

# Evaluation and Selection of Clustering Methods Using a Hybrid Group MCDM

Sasan Barak<sup>1,2</sup>, Taha Mokfi<sup>3</sup>

1. Faculty of Economics, VŠB Technical University of Ostrava, Ostrava 70200, Czech Republic
2. Department of Management Science, Lancaster University Management School, Lancaster University, UK
3. Department of Statistics, University of Central Florida, Orlando, USA (taha.mokfi@ucf.edu)

Corresponding author: Sasan Barak,

Faculty of Economics, VŠB Technical University of Ostrava, Sokolská třída 33, Ostrava 70200, Czech Republic.

Tel: +420722495247,

Email: s.barak@lancaster.ac.uk, sasan.barak@gmail.com

## Acknowledgement:

The research was supported by the Czech Science Foundation (GACR), Project number GA17-22662S. We are deeply thankful to Ali Kamali Mohammadzadeh for helping in the first draft of this research.

## **Abstract**

Due to the lack of objective measures, the evaluation and prioritization of clustering methods is inherently challenging. Since their evaluation generally involves numerous criteria, it can be designed as a multiple criteria decision making (MCDM) problem and using multiple data sets, the problem can be formulated as a group MCDM modeling.

In this paper, a MCDM-based framework is proposed to evaluate and rank a number of clustering methods. The proposed approach employs three group MCDM algorithms and a Borda count method which leads to a comprehensive, robust framework capable of evaluating and ranking multiple clustering models on manifold data sets (cases). Moreover, we introduce a hybrid data clustering algorithm which combines a particle swarm optimization (PSO) algorithm with a K-means clustering algorithm. Finally, a clustering comparison with regard to both external and internal evaluation indicators is another contribution of this paper.

Six clustering methods are compared based on five evaluation measures. The results of comparative experiments on ten data sets indicate the effectiveness of the proposed hybrid clustering method. More importantly, the experimental results vividly demonstrate the effectiveness of the group MCDM-based evaluation on clustering model selection.

**Keywords:** Clustering; MCDM; Group TOPSIS; Group COPRAS; Particle Swarm Optimization.

## **1. Introduction**

In the past few decades, cluster analysis has played a central role in a variety of disciplines, including but not limited to computer sciences, life and medical sciences, earth sciences, social sciences, and economics (Das, Abraham, & Konar, 2008). This sheer usefulness reflects its significance as one of the main steps in data analysis.

Given the countless number of clustering algorithms available, the significance of evaluating these algorithms becomes evident as well. Numerous evaluation metrics have been proposed by researchers hitherto. The evaluation metrics can be intrinsic (internal) or extrinsic (external). Intrinsic metrics measure the closeness of elements within one cluster as well as the distance, or separation, of a cluster's elements from other clusters. Extrinsic metrics, on the other hand, evaluate the output of a given clustering method. Since these metrics measure different

characteristics of clustering methods, a perfect evaluation must include both aforementioned classes.

However, the use of different techniques gives rise to another challenge which is the multitude of different, and sometimes inconsistent, results which makes the evaluation process extremely difficult. Thus, it is of significant importance to aggregate these diverse outcomes into a single, final result. Otherwise, comparing the techniques and selecting the best method would be impossible in many cases.

There are also a few papers that consider the clustering modelling as a multi objective optimization problem (Liu, Li, Zhao, & Liu, 2019; Shang, Zhang, Li, Jiao, & Stolkin, 2019). Mainly these studies take into account the inconsistency in the criteria and enhance the capability of one specific model with a sophisticated approach; however, due to encounters with large numbers of datasets, the critical issue does not lie in developing or finding a complex clustering algorithm, but in the selection of a well-discriminating model. Fulcher, Little, and Jones (2013) discussed how a threshold on the simple model computed for each dataset provides comparable classification/clustering performance on different problems, undermining the need for computing complex classification/clustering algorithms.

Multiple criteria decision making (MCDM) has proven to be a real boon in such cases and can be used to resolve the aforementioned challenge, which is the primary motivation of this paper. MCDM deals with choosing among a finite set of alternatives evaluated on the basis of multiple criteria. It makes preference decisions, such as selecting, ranking, and prioritization over the available alternatives. When the decision making process is conducted by more than one decision maker (DM), a MCDM problem becomes a group MCDM problem. For a clustering evaluation problem, the utilized methods, data sets, and measures are considered as the alternatives, criteria, and DMs, respectively. Therefore, considering the different number of data sets, the problem of interest can be approached as a MCDM problem.

This study, first and foremost, provides a thorough framework to evaluate the clustering algorithms in different benchmark data sets, utilizing both internal and external measures. This framework is formulated as a group MCDM modeling. Since using only one ranking algorithm may not be robust, three MCDM algorithms are provided and the final ranking is aggregated with the Borda

method. To the best of our knowledge, such a comprehensive approach has not been presented hitherto.

In this paper, five different groups of clustering approaches, including partitioning based models, hierarchical, fuzzy, density based, and a novel optimized clustering algorithm are evaluated with external and internal clustering measures. An optimized clustering algorithm is a combination of k-means and particle swarm optimization (PSO) which aims to improve the performance of distance-based clustering algorithms by iteratively trying to optimize a candidate solution with regard to a given measure of quality.

Ergo, the major contributions of this paper are as follows:

- A clustering comparison with regard to both external and internal evaluation indicators
- Proposing a group MCDM (GMCDM) framework in order to evaluate and rank the clustering algorithms
- Presenting a robust ranking via the Borda method for aggregating MCDM algorithms
- Introducing a new hybrid PSO-clustering approach

The rest of the paper is organized as follows. In section 2 an extensive review of the literature is presented. The proposed algorithm is rendered in Section 3. In Section 4, the experimental results on the data sets are demonstrated and finally Section 5 concludes the paper.

## **2. Literature review**

### **2.1. Overview**

The goal of clustering is to categorize an unlabeled data set into groups of objects (Berkhin, 2006). Each group, called a “cluster”, consists of objects that are similar among themselves and dissimilar to objects of other groups according to specific metrics (Fahad, et al., 2014). In a more formal definition, clustering is the act of identifying a structure in an unlabeled data set by organizing data into homogeneous groups where the within-group-object similarity is minimized and the between-group-object dissimilarity is maximized (Liao, 2005). The terms unsupervised learning (Jain & Dubes, 1988), numerical taxonomy (Sneath & Sokal, 1973), vector quantization (Oehler & Gray, 1995), and learning by observation (Michalski & Stepp, 1983) are all synonymous with clustering (Jain, Murty, & Flynn, 1999).

The process of clustering consists of four steps (R. Xu & Wunsch, 2005):

1. Feature extraction and selection: the most representative features are extracted and selected from the original data set;
2. Clustering algorithm design: the clustering algorithm is designed pursuant to the characteristics of the problem;
3. Result evaluation: the clustering result and the validity of the algorithm are evaluated;
4. Result explanation: a pragmatic explanation for the result is provided;

## **2.2. Related works**

This section focuses on the clustering evaluation approaches in previous studies. Since the main goal of this paper is to provide a comprehensive evaluation framework, we review the studies mainly in terms of their evaluation measures and the ways the algorithms are compared to each other.

D. Xu and Tian (2015) analyzed both traditional and modern clustering algorithms. 26 traditional and 45 modern clustering methods were discussed in detail. However, no practical experiment was performed. Hsu (2015) compared three clustering methods, namely cluster-wise regression, model-based clustering and the K-means method using a large data set of buildings in New York. The algorithms were evaluated according to the Jaccard coefficient as a measure of cluster stability and prediction accuracy. The results showed that cluster-wise regression gives accurate predictions but unstable clusters, while the K-means gives more stable clusters but poor predictions in some cases. Das, et al. (2008) proposed an application of differential evolution to the automatic clustering of large unlabeled data sets. Superiority of the proposed method over three other methods including genetic and PSO clustering, and one hierarchical clustering algorithm was demonstrated. Two evaluation metrics, including Davis–Bouldin and Chou–Su (CS) were used in order to compare the algorithms. Despite some inconsistencies regarding the results of the measures, no aggregate ranking approach was utilized. Tzortzis and Likas (2014) presented the MinMax k-Means algorithm, an algorithm that assigns weights to each cluster relative to its variance, and the performance of the proposed algorithm was compared to K-means, k-means++ and Pifs K-means algorithms on six popular data sets. To evaluate the quality of the solutions, the maximum cluster variance and the sum of the cluster variances were utilized. Sanchez, Castillo, Castro, and Melin (2014) compared fuzzy granular gravitational clustering algorithm (FGGCA)

against four other clustering algorithms, namely hierarchical-complete, hierarchical-centroid, K-means, and affinity propagation with regard to classification accuracy and 10 clustering validity indices over a number of popular benchmark data sets, such as Wine, Iris, Seeds, and Glass identification. The results of utilized measures were inconsistent; however, no ranking method is presented and the best algorithm is selected intuitively. Ünlü and Xanthopoulos (2019) employ four different indices including Silhouette, Calinski–Harabasz (CH), Davies–Bouldin, and Consensus to evaluate the correct number of clusters and mentioned that there is no de facto optimal approach for finding this.

