Abstract: The clustering algorithms have evolved over the last decade. With the continuous success of natural inspired algorithms in solving many engineering problems, it is imperative to scrutinize the success of these methods applied to data mining. This paper provides a survey for the data clustering algorithms as of applications of data mining form evolutionary and swarm intelligence perspective. In this paper, We will familiarize the reader with the different evolutionary and swarm intelligence techniques effortlessly. The clustering techniques that will be covered in this paper are differential evolution (DE), genetic algorithms (GA), particle swarm optimization (PSO), and ant colony optimization (ACO). Moreover, an overview of the main characteristics of the clustering algorithms presented in a comparative way.

Keywords: data clustering, k-means, differential evolution (DE), genetic algorithms (GA), particle swarm optimization (PSO), and ant colony optimization (ACO).

1. Introduction

In recent years, the dramatic rise in the use of the web and the improvement in communications in general have transformed our society into one that strongly depends on information. The huge amount of data that is generated by this communication process contains important information that accumulates daily in databases and is not easy to extract. The field of data mining developed as a means of extracting information and knowledge from databases to discover patterns or concepts that are not evident [1]. Machine learning provides the technical basis of data mining by extracting information from the raw data in the databases. The process usually consists of the following: transforming the data to a suitable format, cleaning it, and inferring or making conclusions regarding the data. Clustering aims at decomposing a given set of objects into subgroups or clusters based on similarity. Its main purpose is to identify homogeneous groups by finding similarities between objects regarding their characterizing attributes. From a machine learning perspective, clusters correspond to the hidden patterns in data, the search for clusters is a kind of unsupervised learning, and the resulting system represents a data concept. Data clustering is an important but an extremely difficult problem. Clustering techniques require the definition of a similarity measure between patterns, which is not easy to specify in the absence of any prior knowledge about cluster shapes. The clustering problem has been addressed in many contexts and by researchers in many disciplines; this reflects its broad appeal and usefulness as one of the steps in exploratory data analysis. However, clustering is a difficult problem combinatorially, and differences in assumptions and contexts in different communities has made the transfer of useful generic concepts and methodologies slow to occur [2][3]. A large number of clustering algorithms exist
and most of these algorithms are based on the following two popular clustering techniques: iterative square-error partitional clustering and agglomerative hierarchical [3], yet no single algorithm can adequately handle all sorts of cluster shapes and structures. Each algorithm has its own approach for handling cluster validity, number of clusters, and structure imposed on the data. For many years now, several papers have highlighted the efficiency of approaches inspired from the nature. In particular a variety of algorithms inspired by the evolutionary algorithms and swarm intelligence. The evolutionary algorithms simulate natural evolution that based on natural selection and genetics by combining the fittest individual such as genetic algorithms, evolutionary strategies. The swarm intelligence originally took its inspiration from the biological examples by swarming, flocking and herding phenomena in vertebrates [4][5]. The swarm intelligence techniques as a promising techniques have been applied to clustering optimization problems, the particle swarm optimization (PSO) and ant colony optimization (ACO) [6][7][8]. The purpose of this paper is to provide a comprehensive and systematic description of the influential and important clustering algorithms from evolutionary and swarm intelligence perspective, with emphasis on new advances in recent years and suggested directions of research.

2. What is Clustering?

A large number of clustering definitions can be found in the literature, from simple to elaborate. The simplest definition is shared among all and includes one fundamental concept: the grouping together of similar data items into clusters. A more elaborate definition, for example, is stated in [9], "These obtained clusters should reflect some mechanism at work in the domain from which instances or data points are drawn, a mechanism that causes some instances to bear a stronger resemblance to one another than they do to the remaining instances." A simple, formal, mathematical definition of clustering, as the following: let \( X \in \mathbb{R}^{m \times n} \) a set of data items representing a set of \( m \) points \( x_i \) in \( \mathbb{R}^n \). The goal is to partition \( X \) into \( Q(P) \) groups \( C_k \) such every data that belong to the same group are more "alike" than data in different groups. So that

1. None of the clusters is empty; \( C_k \neq \emptyset \) \( k=1,\ldots,K \)
2. The union of the clusters form the data sets \( \bigcup_{k=1}^{K} C_k = X \)
3. Each sample belongs to a single cluster; \( C_k \cap C_l = \emptyset \), \( k,l=1,\ldots,K \), and \( k \neq l \)

Each of the \( K \) groups is called a cluster. The result of the algorithm is an injective mapping \( X \mapsto C \) of data items \( x_i \) to clusters \( C_k \). The number \( K \) might be pre-assigned by the user or it can be an unknown, determined by the algorithm. The nature of the clustering problem is such that the ideal approach is equivalent to finding the global solution of a non-linear optimization problem. This is usually a difficult task to achieve and combinatorial. As a matter of fact, this is an NP hard problem, e.g. a problem that can not be solved in polynomial time. Naturally, the best way to apply clustering would be to identify all possible clustering and select the best suitable one. Unfortunately, due to limited time and large amount of feature vectors this isn’t usually possible. If we let \( S(m,K) \) denote the number of possible clustering of \( m \) vectors into \( K \) groups it is true that

