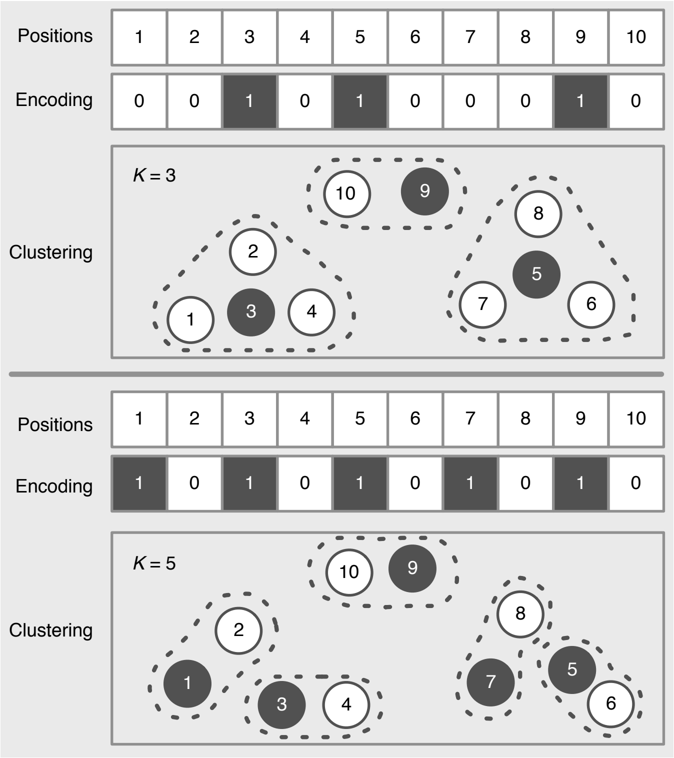


Figure 2: Binary encoding with $K = 3$ and $K = 5$.

2.6.2. Integer Encoding

Similarly to binary encoding, the integer representation codifies all objects in the data set into an integer array. The codification of clustering solutions can be done by two kind of representation: label-based and graph-based encodings, which are detailed below.

- Label-based encoding. Every object in the data set takes an integer value (or cluster label) in the alphabet $\{1, \dots, K_{\max}\}$, where $K_{\max}$ is the maximum number of possible clusters. This integer encoding is naturally redundant, because $K_{\max}!$ different labelings represent the same clustering solution. For instance, for a 5-point data set with $K = 2$, the clustering solution $\mathbf{C} = \{11122\}$ is the same as $\mathbf{C} = \{22211\}$. Therefore, the size of the search space explored by the clustering algorithm is much larger than the original space of solutions [9]. An alternative solution to this problem is to use a renumbering procedure [2]. Figure 3a shows the labeled data set in Figure 1 when $K = 3$ and $K = 5$.
- Graph-based encoding. Each object in the data set can take a single value from the set $\{1, \dots, N\}$. A value j assigned to the ith position means that there is a link between these objects, which are placed in the same cluster. Then, the clustering solution is recovered by identifying all connected components of the directed graph [7]. Figure 3b illustrates the clustering solution of the data set in Figure 1 when $K = 3$ and $K = 5$.

2.6.3. Real Encoding

In real encoding approaches, a clustering solution represents the location of the cluster prototype in the D-dimensional feature space. Let us consider a set of N objects to be clustered into K groups, in which the prototypes of each cluster are codified into a real vector of size $D \times K$. The population-based metaheuristics used in automatic clustering commonly used fixed- or variable-length encoding.

- Fixed-length encoding. All clustering solutions encode a predefined maximum number of prototypes,

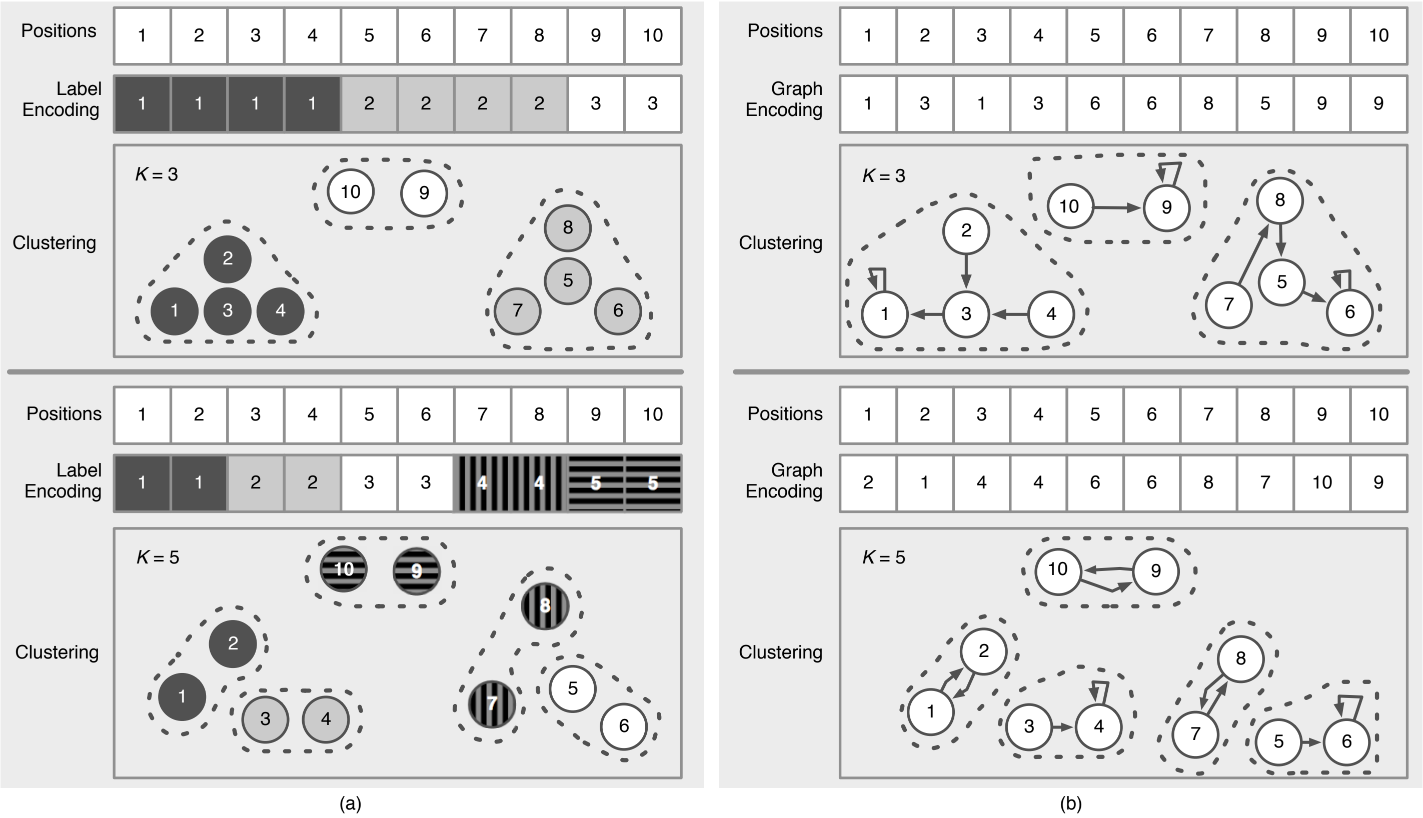


Figure 3: Integer encoding: (a) label-based representation and (b) graph-based representation.

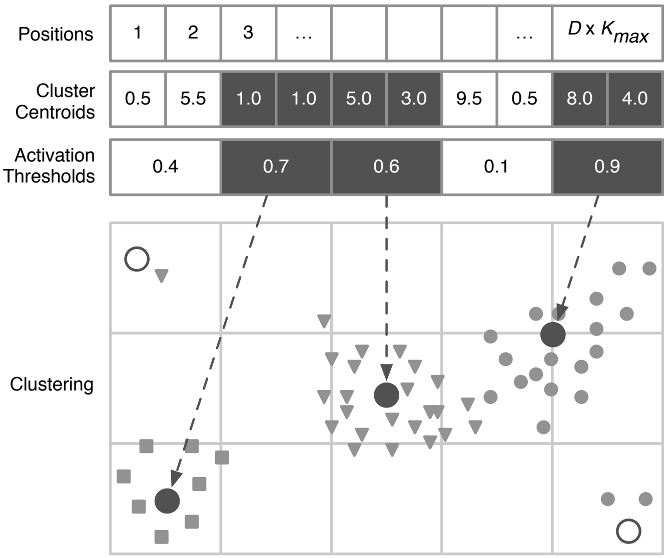


Figure 4: Real encoding: centroid based representation with $K = 3$ and $D = 2$.

$K_{\max}$. Hence, all members of the population maintain the same length throughout the optimization process; that is, $D \times K_{\max}$. Additionally, to determine which prototypes participate in the clustering process at each iteration of the algorithm, a mask vector or activation thresholds are commonly used [43, 44]. Figure 4 shows a fixed-length encoding when $K_{\max} = 5$ but considering the activation of $K = 3$ prototypes.

- Variable-length encoding. Consider a set of P clustering solutions $\mathcal{C} = \{\mathbf{C}_i | i = 1, \ldots, P\}$. Then, the ith clustering solution encodes a particular number of prototypes K_i. Hence, the length of vector solutions is variable; that is, $D \times K_i$. The population-based metaheuristics that use this variable scheme should adjust the search operators to cope with members of different sizes [45, 46].

2.7. Test Data Sets

Nature-inspired automatic clustering algorithms have been successfully applied to distinct areas of engineering and science. Researches commonly use standard data sets, which could be either synthetic or real-life measurements, for validating and comparing the performances of different automatic clustering algorithms.

Synthetic data sets are widely preferred because it is feasible to control distinct features, such as number of clusters, density, size, shape, dimensionality, overlapping, and noise.

Moreover, an important feature is the linear separability between clusters, which mostly determines the complexity of the data set. Three cases are clearly identified: (*i*) linear separability without overlapping, (*ii*) linear separability with overlapping, and (*iii*) non-linear separability without overlapping. The two first cases consider globular or elliptical clusters, such as $\widetilde{S}_1$, $\widetilde{S}_3$, and $\widetilde{S}_7$ data sets in Table 1, whereas the third case involves arbitrary cluster shapes, such as $\widetilde{S}_9$, $\widetilde{S}_{10}$, and $\widetilde{S}_{11}$ data sets in Table 1.

On the other hand, real-life data sets consider features obtained from sensed or measured real-world data, which could be associated, in some cases, to class labels. Well-known data sets, such as the Iris, Breast Cancer, Wine, or Glass, have become standard data sets in the literature. Besides, many authors define particular data sets related to the particular problem that they are attempting to solve with a clustering algorithm designed to cope with specific requirements.

Furthermore, principal real-life applications areas of automatic clustering based on nature-inspired metaheuristics include image segmentation, such as satellite imagery [51, 8, 52, 53, 54, 55], medical image [57, 49], and textured images [47, 43, 96, 97], and gene expression microarray analysis [37, 56, 57, 58, 25].

It is worth mentioning that synthetic data sets are useful to evaluate distinct characteristics and limitations of clustering algorithms. Besides, real-world problems aid to demonstrate the generalization capabilities of the clustering algorithms to distinct research fields. Hence, a complete evaluation of an automatic clustering algorithm should involve synthetic and standard real-life data sets as well as an application field.

Tables 1 and 2 summarize the characteristics of common synthetic and real-life data sets, respectively, used by the automatic clustering algorithms reviewed herein. In such tables, “ID” identifies the data set, “Data set” refers to the original name of the data set, “N” denotes the number of patterns, “D” is the feature space dimensionality, and “K” is the number of groups.

Table 1: Popular synthetic data sets.

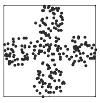
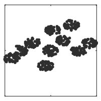
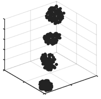
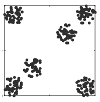
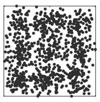
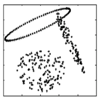
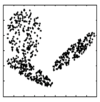
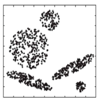
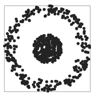
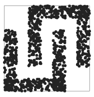
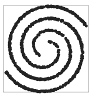
ID	Dataset	N	D	K	Cluster Structures
$\widetilde{S}_1$	Data_5_2	250	2	5	
$\widetilde{S}_2$	Data_10_2	500	2	10	
$\widetilde{S}_3$	Data_4_2	400	3	4	
$\widetilde{S}_4$	Data_6_2	300	2	6	
$\widetilde{S}_5$	Data_9_2	900	2	9	
$\widetilde{S}_6$	Sym_3_2	350	2	3	
$\widetilde{S}_7$	Sym_3_2	600	2	3	
$\widetilde{S}_8$	Sym_5_2	350	2	5	
$\widetilde{S}_9$	Ring_2_2	600	2	2	
$\widetilde{S}_{10}$	Twous_2_2	1000	2	2	
$\widetilde{S}_{11}$	Spirals_2_2	1000	2	2	

Table 2: Standard real-life data sets.

ID	Dataset	N	D	K
$\widetilde{R}_1$	Iris	150	4	3
$\widetilde{R}_2$	Breast Cancer	683	9	2
$\widetilde{R}_3$	Wine	178	13	3
$\widetilde{R}_4$	Glass	214	9	6
$\widetilde{R}_5$	Vowel	990	12	11
$\widetilde{R}_6$	Yeast	1484	8	10
$\widetilde{R}_7$	Heart	297	13	2
$\widetilde{R}_8$	Ruspini	75	2	4

3. Single-Solution Metaheuristics

This section presents a review of single-solution-based metaheuristics for the automatic clustering problem. These kinds of metaheuristics improve a single point solution, evaluated by a single criterion function, and could be viewed as search trajectories through the search space. Such trajectories are updated by iterative procedures that move from the current solution to another one in the search space. Hence, they are mainly oriented toward intensifying the search in local regions (exploitation) [41]. In the next subsections, simulated annealing (SA) and tabu search (TS) approaches applied to automatic clustering are reviewed.

3.1. Simulated Annealing

Simulated annealing (**SA**) is a probabilistic method proposed by Kirkpatrick et al. [59] to find the global minimum of a cost function. SA emulates the physical process whereby a melted solid material (initial state) is gradually cooled until the minimum energy state is reached, that is, when the material structure is “frozen.” An overview of the theoretical and practical aspects of SA is provided in [60].

Bandyopadhyay et al. [61] proposed an algorithm called **SAKM-clustering**, in which objects are redistributed among K clusters probabilistically. In this scheme, points farther away from the cluster center have higher probabilities of migrating to other clusters. SAKM-clustering minimizes the WGS index, which is also used by the k-means algorithm [17]. The result of experiments, which considered three artificial and two real-life data sets, indicated that SAKM-clustering surpassed the performance of k-means.

Later, Bandyopadhyay [44] proposed an unsupervised fuzzy clustering algorithm called **SA-RJMCMC**. This approach automatically discovers the number of clusters by using the homogeneous *reversible jump Markov chain Monte Carlo* (RJMCMC) kernel; thus, this algorithm is able to jump between different dimensions. SA-RJMCMC does the clustering by optimizing the XB index. Its effectiveness, in comparison with the FCM algorithm, was shown in experiments that considered three artificial and two real-life data sets, in which the SA-RJMCMC technique determined the appropriate number of clusters automatically and provided better values of the XB index in almost all the evaluated cases.

3.2. Tabu Search

Tabu search (**TS**), a metaheuristic search procedure proposed by Glover [62], uses flexible-structure memories conditions to strategically constrain and free the search process (embodied in the tabu restrictions and aspiration criteria) and applies memory functions of varying time spans to intensify and diversify the search [63, 64]. TS has obtained optimal and near optimal solutions to a wide variety of classical and practical problems. Although it has not been used directly to solve the automatic clustering problem, TS has been used to guide other methods in escaping local optima by using a tabu memory, such as in the method proposed by Pan and Cheng [65], which is detailed in Section 4.1.3.

3.3. Summary

When the number of groups is assumed *a priori*, single-solution metaheuristics are suitable for finding an adequate partitioning of the data. For instance, applications of SA to clustering with a fixed number of groups can be found in [66, 67, 68]. However, if one considers that the search space increases in automatic clustering, not only the correct partitioning should be found but also the appropriate number of clusters should be determined. Then, it is logical to infer that a single solution would require additional mechanisms to explore the whole search space, which involves the risk of entrapment in local optima. Therefore, because single-solution metaheuristics are exploitation-oriented, they are uncommonly applied to automatic clustering approaches.

4. Population-Based Metaheuristics

Population-based metaheuristics (P-metaheuristics) are iterative processes that attempt to improve a population of solutions. First, the population is randomly initialized. Then, a new population of potential solutions is generated, which could be integrated into the current one by using some selection criteria. The search process is stopped when a given condition is satisfied.

P-metaheuristics are exploration-oriented; that is, they allow a better diversification of the search than single-solution metaheuristics [41]. In this sense, P-metaheuristics based on evolutionary computation (EC), swarm intelligence (SI), and artificial immune systems (AIS) paradigms have become popular to solve the automatic clustering problem.

4.1. Evolutionary Computation

Single-objective EC algorithms, such as evolutionary programming (EP), evolution strategy (ES), genetic algorithm (GA), and differential evolution (DE), have been used for automatic clustering purposes. These approaches are described in detail below.