In Meila and Heckerman (2013), an experimental comparison of several clustering methods including Expectation–Maximization (EM), Classification EM (CEM), and a model-based hierarchical agglomerative clustering (HAC) method on a real-world data set was performed. The log-marginal-likelihood criterion was used to select the best model structure. Moreover, the difference between the number of clusters extracted from the model and the correct number of clusters was utilized to measure how well the models understand the domain under study. The two aforementioned criteria along with classification accuracy, running time and memory requirements were used to evaluate the algorithms. Karaboga and Ozturk (2011) compared a proposed Artificial Bee Colony clustering algorithm against 10 clustering methods on 13 data sets from the UCI Machine Learning Repository. In order to prioritize the algorithms, classification error percentages (CEP) on test data sets were used. The average of CEPs and the sum of the algorithms’ rankings of each problem were used as ranking measures. In Lv, et al. (2016), a density-based clustering algorithm based on a density-based spatial clustering of applications with noise (DBSCAN) method called SB-DBSCAN was proposed. The algorithm was compared with a traditional DBSCAN and IS-DBSCAN method on three data sets from the UCI Machine Learning Repository. The algorithms were compared according to their respective correct rate. However, in real clustering cases, the correct class (label) of the data sets is not available. Therefore, using accuracy is not a general approach for the clustering evaluation. Ding and Fu (2016) propounded a kernel-based fuzzy C-means (KFCM) based on the genetic algorithm. In order to improve the clustering performance of the KFCM algorithm, an improved adaptive genetic algorithm was used to optimize the initial clustering centers. The algorithm was compared to FCM and KFCM methods on a number of test data sets. The correct rate was employed to evaluate the methods. Márquez, Otero, Félix, and García (2018) proposed a strategy for evolving prototype based clustering that

uses a weighting scheme to “progressively forget” old samples. They tested their approach on a simulated database and did not evaluate it on real-world datasets. Duò, Robinson, and Soneson (2018) compared 14 different clustering algorithms on nine public datasets and three simulated datasets. The authors tried to assess the stability, run time and scalability of these 14 clustering methods for finding subpopulations in genetic datasets yet they did not propose any quantitative ranking on the tested algorithms. Yang, Li, Liang, Li, and Xue (2018) proposed a new cluster validity index based on an optimized morphology similarity distance and ReliefF algorithm.

By consideration the aforementioned studies, the choice of which measure to use to evaluate the clustering algorithms is subjective and non-systematic. According to an extensive study by Figueiredo, et al. (2019), the existence of too many different clustering evaluation indexes demonstrate that this field has not converged to a unique evaluation metrics to fairly and accurately evaluate clustering solutions. More importantly, how to assess the clustering efficiency from the result of the measures is the second dilemma. In the cases mentioned, different measures for evaluating the candidate clustering were implemented; however, they do not provide a ranking procedure to deal with the inconsistency on the evaluation measures’ result. This inconsistency is intensified when different data sets are utilized to prove the potential of the provided clustering algorithms. Lack of a comprehensive method to aggregate results and prioritize the algorithms are the main drawback of the studies.

There are other studies in which a number of clustering methods are compared. However, just like the ones that are described in this section, most do not present a comprehensive and objective framework to evaluate and rank the algorithms. Among the few that do, Kou, Peng, and Wang (2014) presented a MCDM-based approach to prioritize and rank six clustering algorithms in financial risk analysis on three data sets. Eleven performance measures were ranked using 3 MCDM methods, namely TOPSIS, DEA, and VIKOR. However, the lack of an integrated approach to combine the results of MCDM methods on the data sets is noticeable. Besides, due to this issue, increasing the number of data sets would lead to confusing results.

We believe that applying a comprehensive framework consisting of both internal and external measures, in addition to providing a robust group MCDM approach to rank the utilized methods, the above mentioned shortcoming can be resolved. This approach not only integrates the results of all methods on all data sets into a final ranking, which is completely clear and comprehensible, but

also allows scholars to use as many data sets as necessary to be able to evaluate the algorithms more precisely.

### 3. Methodology

Three main elements have a crucial role to evaluate a clustering algorithm: The evaluation measures, the data set (or case studies) and the candidate clustering algorithms. This section provides detailed explanations about these three major elements which, combined with the proposed MCDM framework, basically form the methodology of this study.

The proposed method consists of two different experiments, each including several steps. Figure 1 demonstrates the steps of each experiment. In the following sections, the steps are described in detail.

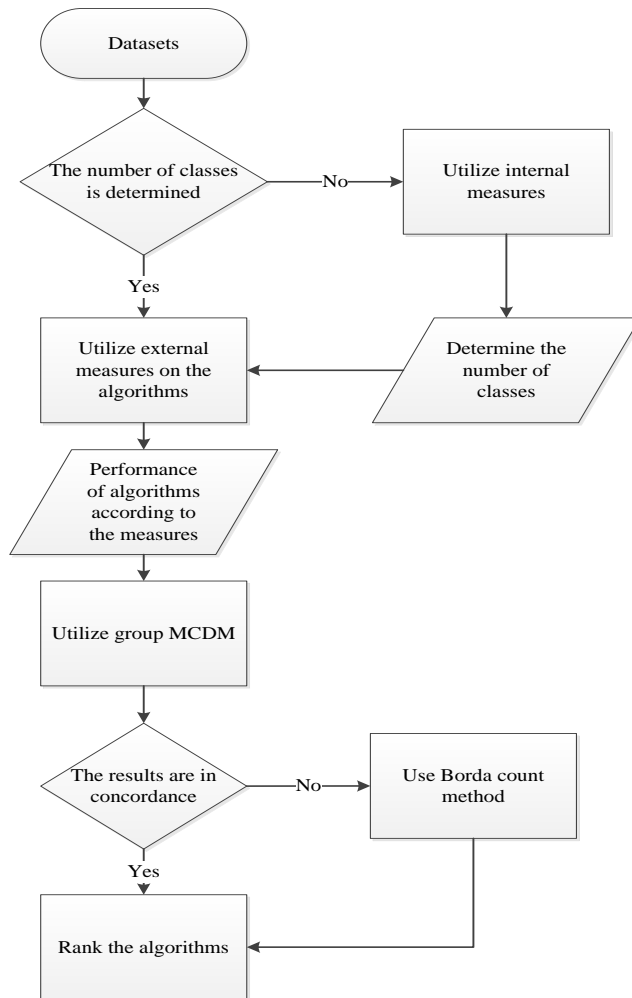


Figure 1. The proposed methodology



### **3.1. Data sets**

The data sets analyzed in this paper are retrieved from the UCI Machine Learning Repository, which was first created in 1987 at UC Irvine. It is a collection of databases that are widely used by scholars for the empirical analysis of algorithms in different fields of study (Bache & Lichman, 2013).

Ten data sets from the databases are utilized to demonstrate the results of the techniques. These data sets are implemented because of their generality and convenience in reproducing the results by other researchers.

### **3.2. Clustering algorithms**

As shown in Figure 1, two different approaches were undertaken to cluster the data sets. In the first one, the number of clusters is a predetermined parameter (equal to the number of classes for each data set), while in the second experiment, each data set was clustered in eight scenarios (from two up to nine clusters). Furthermore, six clustering algorithms are chosen to cluster the aforementioned data sets. We tried to select them in a way that as many clustering approaches as possible have at least one representative among the algorithms. To this end, the utilized algorithms are selected from five partition based, hierarchical, fuzzy theory based, density based, and metaheuristic based clustering classes. This section presents a brief explanation regarding each class, and the utilized algorithms are described in the relevant class.

#### **3.2.1. Partition based clustering algorithms**

The fundamental idea of partition based clustering algorithms is to consider the center of data objects as the center of the corresponding cluster. The partitioning based clustering algorithms split data points into a number of partitions, in the way that each partition represents a cluster. Each cluster must include at least one data point and each data point must belong to exactly one cluster (Fahad, et al., 2014).

K-means, K-medoids, K-modes, PAM, and CLARANS are some well-known algorithms of this class of clusters.

#### **K-means++ clustering algorithm**

Originally proposed in 2007, K-means ++ algorithm is an approximation algorithm for the k-means problem (Arthur & Vassilvitskii, 2007). The main idea of the algorithm is to overcome one of the major shortcomings of k-means algorithms. Since a standard k-means algorithm selects the primary centers in a random manner, this selection can be arbitrarily bad with regard to the objective function. In other words, K-means++ initiates with allocating just one cluster center randomly and subsequently searches for other cluster centers given the first one.

The algorithm consists of 5 steps:

Step 1. Randomly select one cluster center among the data objects.

Step 2. For each data object  $x$ , calculate  $D(x)$ , the Euclidean distance between  $x$  and the center.