1. \( S(m,1)=S(m,m)=1 \)
2. \( S(m,K)=0 \) if \( K > m \)
where the second statement comes from the definition that no cluster may be empty. Actually, it can be shown that the solution for $S$ is the Stirling numbers of the second kind:

$$S(m,K) = \frac{1}{K!} \sum_{i=1}^{K} (-1)^{K-i} \binom{K}{i} i^m$$

It is quite clear that the solution of Equation 1 grows rapidly with $m$ and if the number of desired clusters $K$ is not initially available many different values must be tested and a “raw power” solution becomes impossible. A more efficient solution must be found. The clustering problem can be formulated an optimization and search problem where the result is to find the partition $P^* = \{C_1, \ldots, C_k\}$ that has optimal adequacy with respect to all other feasible solutions $\{p^1, \ldots, p^S(m,K)\}$. This is equivalent to a statistical mathematical function $J_e(P)$ that needs to be optimized, where $J_e(P)$ represents the quality measurement for a partition $P$ given $\forall P J_e(P) \geq 0$. The problem is to find the best solution (i.e., partition) $P^*$ such that:

$$J_e(P^*) = \max_P J_e(P)$$

There are many different ways to express and formulate the clustering problem, as a consequence, the obtained results and its interpretations depend strongly on the way the clustering problem was originally formulated. For example, the clusters or groups that are identified may be exclusive, so that every instance belongs in only one group (i.e., hard clustering). Or, they may be overlapping, meaning that one instance may fall into several clusters (i.e., soft clustering). Or they may be probabilistic, whereby an instance belongs to each group depending on a certain assigned probability. Such statistical approaches make sense in practical situations where no amount of training data is sufficient to make a completely firm decision about cluster memberships. Or they may be hierarchical, such that there is a crude division of the instances into groups at a high level that is further refined into a finer levels. Furthermore, different formulations lead to different algorithms to solve. If we also consider all the “variations” of each different algorithm proposed to solve each different formulation, we end up with a very large family of clustering algorithms [2][3].

3. Stages of Clustering Algorithms

The cluster analysis is primarily a tool for discovering previously hidden structure in a set of unordered objects. In this case one assumes that a ‘true’ or natural grouping exists in the data. However, the assignment of objects to the classes and the description of these classes are unknown. By arranging similar objects into clusters one tries to reconstruct the unknown structure in the hope that every cluster found represents an actual type or category of objects. Clustering methods can also be used for data reduction purposes. Then it is merely aiming at a simplified representation of the set of objects which allows for dealing with a manageable number of homogeneous groups instead of with a vast number of single objects. Only some mathematical criteria can decide on the composition of clusters when classifying data-sets automatically. Therefore clustering methods are endowed with distance functions that measure the dissimilarity of presented example cases, which is equivalent to measuring their similarity. Figure 1, depicts the procedure of cluster analysis with four basic steps [10]. These steps can be listed as follows:
1) **Feature selection or extraction.** Feature selection chooses distinguishing features that should be of use in distinguishing patterns belonging to different clusters, immune to noise, easy to extract and interpret. Moreover, some clustering algorithms do not perform well when applied to large high-dimensional datasets. In particular, several model-based algorithms that are shown to be very efficient on limited size datasets are found unfeasible when large scale datasets are introduced.

2) **Clustering algorithm design or selection.** This step is usually combined with the selection of a corresponding proximity measure and the construction of a criterion function. Almost all clustering algorithms are explicitly or implicitly connected to some definition of proximity measure. Once a proximity measure is chosen, the construction of a clustering criterion function makes the partition of clusters an optimization problem, which is well defined mathematically, and has rich solutions in the literature. Clustering is ubiquitous, and a wealth of clustering algorithms has been developed to solve different problems in specific fields. However, there is no clustering algorithm that can be universally used to solve all problems.

3) **Cluster validation.** The needs for an effective evaluation standards and criteria are important to provide the users with a degree of confidence for the clustering results derived from the used algorithms.

4) **Results interpretation.** The ultimate goal of clustering is to provide users with meaningful insights from the original data, so that they can effectively solve the problems encountered. Experts in the relevant fields interpret the data partition. Further analyzes, even experiments, may be required to guarantee the reliability of extracted knowledge.

![Figure 1. The main steps for clustering algorithms [10].](image)

### 4. Clustering Algorithms

A large number of clustering algorithms exist and most of these algorithms are based on the following two popular clustering techniques (1) agglomerative hierarchical and (2) iterative square-error partitional clustering [3]. Hierarchical clustering groups data objects with a sequence of partitions, either from singleton clusters to a cluster including all individuals or vice versa, while partitional clustering directly divides data objects into some pre-specified number of clusters without the hierarchical structure. The hierarchical algorithms, has several advantages, such as the number of clusters need not to be specified *a priori*, and they are independent of the initial conditions. However,
they are computationally expensive for most of hierarchical clustering algorithms is at least $O(m^2)$ and are not suitable for very large data sets, and they lack robustness and are, hence, sensitive to noise and outliers and the quality of a pure hierarchical clustering method suffers from its inability to perform adjustment once a merge or split decision has been executed. That is, if a particular merge or split decision later turns out to have been a poor choice, the method cannot backtrack and correct it. Popular techniques in the hierarchical category are the single linkage, complete linkage and average linkage algorithms.