4.1.1. Evolutionary Programming

Evolutionary programming (**EP**), conceived by Fogel [69], is a stochastic optimization strategy with emphasis on the behavioral linkage between parents and their offspring. The earliest attempt at automatic clustering based on EP was made by Sarkar et al. [70] in 1996. This algorithm (herein called **EP-clustering**) is based on the minimization of both the WGS and DB indices to explore and exploit the search space, respectively. As the authors pointed out, EP-clustering is applicable when the clusters are crisp and spherical. This method outperformed the k-means algorithm when two artificial and two real-life (English vowel sounds) data sets were considered. Chen et al. [71] presented a clustering algorithm called **GEP-cluster**, which is based on gene expression programming. The main contributions of these authors include: (*i*) a concept called clustering algebra, which makes clustering an algebraic operation; (*ii*) an algorithm for discovering the adequate clustering solution without any prior knowledge; and (*iii*) an automatic method for merging partial solutions. GEP-cluster was tested on two different synthetic data sets; the experimental results shown that it is sensitive to noise and inefficient with high-dimensional data.

4.1.2. Evolution Strategy

Evolution strategy (**ES**) is an optimization technique based on ideas of adaptation and evolution. In the $(\mu+\lambda)$-ES selection strategy, λ generated offspring compete with μ parents; for instance, when (1+1)-ES, the best offspring becomes the parent in the next generation. The basic algorithms and design principles for variation and selection operators, as well as theoretical issues, are presented in a comprehensive review in [72].

Lee and Antonsson [73] used ES for automatic clustering (herein called **ES-clustering**), which implements variable-length genomes and the (10+60)-ES selection strategy. A modification of the mean square error (MSE) was chosen as a fitness function. This algorithm was tested on two 2D artificial data sets, both containing 20 clusters. The results suggested an adequate performance of ES-clustering when the clusters are nonoverlapping and present globular shapes. Although the performance of ES-clustering was tested only with 2D data, the authors concluded that the method is easily generalizable to any dimension.

Recently, Kashan et al. [74] proposed a fuzzy clustering algorithm based on a grouping evolutionary strategy (**FuzzyGES**) with (1+1)-ES selection scheme. The XB index was used as a fitness function to evaluate individuals of variable length. The performance of FuzzyGES was compared with that of FCM and three other clustering algorithms based on particle swarm optimization (PSO), differential evolution (DE), and league championship algorithm (LCA). Computational experiments were carried out by considering 13 2D artificial data sets and the well-known Iris data set. The results indicated that FuzzyGES outperformed all the other algorithms in terms of best, average, and standard deviation values of the XB index.

4.1.3. Genetic Algorithm

Genetic algorithm (**GA**), developed by Holland in the early 1970s [75], emulates the principle of evolution by natural selection stated by Charles Darwin. GA has been applied for clustering purposes and a useful survey about GA-based clustering techniques was done by Sheikh et al. [12].

Because GA is the most common algorithm used to address the automatic clustering problem, this section is divided into four parts according to the encoding scheme used by the depicted clustering approaches.

A. Centroid-based Encoding of Variable Length

In 2001, Tseng and Yang [76] proposed the earliest attempt at automatic clustering based on GA, namely, **CLUSTERING**. First, a nearest neighbor clustering method groups those data points that are close together; then, a set of small clusters is obtained. Next, a genetic clustering algorithm merges the set of small clusters into larger ones. In this step, a weighted difference between the WGS and BGS indices defines the fitness function. Finally, a heuristic strategy is used to find the adequate partition. CLUSTERING was compared with three conventional methods: k-means, complete-link, and single-link. Computational experiments that considered one real-life and two artificial data sets shown that CLUSTERING outperformed its counterparts.

Bandyopadhyay and Mulik [77] presented the nonparametric **VGA-clustering** algorithm, which uses variable-length encoding. A comparative study of three different validity indices was done, including DB, DI, and PBM, where the latter (the one proposed by the authors) finds the correct number of clusters for all the tested data sets (five artificial and one real-life). The same authors proposed a fuzzy variant of the VGA-clustering algorithm, called **Fuzzy-VGA**, which uses the XB index as a fitness function [52]. This algorithm was applied in image segmentation, in which different homogeneous regions constituted the pixel clusters in the intensity space. Similarly to VGA-clustering, Fuzzy-VGA codifies cluster centroids into chromosomes of variable length. The effectiveness of Fuzzy-VGA was shown quantitatively and qualitatively in the segmentation of two IRS (remote sensing imagery) data sets, wherein it surpassed the performance of the FCM algorithm.

Sheng et al. [78] proposed two relevant items: (*i*) an objective function called the weighted sum validity function (WSVF), which is the weighted sum of six normalized validity indices (DB, SI, CH, PBM, and two DI variants), and (*ii*) a hybrid *niching* GA (**HNGA**) for automatic clustering. A *niching* method was developed to preserve both the population diversity with respect to the number of clusters encoded in the individuals and the subpopulation diversity with the same number of clusters during the search. The efficiency of HNGA was compared with a clustering GA [79] and VGA-clustering [77]. The experiments were done with three artificial and three real-life data sets, and the results shown that HNGA consistently and efficiently converged to both the correct number of clusters and the data partitioning. However, HNGA clustering becomes computational expensive for large data sets.

Bandyopadhyay and Saha [36] proposed a symmetry-based genetic clustering algorithm (**VGAPS**) that uses the point symmetry (PS) distance rather than the conventional Euclidean distance, and the Sym-index as a fitness function. The experiments considered five artificial and nine real-life data sets. The results revealed that VGAPS outperformed the GCUK [8] and HNGA [78] algorithms when data sets presented symmetric properties. The same authors adapted the original VGAPS to **Fuzzy-VGAPS** [57] that uses a symmetry-based fuzzy validity index called FSym-index. The effectiveness of Fuzzy-VGAPS was compared with seven different clustering techniques by considering four artificial and four real-life data sets. For all the tested data sets, Fuzzy-VGAPS surpassed its counterparts in terms of finding the correct number of clusters and the adequate clustering. However, both VGAPS and Fuzzy-VGAPS were applied only to data sets containing point symmetric clusters. Besides, because the PS distance determines the nearest neighbors, the method requires a considerable computation time for high-dimensional data sets.

Xiao et al. [80] proposed a quantum-inspired GA for k-means clustering (**KMQGA**). This algorithm is based on quantum computing concepts, in which each individual is represented by a Q-bit string. The state of a Q-bit may be "0" or "1" or any superposition of the two. A collapse operation is applied on the Q-bit string to define a certain state, that is, a binary string, which is subsequently transformed into a real-coded string consisting of several centroids. Finally, one iteration of *k*-means is done to create groups that are evaluated by means of the DB index. In the first experiment, three 2D synthetic data sets containing well-separated globular clusters were used to illustrate the functioning of KMQGA. Besides, four real-life data sets were grouped by KMQGA and VGA-clustering [77], and the proposed technique surpassed the latter in terms of finding the correct number of clusters.

A fuzzy variant of an evolutionary algorithm for relational clustering, called **F-EARFC**, was proposed by Horta et al. [81]. In this algorithm, the patterns are described by a proximity matrix (relational data), where the number of clusters were automatically estimated from such matrix. An individual of variable length encodes cluster prototypes. F-EARFC is equipped with two mutation operators capable of dealing with relational data sets and fuzzy partitions. The authors stated that any fuzzy clustering validity criterion can be used as a fitness function; in this case, the fuzzy SI index was applied. The computational performance of F-EARFC was experimentally assessed in comparison with two pseudo-exhaustive clustering approaches, namely, ordered multiple runs (OMR) and multiple runs (MR). The experiments were run on 18 artificial and two real-life data sets. F-EARFC outperformed both OMR and MR methods in terms of accuracy and computing times.

Liu et al. [82] proposed an automatic genetic clustering algorithm for unknown number of clusters called **AGCUK**. This method adopts a cluster center-based representation, in which an individual is a real-code chromosome of variable length. The fitness of an individual is computed by using the DB index. AGCUK was compared with four automatic clustering algorithms: VGA-clustering [77], GCUK [8], HGA [83], and GA-unsupervised [84]. The experiments considered 50 two-dimensional artificial data sets and a real-life data set known as the Breast Cancer data set. The results revealed that AGCUK outperformed its counterparts in terms of the misclassification rate. Regarding time consumption, AGCUK and GCUK shown similar performances and both were faster than VGA, HGA, and GA-unsupervised techniques.

He and Tan [46] proposed a two-stage genetic clustering algorithm called **TGCA**. It applies two-stage selection and mutation operations to exploit the search capability of the algorithm by changing the probabilities of selection and mutation according to the consistence of the number of clusters in the population. TGCA uses the CH index as a fitness function. Its efficiency was compared with automatic clustering algorithms, including hierarchical agglomerative *k*-means (HKM), automatic spectral algorithm (ANJW), and a standard genetic *k*-means clustering algorithm (SGKC). Experiments were done on four artificial and seven real-life data sets and the results shown that TGCA had the best performance in terms of finding the correct number of clusters and clustering accuracy.

B. Centroid-based Encoding of Fixed Length

Bandyopadhyay and Mulik proposed a clustering algorithm called **GCUK** [8], which uses a fixed-length representation, wherein "do not care" symbols were applied to discard prototypes when a solution (or chromosome) is evaluated by the DB index. The effectiveness of GCUK was shown with the use of four synthetic and two real-life data sets.

Liu et al. [85] developed the **VGA-FCM** technique, which is a hybrid algorithm that combines GA and fuzzy *c*-means by using chromosomes of variable length. This approach was applied to identifying roads in satellite imagery. VGA-FCM encodes the cluster centers in the RGB color space, with the FCM index as the fitness function. The main feature of VGA-FCM is the self-adaptation of crossover and mutation probabilities for every chromosome in the population.

A variant of GA, the hierarchical genetic algorithm (**HGA**), was used by Lai [83] for automatic clustering. In HGA the chromosome is divided into two (or more) structural levels consisting of control and parametric genes, emulating the formulation of the biological DNA structure. Thus, an individual is a real-code chromosome of fixed length that uses control genes within a masker, which is a binary string that activates or deactivates some prototypes. The experimental results from five artificial and two real-life data sets shown the effectiveness of HGA. Later, HGA was applied to automatic medical image segmentation, focusing mainly on computed tomography (CT) and magnetic resonance imaging (MRI) [49]. For every chromosome, the control genes represented the number of regions (i.e., number of clusters), and the parametric genes codified the gray levels (i.e., prototypes). The fitness function was the WGS index. The experiments involved five CT and three MRI images, and the results indicated that HGA was capable of producing more continuous and smoother segmentation outcomes in comparison with other methods, inclusing *k*-means, fuzzy *c*-means, the Otsu method, and the competitive Hopfield neural network.

Chang et al. [53] proposed an approach to automatic clustering based on dynamic *niching* GA with *niche* migration (**DNNM**). In this algorithm, dynamic identification of niches with *niche* migration is done at each generation to automatically evolve the number of clusters and the cluster centroids. The fitness function evaluates the total similarity measure by summing the responses of K radial functions, which is related to the density shape of the data points in the neighborhood (herein called density shape function, DSF). The performance of DNNM was compared with the GA with dynamic niche (DNS), species-conserving GA (SCGA), and the dynamic fitness sharing (DFS) algorithm. The experiments were carry out on five artificial and two real-life data sets, and the experimental results shown that DNNM outperformed the DNS, SCGA, and DFS algorithms in terms of identifying the correct number of clusters.

Pan and Cheng [65] proposed a framework for automatic clustering called **ETSAs**, which was inspired by the concept of tabu search (TS) and the advantages of the genetic operators in evolutionary algorithms. ETSAs evolves the number of clusters by using an adaptive evolutionary mechanism, in which the population competition is generated in parallel by two types of reproduction operations. A fitness function defined by the product of the PBM index and a tabu factor was used to evaluate each individual. For comparison purposes, five exhaustive traditional clustering algorithms were evaluated by considering five artificial and four real-life data sets. The results shown that the exhaustive traditional clustering algorithms did not always find the correct number of clusters. In contrast, ETSAs automatically found the adequate number of clusters in all the tested instances.

C. Label-based Encoding

Hruschka and Ebecken [79] proposed a GA for cluster analysis (herein called **GACA**) that implements label-based representation. The SI index was used as a fitness function for evaluating individuals of fixed length. The efficacy of GACA was shown by the simulation results obtained with four real-life data sets, in which the method found the adequate clustering, making it comparable with the best algorithms described in the literature. Lately, GACA has been further extended, resulting in an improved version called **CGA-I** [86]. This new approach uses label-based encoding and the SI index as a fitness function. To improve the efficiency of CGA-I, three additional variants were developed: CGA-II, CGA-III, and **EAC** [56]. CGA-II introduces the *k*-means algorithm as a local search method within the CGA-I framework. CGA-III changes the fitness function of CGA-II with a simplified SI index. EAC removes the crossover operator in CGA-III and incorporates two mutation operators that are more sophisticated. All of these clustering techniques were tested by using six gene expression data sets. The results revealed an adequate performance for all the algorithms, although statistical comparisons, in terms of time consumption, indicated that EAC is faster than the first three variants of CGA. Besides, the statistical evidence also shown that EAC outperformed a traditional method based on multiple runs of *k*-means.

Ma et al. [37] proposed the **EvoCluster** algorithm, which is based on GA. To facilitate the exchange of information between chromosomes, EvoCluster encodes an entire cluster grouping in a fixed-length chromosome such that each gene represents a single cluster. The WIM index is used as the fitness function, which evaluates the dependency between clusters by using mutual information. For performance comparison, the self-organizing map (SOM) and *k*-means were considered, and one real-life and two simulated data sets (focusing on gene expression microarray) were used for experimentation. The results shown that EvoCluster can be effective and robust even in the presence of noise and missing values. However, EvoCluster is about 12 to 18 times slower than its counterparts.

The bacterial evolutionary algorithm (BEA) is part of the family of EC techniques and it is inspired by the biological phenomenon of microbial evolution. This approach was used by Das et al. [87] to develop automatic clustering with BEA (**ACBEA**). It uses chromosomes of variable length, modified by means of both bacterial mutation and gene transfer operators. The fitness function is the CS index. ACBEA outperformed the GCUK [8] and DCPSO [88] algorithms when applied to artificial and real-life data sets.

D. Binary-based Encoding

A genetic clustering algorithm (**GCA**), composed of the cluster decomposition algorithm (CDA) and the hierarchical cluster merging algorithm (HCMA) was presented by Garai and Chaudhuri [89]. HCMA is an iterative GA-based algorithm that contains a component called adjacent cluster checking algorithm (ACCA). The fitness function is similar to that in the CLUSTERING method [76], which relates the WGS and BGS indices. The performance of GCA in comparison with the CURE, DBScan, and Chameleon clustering methods was shown with the use of one real-life and nine artificial data sets.

Lin et al. [84] proposed another relevant algorithm, called **GA-unsupervised**, which selects cluster centers directly from the data set. The DB index was used to measure the validity of the clusters. The experiments considered 100 2D synthetic data sets with the number of clusters varying between 2 and 10, and the number of data points between 100 and 700. The results shown the superiority of GA-unsupervised over the GCUK approach [8] in terms of clustering quality and time efficiency.

4.1.4. Differential Evolution

Differential evolution (**DE**), proposed by Storn and Price [90], is an evolutionary algorithm that works in the following stages: initialization of vectors, mutation (differential operator), crossover (recombination), and selection. A detailed review of the basic concepts and a survey of the principal variants and applications are given in [91]. Meanwhile, a comprehensive review of DE-based clustering techniques can be found in [92].

Das et al. [43] presented a DE-based approach for automatic clustering called **ACDE**. In this approach, each individual is a real value vector of fixed length that encodes activation thresholds linked to cluster centroids. If a threshold value is larger than 0.5, then its respective centroid is active; otherwise, it is inactive. In relation to the DE framework, the authors proposed varying the scale factor of mutation in a random manner; also, the crossover rate of recombination decreases linearly within a predefined range. ACDE was tested with both the CS and DB indices as fitness functions. Its performance was compared with four clustering techniques: GCUK [8], dynamic clustering PSO (DCPSO) [88], the classical DE algorithm, and the average-link algorithm. The experimental results considered five real-life data sets and an application in image segmentation. ACDE outperformed its counterparts in a statistically meaningful way in terms of classification error. The same authors proposed a fuzzy version of ACDE, called **AFDE**, in which the fitness function becomes a fuzzy point symmetric measure. This algorithm was applied to the segmentation of six real-life grayscale images. Also, a comparison with the Fuzzy-VGA [52] and the fuzzy c-means algorithm was carried out, were the results suggested that AFDE is capable of finding the correct number of clusters consistently. Later, Das and Sil [48] presented a DE-based kernelized fuzzy clustering with neighborhood topology (**KFNDE**), which uses a kernelized XB index as a fitness function. Besides, the authors proposed a weighted mutation operator that combines local and global mutant vectors. The KFNDE was used to segment ten real-life grayscale images. The experimental results shown that it surpassed both the Fuzzy-VGA [52] and the kernel-based fuzzy ant clustering (KFCA) algorithms in terms of finding the correct number of groups and segmentation accuracy.

Maulik and Saha [54] proposed a modified DE-based automatic fuzzy clustering, called **MoDEAFC**, which uses a fixed length representation wherein each individual encodes the centroids and the masker for activating or deactivating a centroid. This algorithm minimizes the XB index to achieve proper clustering. A classical mutation or a global-to-local-based mutation is selected as the mutation operator of the DE algorithm according to a selection value that decreases exponentially within the range between [1, 0.5]. The MoDEAFC was tested on two IRS images, and its performance was compared with the classical DE-based fuzzy clustering (DEAFC), Fuzzy-VGA [52], ACDE [43], and an iterated version of the fuzzy c-means algorithm. The results shown that MoDEAFC reached the correct number of clusters in almost all the executions.