Step 3. Randomly select a data object as a new center, using a weighted probability distribution where a point  $x$  is chosen with probability proportional to  $D(x)^2$ .

Step 4. Repeat Steps 2 and 3 up to the point that  $k$  centers are chosen.

Step 5. Proceed using standard k-means clustering algorithm.

In our research, this algorithm is implemented 50 times with different starting random point and its standard k-means has a maximum iteration of 500.

### **Lloyd's algorithm**

Also known as a Voronoi iteration, Lloyd's algorithm is a closely related k-means algorithm originally used for scalar quantization and was later extended for vector quantization. Nevertheless, it differs from a standard k-means clustering algorithm in that its input is continuous rather than discrete. Hence, Lloyd's clustering algorithm uses Voronoi diagrams rather than detecting the closest center to each point as a k-means does.

In the simplest words, Lloyd's algorithm consist of two main steps (Hamerly & Drake, 2015):

1. Initialize the centers
2. Until the algorithm converges:
  - a. Assign each point to its nearest cluster center
  - b. Move each center to the mean of its assigned points.

Steps 2(a) and 2(b) repeat until centers have not changed within 500 iterations.

### 3.2.2. Hierarchical clustering algorithms

The underlying idea for hierarchical clustering methods is to construct hierarchical relationships among data (Johnson, 1967). Hierarchical clustering algorithms consist of agglomerative and divisive methods. An agglomerative clustering method commences with considering each object as a cluster and recursively merges the most appropriate clusters. A divisive clustering method commences with the whole data set as one cluster and recursively breaks the most appropriate cluster. The process continues until a given stopping criterion is satisfied. However, hierarchical clustering methods have a major disadvantage, which is once a step is performed, it cannot be undone. BIRCH, CURE, ROCK and Chameleon are some instances of hierarchical clustering algorithms (R. Xu & Wunsch, 2005).

#### Ward's clustering method

Originally proposed by Ward in 1963, the Ward clustering method has been widely used and generalized in various ways (Murtagh & Legendre, 2014). Among the agglomerative clustering methods, Ward's method is the only algorithm that is based on a classical sum-of-squares criterion, producing groups that minimize within-group dispersion at each binary fusion (Murtagh & Legendre, 2014). Furthermore, another interesting feature of Ward's method is that it searches for clusters in multivariate Euclidean space.

Ward's method indicates the amount of sum of squares increasing by merging any two clusters. Merging cost,  $\Delta$ , of combining two clusters,  $C_a$  and  $C_b$ , is defined in equation 1:

$$\Delta(C_a, C_b) = \sum_{x \in C_{a \cup b}} d(x, c_{a \cup b}) - \sum_{x \in C_a} d(x, c_a) - \sum_{x \in C_b} d(x, c_b) = \frac{N_a N_b}{N_a + N_b} d(c_a, c_b) \quad (1)$$

where  $x$  is instance data;  $c_a$ ,  $c_b$ , and  $c_{a \cup b}$  are centers of  $C_a$ ,  $C_b$ , and  $C_{a \cup b}$  clusters respectively;  $d$  stands for distance;  $N_a$  and  $N_b$  are the cardinalities of the clusters  $C_a$  and  $C_b$  respectively.

With starting each point as one cluster, the sum of squares starts with zero value. Then, by merging clusters, it grows. The Ward method keeps merging clusters with slow growth in the sum of squares as much as possible.

### 3.2.3. Fuzzy theory based clustering algorithms

Fuzzy theory based clustering algorithms change the discrete value of belonging label into the continuous interval [0, 1]. Among the fuzzy theory based algorithms, FCM, FCS and MM can be enumerated. For more information regarding these, read Baraldi and Blonda (1999).

### Fuzzy K-Means Clustering

In FCM, the membership of data objects to different clusters can vary from 0 to 1. This method starts with randomly generated cluster centers and then each point is assigned to a cluster, according to membership weight that is calculated as follows:

$$W_{ij} = \frac{1}{\sum_{r=1}^K \left( \frac{d(x_i, c_k)}{d(x_i, c_r)} \right)^{\frac{2}{\beta-1}}} \quad (2)$$

where  $d$  denotes distance and  $\beta \in [1, \infty)$  is a predefined constant parameter which controls fuzziness of clusters.

After that, with an iterative gradient descent, the center of clusters is updated by the following formula:

$$c_j = \frac{\sum_{x_i \in C_j} W_{ij}^\beta * x_i}{\sum_{x_i \in C_j} W_{ij}^\beta} \quad (3)$$

The objective function is formulated as the sum of membership weighed Euclidean distances which should be minimized:

$$\Phi = \sum_{j=1}^K \sum_{x_i \in C_j} W_{ij}^\beta * d(x_i, c_j) \quad (4)$$

Convergence can be evaluated with comparison of changes in the cluster centers or membership weight at the end of the iterations.

### 3.2.4. Density based clustering algorithms

Density based clustering algorithms separate data points based on their density, connectivity, and boundary. In such algorithms, dense areas in the data space are considered the clusters, which are separated by areas with low-density, which are considered noise (Birant & Kut, 2007). Density based algorithms are very effective in discovering arbitrary shaped clusters. DBSCAN, DENCLUE, FlockStream, and OPTICS are typical examples of this class.

### **DBSCAN clustering algorithm**

DBSCAN is a very popular clustering algorithm proposed in 1996 (Ester, Kriegel, Sander, & Xu, 1996). It defines the density based on the number of objects in a region of a predetermined radius around a given point.

Using DBSCAN, the points are classified as core points, reachable points and outliers (Aggarwal & Reddy, 2013) as follows:

- A point  $x_i$  is a core point if at least minPts (a pre-defined parameter) points are within distance  $\varepsilon$  of it.
- A point  $x_j$  is directly reachable from  $x_i$  if point  $x_j$  is within distance  $\varepsilon$  from point  $x_i$  and point  $x_i$  is a core point.
- A point  $x_j$  is reachable from  $x_i$  if there is a path  $x_1, x_2, \dots, x_p$  with  $x_1 = x_i$  and  $x_p = x_j$ , where each point is directly reachable from its previous point.
- All points which are not reachable from any other point are outliers.

Based on the above definitions, the DBSCAN algorithm consists of the following steps:

Step 1: Find the  $\varepsilon$  neighbors of each point, and identify the core points with more than minPts neighbors.

Step 2: Find the connected elements of the core points on the neighboring graph, disregarding non-core points.

Step 3: Assign each non-core point to a nearby cluster if the cluster is an  $\varepsilon$  neighbor, otherwise assign it to the noise.

In this research, we use 5 and 0.5 values for minPts and  $\varepsilon$  respectively.

### **3.2.5. Metaheuristic based clustering methods**

The core idea of metaheuristic based clustering algorithms, which are among the state of the art clustering algorithms, is to imitate the changing process of the biological population (D. Xu & Tian, 2015). Since a PSO based clustering is presented in this study, this type of metaheuristic based clustering algorithms is described in more detail below.

### **PSO-K-means clustering algorithm**

Because of some shortcomings of the traditional K-means algorithm, such as sensitivity to select the initial clustering centers and easy converging to local optimization, a PSO based K-means clustering is established (C.-Y. Chen & Ye, 2012; Van der Merwe & Engelbrecht, 2003). The PSO algorithm is a population-based search method that was inspired by simulating the social behaviour of birds and became very popular due to its unique characteristics (Reyes-Sierra & Coello, 2006). A PSO algorithm consists of a swarm of particles in a given space. The position of a particle is demonstrated by a vector which represents a solution. PSO is initialized with a population of randomly positioned particles and searches for the best position with the best fitness (according to some predefined objective function) (R.-M. Chen, 2011).

The main advantage of using k-means as a local search technique in combination with PSO is that the global evolutionary procedure converges more quickly. The evolutionary procedure provides a method for automatically optimizing k-means, in terms of the centroids and the number of clusters. In general, swarm-based algorithms have advantages over some of the other heuristic-based clustering algorithms which can be summarized as follows:

- They are one of the most used algorithms and are reasonably simple to implement (Esmin, Coelho, & Matwin, 2015; Figueiredo, et al., 2019)
- PSO based clustering algorithms can be applied in different applications and domains (Esmin, et al., 2015; Figueiredo, et al., 2019; Nanda & Panda, 2014)
- They have relatively easier parameter selection and have a fast convergence (Figueiredo, et al., 2019; Nanda & Panda, 2014)
- Many papers were proposed in the past two decades when the PSO has been successfully applied to real-world problems (Figueiredo, et al., 2019). Also, hybrid PSO implementations show promising performance and better results in high dimensional data sets (Esmin, et al., 2015)
- Learning time for PSO based algorithms might be large when applying on big data, but they can be easily implemented in parallel frameworks to speed up the learning process (Nouaouria, Boukadoum, & Proulx, 2013)

In the original PSO algorithm, particles are characterized by their positions and velocities and tend to move in solution space with the following rules (Breaban & Luchian, 2011).