The partitional clustering techniques generally attempt iteratively to minimize a cost function. A distinguishing feature of the hierarchical clustering techniques from the partitional ones is that while the former provide a valid clustering of the data at each iteration of the algorithm, the latter do not do so (they provide a valid clustering only on termination of the algorithm). The essential difference in the various types of partitional algorithms lies in the techniques used to model and minimize the cost function. These partitional clustering algorithms determine a grouping solution by maximizing the similarities among objects within the same groups while minimizing the dissimilarities between different groups. Thus, the algorithmic task can be stated as an optimization problem. Statistical criteria that consider the within and the between variance scatter matrices can be used to quantify the goodness of the partitions and to determine the optimal one. Ideally, a clustering algorithm should be simple, efficient and capable of dealing with huge datasets. Moreover, it should be objective and robust for equivalent samples and able to detect different cluster shapes. One of the most basic, straightforward clustering techniques is the classic k-means algorithm, in use for several decades. k-means algorithm is an iterative procedure. The k-means algorithm consist of two primary steps: 1) The assignment step where the instances are placed in the closest class. 2) The re-estimation step where the class centroids are recalculated from the instances assigned to the class. We repeat the two steps until convergence occurs which is when the re-estimation step leads to minimal change in the class centroids [11]. The procedure described in Table 1. Algorithmically, the k-means algorithm in its general form differs only slightly to the EM algorithm though the loss functions and results obtained using them can differ greatly. Variations of k-means where the Euclidean distance function is replaced by another distance have been proposed such as k-medoids, and their variations. In k-means case a cluster is represented by its centroid, which is a mean of points within a cluster. k-medoid the most appropriate data point within a cluster that represent the cluster. Representation by k-medoids has two advantages. First, it presents no limitations on attributes types, and, second, the choice of medoids is dictated by the location of a predominant fraction of points inside a cluster and, therefore, it is lesser sensitive to the presence of outliers. On the other hand, centroids have the advantage of clear geometric and statistical meaning. There are two main approaches described in the literature which can be used to reduce the overall computational requirements of the k-means clustering method especially for the distance calculations: (1) Use the information from the previous iteration to reduce the number of distance calculations. P-cluster is a k-means-based clustering algorithm which exploits the fact that the changes of the assignment of patterns to clusters are relatively few after the first few iterations [12]. It uses a heuristic which determines if the closest prototype of a pattern $E$ has been changed or not by using a simple checks. If the assignment has not changed, no further distance calculations are required. It also uses the fact that the movement of the cluster centroids is small for consecutive iterations (especially after a few iterations). (2) Organize the prototype vectors in a suitable data structure so that finding the closest prototype for a given pattern becomes more efficient [13]. This problem reduces to finding the nearest neighbor problem for a given pattern in the prototype space. The time complexity and space complexity
of $k$-means clustering algorithm is $O(mnK)$ and $O(n + K)$. Since $K$ and $n$ are usually much less than $m$, $k$-means can be used to cluster large data sets.

Table 1. The main steps for $k$-mean clustering algorithm.

1. Choose $K$ initial clusters $c_1, \ldots, c_K$.
2. At the $k^{th}$ iterative step, distribute the samples $X$ among the $K$ clusters using the following relation
   \[ x_i \in C_j \quad \text{if} \quad \|x - C_j^k\| < \|x - C_i^k\| \quad \forall i \text{ and } i \neq j \]
   where $C_j$ denotes the set of samples whose cluster center is $C_j^k$.
3. Compute the new cluster centers $c_j^{k+1}: j = 1, 2, \ldots, K$, at $(k+1)^{th}$ step, such that the sum of the squared distance from all points in $C_j^k$ to the new cluster is minimized. The measure which minimizes this is simply the sample mean of $C_j^k$. Therefore, the new cluster center is given by
   \[ c_j^{k+1} = \frac{1}{m_j} \sum_{x \in C_j^k} x, \quad j = 1, 2, \ldots, K \]
   where $m_j$ is the number of samples in $C_j^k$.
4. Compute the error function:
   \[ J_e = \sum_{j=1}^{K} \sum_{x \in C_j} (x - c_j^{k+1})^2 \]
5. If $J_e$ does not change significantly or cluster membership no longer changes the algorithm has converged and the procedure is terminated. Otherwise go to step 2.

Competitive (or winner–take–all) neural networks are often used to cluster input data. In competitive learning, similar patterns are grouped by the network and represented by a single unit (neuron). This grouping is done automatically based on data correlations. Well-known examples of ANNs used for clustering include Kohonen’s learning vector quantization (LVQ) and self-organizing map (SOM) [14]. SOMs can also be viewed as a constrained version of $k$-means clustering, in which the cluster centers tend to lie in a low-dimensional manifold in the feature or attribute space [15]. Although this algorithm is easy to implement, it has the drawback of being greedy in the sense that it tends to get stuck in a local minimum due to non-convexity of its objective function that usually depends on the initial center provided. This makes this algorithm very sensitive to initial points [16]. Moreover, clustering algorithms usually require selecting a set of parameters, thus turning each application into a set of subjective choices. If no prior knowledge is available, assessing the correct number of clusters (e.g., as required by the $k$-means algorithm), is almost impossible. Several of the most successful algorithms in the field of data mining do not explicitly accept the number of clusters $K$ as an input; however this number is directly derived from their parameters.