4.2. Swarm Intelligence

Studies of social animals and organisms have resulted in a number of computational models of swarm intelligence (SI). In this sense, SI is the property of a system whereby the collective behaviors of unsophisticated agents (interacting locally with their environment and neighboring individuals) cause more complex behaviors, which can be used to solve optimization problems. The basic theory and distinct models of SI are extensively described in [93].

This section outlines automatic clustering algorithms based on computational SI models, specifically focusing on particle swarm optimization (PSO), ant colony optimization (ACO), invasive weed optimization (IWO), and bee colony optimization (BCO) algorithms.

4.2.1. Particle Swarm Optimization

Particle swarm optimization (**PSO**), introduced by Kennedy and Eberhart [94], is a population-based search procedure in which the individuals (referred to as particles) are grouped into a swarm. The particles explore the search space by adjusting their trajectories iteratively according to self-experience and neighboring particles [93]. Although PSO-based clustering was first introduced by Omran et al. [95] in 2005, few attempts at automatic clustering have been proposed as summarized below.

Omran et al. [96] presented a dynamic clustering approach based on PSO, called **DCPSO**, which was developed for image segmentation. The algorithm starts by partitioning the data set into a large number of clusters to reduce the effects of initial conditions. Next, by means a binary PSO, the "best" number of clusters is selected. Then, the centers of the chosen clusters are refined by using k-means. Three fitness functions were tested: DI, TI, and SDbw index. DCPSO was applied on 11 synthetic images and six natural images, and its performance was compared with that of the unsupervised fuzzy approach (UFA) and the SOM technique. The results suggested that DCPSO with the use of the TI consistently achieved the proper range of cluster numbers.

Das et al. [97] proposed a PSO-based segmentation algorithm for automatically grouping image pixels into different regions. The PSO algorithm was modified by incorporating a multi-elitist strategy for searching the global best to prevent premature convergence. In this modified algorithm, called **MEPSO**, a single particle encodes both the cluster centroids and the activation thresholds for activating or deactivating centroids. The fitness of a particle is computed by using a modified XB index, which considers spatial information in the membership function. The experimental results indicated that MEPSO outperformed the Fuzzy-VGA [52] and fuzzy c-means techniques, with the XB index attaining the smallest values when three grayscale images were segmented. Later, the same authors used the MEPSO algorithm for data clustering (herein called **KMEPSO**), which uses a kernelized CS index as a fitness function [98]. The performance of KMEPSO was compared with GCUK, DCPSO, and conventional PSO by considering five artificial and three real-life data sets. The experimental results shown that KMEPSO outperformed the other algorithms in terms of the quality of clustering.

A point symmetry-based PSO clustering algorithm (**PSOPS**) was proposed by Qu et al. [99]. PSOPS evaluates the fitness of a particle by computing the Sym-index. Each particle represents the vector of centroids of variable length. The effectiveness of PSOPS was compared with the conventional PSO clustering based on the Euclidean distance by considering one real-life and two artificial data sets. The experimental results shown that PSOPS is better at detecting the correct clustering when the data sets possess the property of symmetry; however, PSOPS tends to fail in clusters that are not symmetric.

Ouadfel et al. [100] presented an automatic image segmentation algorithm based on a modified version of PSO. This algorithm, called **ACMPSO**, starts by partitioning the image into different regions encoded in each particle of the swarm. Then, three mutation operators (permute, split, and merger) are used to refine the partition obtained from a particle and reduce the number of clusters. The CS index was used as a fitness function to evaluate each particle of variable length. The performance of ACMPSO was compared with that of automatic clustering algorithms, such as DCPSO [96], VGA-clustering [61], and ISODATA, by considering three gray-scale images. The ACMPSO algorithm was found to be better than the other evaluated clustering algorithms in terms of finding the correct number of clusters.

Cura [101] presented a new PSO approach to the clustering problem that is applicable when the number of clusters is either known or unknown. This algorithm (herein called **CPSO**) follows the *gbest* neighborhood topology. CPSO encodes the cluster centroids in particles, which could create new partitions during the optimization procedure by removing or splitting clusters until the assigned number of clusters is reached. This algorithm uses two fitness functions: if the number of clusters is known *a priori*, the WGS index is used; if the number of clusters is unknown, the difference between the WGS and BGS indices is applied. CPSO was tested on two artificial and five real-life data sets; its performance was compared with that of three nonautomatic clustering techniques, with CPSO achieving the best performance in terms of clustering quality. However, when the number of clusters is unknown, its robustness is reduced because the difference between the best and worst fitness values is relatively high for all data sets, even for well-separated ones.

A hybrid approach to automatic clustering involving PSO and GA was proposed by Kuo et al. [102]. In this technique, called **DCPG**, two parents are first created, the first from the PSO and the second generated by using the GA crossover and mutation operators. Then, the next iterative parent is defined by elitist selection until the maximum number of iterations is met. The fitness value of each particle is calculated by using the TI index. Finally, *k*-means is used to adjust the obtained centroids. Four real-life data sets were used in the experimentation, and the results of the DCPG were compared with that of DCPSO [96], ACMPSO [100], and a conventional GA-clustering. For almost all the tested data sets, the DCPG algorithm found a suitable number of clusters. This algorithm involves a higher computational cost because its complexity; however, it requires fewer iterations than its counterparts to find the optimal solution. Recently, the same authors proposed another hybrid PSO-based algorithm called **PSOAC** [103], which was applied to the cell formation (CF) problem. PSOAC adopts a centroid-based representation of variable length to encode both the number of cells and the machine cluster centroids. The fitness of each particle is computed by minimizing the number of exceptional elements (NEE). Besides, a discrete PSO algorithm is utilized to search the number of machine cells and a continuous PSO algorithm is applied to perform machine clustering. The effectiveness of PSOAC was compared with seven clustering algorithms (six non-automatic and one automatic) by considering 20 artificial test problems. The experimental results shown the superiority performance of PSOAC over its counterparts in terms of the best numbers of cells.

4.2.2. Ant Colony Optimization

The ant colony optimization (**ACO**) algorithm, introduced by Dorigo [104], is a probabilistic technique inspired by the behavior of ants in finding paths from the colony to food. A survey of ACO theory is provided in [105] and some recent applications in [106].

Kanade and Hall [107] presented an ant-based clustering algorithm (herein called **Ant-clustering**) that tries to find the adequate number of clusters and produce an initialization for the fuzzy *c*-means algorithm. Initially, the objects are placed randomly on a discrete 2D board. Such a board is a toroidal matrix that allows the ants to travel easily from one end to the other. Then, the ants cluster the objects to form heaps, which are collections of two or more objects; two procedures are proposed for picking and dropping objects to form heaps. After the formation of heaps, the fuzzy *c*-means algorithm is used to refine the clusters obtained in the first stage. Ant-clustering was applied to three well-known real-life data sets (Iris, Wine, and Glass), for which it found the adequate number of clusters in almost all cases. However, Ant-clustering is sensitive to the parameter initialization, for instance, the ants are sensitive to the threshold for deciding when to merge heaps and when to remove items from a heap.

Handl et al. [108] presented an adaptive time-dependent transporter ant for clustering (**ATTA-C**). The authors proposed some modifications to the conventional ACO algorithm to penalize high dissimilarities, to improve the spatial separation between clusters, and to accelerate the clustering process. The fitness value of each clustering solution is calculated by using a neighborhood function (NF). This algorithm was subjected to an experimental evaluation with seven synthetic and seven real-life data sets. The results shown the ability of ATTA-C to automatically identify the number of clusters inherent in a data collection; indeed, the technique is particularly robust for clusters that differ in size and overlap. ATTA-C also outperformed other clustering techniques, such as *k*-means, SOM, hierarchical clustering, and the gap statistic method in identifying the correct number of clusters.

4.2.3. Invasive Weed Optimization

Invasive weed optimization (**IWO**), a relatively recent metaheuristic technique proposed by Mehrabian and Lucas [109], is a numerical stochastic search algorithm that mimics the natural behavior of weed colonization. IWO is inspired by unwanted plants (i.e., weeds), whose vigorous and invasive growth poses a serious menace to cultivated plants, making them a threat to agricultural production. Weeds reproduce rapidly and, thus, quickly increase their population. Their behavior changes with time as the colony becomes dense, leaving a smaller chance of survival for plants that are less fit. The basic design principles, operators, and theoretical issues of IWO are reviewed in [109]. Also, IWO has been gradually used in pattern recognition problems, such as in a recent proposal by Razavi-Far and Palade [110], which attempt to address the problem of pattern classification for multi-class data sets.

The first attempt at automatic clustering based on single-objective IWO was proposed by Chowdhury et al. [111]. In this approach, called **IWO-clustering**, variable-length encoding was used to denote weed strings; thus, each weed in the population encodes cluster centroids. The fitness function used in IWO-clustering was a modified Sym-index. Computational experiments were carried out by considering four data sets (two artificial and two real-life). The results shown that IWO-clustering performed better than the VGAPS [36], GCUK [8], and HNGA [78] algorithms in terms of the Minkowski score.

4.2.4. Bee Colony Optimization

Bee colony optimization (**BCO**) is a swarm intelligence method introduced by Karaboga in 2005 [112] and its performance was analyzed in [113, 114]. The BCO algorithm imitates the behavior of real bees in finding food sources and sharing information with other bees.

The BCO algorithm for clustering analysis was studied in [115, 116, 117]. However, these proposals are limited to data clustering with a known number of groups.

Recently, Kuo et al. [118] presented a kernelized BCO-based automatic clustering algorithm (called **AKC-BCO**), which uses a Gaussian kernel and applies the CS index as a fitness function. In AKC-BCO each encoding solution involves two sections: the threshold values and their corresponding centroids. If a threshold value is greater than 0.5, then its corresponding centroid is active; otherwise, it is inactive. To evaluate the performance of the AKC-BCO algorithm, seven real-life data sets were used. The experimental results shown that AKC-BCO is more stable and accurate than the MEPSO [98], DCPSO [96], and DCPG [102] techniques. Regarding the number of clusters, AKC-BCO found the closest solutions to the actual clusterings. In terms of time consumption, AKC-BCO requires a longer computational time than its counterparts.

4.3. Artificial Immune Systems

Artificial immune system (**AIS**) was inspired by the principles of immunology and the observed immune processes of vertebrates. AIS is highly robust, adaptive, inherently parallel, and self-organized. It has powerful learning and memory capabilities and presents an evolutionary type of response to infectious foreign elements [41]. Most AIS optimization algorithms are based on clonal selection, immune networks, and negative selection algorithms, which are reviewed in [119].

The clonal selection algorithm (**CSA**) has been successfully applied to nonautomatic cluster analysis [120, 121]. Additionally, Liu et al. [45] developed a CSA-based approach to automatic clustering, in which a gene transposon operator based on CSA (called **GTCSA**) is presented. An improved variant of CSA was used to determine the correct number of clusters and the appropriate partitioning. The PBM index was used to define the antibody affinity function. GTCSA was compared with three automatic clustering algorithms, including ACDE [43], VGA-clustering [77], and DCPSO [96]. Computational experiments were carried out by considering 23 data sets (15 artificial and eight real-life). The GTCSA outperformed its counterparts on most of the tested data sets in terms of the Minkowski score.

Recently, R. Liu et al. [122] proposed an immune automatic clustering algorithm named **DLSIAC**. This approach presents a dynamic local search scheme to exploit the neighborhood of each antibody during the evolution process. The antibody encodes both the centroids of clusters and the activation thresholds for activating or deactivating centroids. DLSIAC evaluates the fitness of each antibody by computing the PBM index. Its effectiveness was compared with five automatic clustering techniques TGCA [46], ACDE [43], DCPSO [88], FVGA [52], and MODEAFC [54] by considering 18 artificial and 12 real-life data sets. The DLSIAC algorithm performed better than its counterparts on almost tested data sets in terms of the correct number of clusters and clustering accuracy. Besides, DLSIAC was satisfactorily applied to the segmentation of five texture images.

4.4. Summary

Automatic clustering algorithms based on single-objective nature-inspired metaheuristics were reviewed. Different metaheuristics, validity indices, proximity measures, and encoding schemes influence the quality of the resultant clustering; thus, the selection of the appropriate techniques is important.

Table 3 summarizes relevant details about the 45 automatic clustering algorithms described in Section 4. The clustering techniques are grouped according to the nature-inspired metaheuristics, with corresponding columns for the algorithm name, encoding scheme, objective function (i.e., validity index), test data sets and the application field, and the literature reference of the approach. Also, some symbols denote the type of clustering technique (hard or fuzzy), the proximity measure, and the type of linear separability of the synthetic data sets. Additionally, the symbol over the type of data set denotes if it is popular, $\widetilde{\mathrm{D}}$ (Tables 1 and 2), or particular, $\widehat{\mathrm{D}}$, for the algorithm assessment. Concerning the application field, the notation "IS" indicates image segmentation.

It is remarkable that only three GA-based algorithms, namely DNNM, Fuzzy-VGAP, and GCUK, were tested with a complete evaluation (i.e., synthetic and real data sets as well as an application field). Also, 29 algorithms were evaluated with synthetic data sets with linearly separable clusters for most of them. Besides, six approaches (VGAPS, GCA, KMEPSO, FuzzyGES, DLSIAC, and GTCSA) attempted to solve non-linear separable clusters. However, they obtained inadequate clustering solutions because the objective functions just considered the compactness and separation of clusters represented by their respective single prototypes. In order to overcome this limitation, multiobjective clustering algorithms usually incorporate other objective functions to measure not only the compactness, but also the connectedness of clusters as depicted in next section.

5. Multiobjective Metaheuristics

Nowadays, multiobjective optimization is considered an important discipline in science and engineering. Indeed, Coello lists more than 4000 references on this subject on his website [123].

As the name suggests, a multiobjective optimization problem (MOOP) attempts to optimize more than one objective function simultaneously. The optimal solution for MOOP is not a single solution, such as for single-objective optimization problems, but a set of solutions, denoted as Pareto-optimal solutions. In this sense, a solution is Pareto-optimal if it is impossible to improve a given objective without deteriorating at least another one. Generally, such set of solutions represents the compromise solutions between different conflicting objectives [41].

The success of a clustering algorithm depends mainly on the definition of the objective function to be optimized [124]. However, the optimization of a single clustering criterion is unable to identify the underlying clusters present in the data set and, consequently, leads to poor accuracy [125]. This suggests that the quality of a clustering solution should be evaluated by distinct validity indices rather than a single criterion. Therefore, the clustering task is formulated as a multiobjective problem in which multiple optimization criteria can be set and evaluated simultaneously.

Recently, Bong and Rajeswari [6] reported that the algorithm design, development, and applications of multiobjective nature-inspired metaheuristics for clustering and classification have increased notably during the years 2006 to 2010. Another relevant reference is a book related to multiobjective genetic clustering algorithms published by Maulik et al. [126] .

Summarized in the following subsections are different multiobjective clustering approaches, in which the clustering task is formulated as a MOOP by simultaneously optimizing two or more criterion functions. The main feature of these algorithms is that they automatically partition the data set into an appropriate number of clusters as well as the correct grouping.

5.1. Multiobjective Evolutionary Algorithms

5.1.1. Multiobjective Genetic Algorithm

Research in the area of automatic clustering became popular after the work by Handl and Knowles [7] entitled “Multiobjective Clustering with K determination,” which was published in 2007. Before this work, the same authors proposed the first nonautomatic multiobjective clustering algorithm based on PESA-II, called VIENNA [127]. Later, the authors fine-tuned one of the objectives used in VIENNA and developed a method for automatically determining the number of clusters. Both improvements were incorporated into an enhanced version of VIENNA called **MOCK** [7]. This approach consists of two main stages: clustering and model selection. In the clustering phase, a multiobjective evolutionary algorithm (MOEA) with graph-based representation is used to optimize two fitness functions: the compactness and connectedness of clusters. This produces a set of mutually nondominated clustering solutions, which correspond to different trade-offs between the objectives and the number of clusters. In the model selection stage, MOCK analyzes the shape of the trade-off curve, which is compared with an appropriate null model obtained from random data clustering. Based on this analysis, MOCK provides an estimation of the quality of all individual clustering solutions and determines a set of potentially promising clustering solutions. MOCK outperformed traditional single-objective clustering techniques (k-means, average link, single link, and an ensemble technique) across diverse benchmark data sets. However, Matake et al.[128] shown that the computational cost of MOCK is too high when the technique is applied to large data sets. Thus, they developed a data-clustering algorithm for web mining based on MOCK, where the main contributions are: (*i*) a new automatic k-determination scheme that is able to discover the appropriate number of clusters at low cost, and (*ii*) a new strategy for k-determination that does not require a null model. These improvements allow the application of MOCK to large data sets.

Liu et al. [129] developed an automatic clustering approach based on NSGA-II, called **MOKGA**, which minimizes two objective functions: the number of clusters and the partitioning error. Additionally, a multiple Pareto-optimal front layer ranking method was proposed to maintain a relatively consistent population size in the genetic process. The applicability and effectiveness of MOKGA was evaluated though experiments with the Iris and Ruspini data sets. The authors analyzed the clustering results by using several CVIs, such as SI, C-index, DI, DB, SD index, and SDbw index. Similarly, Bandyopadhyay et al. [25] proposed a **NSGA-II-clustering** technique that simultaneously optimizes two cluster validity measures, the XB and FCM indices. A real encoding with centroid-based representation was used. The experiments results were carried out by considering numeric remote sensing data and IRS satellite images.