- each particle has inertia in its current position;
- particles remember their best position and tend to move toward it;
- particles can find the best position of the population, the global best position, and partially move toward it, too.

The velocity and position of particles are updated with the following equations:

$$v_i^{t+1} = w_v \times v_i^t + w_p \times r_1 \times (Pbest_i^t - P_i^t) + w_g \times r_2 \times (Gbest^t - P_i^t) \quad (5)$$

$$P_i^{t+1} = P_i^t + v_i^{t+1} \quad (6)$$

where  $v_i^t$  and  $v_i^{t+1}$  are the velocity of particle  $i$  in iteration  $t$  and  $t+1$ ,  $P_i^t$  and  $P_i^{t+1}$  are the position of particle  $i$  in iteration  $t$  and  $t+1$ ,  $Pbest_i^t$  is the best position of particle  $i$  in iteration  $t$ ,  $Gbest^t$  is the best position among all the particles in iteration  $t$ ,  $w_p$  and  $w_g$  are constant positive values,  $r_1$  and  $r_2$  are random numbers between 0 and 1, and  $w_v$  is the weight of inertia (see Figure 2).

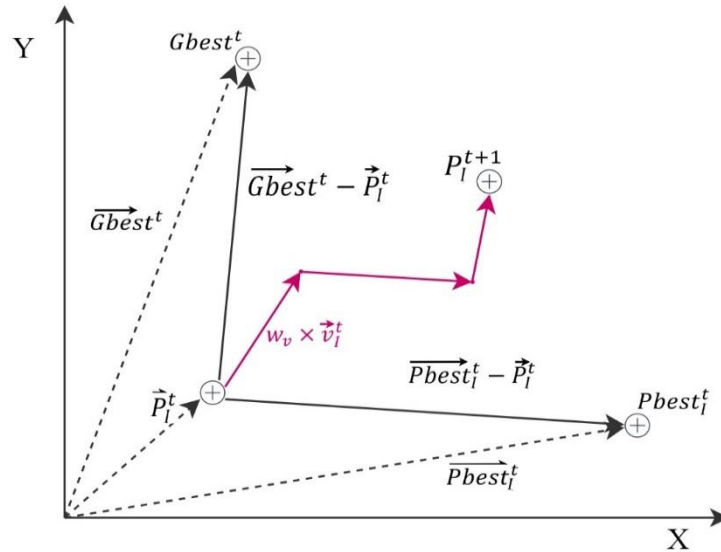


Figure 2. The procedure of updating velocity and position of particles

In order to enhance the performance of the algorithm, in comparison with the original PSO, some modifications were used in the updating procedure which is presented in the Figure 3.

The proposed algorithm uses a powerful global search capability of the PSO algorithm to optimize the selection of the initial clustering centers while dynamically adjusting the inertia weight and other parameters to enhance the performance of the PSO; taking advantage of the fitness variance of the group to decide the conversion timing between the front part of the PSO algorithm and the rear part of the K-means algorithm; setting the variables to monitor the changes of the optimal values of each particle and particle population; and taking the premature convergence particle to the mutation operation in a timely manner. Thus, we can find the global optimum initial clustering centers for K-means algorithm, then the clustering results are not affected by the initial clustering centers, and it will be easy to get a global optimal solution.



<p><b>Input:</b> Data set of <math>n</math> items in <math>D</math>-dimensional space <math>X = (x_1, x_2, \dots, x_n)</math>, the number of clusters <math>K</math>.</p> <p><b>Output:</b> <math>K</math> clusters <math>C = (C_1, C_2, \dots, C_K)</math></p>
<p><b>1. Setting model parameters</b></p> <p><math>\emptyset = 2.05 \Rightarrow w_p = w_g = 1.4962</math>; <i>damping rate</i> (<math>w_v</math>): 0.99</p> <p>Set minimum and maximum value for <math>P_I</math> and <math>v_I</math>:</p> <p>(<math>\min_{P_I}, \max_{P_I}, \min_{v_I}, \max_{v_I}</math>)</p>
<p><b>2. PSO_kmeans algorithm:</b></p> <p><b>While</b> <i>Gbest</i> has not change in <u>max_iteration=500</u>:</p> <p><b># PSO iteration</b></p> <p>For <math>I=1,2,\dots,N</math>, all particles, do:</p> <p style="padding-left: 20px;"><b>Update velocities</b></p> <p style="padding-left: 40px;">* With equation (5) calculate new velocity for all Particles</p> <p style="padding-left: 20px;"><b>Apply velocity limits</b></p> <p style="padding-left: 40px;">* Restrict velocities with <math>\max_{v_I}</math> and <math>\min_{v_I}</math>:</p> <p style="padding-left: 60px;"><math>v_I = \min\{\max\{v_I, \min_{v_I}\}, \max_{v_I}\}</math></p> <p style="padding-left: 20px;"><b>Update positions</b></p> <p style="padding-left: 40px;">* With equation (6) calculate new position for all Particles</p> <p style="padding-left: 20px;"><b>Apply velocity mirror effect</b></p> <p style="padding-left: 40px;">* For any dimension (<math>d</math>) which <math>d</math>th element of <math>P_I</math> is out of solution space (<math>P_{Id} &gt; \max_{P_I}</math> or <math>P_{Id} &lt; \min_{P_I}</math>): Substitute <math>d</math>-th element of velocity with its mirror value <math>v_{Id} = -v_{Id}</math></p> <p style="padding-left: 20px;"><b>Apply positions limits</b></p> <p style="padding-left: 40px;">* Return Particle positions into the solution space:</p> <p style="padding-left: 60px;"><math>P_I = \min\{\max\{P_I, \min_{P_I}\}, \max_{P_I}\}</math></p> <p style="padding-left: 20px;"><b>Update Pbest</b></p> <p style="padding-left: 40px;">* If <math>f(P_I^{t+1}) &lt; f(P_{best}_I^t)</math>:</p> <p style="padding-left: 60px;"><math>best_I^{t+1} = P_I^{t+1}</math></p> <p style="padding-left: 40px;">Else: <math>P_{best}_I^{t+1} = P_{best}_I^t</math></p> <p style="padding-left: 20px;"><b>Update Gbest</b></p> <p style="padding-left: 40px;">Assign best Pbest to the Gbest</p> <p><b># k-means Iterations:</b></p> <p>For all particles, <math>I = 1, 2, \dots, N</math> do:</p> <p>For <math>i = 1, 2, \dots, n</math> do:</p> <p style="padding-left: 20px;">* assign <math>x_i</math> to the nearest cluster <math>C_{jI}</math> :</p> <p style="padding-left: 40px;"><math>\operatorname{argmin} \{d(x_i, c_{jI})\}_{j=1,2,\dots,K}</math></p> <p style="padding-left: 20px;">* Update center of clusters:</p> <p style="padding-left: 40px;">○ <math>c_{jI} = \frac{1}{ C_{jI} } \sum_{x_i \in C_{jI}} x_i, j = 1, 2, \dots, K</math></p> <p style="padding-left: 40px;">○</p> <p><b># Adjust the velocity with the inertia damping rate</b></p>

Figure 3. The pseudo code of hybrid k-means clustering with PSO

The following steps briefly explain the proposed hybrid k-means clustering with PSO as indicated in Figure 3. First of all, to assign initial particles values, we run k-means for N times to create a population with N particles. After a k-mean convergence, we set results as the initial position of the particles.

$$population = [P_1, P_2, \dots, P_N]$$

which means we have N particles,  $P_1, P_2, \dots, P_N$ , each one is an aggregate of K position of cluster centers  $\{c_{1I}, c_{2I}, \dots, c_{KI}\}, I = 1, 2, \dots, N$ . Then, an iterative process which contains PSO and k-means algorithms starts until reaching stable clusters in a predefined maximum iteration.

In PSO iteration, first, we update the velocity of particles with the equation (5). For the first iteration, we consider zero velocity for all particles, initial positions as the best, and cost of the best position among the population as the *Gbest*.

After each velocity updating, we limit velocity with our predefined limit values. Then, with equation (6) the position of particles is updated. New positions could be out of the solution space which can lead to wrong solutions. So, for enhancing the method, we inverse the wrong direction of particle movements. To do this, in our D-dimensional solution space, we substitute velocities with their mirror value, in dimensions in which particles' position is out of the solution space. Actually, by implementing this mirror effect, we use intelligent velocities for the next iteration.

Besides, the positions of particles are bonded with predefined min and max values. In this research, we choose the following values as bond values:

$$max\_v_I = 0.1 * (\max\_P_I - \min\_P_I) \quad (7)$$

$$\min\_v_I = -max\_v_I \quad (8)$$

where  $max\_P_I$  and  $min\_P_I$  are the maximum and minimum position value of our dataset.