5. Evolutionary Clustering Techniques

The different clustering methods, in general, try to optimize some measure of goodness of a clustering solution either explicitly or implicitly. The clustering problem then reduces to one of searching for an appropriate number of suitable partitions such that some goodness measure is optimized. For example,
the k-means algorithm performs a local optimization of the $J_c$ measure. The iteratively optimal procedure of k-means cannot guarantee convergence to a global optimum. It may be noted that searching the exhaustive set of all possible partitions of the data is prohibitively exhaustive since the search space is huge and complex, with numerous local optima. Simple local search techniques, like hill-climbing algorithms, are utilized to find the partitions, but they are easily stuck in local minima and therefore cannot guarantee optimality. As an alternative to the problem of local minimum, stochastic methods, simulated annealing, differential evolution, evolutionary strategies or genetic algorithms and other evolutionary algorithms. Where, the clustering can be regarded as a category of optimization problems. In evolutionary algorithms, the main idea is to create a population of candidate solutions to an optimization problem, which is iteratively refined by alteration and selection of good solutions for the next iteration. Candidate solutions are selected according to the so-called fitness function, which evaluates their quality with respect to the optimization problem as shown in Table 2. Selection, recombination, and mutation are the most widely used evolutionary operators. The selection operator ensures the continuity of the population by favoring the best individuals in the next generation. The recombination and mutation operators support the diversity of the population by exerting perturbations on the individuals.

**Table 2. The main steps for any evolutionary algorithm**

<table>
<thead>
<tr>
<th>Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>t = 0</td>
</tr>
<tr>
<td>Initialise P(t)</td>
</tr>
<tr>
<td>Evaluate P(t)</td>
</tr>
<tr>
<td>while (not done)</td>
</tr>
<tr>
<td>t = t + 1</td>
</tr>
<tr>
<td>select P(t) from P(t−1)</td>
</tr>
<tr>
<td>alter P(t)</td>
</tr>
<tr>
<td>evaluate P(t)</td>
</tr>
<tr>
<td>return best</td>
</tr>
</tbody>
</table>

Lee and Antonsson [17], proposed evolution strategies implements variable length genomes that allow the algorithm to effectively search for both optimal cluster center positions and cluster number. Das et al., [18], modified the conventional DE algorithm from its classical form to improve its convergence properties and used a novel representation scheme for the search variables in order to determine the optimal number of clusters as shown in Table 3. Among many evolutionary algorithms, genetic algorithms are the most popular approaches applied in cluster analysis. For example, the genetic algorithms have been applied to partitioning clustering in many ways, which can be grouped into three main categories [19][20][21]:

1. Direct encoding of the object cluster association: The idea in this approach is to use a genetic encoding that allocates directly $m$ objects to $K$ clusters, such that each candidate solution consists of $m$ genes with integer values in the interval $[1,K]$. When there are $K$ clusters, there are $K!$ different chromosomes corresponding to each $K$-partition of the data. This increases the effective search space size by a factor of $K!$. However, the representation scheme has a major drawback because of its redundancy [3].

2. Encoding of cluster separating boundaries, which divide the attribute feature space to separate the clusters. Bandyopadhyay and Maulik [22] is to determine the boundaries between clusters by connected linear segments instead of rigid planes.
Table 3. The main steps for DE clustering algorithm.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Initialize each chromosome $p$ to contain $K$ number of randomly selected cluster centers $c_{p1}, c_{p2}, \ldots, c_{pK}$ and $K$ (randomly chosen) activation thresholds in $[0, 1]$.</td>
</tr>
<tr>
<td>2</td>
<td>Repeat</td>
</tr>
<tr>
<td>3</td>
<td>for each chromosome $p$ in population do</td>
</tr>
<tr>
<td>4</td>
<td>if $T_{ij} &gt; 0.5$ then the $j^{th}$ cluster center $c_{ij}$ is active else $c_{ij}$ is inactive</td>
</tr>
<tr>
<td>5</td>
<td>for each pattern $x_i$ in the data set $X$ do</td>
</tr>
<tr>
<td>6</td>
<td>calculate Euclidean distance of $x_i$ with all cluster centroids</td>
</tr>
<tr>
<td>7</td>
<td>assign $x_i$ to the cluster that have nearest centroid to $x_i$</td>
</tr>
<tr>
<td>8</td>
<td>check if the number of data points belonging to any cluster center $c_{ij}$ is less than</td>
</tr>
<tr>
<td>9</td>
<td>If so, update the cluster centers of the chromosome using the concept of average.</td>
</tr>
<tr>
<td>10</td>
<td>end for</td>
</tr>
<tr>
<td>11</td>
<td>calculate the fitness function $f(C_p)$</td>
</tr>
<tr>
<td>12</td>
<td>end for</td>
</tr>
<tr>
<td>13</td>
<td>Update the cluster centroids according to DE rules.</td>
</tr>
<tr>
<td>14</td>
<td>Until termination (condition).</td>
</tr>
</tbody>
</table>

3. Centroid/medoid and variation parameter encoding for each cluster. The idea is to determine a representation point for each cluster and to allocate each object to the cluster with the nearest representation point, where 'nearest' refers to a distance measure, such as Euclidean distance. The fitness of a candidate solution is then computed as the adequacy of the identified partition according to a statistical criterion, such as the Marriott or variance ratio criterion. Many studies have shown that this approach is more robust in converging towards the optimal partition than classic partitional algorithms [16][22][21].