Recently, NSGA-II was adopted as the basis for the multiobjective evolutionary approach based on soft subspace clustering (**MOEASSC**) proposed by Xia et al. [130], which simultaneously minimizes both the WGS and the information on the negative weight entropy plus separation between clusters. Additionally, to select the best solution and the number of clusters, an indicator called the projection similarity validity index (PSVIndex) was used,

Table 3: Automatic clustering approaches based on single-objective metaheuristics. “Encoding Scheme” indicates whether the algorithm involves centroid-based (C) encoding of variable length (VL) or fixed length (FL), label-based encoding (L), and binary encoding (B); “Objective Function” refers to the validity index as defined in Section 2.4; “Synthetic Data”, “Real Data”, and “Application Field” denote the type of data set to evaluate the algorithm and the number of data sets in parenthesis; and “Ref.” indicates the literature reference of the approach.

Metaheuristic	Algorithm Name	Encoding Scheme	Objective Function	Synthetic Data	Real Data	Application Field	Ref.
GA	TGCA$_{\mathcal{H}}$	C-VL	CH$^{\triangleright}$	$\widehat{S}^{\bullet}_{(4)}$	$\widetilde{R}_{1,2,4}$, $\widehat{R}_{(4)}$	–	[46]
	AGUCK$_{\mathcal{H}}$	C-VL	DB$^{\triangleright}$	$\widehat{S}^{\bullet\blacktriangledown}_{(100)}$	–	–	[82]
	F-EARFC$_{\mathcal{F}}$	C-VL	SI$^{\triangleright}$	$\widehat{S}^{\bullet\blacktriangledown}_{(18)}$	$\widehat{R}_{(2)}$	–	[81]
	KMQGA$_{\mathcal{H}}$	C-VL	DB$^{\triangleright}$	$\widehat{S}^{\bullet}_{(3)}$	$\widetilde{R}_{1,3,4,7}$	–	[80]
	DNNM$_{\mathcal{H}}$	C-FL	DSF$^{\triangleright}$	$\widetilde{S}_{1,4}$, $\widehat{S}^{\bullet\blacktriangledown}_{(3)}$	$\widetilde{R}_{1,2}$	IS: Satellite$_{(1)}$	[53]
	Fuzzy-VGAPS$_{\mathcal{F}}$	C-VL	Sym$^{\triangleleft}$	$\widetilde{S}_{2,5-7}$, $\widehat{S}^{\bullet\blacktriangledown}_{(1)}$	$\widetilde{R}_{1,2,4}$	IS: Medical$_{(1)}$	[57]
	HEA$_{\mathcal{H}}$	B	WGS$^{\triangleright}$	–	–	IS: Medical$_{(5)}$	[49]
	ACBEA$_{\mathcal{H}}$	L	CS$^{\triangleright}$	–	$\widetilde{R}_{1-4}$	–	[87]
	VGAPS$_{\mathcal{H}}$	C-VL	Sym$^{\triangleleft}$	$\widetilde{S}_{1,6,8}$, $\widehat{S}^{\bullet\blacktriangledown\star}_{(6)}$	$\widetilde{R}_{1,2,4}$, $\widehat{R}_{(2)}$	–	[36]
	ETSA$_{\mathcal{H}}$	C-FL	PBM$^{\triangleright}$	$\widehat{S}^{\bullet\blacktriangledown}_{(5)}$	$\widetilde{R}_{1,2}$, $\widehat{R}_{(2)}$	–	[65]
	EvoCluster$_{\mathcal{H}}$	L	WIM$^{\triangleright}$	$\widehat{S}_{(1)}$	–	Microarray$_{(2)}$	[37]
	EAC$_{\mathcal{H}}$	L	SI$^{\triangleright}$	–	$\widetilde{R}_{6}$	Microarray$_{(5)}$	[56]
	HGA$_{\mathcal{H}}$	C-FL	DB$^{\triangleright}$	$\widetilde{S}_{1}$, $\widehat{S}^{\bullet\blacktriangledown}_{(4)}$	$\widetilde{R}_{1,2}$	–	[83]
	HNGA$_{\mathcal{H}}$	C-VL	WSVF$^{\triangleright}$	$\widehat{S}^{\bullet\blacktriangledown}_{(3)}$	$\widetilde{R}_{1,2}$	–	[78]
	GA-clustering$_{\mathcal{H}}$	B	DB$^{\triangleright}$	$\widehat{S}^{\bullet\blacktriangledown}_{(100)}$	–	–	[84]
	GCA$_{\mathcal{H}}$	B	WGS$^{\triangleright}$	$\widehat{S}^{\bullet\blacktriangledown\star}_{(9)}$	$\widetilde{R}_{1}$	–	[89]
	CGA$_{\mathcal{H}}$	L	SI$^{\triangleright}$	$\widehat{S}^{\bullet}_{(1)}$	–	–	[86]
	VGA-FCM$_{\mathcal{F}}$	C-FL	FCM$^{\triangleright}$	–	–	IS: Satellite$_{(1)}$	[85]
	Fuzzy-VGA$_{\mathcal{F}}$	C-VL	XB$^{\triangleright}$	–	–	IS: Satellite$_{(2)}$	[52]
	GACA$_{\mathcal{H}}$	L	SI$^{\triangleright}$	$\widehat{S}^{\bullet}_{(2)}$	$\widetilde{R}_{1,2}$	–	[79]
	GCUK$_{\mathcal{H}}$	C-FL	DB$^{\triangleright}$	$\widetilde{S}_{1,3,4}$, $\widehat{S}^{\bullet}_{(1)}$	$\widetilde{R}_{1,2}$	IS: Satellite$_{(1)}$	[8]
	VGA-clustering$_{\mathcal{H}}$	C-VL	PBM$^{\triangleright}$	$\widetilde{S}_{1-3}$	$\widetilde{R}_{1,2}$	–	[77]
	CLUSTERING$_{\mathcal{H}}$	C-VL	WGS$^{\triangleright}$	$\widehat{S}^{\bullet\blacktriangledown}_{(2)}$	–	–	[76]
DE	MoDEAFC$_{\mathcal{F}}$	C-FL	XB$^{\triangleright}$	–	–	IS: Satellite$_{(2)}$	[54]
	KFNDE$_{\mathcal{F}}$	C-FL	XB$^{\diamond}$	–	–	IS: Natural$_{(10)}$	[48]
	AFDE$_{\mathcal{H}}$	C-FL	Sym$^{\triangleleft}$	–	–	IS: Natural$_{(6)}$	[47]
	ACDE$_{\mathcal{H}}$	C-FL	DB$^{\triangleright}$	–	$\widetilde{R}_{1-5}$	IS: Natural$_{(6)}$	[43]
PSO	PSOAC$_{\mathcal{H}}$	C-VL	NEE$^{\triangleright}$	$\widehat{S}_{(3)}$	$\widehat{R}_{(17)}$	–	[103]
	CPSO $_{\mathcal{H}}$	C-FL	WGS$^{\triangleright}$, BGS$^{\triangleright}$	$\widehat{S}^{\bullet\blacktriangledown}_{(2)}$	$\widetilde{R}_{1,3,4}$, $\widehat{R}_{(2)}$	–	[101]
	DCPG$_{\mathcal{H}}$	C-FL	TI$^{\triangleright}$	–	$\widetilde{R}_{1,3-5}$	–	[102]
	PSOPS$_{\mathcal{H}}$	C-VL	Sym$^{\triangleleft}$	$\widehat{S}^{\bullet}_{(2)}$	$\widetilde{R}_{1}$	–	[99]
	ACMPSO$_{\mathcal{H}}$	C-FL	CS$^{\triangleright}$	–	–	IS: Natural$_{(3)}$	[100]
	KMEPSO$_{\mathcal{H}}$	C-FL	CS$^{\diamond}$	$\widetilde{S}_{1,3}$, $\widehat{S}^{\bullet\blacktriangledown\star}_{(3)}$	$\widetilde{R}_{2-4}$, $\widehat{R}_{(2)}$	–	[98]
	DCPSO$_{\mathcal{H}}$	C-FL	TI$^{\triangleright}$	–	–	IS: Natural$_{(21)}$	[88, 96]
	MEPSO$_{\mathcal{F}}$	C-FL	XB$^{\triangleright}$	–	–	IS: Natural$_{(3)}$	[97]
ACO	ATTA-C$_{\mathcal{H}}$	C-FL	NF$^{\triangleright}$	–	$\widetilde{R}_{1,2}$, $\widehat{R}_{(5)}$	–	[108]
	Ant-clustering$_{\mathcal{F}}$	C-FL	FCM$^{\triangleright}$	–	$\widetilde{R}_{1,3,4}$	–	[107]
IWO	IWO-clustering$_{\mathcal{H}}$	C-VL	Sym$^{\triangleleft}$	$\widetilde{S}_{3,4}$	$\widetilde{R}_{1,4}$	–	[111]
ES	FuzzyGES$_{\mathcal{F}}$	C-VL	XB$^{\triangleright}$	$\widehat{S}^{\bullet\blacktriangledown\star}_{(13)}$	$\widetilde{R}_{1}$	–	[74]
	ES-clustering$_{\mathcal{H}}$	C-VL	WGS$^{\triangleright}$	$\widehat{S}^{\bullet\blacktriangledown}_{(2)}$	–	–	[73]
EP	GEP$_{\mathcal{H}}$	C-FL	WGS$^{\triangleright}$	$\widehat{S}^{\bullet}_{(2)}$	–	–	[71]
	EP-clustering$_{\mathcal{H}}$	C-FL	DB$^{\triangleright}$	$\widehat{S}^{\bullet\blacktriangledown}_{(3)}$	–	–	[70]
CSA	DLSIAC$_{\mathcal{F}}$	C-FL	PBM$^{\triangleright}$	$\widehat{S}^{\bullet\blacktriangledown\star}_{(18)}$	$\widehat{R}_{(12)}$	–	[122]
	GTCSA$_{\mathcal{H}}$	C-VL	PBM$^{\triangleright}$	$\widetilde{S}_{2,3}$, $\widehat{S}^{\bullet\blacktriangledown\star}_{(13)}$	$\widetilde{R}_{1-4}$, $\widehat{R}_{(4)}$	–	[45]
BCO	AKC-BCO$_{\mathcal{H}}$	C-FL	CS$^{\diamond}$	–	$\widetilde{R}_{1,3,4}$, $\widehat{R}_{(4)}$	–	[118]

Type of clustering technique: hard ($\mathcal{H}$) and fuzzy ($\mathcal{F}$).
Type of proximity measures: euclidean ($\triangleright$), point symmetry ($\triangleleft$), and kernel-based measure ($\diamond$).
Type of synthetic data set: linearly separable ($\bullet$), linearly separable with overlapping ($\blacktriangledown$), and non-linearly separable ($\star$)

which assumes that objects belonging to the same cluster are similar in relevant dimensions. MOEASSC was compared with four clustering algorithms by considering three artificial and 13 real-life data sets. The results shown that MOEASSC has better clustering accuracy than its counterparts.

5.1.2. Multiobjective Differential Evolution

Due to its success in solving single-objective optimization problems in continuous search spaces, differential evolution (DE) has been extended to MOOP. A comprehensive review of DE for multiobjective optimization is given in [131].

Suresh et al. [58, 132] applied DE to the task of automatic clustering in a multiobjective optimization framework. The study compared the performances of two multiobjective variants of DE, namely, **MODE** and **DEMO**, over the fuzzy clustering problem, in which the XB and FCM indices were optimized. A real centroid-based encoding of the search variables was used. The gap statistic was applied in choosing the most interesting solutions from the Pareto front. The final clustering quality was evaluated by considering two measures: the adjusted Rand index and the SI index. The performance of MODE and DEMO were compared with that of the NSGA-II clustering [25] and MOCK [7] techniques. The experimental results based on six artificial and four real life data sets with different complexities indicated that MODE and DEMO produced better final clustering solutions that of MOCK and NSGA-II clustering for all instances.

Saha et al. [133] proposed another fuzzy clustering technique based on MODE. This technique, called **MOMoDEFC**, encodes the cluster centers and optimizes the XB and FCM validity measures. The performance of MOMoDEFC was compared with that of multiobjective DE-based fuzzy clustering, multiobjective GA-based fuzzy clustering, single-objective DE-based fuzzy clustering, and the fuzzy *c*-means algorithm. Computational experiments were carried out by considering two artificial and four real-life data sets and a statistical significance test shown the superiority of MOMoDEFC.

Zhong and Zhang [55] also optimized the XB and FCM indices by means of a two-layer fuzzy clustering system based on multiobjective DE. This algorithm, called **AFCMDE**, has both optimization and clustering layers. It was applied to IRS data clustering, in which the same land types may have different spectral curves, whereas different land types could have similar curves. The clustering results of the fuzzy *c*-means algorithm, ACDE [43], and MoDEAFC [54] were compared with that of AFCMDE, where the latter shown better classification accuracy.

The **GADE** algorithm, developed by Kundu et al. [134], is a hybrid clustering approach based on MODE and GA. GADE incorporates some GA operators into the MODE framework. The XB and FCM indices were used as the objective functions to be optimized. Six synthetic and four real-life data sets were used to evaluate the performance of GADE, and the resultant solutions were compared with those obtained from MOCK [7] and NSGA-II clustering [25]. GADE shown the best performance in terms of the adjusted Rand index and the SI index.

5.2. Multiobjective Swarm Intelligence

The improvement of swarm intelligence methods for applications in multiobjective problems is an emergent research area. In this regard, Nedjah et al. [135] presented a review about theories and experiences in multiobjective swarm intelligent systems. Additionally, a collection of contributions in the fields of multiobjective optimization, swarm intelligence, and data mining can be reviewed in [136]. In line with this, different multiobjective swarm intelligence algorithms for automatic clustering are described below.

5.2.1. Multiobjective Particle Swarm Optimization

A clustering methodology based on a multiobjective PSO (**MOPSO**) framework was presented by Paoli et al. [137]. This approach was proposed to solve three different issues related to the segmentation of hyperspectral images: (*i*) the estimation of class statistical parameters, (*ii*) the detection of the best discriminative bands without requiring their number *a priori*, and (*iii*) the estimation of the number of data classes (number of clusters) characterizing the image being considered. MOPSO is guided by three optimization criteria: the log-likelihood function (LF), the Bhattacharyya statistical distance (BD) between classes, and the minimum description length (MDL). An experimental analysis was done on both simulated and real hyperspectral images. In general, MOPSO provided satisfactory classification accuracy while reducing the number of bands used for the classification task. However, its main drawback is the time required for the optimization process, which may reach several hours depending on the image size.

Nanda and Panda [124] introduced a hybrid multiobjective algorithm called **MOIMPSO**. This approach combines operators from the clonal selection principle (CSP) and PSO, such as velocity and position update, cloning, hypermutation, and re-selection of immune cells, which are applied to the entire swarm. Two objective functions, WGS and connectivity, were simultaneously optimized to evaluate the effectiveness of the solutions in the multiobjective domain. MOIMPSO provides a single best solution from the Pareto-optimal front, which mostly satisfies the user requirement. Furthermore, the *k*-means algorithm is incorporated at the beginning to provide effective initial solutions. MOIMPSO was compared with the MOCK [108], MOPSO [137], and MOCLONAL [125] approaches through computational experiments that considered five artificial and six real-life data sets. MOIMPSO provided more accurate results at high computational cost. The MOCK and MOCLONAL algorithms attained reasonably good solutions in the majority of data sets, but the solutions reached by MOCLONAL algorithm were more consistent. All four algorithms provided good results for nonoverlapping data sets; however, the proposed MOIMPSO performed better in cases with overlapping groups.

Recently, Abubaker et al. [138] presented a hybrid multiobjective clustering algorithm called **MOPSOSA**. This algorithm combines features from PSO and simulated annealing (SA). MOPSOSA optimizes simultaneously three objective functions: (*i*) the DB index based on the Euclidean distance, (*ii*) the Sym index based on the point symmetry distance, and (*iii*) the Conn-index which is measured by using the relative neighborhood graph concept. The performance of MOPSOSA was compared with six automatic clustering algorithms (GenClustMOO [139], GenClustPESA2 [139], MOCK [7], VGAPS [36]) and two traditional approaches (*k*-means and single-link). In addition, 14 artificial and five real-life data sets were considered. The experimental results demonstrated the superior performance of MOPSOSA over their counterparts in terms of the mean F-measure.

5.2.2. Multiobjective Invasive Weed Optimization

Recently, Kundu et al. [140] extended the IWO algorithm for multiobjective optimization. The new approach (**MOIWO**) was used by Liu et al. [141] to solve the automatic clustering problem in which the number of clusters is uncertain. MOIWO optimizes two validity functions, the XB and FCM indices, and adopts a variable length real-coded scheme in which weed strings encode cluster centroids. To maintain the diversity of the weeds, a mechanism called feedback update is introduced to update those individuals whose corresponding number of centroids has been eliminated in one generation. Finally, the SI index is used to select the best solution. Computational experiments considered 15 artificial and four real-life data sets. The results indicated an adequate performance for MOIWO in spherical data sets for which it determined the correct number of clusters and the appropriate partition. However, in non-globular data sets, MOIWO was unable to select the correct clustering from the Pareto-optimal solutions.