After updating the position and velocity of particles intelligently, we update best personal and global values. By utilizing a Euclidean distance objective (cost) function, we evaluate each particle:

$$f(P_I) = \sum_{j=1}^K \sum_{x \in I} d(x, c_{jI}), I = 1, 2, \dots, N \quad (9)$$

where  $x$  is an instance,  $c_j$  is the center of  $C_j$  and  $d$  denotes distance.

Individual and global best positions are updated by comparing the newly evaluated fitness against the previous individual and global best values and replacing the best fitness values and positions as necessary.

If the current cost of the new particle is the best in comparison with the old one ( $Pbest_i^t$ ), then we keep this new position as  $Pbest_i^{t+1}$ . The particle which has the best cost among all particles is selected and if this cost is better than the old  $Gbest^t$  value, then its position will be assigned as the new  $Gbest^{t+1}$ . Otherwise, the old  $Pbest$  and  $Gbest$  will remain as the best particle and global position, respectively. To alter the PSO iteration, we run K-mean for all particles and reassign instances ( $x_i$ ) to the new centres. After updating clustering, we can update the centers' position by averaging values of instances that belong to the cluster. Actually, in each iteration, the Gbest which has the best cost of our solution between particles should be checked to see whether it has reached a constant position or not.

For setting the model parameters, Clerc and Kennedy (2002) demonstrated that the implementation of properly defined constriction coefficients could significantly improve the performance of PSO. Accordingly, the parameters  $w_p$  and  $w_g$  are set as presented in equations 10 and 11:

$$w_p = w_g = h\phi \quad (10)$$

$$h = \frac{2}{\phi - 2 + \sqrt{\phi^2 - 4\phi}} \quad (11)$$

while considering certain conditions, different values can be assigned to the variable  $\phi$ , in this study the assigned value for  $\phi$  is equal to 2.05. For more information about the mathematical foundations of constriction coefficients, read Clerc and Kennedy (2002)

**Inertia Weight Damping Ratio:** Inertia weight,  $w_v$ , is a control parameter used to adjust the impact of the previous velocities on the current one. Thus, it affects the tradeoff between the particles' global and local exploration abilities. It is recommended that large inertia weights be used for the initial stages that can be gradually decreased over time (iterations). To this end, a damping ratio equal to 0.99 has been considered in the algorithm.

### 3.3. Measures

Since clustering is unsupervised learning, it is extremely important to evaluate the strengths and weaknesses of any given clustering algorithm compared to others. To this end, we need suitable methods for assessing clustering stability and tendency.

Similarity and Distance (dissimilarity) are two foundations for constructing clustering algorithms (D. Xu & Tian, 2015). In most cases, distance is a more suitable measure for dealing with

quantitative data features. For qualitative data features, on the other hand, similarity (e.g. Jaccard similarity) is usually preferred (R. Xu & Wunsch, 2005). There are some well-known measures that different studies have used for their comparison and evaluation of proposed clustering techniques. After a comprehensive literature review, Figueiredo, et al. (2019) mentioned that there are some studies that showed some of the metrics such as Davies–Bouldin, Rand-index, F-measure, Entropy-based and Mutual Information are more suitable for assessing the quality of clustering results. For example, Dunn and Davies-Bouldin (Karmitsa, Bagirov, & Taheri, 2018; Wiwie, Baumbach, & Röttger, 2015), Rand (Karmitsa, et al., 2018; Ros & Guillaume, 2019; Wiwie, et al., 2015), Jaccard (Pham, Siarry, & Oulhadj, 2018; Wiwie, et al., 2015), and Fowlkes-Mallows (Wiwie, et al., 2015) are used in different studies to evaluate the results.

In general, the methods for clustering validation can be categorized into two classes:

*External measures:* An external evaluation compares the predicting labels with the actual labels. Though this form of assessment is objective, it requires a priori data structure, (Kou, et al., 2014). Some instances of external measures are the Rand indicator, adjusted methods for Rand indicator, Jaccard indicator, Fowlkes–Mallows indicator, and Confusion matrix;

*Internal measures:* Internal measures judge clustering algorithms based on the resulting clusters' structure. An algorithm is considered suitable if the resulting clusters demonstrate high intra-class and low inter-class similarities. Some instances of internal measures are the Davies–Bouldin indicator, Dunn indicator and Silhouette coefficient;

Due to utilizing different approaches in designing the experiments, their evaluation are also distinct. In the first experiment, five external measures, including the Rand index, Hubert and Arabie's adjusted Rand index, Morey and Agresti adjusted Rand index, Fowlkes and Mallows index, and Jaccard coefficient are utilized. To evaluate the second experiment; however, two internal measures, namely Dunn's index and the Davies\_Bouldin index are employed, and the optimal number of clusters is obtained based on the results of the internal measures. Afterwards, the external measures used in the first experiment are employed to evaluate the algorithms.

In the following, the measures utilized in this study are illustrated.

### **3.3.1. Rand Index**

The Rand index always lies between 0 and 1. If the two partitions agree perfectly, it takes the value 1 (Yeung & Ruzzo, 2001). This index measures the percentage of decisions that are correct and penalizes both false negative and false positive decisions with equal weight.

Let  $C(x_i)$  and  $C'(x_i)$  define the class and the cluster of an instance  $x_i$ , then SS, SD, DS, and DD are defined as follows:

$$SS = [(x_i, x_j) | C(x_i) = C(x_j) \text{ and } C'(x_i) = C'(x_j)]$$

$$SD = [(x_i, x_j) | C(x_i) = C(x_j) \text{ and } C'(x_i) \neq C'(x_j)]$$

$$DS = [(x_i, x_j) | C(x_i) \neq C(x_j) \text{ and } C'(x_i) = C'(x_j)]$$

$$DD = [(x_i, x_j) | C(x_i) \neq C(x_j) \text{ and } C'(x_i) \neq C'(x_j)]$$

The Rand index is defined as equation 12:

$$R = \frac{|SS| + |DD|}{|SS| + |SD| + |DS| + |DD|} \quad (12)$$

### 3.3.2. Fowlkes and Mallows index (FM)

This measure could be either between two clustering results or a clustering and a benchmark. Since the index is proportional to the number of true positives, a higher value indicates a greater similarity between the clusters and the benchmark. One of the advantages of this index is that it performs in a satisfactory way in case of adding noise to a data set (Fowlkes & Mallows, 1983). Fowlkes and Mallows' index is defined as equation 13:

$$FM = \sqrt{\frac{|SS|}{|SS| + |SD|} \times \frac{|SS|}{|SS| + |DS|}} \quad (13)$$

### 3.3.3. Jaccard coefficient

The Jaccard coefficient is mostly used in order to measure the similarity of clusters (Hennig, 2008) and similarity is calculated with equation 14 (Huang, 2008):

$$J = \frac{|SS|}{|SS| + |SD| + |DS|} \quad (14)$$

### 3.3.4. Morey and Agresti adjusted Rand index (MA)

Morey and Agresti (1984) demonstrated that the Rand Index is extremely dependent upon the number of clusters. In order to resolve this shortcoming, they proposed a modified version known as the Morey and Agresti adjusted Rand index.

For two partition  $C$  and  $C'$  with  $K$  and  $K'$  subset clusters respectively, contingency Table1 can be created to show the overlap between them:

Table 1: Co-occurrence between two partitions

		$C'$					
		Cluster	$C'_1$	$C'_2$	.....	$C'_{K'}$	Sums
$C$	$C_1$	$n_{11}$	$n_{12}$	.....	$n_{1K'}$	$n_{1*}$	
	$C_2$	$n_{21}$	$n_{22}$	.....	$n_{2K'}$	$n_{2*}$	
	$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	
	$C_K$	$n_{K1}$	$n_{K2}$	.....	$n_{KK'}$	$n_{K*}$	
Sums		$n_{*1}$	$n_{*2}$	.....	$n_{*K'}$	$n_{**} = n$	

where  $n_{ij}$  indicates the number of instances which belong to the  $i^{\text{th}}$  cluster of  $C$  and the  $j^{\text{th}}$  cluster of  $C'$ .

Morey and Agresti's adjusted Rand index is defined as follows:

$$MA = \frac{\sum_{i=1}^K \sum_{j=1}^{K'} \binom{n_{ij}}{2} - t_3}{\frac{1}{2}(t_1 + t_2) - t_3} \quad (15)$$

where:

$$t_1 = \sum_{i=1}^K \binom{n_{i*}}{2}, \quad t_2 = \sum_{j=1}^{K'} \binom{n_{*j}}{2}, \quad t_3 = \frac{t_1 * t_2}{n^2} \quad (16)$$

$n_{i*}$  and  $n_{*j}$  are respectively the row and the column marginal of the contingency table.