Finally, some authors introduced hybrid clustering algorithms, which combine classic clustering techniques with evolutionary algorithms. In a hybrid approach proposed in Babu and Murty [23], the GA is used only to find good initial cluster centers and the $k$-means algorithm is applied to find the final partition. Krishna and Murthy [24] introduced a GA with the direct encoding of object-cluster associations and applied $k$-means to determine the quality of the GA candidate solutions. For this, each GA candidate solution is used as a starting point for a $k$-means run. The quality of the solution found by the $k$-means run is then used as the fitness of the GA candidate solution. as well as applications of simulated annealing and other local search methods to clustering. In Tseng and Yang [25][26], the algorithm CLUSTERING includes a heuristic scheme for estimating the appropriate number of clusters in the data. It also uses a nearest-neighbor algorithm to divide data into small subsets, before GAs-based clustering, in order to reduce the computational complexity. In Babu and Murty [27], evolutionary strategies were used on both hard and fuzzy clustering problems and evolutionary programming has been used to evolve fuzzy min-max clusters. It has been observed that they perform better than their classical counterparts, the $k$-means algorithm and the fuzzy $c$-means algorithm. The main drawback that plagues the search techniques-based clustering algorithms is the parameter selection. These search techniques introduce more parameters than other methods (like $k$-means). There are no theoretic guidelines to select the appropriate and effective parameters; specially all of these approaches suffer from sensitivity to control parameter selection. For each specific problem, one has to tune the parameter values to suit the application [3]. Another problem is the computational complexity paid for the convergence to global optima. High computational requirement limits their applications in large-scale
data sets [10]. The time complexity and space complexity of evolutionary algorithms is $O(mnKP)$ and $O(P + m + K)$, where $P$ is the population size.

6. From Evolutionary to Swarm Intelligence Clustering Techniques

Recently, the swarm intelligence techniques as a promising techniques have been applied to clustering optimization problems, the particle swarm optimization (PSO) and ant colony optimization (ACO) [3][6][7][8]. These studies are often inspired by the observation of social insects and other animal societies [4][5][28]. These algorithms mainly stochastic search and optimization techniques, guided by the principles of collective behavior and self-organization of insect swarms. They are efficient, adaptive and robust search methods producing near optimal solutions and have a large amount of implicit parallelism. On the other hand, data clustering may be well formulated as a global optimization problem. The PSO-based clustering algorithm was first introduced by Omran et al. in [29]. The results of Omran et al. [29][30][31][32] showed that PSO based method outperformed k-means, FCM and a few other state-of-the-art clustering algorithms. In their method, Omran et al. used a quantization error based fitness measure for judging the performance of a clustering algorithm. The quantization error is defined as:

$$J_e = \frac{1}{K} \sum_{j=1}^{K} \sum_{x_i \in C_j} d(x_i - c_j)m_j$$  \hspace{1cm} (3)

where $c_j$ is the $j^{th}$ cluster center and $m_j$ is the number of data points belonging to the $j^{th}$ cluster. Each particle in the PSO algorithm represents a possible set of $K$ cluster centroids as vector $C_{pj}$ refers to the $j^{th}$ cluster centroid vector of the $p^{th}$ particle. The quality of each particle is measured by the following fitness function:

$$f(C_p, Z) = w_1 d_{max}(Z, C_p) + w_2 (x_{max} - d_{min}(C_p)) + w_3 J_e$$  \hspace{1cm} (4)

In the above expression, $x_{max}$ is the maximum feature value in the dataset and $Z$ is a matrix representing the matrix representing the assignment of the patterns to the clusters of the $p^{th}$ particle. Each element $z_{pi,j}$ indicates whether the pattern $x_i$ belongs to cluster $C_{pj}$ of $p^{th}$ particle. The user-defined constants $w_1$, $w_2$, and $w_3$ are used to weigh the contributions from different sub-objectives. In addition,

$$d_{max}(Z, C_p) = \max_{j=1,...,K} \left\{ \sum_{z_{pi,j} \in C_{pj}} d(z_{pi,j}, C_{pj}) / m_{pj} \right\}$$ \hspace{1cm} (5)

and,

$$d_{min}(C_p) = \min_{z_{pj1,j}, z_{pj2,j} \neq j} \{d(C_{pj1}, C_{pj2})\}$$ \hspace{1cm} (6)

is the minimum Euclidean distance between any pair of clusters. In the above, $m_{pj}$ is the number of patterns that belong to cluster $C_{pj}$ of particle $p$. The fitness function is a multi-objective optimization problem, which minimizes the intra-cluster distance, maximizes inter-cluster separation, and reduces
the quantization error. The PSO clustering algorithm is summarized in Table 4. Van der Merwe and Engelbrecht [33] hybridized this approach with the k-means algorithm for clustering general datasets. A single particle of the swarm is initialized with the result of the k-means algorithm. The rest of the swarm is initialized randomly. In 2003, Xiao et al [34] used a new approach based on the synergism of the PSO and the Self Organizing Maps (SOM) for clustering gene expression data. They got promising results by applying the hybrid SOM-PSO algorithm over the gene expression data of Yeast and rat Hepatocytes. Finally, Omran et al. [35] applied the Differential Evolution (DE) algorithm for clustering problem [36]. Paterlini and Krink [20] have compared the performance of k-means, GA, PSO and DE for a representative point evaluation approach to partitional clustering. The results show that PSO and DE outperformed the k-means algorithm. Their work is focused on non-automatic clustering with a pre-assigned number of clusters.