5.3. Multiobjective Artificial Immune System

The first multiobjective technique that used AIS operators was proposed by Yoo and Hajela [142] in 1999. However, the development of automatic clustering algorithms that use the multiobjective artificial immune system (MOAIS) is relatively recent; in this regard, the works of Ma et al. [143] and Nanda and Panda [125] are the most relevant. These approaches considered two complementary objectives based on the compactness and connectedness of clusters. Both objective functions were used originally by the MOCK algorithm [108].

Ma et al. [143] developed the immune dominance clonal multiobjective clustering algorithm (**IDCMC**) based on clonal selection with immune dominance and clone anergy. This approach divides a single population into three subpopulations according to three different measurements and adopts distinct evolution and selection strategies for every sub-population. The experiments considered six artificial data sets with manifold structures and handwritten digit data sets. The results revealed that IDCMC solved most of the tested problems, outperforming MOCK [108] and *k*-means in terms of the adjusted Rand index.

Similarly, Nanda and Panda [125] proposed an automatic multiobjective clustering algorithm based on the clonal selection algorithm. This algorithm, called **MOCLONAL**, uses the operators of cloning, hypermutation, and re-selection of immune cells. Furthermore, the *k*-means algorithm is incorporated at the beginning to provide initial solutions, which are then guided by MOCLONAL. The performance of MOCLONAL compared with MOCK [108] by considering one artificial and two real-life data sets. The experimental results shown the superiority performance of MOCLONAL over MOCK [108] in terms of the Minkowski score achieved over fifty generations.

5.4. Multiobjective Simulated Annealing

In this section, developments related to multiobjective simulated annealing (SA) algorithms for automatic clustering are surveyed. Bandyopadhyay et al. [144] proposed an archived multiobjective simulated annealing algorithm, called AMOSA. In contrast to most multiobjective algorithms, AMOSA selects dominated solutions with a probability that is dependent on the amount of domination measured in terms of the hypervolume between two solutions in the objective space. This algorithm was used by Saha and Bandyopadhyay as the underlying optimization strategy in three automatic clustering approaches, namely, SSym-AMOSA [145], VAMOSA [146], and GenClustMOO [139].

SSym-AMOSA [145] simultaneously optimizes two objective functions, one reflecting the total cluster symmetry and the other reflecting the stability of the obtained partitions over different bootstrap samples of the data set. Additionally, the authors proposed a semi-supervised method [68] to select the best solution from a large number of nondominated solutions on the final Pareto front. In experiments considering eight artificial and six real-life data sets, the performance of SSym-AMOSA was compared with those of four existing clustering techniques: MOCK [108], VGAPS [36], GCUK [8], and HNGA [78]. The results shown that SSym-AMOSA was able to determine the correct number of clusters for almost all the tested data sets.

In the **VAMOSA** [146] technique, both the XB and Sym indices are defined as objective functions to determine the appropriate number of clusters. Moreover, the semi-supervised method used by SSym-AMOSA is also used to select the best solutions from the Pareto front. The effectiveness of VAMOSA was tested in seven artificial and six real-life data sets. The experimental results suggested that VAMOSA outperformed the MOCK [108], VGAPS [36], and GCUK [8] algorithms. Note that VAMOSA as well as SSym-AMOSA are capable to detect clusters with point symmetry but not those that are nonsymmetrical.

GenClustMOO [139] is a recent approach that optimizes three objective functions: (*i*) the total compactness of the partitioning based on the Euclidean distance, (*ii*) the total symmetry of the clusters based on the point symmetry distance, and (*iii*) the cluster connectedness, which is measured by using the relative neighborhood graph concept. Because AMOSA provides a set of Pareto-optimal solutions, a new method was also developed to determine a single solution from such set. GenClustMOO applies a mechanism to merge clusters from a variable number of global clusters. Its effectiveness was compared with MOCK [108], VGAPS [36], *k*-means, and a single linkage clustering by considering 19 artificial and seven real-life data sets. The experimental results shown that GenClustMOO was able to detect the correct number of clusters and the appropriate partitioning in data sets having either well-separated clusters with arbitrary shapes or symmetrical clusters with or without overlap.

5.5. Summary

Table 4 summarizes relevant features of the 18 aforementioned automatic clustering algorithms based on multiobjective metaheuristics. The columns provide the basic metaheuristic, algorithm name, encoding scheme, objective function (i.e., validity index), test data sets and the application field, and the literature reference of the approach. Also, some symbols denote the type of clustering technique (hard or fuzzy), the proximity measure, and the type of linear separability of the synthetic data sets. Additionally, the symbol over the type of data set denotes if it is popular, $\widetilde{\text{D}}$ (Tables 1 and 2), or particular, $\widehat{\text{D}}$, for the algorithm assessment. Concerning the application field, the notation "IS" indicates image segmentation.

Notice that only MOMoDEFC and MOIWO were tested with a complete evaluation involving synthetic and real data sets as well as an application field. Also, 14 algorithms used synthetic data sets to evaluate their performances, mostly considering linear separable clusters. However, only four algorithms (MOCK, mMOCK, MOPSOSA, and GenClustMOO) considered non-linear separable clusters and they obtained better results than single-objective approaches. It is notable that such algorithms measured both the connectedness (with Conn index) and the compactness or "balancedness" (with WGS, PBM, and DB indices) of clusters. To deal with the connectivity of data, MOCK and mMOCK algorithms used a graph-based encoding scheme, whereas MOPSOSA utilized a label-based encoding scheme. Note that these algorithms codified the entire data set points to represent a clustering solution. On the other hand, GenClustMOO approach used a centroid-based encoding scheme, where each cluster is represented by multiple prototypes, which is shorter than encoding schemes based on graphs and labels. Besides, MOPSOSA and GenClustMOO incorporated a third objective function, namely Sym index, to discover symmetric shaped clusters, such as lines, ellipses, etc.

6. Discussion

Research about automatic clustering based on nature-inspired metaheuristics is relatively recent and continuously new concepts and applications emerge. Figure 5 illustrates the timeline of automatic clustering approaches based on the nature-inspired metaheuristics depicted herein. Evolutionary computation is noticeably the most commonly used paradigm for automatic clustering. However, in recent years, swarm intelligence and artificial immune system have become popular to solve the clustering problem. Also, in the last five years, there has been a strong tendency to use multiobjective metaheuristics and hybrid approaches. Single-solution metaheuristics, which were used in the first attempts of automatic clustering algorithms, are rarely used today.

From the 65 articles reviewed herein, Figure 6 shows the fraction of clustering algorithms in function of both the optimization scheme and the metaheuristic technique. In Figure 6a, it is observed a very low percentage of clustering algorithms (3 %) involving single-solution metaheuristics. This is mainly because these kinds of methods only evaluate a single clustering solution. However, if the clustering problem has multiple potential solutions, single-solution metaheuristics could not be appropriated. Thus, population-based metaheuristics have been preferred because they are able to find multiple candidate clustering solutions at the same time. Indeed, single-objective population-based metaheuristics represent the 69 % of the surveyed algorithms. Moreover, the ability of finding multiple solutions makes population-based metaheuristics adequate to solve multiobjective optimization problems. In this regard, nowadays, there is a strong tendency (28 %) in using multiobjective algorithms to address the problem of automatic clustering. Figure 6b evidences that several nature-inspired metaheuristcs belonging to different computational intelligence techniques have been used. It is notable that approaches related to evolutionary computation paradigm have been used in 61 % of the proposals, where the GA is the most used (43 %). This is because the GA has more than 40 years in the community (since the early 1970s) and several GA variants have proofed their capabilities to solve a wide range of real-world optimization problems. Nevertheless, more recent techniques like PSO (17 %) and DE (12 %) algorithms have become popular because they are relatively simple to implement and could solve complex optimization problems in reasonable computation time.

To evaluate the clustering performance of automatic clustering algorithms, distinct data sets in different scenarios should be used. Commonly, real-world data sets adopted by the community as benchmark cases (listed in Table 2) are used to assess and compare objectively distinct clustering approaches. Moreover, some applications fields, such as image segmentation and microarray data analysis, are frequently addressed because the number of groups is difficult to known *a priori*. Although real-world data are useful to demonstrate the capabilities of the clus-

Table 4: Automatic clustering approaches based on multiobjective metaheuristics. “Encoding Scheme” indicates whether the algorithm involves centroid-based (C) encoding of variable length (VL) or fixed length (FL), label-based encoding (L), and graph-based encoding (G); “Objective Functions” refers to the validity index as defined in defined in Section 2.4; “Synthetic Data”, “Real Data”, and “Application Field” denote the type of data set to evaluate the algorithm and the number of data sets in parenthesis; and “Ref.” indicates the literature reference of the approach.

Metaheuristic	Algorithm Name	Encoding Scheme	Objective Functions	Synthetic Data	Real Data	Application Field	Ref.
MOEA	MOEASSC$_{\mathcal{H}}$	C-FL	WGS$^{\triangleright}$ & Entropy$^{\triangleright}$	$\widehat{\mathrm{S}}^{\bullet\blacktriangledown}_{(3)}$	$\widetilde{\mathrm{R}}_{2}$, $\widehat{\mathrm{R}}_{(10)}$	–	[130]
	MOCK$_{\mathcal{H}}$	G	WGS$^{\triangleright}$ & Conn	$\widehat{\mathrm{S}}^{\bullet\blacktriangledown\star}_{(37)}$	–	–	[7]
	mMOCK$_{\mathcal{H}}$	G	WGS$^{\triangleright}$ & Conn	$\widehat{\mathrm{S}}^{\bullet\blacktriangledown\star}_{(9)}$	–	–	[128]
	NSGA-IIc$_{\mathcal{F}}$	C-VL	XB$^{\triangleright}$ & FCM$^{\triangleright}$	–	–	IS: Satellite$_{(3)}$	[25]
	MOKGA$_{\mathcal{H}}$	L	WGS$^{\triangleright}$ & K	–	$\widetilde{\mathrm{R}}_{1,8}$	–	[129]
MODE	AFCMDE$_{\mathcal{F}}$	C-FL	XB$^{\triangleright}$ & FCM$^{\triangleright}$	–	–	IS: Satellite$_{(3)}$	[55]
	MOMoDEFC$_{\mathcal{F}}$	C-VL	XB$^{\triangleright}$ & FCM$^{\triangleright}$	$\widetilde{\mathrm{S}}_{1,5}$	$\widetilde{\mathrm{R}}_{1-4}$	IS: Satellite$_{(2)}$	[133]
	MODE$_{\mathcal{F}}$	C-FL	XB$^{\triangleright}$ & FCM$^{\triangleright}$	$\widehat{\mathrm{S}}^{\bullet\blacktriangledown}_{(6)}$	$\widetilde{\mathrm{R}}_{1-3,6}$	–	[58]
	GADE$_{\mathcal{H}}$	C-FL	XB$^{\triangleright}$ & FCM$^{\triangleright}$	$\widehat{\mathrm{S}}^{\bullet\blacktriangledown}_{(6)}$	$\widetilde{\mathrm{R}}_{1-3,6}$	–	[134]
MOPSO	MOPSOSA$_{\mathcal{H}}$	L	DB, Sym, & Conn	$\widetilde{\mathrm{S}}_{1-5}$, $\widehat{\mathrm{S}}^{\bullet\blacktriangledown\star}_{(6)}$	$\widetilde{\mathrm{R}}_{1,2}$, $\widehat{\mathrm{R}}_{(9)}$	–	[138]
	MOIMPSO$_{\mathcal{H}}$	L	WGS$^{\triangleright}$ & Conn	$\widetilde{\mathrm{S}}_{1-4}$	$\widetilde{\mathrm{R}}_{1}$, $\widehat{\mathrm{R}}_{(5)}$	–	[124]
	MOPSO$_{\mathcal{F}}$	C-VL	LF, BD, & MDL	–	–	IS: Satellite$_{(2)}$	[137]
MOSA	GenClustMOO$_{\mathcal{H}}$	C-VL	Sym$^{\triangleleft}$, Conn$^{\circ}$, & PBM$^{\triangleright}$	$\widetilde{\mathrm{S}}_{1-8}$, $\widehat{\mathrm{S}}^{\bullet\blacktriangledown\star}_{(11)}$	$\widetilde{\mathrm{R}}_{1-4}$, $\widehat{\mathrm{R}}_{(3)}$	–	[139]
	VAMOSA$_{\mathcal{H}}$	C-VL	XB$^{\triangleright}$ & Sym$^{\triangleleft}$	$\widetilde{\mathrm{S}}_{1,2,6,8}$, $\widehat{\mathrm{S}}^{\bullet\blacktriangledown}_{(3)}$	$\widetilde{\mathrm{R}}_{1-3}$, $\widehat{\mathrm{R}}_{(3)}$	–	[145]
	SSym-AMOSA$_{\mathcal{H}}$	C-VL	Variants of Sym$^{\triangleleft}$	$\widetilde{\mathrm{S}}_{1,2,6,8}$, $\widehat{\mathrm{S}}^{\bullet\blacktriangledown}_{(4)}$	$\widetilde{\mathrm{R}}_{1-3}$, $\widehat{\mathrm{R}}_{(3)}$	–	[146]
MOAIS	MOCLONAL$_{\mathcal{H}}$	L	WGS$^{\triangleright}$ & Conn	$\widetilde{\mathrm{S}}_{1}$	$\widetilde{\mathrm{R}}_{1,2}$	–	[125]
	IDCMC$_{\mathcal{F}}$	L	WGS$^{\triangleright}$ & Conn	$\widehat{\mathrm{S}}^{\bullet\blacktriangledown}_{(6)}$	–	–	[143]
MOIWO	MOIWO$_{\mathcal{H}}$	C-VL	XB$^{\triangleright}$ & FCM$^{\triangleright}$	$\widehat{\mathrm{S}}^{\bullet\blacktriangledown}_{(15)}$	$\widetilde{\mathrm{R}}_{1-3}$	Microarray$_{(1)}$	[141]

Type of clustering technique: hard ($\mathcal{H}$) and fuzzy ($\mathcal{F}$).
Type of proximity measures: euclidean ($\triangleright$), point symmetry ($\triangleleft$), and Chebyshev distance ($\circ$).
Type of synthetic data set: linearly separable ($\bullet$), linearly separable with overlapping ($\blacktriangledown$), and non-linearly separable ($\star$)

tering algorithm to different sources of information, it is infeasible to control distinct characteristics of the clusters to evaluate the actual limitations and advantages of the clustering algorithms. For this reason, almost 70 % of works revised herein used synthetic data sets in their evaluations because different features can be controlled such as number of clusters, overlapping, density, size, noise, shape, dimensionality, etc. However, the linear separability of the clusters could be the feature that mostly defines the complexity of data set. Herein, three cases to simulate data sets were identified: (1) linear separability without overlapping, (2) near separability with overlapping, and (3) non-linear separability without overlapping.

In general, it was observed that the automatic clustering approaches based on single-objective metaheuristics are capable to cluster adequately data sets of cases 1 and 2; however, few of them attempted to solve data sets of case 3, although their clustering quality was poor. This limitation is because the objective functions (e.g., XB, DB, PBM, Dunn, etc.) only measure the balance between compactness of the cluster around a single prototype and the separation among cluster prototypes. Consequently, clustering techniques based on single-objective metaheuristics assume that the cluster prototypes are sufficiently distant from each other in the feature space. Contrarily, when clusters are non-linear separable, their prototypes could approximately share the same region in the feature space.

In order to adequately discover non-linearly separable clusters, recent multiobjective clustering approaches incorporate additional objective functions to measure the connectivity (Conn index) and the symmetry (Sym index) of clusters, although the encoding of clustering solutions becomes more complex. For instance, MOCK and mMOCK algorithms use a graph-label based encoding scheme; however, the main disadvantage is that the entire data set is codified to represent a single clustering solution, which is computationally expensive. In this sense, recently, the GenClustMOO algorithm has been proposed, which uses a new centroid-based encoding scheme in which multiple prototypes are used to represent a single cluster. This new representation of clustering solutions allows the appropriate partitioning of data sets containing both linearly and non-linearly separable clusters.

Based on these observations, if the clustering problem is linearly separable, a clustering approach based on single-objective metaheuristics could be sufficient to obtain adequate clustering solutions. Approaches such as Fuzzy-VGAPS and GCUK could be considered of general purpose algorithms, because they have been tested under distinct scenarios, including synthetic and real-life data sets as well as image segmentation problems. On the other hand, if the clustering problem is non-linearly separable, a multiobjective clustering approach is recommended because, in general, they attempt to optimize the compactness and the connectedness of clusters simultaneously. In this sense, GenClustMOO algorithm seems to have an acceptable trade-off between clustering quality and computational performance.

7. Future Directions

Automatic clustering is an open field with several research directions and challenges that involve the following issues:

- To solve complex automatic clustering problems, it is necessary to design mechanisms for discovering intrinsic characteristics of the input data in order to choose the appropriate optimization scheme, that is, single-objective or multiobjective optimization. If the input data is linearly separable, a single-objective clustering algorithm is sufficient to efficiently solve the clustering problem. However, if the input data is non-linearly separable, a multiobjective clustering algorithm should be applied.

- Further work is needed to investigate different objective functions (i.e., validity indices) in multiobjective optimization for automatic clustering. Moreover, the selection of the best single solution from the Pareto-optimal front is an important open problem.

- Novel nature-inspired metaheuristics are continuously proposed in the literature as potential approaches to solving the clustering problem. Techniques such as bacterial foraging optimization [147], firefly optimization [148], gravitational search algorithm [149], etc., have been applied in nonautomatic clustering analysis; however, they could be extended to solve the automatic clustering problem. Table 5 lists some nature-inspired metaheuristics that could be further applied in automatic clustering.