### 3.3.5 Hubert and Arabie's adjusted Rand index (HA)

Proposed by Hubert and Arabie in 1985 and sometimes referred to simply as the Adjusted Rand index, this index allows a more appropriate quantitative evaluation of the consistency between two partitions (Carrico, et al., 2006). It is defined in equations 17 and 18:

$$HA = \frac{\sum_{i=1}^K \sum_{j=1}^{K'} \binom{n_{ij}}{2} - t'_3}{\frac{1}{2}(t_1 + t_2) - t'_3} \quad (17)$$

Where

$$t'_3 = \frac{t_1 * t_2}{\binom{n}{2}} \quad (18)$$

### 3.3.6 Dunn's index

Introduced by Dunn in 1974, Dunn's index is an internal evaluation scheme. The objective is to identify compact sets of clusters, where the means of different clusters are as far apart as possible. Dunn's index is defined in equation 19:

$$DI = \frac{\min_{1 \leq i \leq j \leq K} d(c_i, c_j)}{\max_{1 \leq s \leq K} \Delta_s} \quad (19)$$

where  $c_i$  and  $c_j$  are centres of  $C_i$  and  $C_j$  clusters respectively and  $\Delta_s$  is the maximum distance between two instances belonging to cluster  $C_s$ .

### 3.3.7 Davies\_Bouldin index

Davies\_Bouldin index is an internal evaluation measure. This method is a function of the ratio of the summation of within-cluster scatter to between-cluster separation, and as we want to maximize the between-cluster separation and minimize the within-cluster scatter, the aim is to minimize this measure (Ray & Turi, 1999). The index is defined in equation 20:

$$DB(C) = \frac{1}{K} \sum_{i=1, i \neq j}^K \max \left\{ \frac{\bar{d}(C_i) + \bar{d}(C_j)}{d(c_i, c_j)} \right\} \quad (20)$$

where  $\bar{d}(C_i)$  is the measure of scatter within the cluster for cluster  $C_i$  which is the average distance of all instances in cluster  $C_i$  to their centre  $c_i$ , similarly  $\bar{d}(C_j)$  is the measure of scatter within the cluster for cluster  $C_j$ ,  $d(c_i, c_j)$  is a distance of two cluster centres  $c_i$  and  $c_j$  that is called the inter-cluster distance, and  $K$  is the number of clusters.

It is worth mentioning that both lists of clustering algorithms and clustering measures have more potential candidate methods (e.g. graph based clustering algorithms (chameleon), or normalized mutual information criterion), but we select based on the popularity of the algorithms in the literature and the robustness of the methods in different datasets. Moreover, our proposed approach

can be computed in any cases, with no limitation the number of clustering algorithms or clustering measures.

### **3.4. Group MCDM**

MCDM algorithms deal with the decision making process in the presence of multiple, and usually conflicting, objectives. In most cases, the objectives are conflicting and hence, the solution is highly contingent upon the preferences of the decision-maker some level of compromise is inevitable (Pohekar & Ramachandran, 2004).

In general, the steps of MCDM are as follows (Opricovic & Tzeng, 2004):

1. Establishing system evaluation criteria;
2. Developing alternatives for attaining the goals;
3. Evaluating alternatives in terms of criteria;
4. Applying a normative multi criteria analysis method;
5. Accepting one alternative as optimal;

In this research, since five evaluation measures are used for evaluation of clustering algorithms, it can be modeled as a MCDM problem. Thus, MCDM techniques can be used to select the best algorithm. Moreover, evaluation of each data set in clustering models' selection is an independent MCDM problem, and since we have 10 data sets, utilizing a group MCDM approach seems to be a suitable approach for comparing the models.

In the other words, 10 standard data sets are utilized as the case studies and five measures are calculated on each data set. If we consider each data set as an expert and each measure as a criteria, the problem becomes a group MCDM.

In this study, a number of group MCDM methods was nominated and eventually, three methods including TOPSIS, COPRAS, and WSM were selected. The reason we have selected the aforementioned methods is that these algorithms have the least correlation among all the candidate MCDM algorithms.

#### **3.4.1. Group TOPSIS**

Among the vast number of MCDM methods, TOPSIS (technique for order preference by similarity to ideal solution) is one of the most applied ones. Developed by Hwang and Yoon (Hwang &



Yoon, 1981), this technique suggests that the ranking of alternatives should be based on the shortest distance from the positive ideal solution (PIS) and the farthest distance from the negative ideal solution (NIS) (Opricovic & Tzeng, 2004). TOPSIS takes the distances to both PIS and NIS into account, and the preference order is ranked by combining the distances (Shih, 2008).

In the case of group decision making; however, the alternatives are evaluated by more than one DM. In other words, one multi-attribute object is represented in a number of versions. For instance, object characteristics have been measured in different conditions, and a number of experts independently evaluated objects upon intended criteria (Petrovsky, 2014).

It should be noted that in group decision making, DMs' opinions may have different values (weights) and DMs can form different judgments on the criteria. The aforementioned features create more complication rather than simple decision making process.

The final decision is made through aggregating group members' preferences on alternatives with regard to their weights and judgments on selected criteria. Most of the group decision making methods use utility aggregation strategies to derive a consensus preference (Krohling & Campanharo, 2011).

Wei-guo and Hong (2007) extends the TOPSIS framework to group decision making situations by presenting two notions: group positive-ideal and group negative-ideal. In this study, a group TOPSIS method is utilized based on the same concept.

Consider a MCDM problem characterized by the following matrix:

$$\begin{matrix} & F_1 & F_2 & \dots & F_n \\ \begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{matrix} & \begin{bmatrix} z_{11} & z_{12} & \dots & z_{1n} \\ z_{21} & z_{22} & \dots & z_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ z_{m1} & z_{m2} & \dots & z_{mn} \end{bmatrix} \end{matrix}$$

$$W = [w_1, w_2, \dots, w_n]$$

where  $A_1, A_2, \dots, A_m$  are possible alternatives that decision makers can choose,  $F_1, F_2, \dots, F_n$  are sets of criteria,  $z_{ij}$  is the rating of alternative  $A_i$  with respect to criterion  $F_j$ , and  $w_j$  is weight of criterion  $F_j$ .

The procedure of group TOPSIS consists of the following steps:

Step 1: After calculating all criteria for each data set, normalize the resulting (decision) matrix (see equation 21).

$$r_{ij} = \frac{z_{ij}}{\sqrt{\sum_{i=1}^m z_{ij}^2}} \quad i = 1, 2, \dots, m \text{ and } j = 1, 2, \dots, n. \quad (21)$$

Step 2: Create weighted normalized matrix:

$$t_{ij} = r_{ij} * w_j \quad i = 1, 2, \dots, m \text{ and } j = 1, 2, \dots, n. \quad (22)$$

Step 3: Determine the positive ideal,  $I^+$ , and negative ideal,  $I^-$ , solutions as follows:

$$I^+ = \{(max(t_{ij} | i = 1, 2, \dots, m) | j \in J^-), (min(t_{ij} | i = 1, 2, \dots, m) | j \in J^+)\} = \{t_j^+ | j = 1, 2, \dots, n\} \quad (23)$$

$$I^- = \{(min(t_{ij} | i = 1, 2, \dots, m) | j \in J^-), (max(t_{ij} | i = 1, 2, \dots, m) | j \in J^+)\} = \{t_j^- | j = 1, 2, \dots, n\} \quad (24)$$

Where  $J^-$  and  $J^+$  are the index set of the benefit and cost attributes, respectively.

Step 4: Using the Euclidean distance, calculate distance between target alternative  $i$  and positive and negative ideal solutions:

$$d_i^+ = \sqrt{\sum_{j=1}^n (t_{ij} - t_j^+)^2}, i = 1, 2, \dots, m \quad (25)$$

$$d_i^- = \sqrt{\sum_{j=1}^n (t_{ij} - t_j^-)^2}, i = 1, 2, \dots, m \quad (26)$$

Step 4: By averaging individual separation measures, calculate group separation measures ( $gd_i^+$  and  $gd_i^-$ )

Step 5: Calculate the relative closeness to the ideal solution.

$$C\_index = \frac{gd_i^+}{gd_i^- + gd_i^+} \quad i = 1, 2, \dots, m \quad (27)$$

Step 6: Rank the alternatives according to the C\_index.

### 3.4.1. Group COPRAS

The method of multiple criteria complex proportional assessment (COPRAS) is a MCDM algorithm for multi-criteria evaluation of both maximizing and minimizing criteria values (Kaklauskas, et al., 2006). Ranking alternatives using COPRAS assumes dependence of the

significance and priority of alternatives on a system of criteria. The process of the group COPRAS method used in this study consists of the following steps:

Step 1: Construct the normalized decision-making matrix.

For normalization in the COPRAS method the following formula is used:

$$r_{ij} = \frac{z_{ij}}{\sum_{i=1}^m z_{ij}}, i = 1, 2, \dots, m \text{ and } j = 1, 2, \dots, n \quad (28)$$

where  $z_{ij}$  is the result of the  $i$ th alternative with respect to the  $j$ th criterion,  $r_{ij}$  is its normalized value, and  $m$  and  $n$  are number of alternatives and criteria respectively.