Table 4. The main steps for PSO clustering algorithm.

initialize each particle $p$ with $K$ random clusters centers $c_{p1}, c_{p2}, \ldots, c_{pK}$.
repeat
  for each particle $p$ in particles do
    for each pattern $x_i$ in the data set $X$ do
      calculate Euclidean distance of $x_i$ with all cluster centroids
      assign $x_i$ to the cluster that have nearest centroid to $x_i$
    end for
    calculate the fitness function $f(C_p, Z)$
  end for
  Find the personal best and global best position of each particle.
  Update the cluster centroids according to updating velocity and coordinate of PSO.
Until termination (condition).

Cui and Potok [37] proposed a PSO based hybrid document clustering algorithm. The PSO clustering algorithm performs a globalized search in the entire solution space. In the experiments, they applied the PSO, k-means and a hybrid PSO clustering algorithm on four different text document datasets. The results illustrate that the hybrid PSO algorithm can generate more compact clustering results than the k-means algorithm. Fun and Chen [38] introduced an evolutionary PSO learning-based method to optimally cluster $N$ data points into $K$ clusters. The hybrid PSO and k-means, with a novel alternative metric algorithm is called Alternative KPSO-clustering (AKPSO) method. This is developed to automatically detect the cluster centers of geometrical structure data sets. Simulation results compared with some well-known clustering methods demonstrate the robustness and efficiency of the novel AKPSO method.

The ACO [39][40], which focuses on discrete optimization problems, have been used in clustering algorithms [41]. Deneubourg et al. [42] proposed an agent–based model to explain the clustering behavior of real ants. In this model, artificial ants (or agents) are moving randomly on a square grid of cells on which some items are scattered. Each cell can only contain a single item and each ant can move the items on the grid by picking up and dropping these items with a certain probability which depends on an estimation of the density of items of the same type in the neighborhood. Lumer and Faieta [43] extended the model of Deneubourg et al., [42] using a dissimilarity–based evaluation of the local density, in order to make it suitable for data clustering. In their model, objects represent data items that
belong to a database. These items are randomly scattered on a periodic square grid on which randomly moving agents group them according to their similarity. In order to do that, a similarity (or dissimilarity) measure between pairs of data items is needed to compute the probabilities of picking and dropping data elements on the grid. In their model, the probability of picking a data element \( x_i \) is defined as

\[
P_p(x_i) = \left( \frac{k_p}{k_p + f(x_i)} \right)^2
\]

(7)

where \( k_p \) is a constant and \( f(x_i) \) is a similarity density measure with respect to element \( x_i \). Likewise, the probability of dropping a data element is given by

\[
P_d(x_i) = \begin{cases} 
2f(x_i) & \text{if } f(x_i) < k_d \\
1 & \text{otherwise}
\end{cases}
\]

(8)

where \( k_d \) is a constant. The similarity density \( f(x_i) \) for an element \( x_i \), at a particular grid location \( \tau \), is defined as

\[
f(x_i) = \max_{s} \left\{ \frac{1}{s^2} \sum_{x_{j \in \text{neigh}(	au)}} \left( 1 - \frac{d(x_i, x_j)}{\alpha} \right)^2 \right\}
\]

(9)

where \( s^2 \) is the size of the perception area \( \text{neigh}(\tau) \), centered at the location of the agent and \( \alpha \) is a scaling factor of the dissimilarity measure \( d(x_i, x_j) \) between elements \( x_i \) and \( x_j \). Generally, the size of the neighborhood is \( 3 \times 3 \). Probability of picking up data items is more when the object are either isolated or surrounded by dissimilar items. They trend to drop them in the vicinity of similar ones. In this way, a clustering of the elements on the grid is obtained. By following these rules, objects that are near each other in the feature space will be likely to be dropped in neighboring positions in the work space. After an initial period of random activity, a small tentative cluster of few similar objects will form. This pre-cluster acts as a stigmergic beacon so that the probability of dropping new, similar objects near it is greater than anywhere else on the workspace. This leads to a positive feedback cycle which increases the size of the cluster, until the clustering process is complete. The ant-based clustering algorithms, as a general rule, can be considered as non-hierarchical, hard, agglomerative clustering methods as shown in Table 5.

After the first appearance of this algorithm, many other variations of it have been proposed to improve its output quality [44][41], its convergence speed [45][46]. In a similar way, in Kanade and Hall [47][48] presented a hybridization of the ant systems with the classical FCM algorithm to determine the number of clusters in a given dataset automatically. In their fuzzy ant algorithm, at first the ant based clustering is used to create raw clusters and then these clusters are refined using the FCM algorithm. Initially the ants move the individual data objects to form heaps. The centroids of these heaps are taken as the initial cluster centers and the FCM algorithm is used to refine these clusters. In the second stage the objects obtained from the FCM algorithm are hardened according to the maximum membership criteria to form new heaps. These new heaps are then sometimes moved and merged by the ants. The final clusters formed are refined by using the FCM algorithm.
Table 5. The main steps for basic ant clustering algorithm.

```plaintext
randomly scatter data items on the toroidal grid
for j = 1 to #agents do
    i := random_select(free data items)
    Pick_up(agent(j), i)
    g := random_select(empty grid locations)
    place_agent(agent(j), g)
end for
for it_ctr = 1 to #iterations do
    j := random_select(all agents)
    step(agent(j), stepsize)
    i := carried_item(agent(j))
    drop := drop_item?(f(i))  // see equations 8 and 9
    if drop = TRUE then
        drop(agent(j), i)
        pick := FALSE
        while pick = FALSE do
            i := random_select(free data items)
            pick := pick_item?(f(i))  // see equations 7 and 9
        end while
        pick_up(agent(j), i)
    end if
end for
```