- Hybrid nature-inspired algorithms combine the advantages of two or more metaheuristics and, in general, provide better solutions than an individual process [124]. However, only 8 % of all reviewed approaches are hybrid techniques. Therefore, we consider it convenient to develop other hybrid approaches to efficiently solve complex clustering problems.

- Although distinct synthetic and real-life data sets have been used systematically in the literature, there is no recognized benchmark that the research community could use to evaluate and compare automatic clustering approaches. This benchmark should include diverse cluster properties, such as shape, size, dimensionality, density, noise, linear and non-linear separability, etc.

8. Conclusion

This paper provides an up-to-date review of nature-inspired metaheuristic algorithms for automatic clustering.

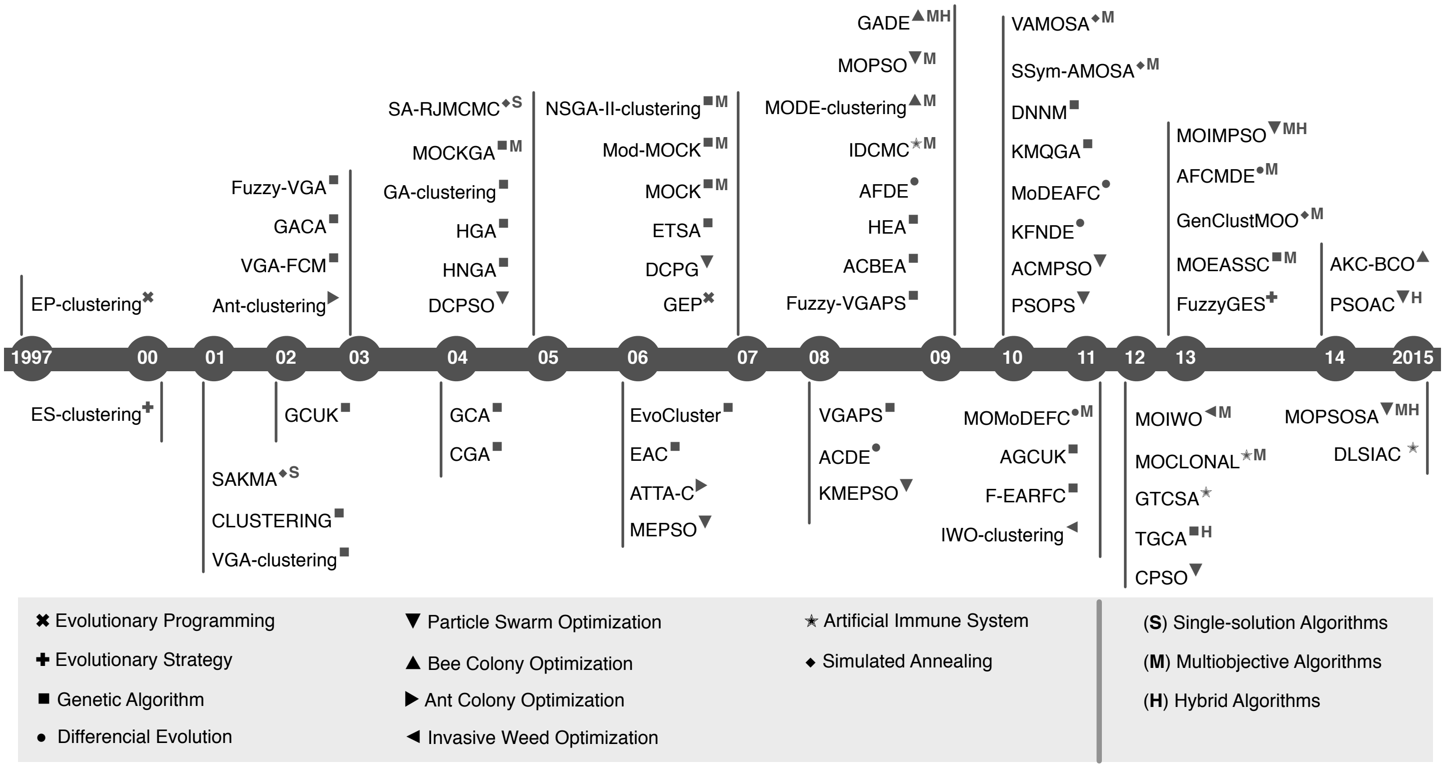
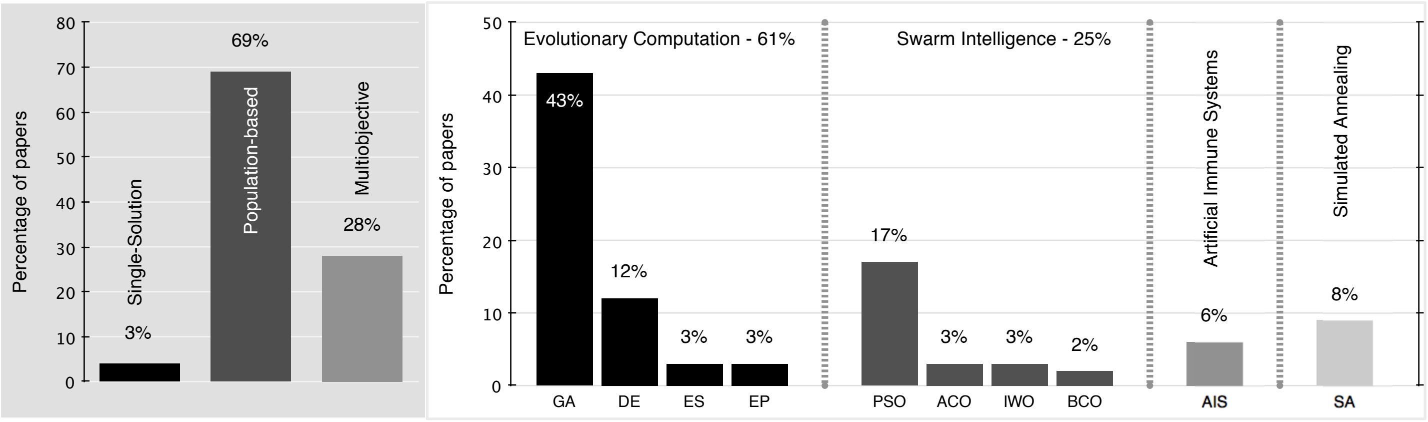


Figure 5: Timeline of automatic clustering algorithms based on nature-inspired metaheuristics.



(a)

(b)

Figure 6: Percentages of papers reviewed herein: (a) optimization scheme and (b) metaheuristic technique.

Table 5: Some nature-inspired metaheuristics that can be applied to automatic clustering.

Metaheuristic Name	Authors	Nature Inspiration	Multiobjective Formulation	Non-Automatic Clustering
Bacterial Foraging Optimization (BFO)	Passino [150]	Imitates the foraging strategies of E. coli bacteria for finding food	Yes [151]	Wan et al. [147]
Fish Swarm Optimization (FSO)	Li et al. [152]	Imitates the fish behaviors such as praying, swarming and following behaviors	Yes [153]	Cheng et al. [154]
Shuffled Frog-Leaping Algorithm (SFL)	Eusuff et al. [155]	Inspired from the behavior of frogs seeking for food in a pond	Yes [156]	Amiri et al. [157]
Cat Swarm Optimization (CSO)	Chu et al. [158]	Imitates the natural hunting skills of cats such, they remain always alert and move very slowly	Yes [159]	Santosa et al. [160]
Firefly Optimization (FO)	Yang [161]	Imitates the foraging behavior of fireflies such as glow brighter to attract prey and to share food	Yes [148]	Senthilnath et al. [162]
Gravitational Search Algorithm (GSA)	Rashedi [163]	It is based on the law of gravity which follows the principles of Newton's laws of gravity	Yes [149]	Hatamlou et al. [164]
Bat Algorithm (BA)	Yang [165]	Inspired from the echolocation behavior of bats	Yes [166]	Sood et al. [167]
Krill Herd Algorithm (KHA)	Gandomi et al. [168]	Simulates the herding of krill swarms in response to specific biological and environmental processes	No	Singh et al. [169]
Black Hole Algorithm (BHA)	Hatamlou [170]	Inspired from the black hole phenomenon which refers to a great amount of matter concentrated into a very small area	No	Hatamlou [170]

The promising solutions to automatic clustering are helpful in that they do not require *a priori* information about the actual number of clusters present in the data set. It is important to keep in mind that the success of a nature-inspired metaheuristic for automatic clustering is highly dependent on how it has been designed, for instance, its encoding scheme, objective function (or validity index), proximity measure, etc. One can note that multiobjective algorithms that consider multiple clustering validity criteria are suitable over single-objective clustering algorithms because they provide the flexibility to select the desired solution from a set of optimal solutions. Also, multiobjective approaches are capable to efficiently deal with non-linearly separable clusters. Although multiobjective algorithms for automatic clustering have already been used, we consider this research topic to be currently underexplored in the literature.

The 65 approaches studied herein reveals the importance of automatic clustering in cluster analysis field. Despite all the authors reported successful performances of their respective algorithms in relation to their counterparts, currently, there is no an extensive comparative study to demonstrate the superiority of a specific clustering approach or nature-inspired metaheuristic. Such comparative study should contemplate at least a common benchmark data or application field, several representative automatic clustering algorithms, and different nature-inspired metaheuristics, just to name a few. Hence, the best way to solve the automatic clustering problem is still an open challenge.

9. Acknowledgements

The authors would like to thank the support from CONACyT Mexico through a scholarship to pursue doctoral studies at the Information Technology Laboratory, Cinvestav-Tamaulipas.

10. References

References

[1] S. Theodoridis, K. Koutrumbas, Pattern Recognition, 4th Edition, Elsevier Inc., 2009.

[2] E. Falkenauer, Genetic Algorithms and Grouping Problems, John Wiley and Sons Ltd, Chichester, England, 1998.

[3] M. C. Cowgill, R. J. Harvey, L. T. Watson, A Genetic Algorithm Approach to Cluster Analysis, Computers & Mathematics with Applications 37 (7) (1999) 99–108. doi:10.1016/S0898-1221(99)00090-5.

[4] A. K. Jain, Data Clustering: 50 Years Beyond K-means, Pattern Recognition Letters 31 (8) (2010) 651–666. doi:10.1016/j.patrec.2009.09.011.

[5] R. Xu, D. C. Wunsch, Survey of Clustering Algorithms, IEEE Transactions on Neural Networks 16 (3) (2005) 645–78. doi:10.1109/TNN.2005.845141.

[6] C. W. Bong, M. Rajeswari, Multi-Objective Nature-Inspired Clustering and Classification Techniques for Image Segmentation, Applied Soft Computing 11 (4) (2011) 3271–3282. doi:10.1016/j.asoc.2011.01.014.

[7] J. Handl, J. Knowles, An Evolutionary Approach to Multiobjective Clustering, IEEE Transactions on Evolutionary Computation 11 (1) (2007) 56–76. doi:10.1109/TEVC.2006.877146.

[8] S. Bandyopadhyay, U. Maulik, Genetic Clustering for Automatic Evolution of Clusters and Application to Image Classification, Pattern Recognition 35 (6) (2002) 1197–1208. doi:10.1016/S0031-3203(01)00108-X.

[9] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, A. C. P. L. de Carvalho, A Survey of Evolutionary Algorithms for Clustering, IEEE Transactions on Systems, Man, and Cybernetics, Part C 39 (2) (2009) 133–155. doi:10.1109/TSMCC.2008.2007252.

[10] A. P. Engelbrecht, Computational Intelligence: An Introduction, 2nd Edition, Jonhn Wiley and Sons Ltd, 2007.

[11] J. Handl, B. Meyer, Ant-based and Swarm-based Clustering, Swarm Intelligence 1 (2) (2007) 95–113. doi:10.1007/s11721-007-0008-7.

[12] R. H. Sheikh, M. M. Raghuwanshi, A. N. Jaiswal, Genetic Algorithm Based Clustering: A Survey, in: 2008 First International Conference on Emerging Trends in Engineering and Technology, Vol. 2, IEEE, Nagpur, India, 2008, pp. 314–319. doi:10.1109/ICETET.2008.48.

[13] S. Rana, S. Jasola, R. Kumar, A Review on Particle Swarm Optimization Algorithms and their Applications to Data Clustering, Artificial Intelligence Review 35 (3) (2010) 211–222. doi:10.1007/s10462-010-9191-9.

[14] S. J. Nanda, G. Panda, A Survey on Nature Inspired Metaheuristic Algorithms for Partitional Clustering, Swarm and Evolutionary Computation 16 (2014) 1–18. doi:10.1016/j.swevo.2013.11.003.

[15] S. Alam, G. Dobbie, Y. S. Koh, P. Riddle, S. Ur Rehman, Research on Particle Swarm Optimization based Clustering: A Systematic Review of Literature and Techniques, Swarm and Evolutionary Computation 17 (2014) 1–13. doi:10.1016/j.swevo.2014.02.001.

[16] R. Xu, D. C. Wunsch, Clustering, Oxford, Wiley, 2008.

[17] J. A. Hartigan, M. A. Wong, Algorithm AS 136: A K-Means Clustering Algorithm, Applied Statistics 28 (1) (1979) 100–108. doi:10.2307/2346830.

[18] J. MacQueen, Some Methods for Classification and Analysis of Multivariate Observations, in: Fifth Berkeley Symposium on Mathematics, Statistics and Probability, University of California Press, 1967, pp. 91–110.

[19] D. Pelleg, A. Moore, X-means: Extending K-means with Efficient Estimation of the Number of Cluster, in: Proceedings of the 17th International Conference on Machine Learning, 2000, pp. 277–281.

[20] G. Hamerly, C. Elkan, Learning the K in K-Means, in: Advances in Neural Information Processing Systems, MIT Cambridge Press, 2003, pp. 281–288.

[21] J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Springer US, Boston, MA, 1981. doi:10.1007/978-1-4757-0450-1.

[22] A. H. Fielding, Cluster and Classification Techniques for the Biosciences, Cambridge University Press, New York, USA, 2007.

[23] J. Hopcroft, R. Kannan, Clustering, in: Foundations of Data Science, Microsoft Research, 2014, Ch. 8, pp. 260–300.

[24] M. C. Su, C. H. Chou, A Modified Version of the K-Means Algorithm with a Distance Based on Cluster Symmetry, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 674–680. doi:10.1109/34.927466.

[25] S. Bandyopadhyay, U. Maulik, A. Mukhopadhyay, Multiobjective Genetic Clustering for Pixel Classification in Remote Sensing Imagery, IEEE Transactions on Geoscience and Remote Sensing 45 (5) (2007) 1506–1511. doi:10.1109/TGRS.2007.892604.

[26] X. L. Xie, G. Beni, A validity Measure for Fuzzy Clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (8) (1991) 841–847.

[27] M. Kim, R. S. Ramakrishna, New Indices for Cluster Validity Assessment, Pattern Recognition Letters 26 (15) (2005) 2353–2363. doi:10.1016/j.patrec.2005.04.007.

[28] J. C. Dunn, A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, Journal of Cybernetics 3 (3) (1973) 32–57. doi:10.1080/01969727308546046.

[29] J. C. Bezdek, N. R. Pal, Some New Indexes of Cluster Validity, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 28 (3) (1998) 301–15. doi:10.1109/3477.678624.

[30] T. Calinski, J. Harabasz, A Dendrite Method for Cluster Analysis, Communications in Statistics - Theory and Methods 3 (1) (1974) 1–27.

[31] D. L. Davies, D. W. Boulding, A Cluster Separation Measure, IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (2) (1979) 224–227.

[32] P. J. Rousseeuw, Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis, Journal of Computational and Applied Mathematics 20 (1987) 53–65. doi:10.1016/0377-0427(87)90125-7.

[33] R. H. Turi, Clustering-based Colour Image Segmentation, Ph.D. thesis, Monash University (2001).
URL http://www.csse.monash.edu.au/~roset/thesis/thesis.pdf

[34] U. Maulik, S. Bandyopadhyay, Performance Evaluation of Some Clustering Algorithms and Validity Indices, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (12) (2002) 1650–1654.

[35] C. H. Chou, M. C. Su, E. Lai, A New Cluster Validity Measure and its Application to Image Compression, Pattern Analysis and Applications 7 (2) (2004) 205–220. doi:10.1007/s10044-004-0218-1.

[36] S. Bandyopadhyay, S. Saha, A Point Symmetry-Based Clustering Technique for Automatic Evolution of Clusters, IEEE Transactions on Knowledge and Data Engineering 20 (11) (2008) 1441–1457. doi:10.1109/TKDE.2008.79.

[37] P. C. H. Ma, K. C. C. Chan, X. Yao, D. K. Y. Chiu, An Evolutionary Clustering Algorithm for Gene Expression Microarray Data Analysis, IEEE Transactions on Evolutionary Computation 10 (3) (2006) 296–314. doi:10.1109/TEVC.2005.859371.

[38] A. Ferligoj, V. Batagelj, Direct Multicriteria Clustering Algorithms, Journal of Classification 9 (1) (1992) 43–61. doi:10.1007/BF02618467.

[39] K. Deb, Multi-Objective Optimization using Evolutionary Algorithms, John Wiley and Sons Ltd, 2009.

[40] D. Whitley, A Genetic Algorithm Tutorial, Statistics and Computing 4 (1994) 65–85.