Step 2: Calculate the sums of weighed normalized criteria values for positive criteria (criteria that are preferred to be maximized) and negative value;

$P_{+i}$  are positive and  $P_{-i}$  are negative criteria calculated as follows:

$$P_{+i} = \sum_{j=1}^S r_{ij} * w_j, i = 1, 2, \dots, m \quad (29)$$

$$P_{-i} = \sum_{j=S+1}^n r_{ij} * w_j, i = 1, 2, \dots, m \quad (30)$$

where  $S$  is the number of positive criteria;  $n$  is total number of criteria; and  $w_j$  is weight of the  $j$ -th criterion.

Step 3: Calculate the relative weight of each alternative

The relative weight  $Q_i$  of  $i$ -th alternative is calculated as follows:

$$Q_i = P_{+i} + \frac{\sum_{i=1}^m P_{-i}}{P_{-i} \sum_{i=1}^m P_{-i}} \quad (31)$$

Step 4: Calculate the group relative weight.

The group relative weight of each alternative is determined by calculating the arithmetic means of relative weights for each alternative.

Step 5: Rank the alternatives.

The priority of alternatives is determined based on their relative weight. The alternative with a higher relative weight is assigned a higher rank.

### 3.4.1. Group WSM

The Weighted Sum Model (WSM) is one of the earliest and robust known MCDM methods. Suppose that there exist  $m$  alternatives,  $n$  decision criteria, and  $w_j$  indicates the relative weight of the criterion  $F_j$ . Furthermore,  $z_{ij}$  is the performance value of alternative  $A_i$  in terms of criterion  $F_j$ . Then, the WSM score is calculated as follows (Triantaphyllou & Mann, 1989):

$$A_i^{WSM\ Score} = \sum_{j=1}^n w_j z_{ij} \quad i = 1, 2, \dots, m. \quad (32)$$

For group WSM, the group WSM score of each alternative is equivalent to the arithmetic mean of all WSM scores.

### 3.5. Borda count method

The Borda count is a preference based election procedure in which voters rank the alternatives in order of preference (Van Erp & Schomaker, 2000). It ranks the alternatives and elects a representative by aggregating the sum of each alternative's weighted preference scores. The steps are as follows:

Step 1: Each alternative is ranked by each voter.

Step 2: Each alternative is assigned a number of points equal to the number of alternatives which are ranked lower than them, so that an alternative receives  $n - 1$  points for the first preference,  $n - 2$  for the second, and so on. This procedure is done for each voter.

Step 3: For each alternative, the sum of its points is calculated.

Step 4: Based on the total score of alternatives, which are calculated in step 3, the alternatives would be ranked.

We use the Borda count method to assign an overall rank to the algorithms.

It is valuable to mention that it is not recommended to use one specific clustering model for all different datasets, while it is known that k-means and its variations (as well as k-medoids, PAM etc.) may be effective only in the case of spherical point clouds, and clusters of more complex shapes are better identified by density-based algorithms (like DBSCAN, nearest neighbors clustering etc.). That is why we are interested in find the suitable clustering model for each dataset. However, with the concept of group MCDM, we can explicitly evaluate which algorithm generally

works well within the benchmarks, and if one implements that clustering model instead of analysing all the models one by one, then it will have more probability to obtain the best clusters. Therefore, group MCDM is valuable in big datasets for decreasing the computation time.

#### 4. Experimental Results

The experiment design conducted in this paper consists of two parts. The first experiment contains the results of the comparison of six clustering methods using external validation, based on actual number of classes in each data set. Figure 4 illustrates the tools used in experiment 1. In the second experiment; however, all the data sets are clustered from two to nine clusters. For each clustering output we calculated two internal measures including the Dunn and Davies-Bouldin measures. Based on the results of these two measures, the number of clusters for the data sets are determined. Then, according to the determined number of clusters, the external measures are applied (see Figure 5 for the tools used in experiment 2).

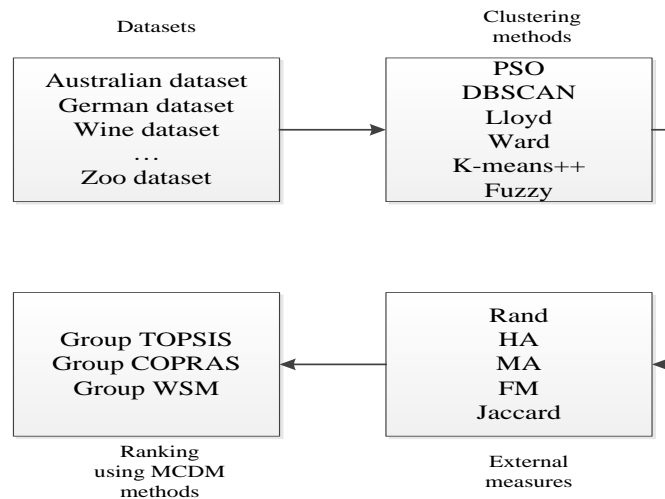


Figure 4. Experiment 1 analytical tools

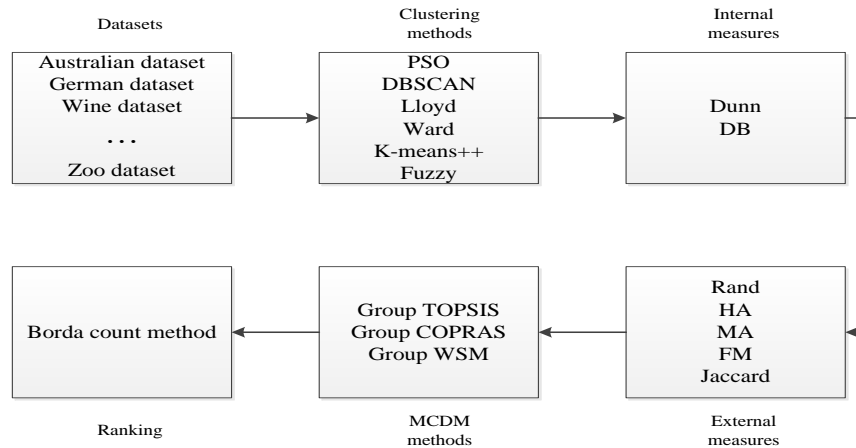


Figure 5. Experiment 2 analytical tools

In order to illustrate the methodology, in both sections we used 10 different data sets from UCI (Bache & Lichman, 2013) and clustered them using six methods, namely PSO clustering, Lloyd's algorithm, DBSCAN algorithm, Fuzzy clustering, K-means ++, and Ward method. The R programming software is used for both experiments. For the sake of the generality of our study, a variety of data such as a large size TV data set (TV News Channel Commercial Detection Dataset) (Vyas, Kannao, Bhargava, & Guha, 2014), a medium size data set (Mushroom) (Schlimmer, 1981), large number of attributes (Heart Disease) and large number of actual classes (Mushroom) have been considered. Table 2 briefly demonstrates the specification of each data set.

Table 2: The specifications of data sets

NAME	NUMBER OF RECORDS	NUMBER OF ATTRIBUTES	ATTRIBUTE CHARACTERISTICS	ACTUAL CLASSES
IRIS	150	4	Real	3
AUSTRALIAN CREDIT APPROVAL	690	14	Categorical, Integer, Real	2
BREAST CANCER WISCONSIN	688	10	Integer	2
GLASS IDENTIFICATION	214	10	Real	6
HEART DISEASE	303	75	Categorical, Integer, Real	4
GERMAN CREDIT DATA	1000	20	Categorical, Integer	2
TV NEWS CHANNEL COMMERCIAL DETECTION DATASET	129685	12	Real	2
PIMA	768	8	Integer, Real	2
ZOO	101	17	Categorical, Integer	10
MUSHROOM	8124	22	Categorical	23

#### 4.1. External validation

As mentioned in section 2, external evaluation compares the predicting labels with the actual labels. So, in this step we clustered all the 10 data sets using six different methods. Then we compared the cluster labels with the actual labels.

Among the six clustering algorithms in this experiment just Fuzzy and Ward are not sensitive to the starting points (initial centers) but others are. In order to reduce the impact of initial centers, we ran all the algorithms 10 times with different random seeds for each run. Using this approach allows us to estimate the performance of the algorithms with higher precision.

Eventually, five different indices, Rand, HA, MA, FM and Jaccard coefficient, were used for the comparison to compare the actual class feature with the results of clustering labels. Table 3, contains the values of five indices on the six clustering methods over the utilized data sets. For each data set, the bold values show the top two techniques and the gray cells represent the bottom two methods based on each evaluation indices.