Later, Gu and Hall [49], proposed fuzzy ants clustering algorithm with a kernel distance metric reformulation and the goodness of the partition is evaluated by using the kernel Xie-Beni criterion after an epoch. Moreover, a number of modifications have been introduced to the basic ant based clustering scheme that improve the quality of the clustering, the speed of convergence and, in particular, the spatial separation between clusters on the grid, which is essential for the scheme of cluster retrieval. A detailed description of the variants and results on the qualitative performance gains afforded by these extensions are provided in [50]. Monmarche et al., [51] proposed an algorithm where several objects are allowed to be on the same cell of the workspace grid. Each cell with one or more objects together corresponds to a cluster. Each ant is also capable of carrying more than one object at a time. In this way, a kind of hierarchical clustering is implemented, where an ant carries an entire heap of objects. Another contribution of Monmarche was to hybridize the ant-based clustering algorithm with k-means algorithm and compared it to traditional k-means on various data sets, using the classification error for evaluation purposes. Ramos et al. [52] proposed ACLUSTER algorithm, which modified the ant-based clustering by changing the movement paradigm. While the previous works all relied on random moving ants, his ants would move according to a trail of pheromones left on clustering formations. This would reduce the exploration of empty areas, where the pheromone would eventually evaporate. In that sense, bio-inspired spatial transition probabilities are incorporated into the system, avoiding randomly moving agents, which encourage the distributed algorithm to explore regions manifestly without interest. The strategy allows guiding ants to find clusters of objects in an adaptive way. Hartmann [53] tried a different approach to the ant clustering algorithm, by using evolution to train both the system's disparity function and move policy. Each ant would have a neural network which would take the objects of its vicinity as input, and return the move action, and the pick up or drop action, as outputs. By changing the evolutionary system fitness function.
7. Comparison of Clustering Algorithms

This section offers an overview of the main characteristics of the clustering algorithms presented in a comparative way. We consider the evolutionary and swarm clustering techniques for data mining that have been discussed earlier. The desired characteristics of a clustering algorithm depend on the particular problem under consideration. However, the evaluation of clustering algorithms is done along a set of evaluation criteria. These include, scalability, minimum requirements for input parameters, dealing with noise and outliers, an ability to estimate any parameters automatically, find arbitrary-shaped clusters, sensitivity to control parameters, and Searching Strategy. This result is summarized in Table 6.

- **Scalability**: Clustering techniques for large sets of data must be scalable, both in terms of speed and space. It is usual for a database to contain millions of records, and thus, any clustering algorithm used should have linear or near linear time complexity to handle such large data sets. Some clustering techniques use statistical sampling. Nonetheless, there are cases, e.g., situations where relatively rare points have a dramatic effect on the final clustering, where a sampling is insufficient. Furthermore, clustering techniques for databases cannot assume that all the data will fit in main memory or that data elements can be randomly accessed. These algorithms are, likewise, infeasible for large data sets. Accessing data points sequentially and not being dependent on having all the data in main memory at once are important characteristics for scalability. One of the problems of the evolutionary and swarm-based techniques-based clustering algorithms is the computational complexity paid for the convergence to global optima. High computational requirement limits their applications in large-scale data sets.

- **Minimum requirements for input parameters**: Many clustering algorithms require some user-defined parameters, such as the number of clusters, in order to analyze the data. However, with large data sets and higher dimensionalities, it is desirable that a method require only limited guidance from the user, in order to avoid bias over the result. The main drawback that plagues the evolutionary and swarm-based techniques-based clustering algorithms is the parameter selection. More often than not, search techniques introduce more parameters than other methods (like k-means). There are no theoretic guidelines to select the appropriate and effective parameters.