[41] E. G. Talbi, Metaheuristics from Design to Implementation, John Wiley and Sons, 2009.

[42] N. R. Pal, J. C. Bezdek, On Cluster Validity for the Fuzzy C-Means Model, IEEE Transactions on Fuzzy Systems 3 (3) (1995) 370–379. doi:10.1109/91.413225.

[43] S. Das, A. Abraham, A. Konar, Automatic Clustering using an Improved Differential Evolution Algorithm, Transactions on Systems, Man and Cybernetics 38 (1) (2008) 218–237. doi:10.1109/TSMCA.2007.909595.

[44] S. Bandyopadhyay, Simulated Annealing Using a Reversible Jump Markov Chain Monte Carlo Algorithm for Fuzzy Clustering, IEEE Transactions on Knowledge and Data Engineering 17 (4) (2005) 479–490. doi:10.1109/TKDE.2005.64.

[45] R. Liu, L. Jiao, X. Zhang, Y. Li, Gene Transposon Based Clone Selection Algorithm for Automatic Clustering, Information Sciences 204 (2) (2012) 1–22. doi:10.1016/j.ins.2012.03.021.

[46] H. He, Y. Tan, A Two-Stage Genetic Algorithm for Automatic Clustering, Neurocomputing 81 (2012) 49–59. doi:10.1016/j.neucom.2011.11.001.

[47] S. Das, A. Konar, Automatic Image Pixel Clustering with an Improved Differential Evolution, Applied Soft Computing 9 (1) (2009) 226–236. doi:10.1016/j.asoc.2007.12.008.

[48] S. Das, S. Sil, Kernel-Induced Fuzzy Clustering of Im-

age Pixels with an Improved Differential Evolution Algorithm, Information Sciences 180 (8) (2010) 1237–1256. doi:10.1016/j.ins.2009.11.041.

[49] C. C. Lai, C. Y. Chang, A Hierarchical Evolutionary Algorithm for Automatic Medical Image Segmentation, Expert Systems with Applications 36 (1) (2009) 248–259. doi:10.1016/j.eswa.2007.09.003.

[50] S. Saha, S. Bandyopadhyay, A New Line Symmetry Distance and Its Application to Data Clustering, Journal of Computer Science and Technology 24 (3) (2009) 544–556. doi:10.1007/s11390-009-9244-1.

[51] S. Bandyopadhyay, S. K. Pal, Pixel Classification Using Variable String Genetic Algorithms with Chromosome Differentiation, IEEE Transactions on Geoscience and Remote Sensing 39 (2) (2001) 303–308. doi:10.1109/36.905238.

[52] U. Maulik, S. Bandyopadhyay, Fuzzy Partitioning Using a Real-Coded Variable-Length Genetic Algorithm for Pixel Classification, IEEE Transactions on Geoscience and Remote Sensing 41 (5) (2003) 1075–1081. doi:10.1109/TGRS.2003.810924.

[53] D. X. Chang, X. D. Zhang, C. W. Zheng, D. M. Zhang, A Robust Dynamic Niching Genetic Algorithm With Niche Migration for Automatic Clustering Problem, Pattern Recognition 43 (4) (2010) 1346–1360. doi:10.1016/j.patcog.2009.10.020.

[54] U. Maulik, I. Saha, Automatic Fuzzy Clustering Using Modified Differential Evolution for Image Classification, IEEE Transactions on Geoscience and Remote Sensing 48 (9) (2010) 3503–3510. doi:10.1109/TGRS.2010.2047020.

[55] Y. Zhong, S. Zhang, L. Zhang, Automatic Fuzzy Clustering Based on Adaptive Multi-Objective Differential Evolution for Remote Sensing Imagery, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 6 (99) (2013) 1–12. doi:10.1109/JSTARS.2013.2240655.

[56] E. R. Hruschka, R. J. G. B. Campello, L. N. de Castro, Evolving Clusters in Gene-Expression Data, Information Sciences 176 (13) (2006) 1898–1927. doi:10.1016/j.ins.2005.07.015.

[57] S. Saha, S. Bandyopadhyay, A New Point Symmetry Based Fuzzy Genetic Clustering Technique for Automatic Evolution of Clusters, Information Sciences 179 (19) (2009) 3230–3246. doi:10.1016/j.ins.2009.06.013.

[58] K. Suresh, D. Kundu, S. Ghosh, S. Das, A. Abraham, S. Y. Han, Multi-Objective Differential Evolution for Automatic Clustering with Application to Micro-array Data Analysis, Sensors 9 (5) (2009) 3981–4004. doi:10.3390/s90503981.

[59] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, Optimization by Simulated Annealing, Science 220 (4598) (1983) 671–680. doi:10.2307/1690046.

[60] S. P. Brooks, B. J. T. Morgan, Optimization Using Simulated Annealing, The Statistician 44 (2) (1995) 241–257. doi:10.2307/2348448.

[61] S. Bandyopadhyay, U. Maulik, M. K. Pakhira, Clustering Using Simulated Annealing with Probabilistic Redistribution, International Journal of Pattern Recognition and Artificial Intelligence 15 (02) (2001) 269–285. doi:10.1142/S0218001401000927.

[62] F. Glover, Tabu Search: A Tutorial, Interfaces 20 (4) (1990) 74–94. doi:10.1287/inte.20.4.74.

[63] F. Glover, Tabu Search - Part I, ORSA Journal on Computing 1 (3) (1989) 190–206. doi:10.1287/ijoc.1.3.190.

[64] F. Glover, Tabu Search - Part II, ORSA Journal on Computing 2 (1) (1990) 4–32. doi:10.1287/ijoc.2.1.4.

[65] S. M. Pan, K. S. Cheng, Evolution-Based Tabu Search Approach to Automatic Clustering, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 37 (5) (2007) 827–838. doi:10.1109/TSMCC.2007.900666.

[66] S. Z. Selim, K. Alsultan, A Simulated Annealing Algorithm for the Clustering Problem, Pattern Recognition 24 (10) (1991) 1003–1008. doi:10.1016/0031-3203(91)90097-O.

[67] D. E. Brown, C. L. Huntley, A Practical Application of Simulated Annealing to Clustering, Pattern Recognition 25 (4) (1992) 401–412. doi:10.1016/0031-3203(92)90088-Z.

[68] S. Saha, S. Bandyopadhyay, A New Multiobjective Simulated Annealing Based Clustering Technique Using Symmetry, Pattern Recognition Letters 30 (15) (2009) 1392–1403. doi:10.1016/j.patrec.2009.07.015.

[69] L. J. Fogel, A. J. Owens, M. J. Walsh, Artificial Intelligence through Simulated Evolution, John Wiley and Sons, 1966.

[70] M. Sarkar, B. Yegnanarayana, D. Khemani, A Clustering Algorithm Using an Evolutionary Programming-Based Approach, Pattern Recognition Letters 18 (10) (1997) 975–986. doi:10.1016/S0167-8655(97)00122-0.

[71] Y. Chen, C. Tang, J. Zhu, C. Li, S. Qiao, R. Li, J. Wu, Clustering without Prior Knowledge Based on Gene Expression Programming, in: Third International Conference on Natural Computation (ICNC 2007), IEEE, Haikou, China, 2007, pp. 451–455. doi:10.1109/ICNC.2007.302.

[72] H. G. Beyer, H. P. Schwefel, Evolution Strategies - A Comprehensive Introduction, Natural Computing 1 (1) (2002) 3–52.

[73] C. Y. Lee, E. K. Antonsson, Dynamic Partitional Clustering Using Evolution Strategies, in: 26th Annual Conference of the IEEE Industrial Electronics Society. IECON 2000., Vol. 4, IEEE, Nagoya, Japan, 2000, pp. 2716–2721. doi:10.1109/IECON.2000.972427.

[74] H. Kashan, B. Rezaee, S. Karimiyan, An Efficient Approach for Unsupervised Fuzzy Clustering based on Grouping Evolution Strategies, Pattern Recognition 46 (5) (2013) 1240–1254. doi:10.1016/j.patcog.2012.11.010.

[75] J. H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Michigan, USA, 1975.

[76] L. Y. Tseng, S. B. Yang, A Genetic Approach to the Automatic Clustering Problem, Pattern Recognition 34 (2) (2001) 415–424. doi:10.1016/S0031-3203(00)00005-4.

[77] S. Bandyopadhyay, U. Maulik, Nonparametric Genetic Clustering: Comparison of Validity Indices, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 31 (1) (2001) 120–125. doi:10.1109/5326.923275.

[78] W. Sheng, S. Swift, L. Zhang, X. Liu, A Weighted Sum Validity Function for Clustering With a Hybrid Niching Genetic Algorithm, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 35 (6) (2005) 1156–1167. doi:10.1109/TSMCB.2005.850173.

[79] E. R. Hruschka, N. F. F. Ebecken, A Genetic Algorithm for Cluster Analysis, Intelligent Data Analysis 7 (1) (2003) 15–25.

[80] J. Xiao, Y. Yan, J. Zhang, Y. Tang, A Quantum-Inspired Genetic Algorithm for K-Means Clustering, Expert Systems with Applications 37 (7) (2010) 4966–4973. doi:10.1016/j.eswa.2009.12.017.

[81] D. Horta, I. C. de Andrade, R. J. G. B. Campello, Evolutionary Fuzzy Clustering of Relational Data, Theoretical Computer Science 412 (42) (2011) 5854–5870. doi:10.1016/j.tcs.2011.05.039.

[82] Y. Liu, X. Wu, Y. Shen, Automatic Clustering Using Genetic Algorithms, Applied Mathematics and Computation 218 (4) (2011) 1267–1279. doi:10.1016/j.amc.2011.06.007.

[83] C. C. Lai, A Novel Clustering Approach using Hierarchical Genetic Algorithms, Intelligent Automation & Soft Computing 11 (3) (2005) 143–153. doi:10.1080/10798587.2005.10642900.

[84] H. J. Lin, F. W. Yang, Y. T. Kao, An Efficient GA-based Clustering Technique, Tamkang Journal of Science and Engineering 8 (2) (2005) 113–122.

[85] H. Liu, J. Li, M. A. Chapman, Automated Road Extraction from Satellite Imagery Using Hybrid Genetic Algorithms and Cluster Analysis, Journal of Environmental Informatics 1 (2) (2003) 40–47.

[86] E. R. Hruschka, R. J. G. B. Campello, L. N. de Castro, Improving the Efficiency of a Clustering Genetic Algorithm, in: Advances in Artificial Intelligence - IBERAMIA 2004, Springer Berlin Heidelberg, Puebla, Mexico, 2004, Ch. Genetic Al, pp. 861–870. doi:10.1007/978-3-540-30498-2_86.

[87] S. Das, A. Chowdhury, A. Abraham, A Bacterial Evolutionary Algorithm for Automatic Data Clustering, in: 2009 IEEE Congress on Evolutionary Computation, IEEE, Trondheim, Norway, 2009, pp. 2403–2410. doi:10.1109/CEC.2009.4983241.

[88] M. Omran, A. P. Engelbrecht, A. Salman, Dynamic Clustering Using Particle Swarm Optimization with Application in

Unsupervised Image Segmentation, International Journal of Computer, Information Science and Engineering 1 (9) (2007) 232–237.

[89] G. Garai, B. B. Chaudhuri, A Novel Genetic Algorithm for Automatic Clustering, Pattern Recognition Letters 25 (2) (2004) 173–187. doi:10.1016/j.patrec.2003.09.012.

[90] R. Storn, K. Price, Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization 11 (4) (1997) 341–359.

[91] S. Das, P. N. Suganthan, Differential Evolution: A Survey of the State-of-the-Art, IEEE Transactions on Evolutionary Computation 15 (1) (2011) 4–31. doi:10.1109/TEVC.2010.2059031.

[92] S. Das, A. Abraham, A. Konar, Metaheuristic Clustering, Vol. 178 of Studies in Computational Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. doi:10.1007/978-3-540-93964-1.

[93] A. P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, John Wiley and Sons Ltd, 2005.

[94] J. Kennedy, R. Eberhart, Particle Swarm Optimization, in: International Conference on Neural Networks, Vol. 4, IEEE, Perth, Australia, 1995, pp. 1942–1948. doi:10.1109/ICNN.1995.488968.

[95] M. Omran, A. P. Engelbrecht, A. Salman, Particle Swarm Optimization Method for Image Clustering, International Journal of Pattern Recognition and Artificial Intelligence 19 (03) (2005) 297–321. doi:10.1142/S0218001405004083.

[96] M. Omran, A. Salman, A. P. Engelbrecht, Dynamic Clustering using Particle Swarm Optimization with Application in Image Segmentation, Pattern Analysis and Applications 8 (4) (2005) 332–344. doi:10.1007/s10044-005-0015-5.

[97] S. Das, A. Abraham, A. Konar, Spatial Information Based Image Segmentation Using a Modified Particle Swarm Optimization Algorithm, in: Sixth International Conference on Intelligent Systems Design and Applications, Vol. 2, IEEE, Jinan, China, 2006, pp. 438–444. doi:10.1109/ISDA.2006.253877.

[98] S. Das, A. Abraham, A. Konar, Automatic Kernel Clustering with a Multi-Elitist Particle Swarm Optimization Algorithm, Pattern Recognition Letters 29 (5) (2008) 688–699. doi:10.1016/j.patrec.2007.12.002.

[99] J. Qu, Z. Shao, X. Liu, Mixed PSO Clustering Algorithm Using Point Symmetry Distance, Journal of Computational Information Systems 6 (6) (2010) 2027–2035.

[100] S. Ouadfel, M. Batouche, A. Taleb-Ahmed, A Modified Particle Swarm Optimization Algorithm for Automatic Image Clustering, in: Proceedings of the 1st International Symposium on Modeling and Implementing Complex Systems, Constantine, Algeria, 2010.

[101] T. Cura, A Particle Swarm Optimization Approach to Clustering, Expert Systems with Applications 39 (1) (2012) 1582–1588. doi:10.1016/j.eswa.2011.07.123.

[102] R. J. Kuo, Y. J. Syu, Z.-Y. Chen, F. C. Tien, Integration of Particle Swarm Optimization and Genetic Algorithm for Dynamic Clustering, Information Sciences 195 (2012) 124–140. doi:10.1016/j.ins.2012.01.021.

[103] Y. Kao, C.-C. Chen, Automatic Clustering for Generalised Cell Formation Using a Hybrid Particle Swarm Optimisation, International Journal of Production Research 52 (12) (2014) 3466–3484. doi:10.1080/00207543.2013.867085.

[104] M. Dorigo, V. Maniezzo, A. Colorni, Ant System: Optimization by a Colony of Cooperating Agents, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 26 (1) (1996) 29–41. doi:10.1109/3477.484436.

[105] M. Dorigo, C. Blum, Ant Colony Optimization Theory: A Survey, Theoretical Computer Science 344 (2-3) (2005) 243–278. doi:10.1016/j.tcs.2005.05.020.

[106] B. Chandra Mohan, R. Baskaran, A Survey: Ant Colony Optimization based Recent Research and Implementation on Several Engineering Domain, Expert Systems with Applications 39 (4) (2012) 4618–4627. doi:10.1016/j.eswa.2011.09.076.

[107] P. M. Kanade, L. O. Hall, Fuzzy Ants as a Clustering Concept, in: 22nd International Conference of the North American Fuzzy Information Processing Society, NAFIPS 2003, IEEE, 2003, pp. 227–232. doi:10.1109/NAFIPS.2003.1226787.

[108] J. Handl, J. Knowles, M. Dorigo, Ant-based Clustering and Topographic Mapping, Artificial life 12 (1) (2006) 35–61. doi:10.1162/106454606775186400.

[109] A. R. Mehrabian, C. Lucas, A Novel Numerical Optimization Algorithm Inspired from Weed Colonization, Ecological Informatics 1 (4) (2006) 355–366. doi:10.1016/j.ecoinf.2006.07.003.

[110] R. Razavi-Far, V. Palade, E. Zio, Invasive Weed Classification, Neural Computing and Applicationsdoi:10.1007/s00521-014-1656-3.

[111] A. Chowdhury, S. Bose, S. Das, Automatic Clustering Based on Invasive Weed Optimization Algorithm, in: Swarm, Evolutionary, and Memetic Computing, Springer Berlin Heidelberg, Pradesh, India, 2011, pp. 105–112. doi:10.1007/978-3-642-27242-4_13.

[112] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Tech. rep., Erciyes University, Kayseri, Turkey (2005).

[113] D. Karaboga, B. Basturk, On the Performance of Artificial Bee Colony (ABC) Algorithm, Applied Soft Computing 8 (1) (2008) 687–697. doi:10.1016/j.asoc.2007.05.007.

[114] D. Karaboga, B. Akay, A Comparative Study of Artificial Bee Colony algorithm, Applied Mathematics and Computation 214 (1) (2009) 108–132. doi:10.1016/j.amc.2009.03.090.

[115] C. Zhang, D. Ouyang, J. Ning, An Artificial Bee Colony Approach for Clustering, Expert Systems with Applications 37 (7) (2010) 4761–4767. doi:10.1016/j.eswa.2009.11.003.

[116] W. Zou, Y. Zhu, H. Chen, X. Sui, A Clustering Approach Using Cooperative Artificial Bee Colony Algorithm, Discrete Dynamics in Nature and Society 2010 (2010) 1–16. doi:10.1155/2010/459796.

[117] D. Karaboga, C. Ozturk, A novel clustering approach: Artificial Bee Colony (ABC) algorithm, Applied Soft Computing 11 (1) (2011) 652–657. doi:10.1016/j.asoc.2009.12.025.

[118] R. J. Kuo, Y. D. Huang, C.-C. Lin, Y.-H. Wu, F. E. Zulvia, Automatic Kernel Clustering with Bee Colony Optimization Algorithm, Information Sciences 283 (2014) 107–122. doi:10.1016/j.ins.2014.06.019.

[119] J. Timmis, A. Hone, T. Stibor, E. Clark, Theoretical Advances in Artificial Immune Systems, Theoretical Computer Science 403 (1) (2008) 11–32. doi:10.1016/j.tcs.2008.02.011.

[120] R. Liu, X. Zhang, N. Yang, Q. Lei, L. Jiao, Immunodomaince based Clonal Selection Clustering Algorithm, Applied Soft Computing 12 (1) (2012) 302–312. doi:10.1016/j.asoc.2011.08.042.

[121] S. Gou, X. Zhuang, Y. Li, C. Xu, L. Jiao, Multi-elitist Immune Clonal Quantum Clustering Algorithm, Neurocomputing 101 (2013) 275–289. doi:10.1016/j.neucom.2012.08.022.

[122] R. Liu, B. Zhu, R. Bian, Y. Ma, L. Jiao, Dynamic Local Search based Immune Automatic Clustering Algorithm and its Applications, Applied Soft Computing 27 (2015) 250–268. doi:10.1016/j.asoc.2014.11.026.

[123] C. A. Coello Coello, List of References on Evolutionary Multiobjective Optimization (2010).
URL http://www.lania.mx/~ccoello/EMOO/EMOObib.html

[124] S. J. Nanda, G. Panda, Automatic Clustering Algorithm Based on Multi-Objective Immunized PSO to Classify Actions of 3D Human Models, Engineering Applications of Artificial Intelligence 26 (5-6) (2013) 1429–1441. doi:10.1016/j.engappai.2012.11.008.

[125] S. J. Nanda, G. Panda, Automatic Clustering Using MOCLONAL for Classifying Actions of 3D Human Models, in: Symposium on Humanities, Science and Engineering Research, IEEE, Kuala Lumpur, 2012, pp. 945–950. doi:10.1109/SHUSER.2012.6269011.

[126] U. Maulik, S. Bandyopadhyay, A. Mukhopadhyay, Multiobjective Genetic Algorithms for Clustering, Springer Berlin Heidelberg, 2011. doi:10.1007/978-3-642-16615-0.

[127] J. Handl, J. Knowles, Evolutionary Multiobjective Clustering, in: Proceedings of the 8th International Conference on Parallel Problem., Springer Berlin Heidelberg, Birmingham, UK., 2004, pp. 1081–1091. doi:10.1007/978-3-540-30217-9_109.

[128] N. Matake, T. Hiroyasu, M. Miki, T. Senda, Multiobjective Clustering with Automatic K-Determination for Large-Scale Data, in: Proceedings of the 9th annual conference on Genetic and evolutionary computation - GECCO '07, ACM Press, New York, USA, 2007, pp. 861–868. doi:10.1145/1276958.1277126.

[129] Y. Liu, T. Ozyer, R. Alhajj, K. Barker, Integrating Multi-Objective Genetic Algorithm and Validity Analysis for Locating and Ranking Alternative Clustering, Informatica 29 (1) (2005) 33–40.

[130] H. Xia, J. Zhuang, D. Yu, Novel Soft Subspace Clustering with Multi-Objective Evolutionary Approach for High-Dimensional Data, Pattern Recognition 46 (9) (2013) 2562–2575. doi:10.1016/j.patcog.2013.02.005.

[131] C. A. Coello Coello, A. Hernández Aguirre, E. Zitzler, DEMO: Differential Evolution for Multiobjective Optimization, in: Evolutionary Multi-Criterion Optimization, Springer Berlin Heidelberg, 2005, pp. 520–533. doi:10.1007/978-3-540-31880-4_36.

[132] K. Suresh, D. Kundu, S. Ghosh, S. Das, A. Abraham, Automatic Clustering with Multi-Objective Differential Evolution Algorithms, in: 2009 IEEE Congress on Evolutionary Computation, IEEE, 2009, pp. 2590–2597. doi:10.1109/CEC.2009.4983267.

[133] I. Saha, U. Maulik, D. Plewczynski, A New Multi-Objective Technique for Differential Fuzzy Clustering, Applied Soft Computing 11 (2) (2011) 2765–2776. doi:10.1016/j.asoc.2010.11.007.

[134] D. Kundu, K. Suresh, S. Ghosh, Automatic Clustering Using a Synergy of Genetic Algorithm and Multi-objective Differential Evolution, in: Hybrid Artificial Intelligence Systems, Springer Berlin Heidelberg, Salamanca, Spain, 2009, pp. 177–186. doi:10.1007/978-3-642-02319-4_21.

[135] N. Nedjah, L. S. Coelho, L. M. Mourelle, Multi-Objective Swarm Intelligent Systems, Springer Berlin Heidelberg, 2010.

[136] C. A. Coello Coello, S. Dehuri, S. Ghosh, Swarm Intelligence for Multi-objective Problems in Data Mining, Springer Berlin Heidelberg, 2010.

[137] A. Paoli, F. Melgani, E. Pasolli, Clustering of Hyperspectral Images Based on Multiobjective Particle Swarm Optimization, IEEE Transactions on Geoscience and Remote Sensing 47 (12) (2009) 4175–4188. doi:10.1109/TGRS.2009.2023666.

[138] A. Abubaker, A. Baharum, M. Alrefaei, Automatic Clustering Using Multi-objective Particle Swarm and Simulated Annealing, Plos One 10 (7) (2015) 1–23. doi:10.1371/journal.pone.0130995.

[139] S. Saha, S. Bandyopadhyay, A Generalized Automatic Clustering Algorithm in a Multiobjective Framework, Applied Soft Computing 13 (1) (2013) 89–108. doi:10.1016/j.asoc.2012.08.005.

[140] D. Kundu, K. Suresh, S. Ghosh, S. Das, B. K. Panigrahi, S. Das, Multi-objective Optimization with Artificial Weed Colonies, Information Sciences 181 (12) (2011) 2441–2454. doi:10.1016/j.ins.2010.09.026.

[141] R. Liu, X. Wang, X. Li, Y. Zhang, Multi-Objective Invasive Weed Optimization Algortihm for Clustering, in: 2012 IEEE Congress on Evolutionary Computation, IEEE, Brisbane, Australia, 2012, pp. 1–8. doi:10.1109/CEC.2012.6256540.

[142] J. Yoo, P. Hajela, Immune Network Simulations in Multicriterion Design, Structural Optimization 18 (2-3) (1999) 85–94. doi:10.1007/BF01195983.

[143] W. Ma, L. Jiao, M. Gong, Immunodominance and Clonal Selection Inspired Multiobjective Clustering, Progress in Natural Science 19 (6) (2009) 751–758. doi:10.1016/j.pnsc.2008.08.004.

[144] S. Bandyopadhyay, S. Saha, U. Maulik, K. Deb, A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA, IEEE Transactions on Evolutionary Computation 12 (3) (2008) 269–283. doi:10.1109/TEVC.2007.900837.

[145] S. Saha, S. Bandyopadhyay, A New Multiobjective Clustering Technique Based on the Concepts of Stability and Symmetry, Knowledge and Information Systems 23 (1) (2009) 1–27. doi:10.1007/s10115-009-0204-4.

[146] S. Saha, S. Bandyopadhyay, A Symmetry Based Multiobjective Clustering Technique for Automatic Evolution of Clusters, Pattern Recognition 43 (3) (2010) 738–751. doi:10.1016/j.patcog.2009.07.004.

[147] M. Wan, L. Li, J. Xiao, C. Wang, Y. Yang, Data Clustering using Bacterial Foraging Optimization, Journal of Intelligent Information Systems 38 (2) (2011) 321–341. doi:10.1007/s10844-011-0158-3.

[148] X. S. Yang, Multiobjective Firefly Algorithm for Continuous Optimization, Engineering with Computers 29 (2) (2012) 175–184. doi:10.1007/s00366-012-0254-1.

[149] H. R. Hassanzadeh, M. Rouhani, A Multi-Objective Gravitational Search Algorithm, in: 2nd International Conference on Computational Intelligence, Communication Systems and Networks, IEEE, Liverpool, UK, 2010, pp. 7–12. doi:10.1109/CICSyN.2010.32.

[150] K. Passino, Biomimicry of Bacterial Foraging for Distributed Optimization and Control, IEEE Control Systems Magazine 22 (3) (2002) 52–67. doi:10.1109/MCS.2002.1004010.

[151] B. Niu, H. Wang, L. Tan, J. Xu, Multi-Objective Optimization Using BFO Algorithm, in: 7th International Conference on Intelligent Computing, Springer Berlin Heidelberg, Zhengzhou, China, 2011, pp. 582–587.

[152] X. L. Li, Z. J. Shao, J. X. Qian, An Optimizing Method Based on Autonomous Animates: Fish-Swarm Algorithm, System Engineering Theory and Practice 22 (11) (2003) 32–38.

[153] M. Jiang, K. Zhu, Multiobjective Optimization by Artificial Fish Swarm Algorithm, in: 2011 IEEE International Conference on Computer Science and Automation Engineering, IEEE, Shanghai, China, 2011, pp. 506–511. doi:10.1109/CSAE.2011.5952729.

[154] Y. Cheng, M. Jiang, D. Yuan, Novel Clustering Algorithms Based on Improved Artificial Fish Swarm Algorithm, in: 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, IEEE, Tianjin, China, 2009, pp. 141–145. doi:10.1109/FSKD.2009.534.

[155] M. Eusuff, K. Lansey, F. Pasha, Shuffled Frog-Leaping Algorithm: A Memetic Meta-Heuristic for Discrete Optimization, Engineering Optimization 38 (2) (2006) 129–154. doi:10.1080/03052150500384759.

[156] J. Liu, Z. Li, X. Hu, Y. Chen, F. Liu, Multi-Objective Dynamic Population Shuffled Frog-Leaping Biclustering of Microarray Data, BMC genomics 13 (3) (2012) 1–11. doi:10.1186/1471-2164-13-S3-S6.

[157] B. Amiri, M. Fathian, A. Maroosi, Application of Shuffled Frog-Leaping Algorithm on Clustering, The International Journal of Advanced Manufacturing Technology 45 (1-2) (2009) 199–209. doi:10.1007/s00170-009-1958-2.

[158] S. C. Chu, P. W. Tsai, Computational Intelligence Based on the Behavior of Cats, International Journal of Innovative Computing, Information and Control 3 (1) (2007) 163–173.

[159] P. M. Pradhan, G. Panda, Solving Multiobjective Problems using Cat Swarm Optimization, Expert Systems with Applications 39 (3) (2012) 2956–2964. doi:10.1016/j.eswa.2011.08.157.

[160] B. Santosa, M. K. Ningrum, Cat Swarm Optimization for Clustering, in: 2009 International Conference of Soft Computing and Pattern Recognition, IEEE, Malacca, Malaysia, 2009, pp. 54–59. doi:10.1109/SoCPaR.2009.23.

[161] X. S. Yang, Nature-Inspired Metaheuristics Algorithms, Luniver Press, United Kingdom, 2010.

[162] J. Senthilnath, S. N. Omkar, V. Mani, Clustering using Firefly Algorithm: Performance Study, Swarm and Evolutionary Computation 1 (3) (2011) 164–171. doi:10.1016/j.swevo.2011.06.003.

[163] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A Gravitational Search Algorithm, Information Sciences 179 (13) (2009) 2232–2248. doi:10.1016/j.ins.2009.03.004.

[164] A. Hatamlou, S. Abdullah, H. Nezamabadi-pour, Application of Gravitational Search Algorithm on Data Clustering, in: 6th International Conference in Rough Sets and Knowledge Technology, Springer Berlin Heidelberg, Banff, Canada, 2011, pp. 337–346. doi:10.1007/978-3-642-24425-4_44.

[165] X. S. Yang, A. H. Gandomi, Bat Algorithm: A Novel Approach for Global Engineering Optimization, Engineering Computations 29 (5) (2012) 464–483. doi:10.1108/02644401211235834.

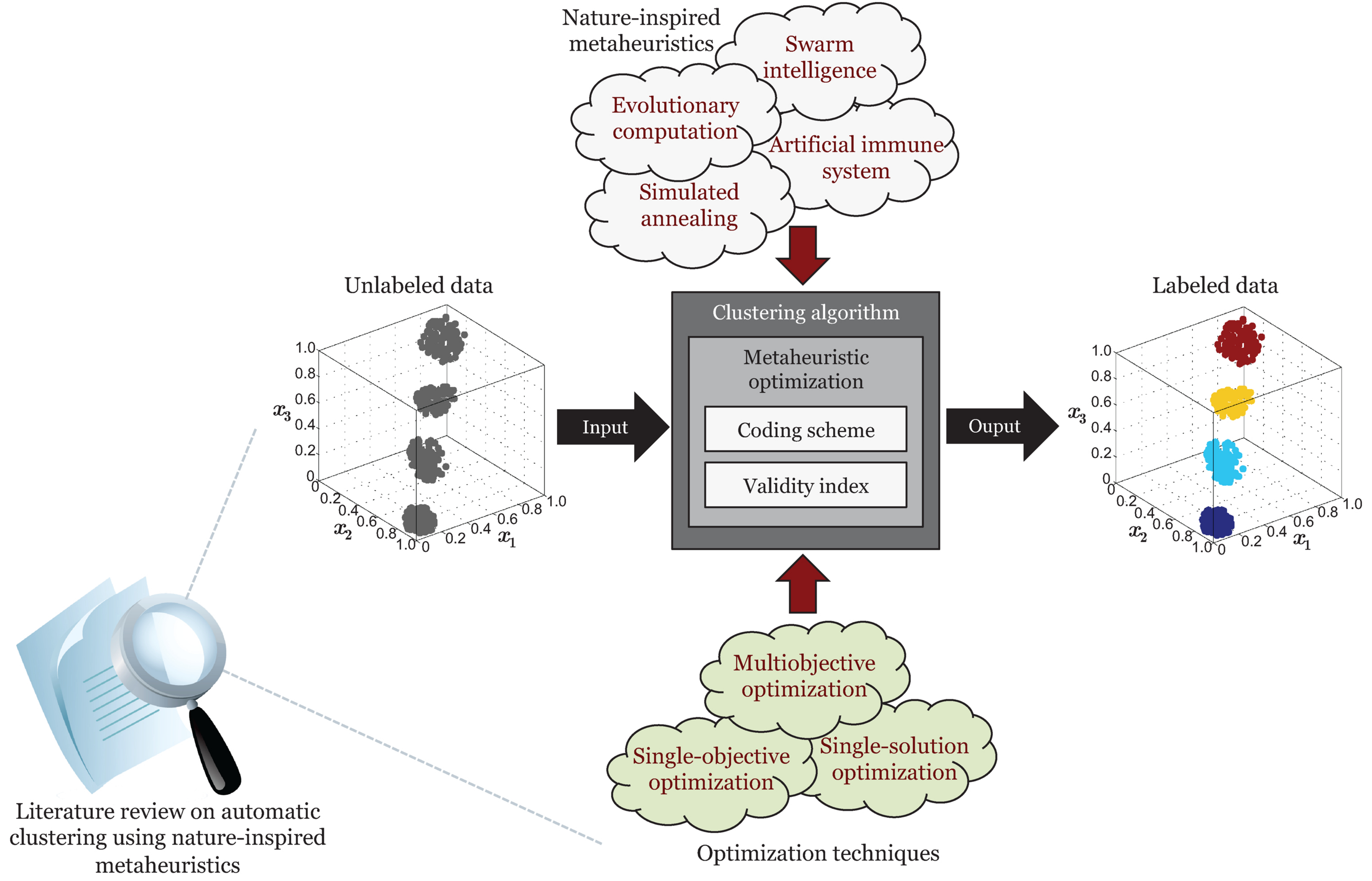
[166] X. S. Yang, Bat Algorithm for Multi-objective Optimisation, International Journal of Bio-Inspired Computation 3 (5) (2011) 267–274.

[167] M. Sood, S. Bansal, K-Medoids Clustering Technique using Bat Algorithm, International Journal of Applied Information Systems 5 (8) (2013) 20–22. doi:10.5120/ijais13-450965.

[168] A. H. Gandomi, A. H. Alavi, Krill Herd: A New Bio-Inspired Optimization Algorithm, Communications in Nonlinear Science and Numerical Simulation 17 (12) (2012) 4831–4845. doi:10.1016/j.cnsns.2012.05.010.

[169] V. Singh, M. Sood, Krill Herd Clustering Algorithm using DBSCAN Technique, International Journal of Computer Science & Engineering Technology 4 (3) (2013) 197–201.

[170] A. Hatamlou, Black Hole: A New Heuristic Optimization Approach for Data Clustering, Information Sciences 222 (2013) 175–184. doi:10.1016/j.ins.2012.08.023.


Nature-inspired metaheuristics
Swarm intelligence
Evolutionary computation
Artificial immune system
Simulated annealing
Unlabeled data
Clustering algorithm
Metaheuristic optimization
Coding scheme
Validity index
Labeled data
Input
Ouput
x_1
x_2
x_3
0
0.2
0.4
0.6
0.8
1.0
Multiobjective optimization
Single-objective optimization
Single-solution optimization
Optimization techniques
Literature review on automatic clustering using nature-inspired metaheuristics