- **Dealing with noise and outliers**: A point which is noise or is simply an atypical point (outlier) can often distort a clustering algorithm. By applying tests that determine if a particular point really belongs to a given cluster, some algorithms can detect noise and outliers and delete them or otherwise eliminate their negative effects. This processing can occur either while the clustering process is taking place or as a post-processing step. However, in some instances, points cannot be discarded and must be clustered as well as possible. In such cases, it is important to make sure that these points do not distort the clustering process for the majority of the points. The sensitivity of different algorithms to noise, which inherently exists in the data, is also a major contribution to the difference between their results. Partition into clusters should reflect the underlying structure of the data. In some cases, some structure exists in the data by chance: a slight modification of the data would wipe this structure out. In such a case, the identified structure may be due only to noise. The k-means is sensitive to outliers and noise. Even if an object is quite far away from the cluster centroid, it is still forced into a cluster and, thus, distorts the cluster shapes. The evolutionary and particle swarm optimization algorithms like the k-mean algorithm. On the other side the ant-based clustering algorithms produce an arbitrary shape clusters.
An ability to estimate any parameters automatically: Many clustering algorithms require some parameters, such as the number of clusters, the size of clusters, or the density of clusters. Many clustering algorithms take the number of clusters as a parameter. This can be a useful feature in certain instances, e.g., when using a clustering algorithm to create a balanced tree for nearest neighbor lookup, or when using clustering for compression. However, this is not generally good, since the number of clusters parameterized may not match the “real” number of clusters. An algorithm that requires the number of clusters up front can always be run multiple times. Assuming that there is some way to compare the quality of the clusters produced, it is then possible to empirically determine the best number of clusters. Of course, this increases the amount of computation required. Likewise, it is often difficult to estimate the proper values for other parameters of clustering algorithms. In general, the parameters of a clustering algorithm may identify areas of weakness. In the best case, the results produced by a clustering algorithm will be relatively insensitive to modest changes in the parameter values. The k-means explicitly accept the number of clusters K as an input, while the evolutionary and particle swarm optimization algorithms explicitly accept the number of clusters K or implicitly determine the best number of clusters. Finally, the ant-based clustering algorithm determine the best number of clusters automatically.

Find arbitrary-shaped clusters: The shape usually corresponds to the kinds of clusters an algorithm can find and we should consider this as a very important thing when choosing a method, since we want to be as general as possible. Different types of algorithms will be biased towards finding different types of cluster structures/shapes and it is not always an easy task to determine the shape or the corresponding bias. Every specific method can perform well on one data set, but very poorly on another, depending on the size and dimensionality of the data as well as the objective function and structures used. For example, the k-means faces two difficulty when clusters have widely different sizes or have convex shapes. The difficulty in these two situations is that the k-means objective function is a mismatch for the kind of clusters we are trying to find. The k-means objective function is minimized by globular clusters of equal size or by clusters that are well separated. This by producing hyperspherical in shape clusters and well-separated in the feature space. If the Mahalanobis distance is used, then the algorithm is even able to detect hyperellipsoidal shaped clusters. The evolutionary and particle swarm optimization algorithms likes the k-mean algorithm since they are searched for the centroid. On the other side the ant-based clustering algorithms produce an arbitrary shape clusters.

Sensitivity to control parameters: The clustering algorithms are dependent on the control parameters, in order to analyze the data. There are no theoretic guidelines to select the appropriate and effective parameters values, specially all of the clustering algorithms suffer from sensitivity to control parameter selection. The main drawback that plagues the evolutionary and swarm-based techniques-based clustering algorithms is the large number of control parameters. For example, the ant-based clustering algorithm has a number of user defined parameters (number of ants, size of the field, movement policies, etc), whose precise effects on the performance are not yet fully known. However, the DE has less number of control parameters.

Searching Strategy: The clustering algorithms have different search strategies and work differently. The k-mean algorithm is easy to implement, it has the drawback of being greedy in the sense that it tends to get stuck in a local minimum due to non-convexity of its objective function that usually depends on the initial center provided. This makes this algorithm very sensitive to initial points. As an alternative to the problem of local minimum, an evolutionary algorithms and
swarm-based clustering algorithms are used. Also, an evolutionary algorithms works in competitive manner while swarm-based clustering algorithms in cooperative manner.

Table 6. The comparison among clustering algorithms.

<table>
<thead>
<tr>
<th></th>
<th>k-mean</th>
<th>Genetic Algorithms</th>
<th>Differential Evolution</th>
<th>Particle Swarm Optimization</th>
<th>Ant Colony</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalability</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Time Complexity</td>
<td>$O(mnK)$</td>
<td>$O(mnKP)$</td>
<td>$O(mnKP)$</td>
<td>$O(mnKP)$</td>
<td>NA</td>
</tr>
<tr>
<td>Space Complexity</td>
<td>$O(m + K)$</td>
<td>$O(P + m + K)$</td>
<td>$O(P + m + K)$</td>
<td>$O(P + m + K)$</td>
<td>NA</td>
</tr>
<tr>
<td>Handling Outlier/Noise</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Minimum requirements for Tuning Parameters</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Search Strategy</td>
<td>Greedy/Local</td>
<td>Competitive/Global</td>
<td>Competitive/Global</td>
<td>Cooperative/Global</td>
<td>Cooperative/Global</td>
</tr>
<tr>
<td>Sensitivity to control parameters</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Arbitrary-Shaped Clusters</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

8. Discussion and Conclusions

In this study we have offers an overview of the main characteristics of the clustering algorithms presented in a comparative way. We consider the evolutionary and swarm clustering techniques for data mining. The desired characteristics of a clustering algorithm depend on the particular problem under consideration. The study shows that the swarm-based clustering algorithms mainly stochastic search and optimization techniques, guided by the principles of collective behavior and self organization of insect swarms. They are efficient, adaptive and robust search methods producing near optimal solutions and have a large amount of implicit parallelism. The recent developments in the ant-based clustering technique show that it is competitive when clustering datasets with high dimensionality or databases where the number of clusters is not previously known. However, the main drawback that plagues the evolutionary and swarm-based techniques-based clustering algorithms is the large number of control parameters. For example, the ant-based clustering algorithm has a number of user defined parameters (number of ants, size of the field, movement policies, etc), whose precise effects on the performance are not yet fully known.

References