Table 3: The values of 5 indices on all of 6 clustering methods over the utilized data sets (Experiment 1)

	Methods	Rand	HA	MA	FM	Jaccard
Australia	PSO clustering	0.692	0.384	0.385	0.695	0.537
	DBSCAN	0.621	0.243	0.244	0.629	0.465
	Lloyd's K-means	0.621	0.243	0.244	0.629	0.465
	Ward	0.608	0.216	0.218	0.614	0.443
	K-means ++	<b>0.752</b>	<b>0.504</b>	<b>0.504</b>	<b>0.753</b>	<b>0.604</b>
	Fuzzy clustering	<b>0.703</b>	<b>0.406</b>	<b>0.407</b>	<b>0.704</b>	<b>0.544</b>
Breast Cancer	PSO clustering	<b>0.926</b>	<b>0.850</b>	<b>0.850</b>	<b>0.932</b>	<b>0.873</b>
	DBSCAN	0.920	0.839	0.839	0.928	0.865
	Lloyd's K-means	0.920	0.839	0.839	0.928	0.865
	Ward	<b>0.926</b>	<b>0.850</b>	<b>0.851</b>	<b>0.931</b>	<b>0.871</b>
	K-means ++	0.920	0.839	0.839	0.928	0.865
	Fuzzy clustering	0.910	0.818	0.818	0.919	0.849
Glass	PSO clustering	0.804	<b>0.448</b>	<b>0.457</b>	<b>0.578</b>	<b>0.402</b>
	DBSCAN	0.762	0.334	0.345	0.490	0.323
	Lloyd's K-means	0.756	0.319	0.329	0.479	0.314
	Ward	<b>0.852</b>	<b>0.587</b>	<b>0.593</b>	<b>0.686</b>	<b>0.517</b>
	K-means ++	0.786	0.424	0.432	0.566	0.395
	Fuzzy clustering	<b>0.807</b>	0.443	0.452	0.571	0.390
Heart	PSO clustering	0.630	0.112	0.119	0.360	0.212
	DBSCAN	0.634	0.121	0.128	0.367	0.217
	Lloyd's K-means	0.633	0.121	0.128	0.369	0.220
	Ward	0.583	0.014	0.021	0.300	0.173
	K-means ++	<b>0.646</b>	<b>0.147</b>	<b>0.154</b>	<b>0.385</b>	<b>0.229</b>
	Fuzzy clustering	<b>0.656</b>	<b>0.224</b>	<b>0.229</b>	<b>0.480</b>	<b>0.315</b>
German	PSO clustering	0.514	0.012	0.013	0.571	0.400
	DBSCAN	0.527	0.023	0.024	0.597	0.425
	Lloyd's K-means	0.527	0.023	0.024	0.598	0.426
	Ward	<b>0.540</b>	<b>0.044</b>	<b>0.045</b>	<b>0.616</b>	<b>0.444</b>
	K-means ++	<b>0.539</b>	<b>0.051</b>	<b>0.052</b>	<b>0.606</b>	<b>0.435</b>
	Fuzzy clustering	0.500	-0.001	0.000	0.538	0.367
Iris	PSO clustering	<b>0.884</b>	<b>0.739</b>	<b>0.743</b>	<b>0.826</b>	<b>0.704</b>
	DBSCAN	0.843	0.660	0.664	0.782	0.643
	Lloyd's K-means	0.843	0.658	0.662	0.780	0.641
	Ward	0.874	0.720	0.723	0.816	0.688
	K-means ++	0.874	0.716	0.720	0.811	0.682
	Fuzzy clustering	<b>0.880</b>	<b>0.729</b>	<b>0.732</b>	<b>0.819</b>	<b>0.693</b>
TV News	PSO clustering	<b>0.568</b>	<b>0.135</b>	<b>0.138</b>	<b>0.621</b>	<b>0.446</b>
	DBSCAN	0.532	0.052	0.055	0.596	0.422
	Lloyd's K-means	0.530	0.048	0.052	0.595	0.421
	Ward	0.500	-0.086	-0.083	0.610	<b>0.438</b>
	K-means ++	0.532	0.052	0.055	0.596	0.422
	Fuzzy clustering	<b>0.557</b>	<b>0.111</b>	<b>0.114</b>	<b>0.612</b>	0.437
PIMA	PSO clustering	<b>0.557</b>	<b>0.108</b>	<b>0.109</b>	0.591	0.419
	DBSCAN	0.556	0.102	0.103	<b>0.597</b>	<b>0.426</b>
	Lloyd's K-means	<b>0.556</b>	0.102	0.103	<b>0.598</b>	<b>0.426</b>
	Ward	0.537	0.066	0.067	0.576	0.404
	K-means ++	0.556	0.102	0.103	<b>0.597</b>	<b>0.426</b>
	Fuzzy clustering	0.555	<b>0.107</b>	<b>0.108</b>	0.582	0.410
Mushroom	PSO clustering	0.938	0.860	0.862	0.907	0.830
	DBSCAN	0.938	0.860	0.862	0.907	0.830
	Lloyd's K-means	0.904	0.793	0.795	0.869	0.773
	Ward	<b>0.955</b>	<b>0.899</b>	<b>0.900</b>	<b>0.933</b>	<b>0.875</b>
	K-means ++	<b>0.941</b>	<b>0.869</b>	<b>0.870</b>	<b>0.913</b>	<b>0.839</b>
	Fuzzy clustering	0.933	0.850	0.851	0.900	0.818
Zoo	PSO clustering	0.873	0.601	0.617	0.688	0.516
	DBSCAN	0.881	0.627	0.641	0.709	<b>0.543</b>
	Lloyd's K-means	0.850	0.545	0.562	0.644	0.477
	Ward	<b>0.904</b>	<b>0.697</b>	<b>0.709</b>	<b>0.768</b>	<b>0.606</b>
	K-means ++	None	None	None	None	None
	Fuzzy clustering	<b>0.885</b>	<b>0.636</b>	<b>0.650</b>	<b>0.718</b>	0.543

For instance, regarding the breast cancer data set, PSO and Ward algorithms outperformed other methods by having a higher Rand index of 0.926. Thus, one can claim that there is 0.926



probability that these two methods agree on a randomly chosen pair of cluster labels and actual class label. On this data set, Lloyd’s algorithm, DBSCAN, and K-means++ have very close clustering outputs as demonstrated in Figure 6. Finally, the Fuzzy clustering demonstrated the lowest performance on this data set as it has fallen in the bottom of the chart.

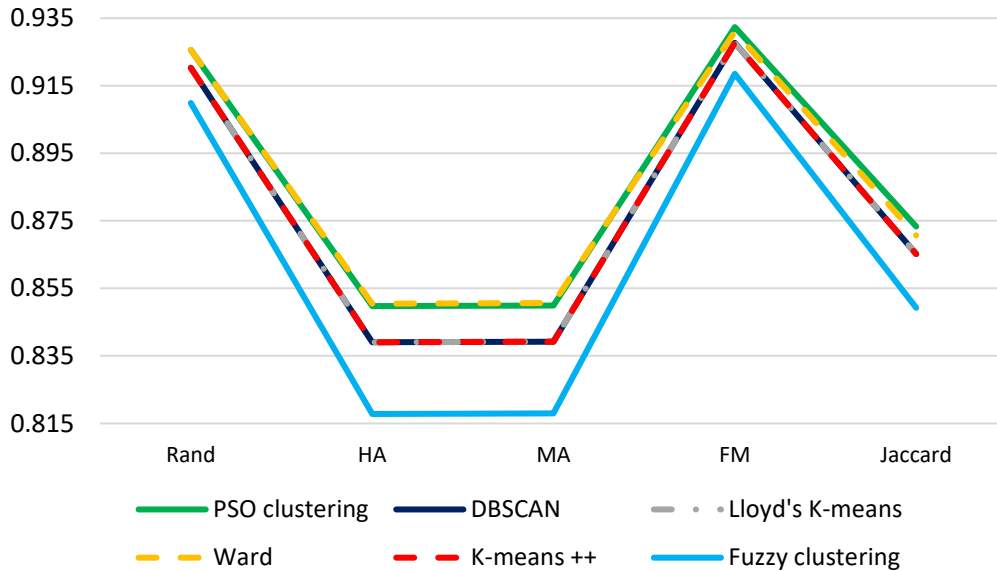


Figure 6. The performance of clustering methods according to external measures

Overall, it seems that Fuzzy clustering, PSO clustering, and K-means++ provide better results over different data sets. Although K-means++ has a good overall performance, it failed on the Zoo data set. Basically, K-means++ was not able to find different initial points for centers in this data set since the Zoo data set contains several Boolean features which makes it harder for K-means++ to find various starting points. Since evaluating the algorithms is hard at this stage, we utilized a MCDM approach to address the algorithm selection. The following section describes the resulted scoring output.

#### 4.1.1. Using MCDM to select the best Algorithm

As mentioned in section 3, the utilized algorithms are ranked using three MCDM methods for each data set. The results are shown in appendix A.

From the results it can be seen that regarding all MCDMs, the PSO algorithm outperformed the other algorithms in four data sets. Moreover, Ward and K-means++ algorithms outperformed the other algorithms regarding three and two data sets, respectively. Finally, regarding the Heart

disease data set, the Fuzzy algorithm outperformed the other algorithms. Lloyd and DBSCAN algorithms, however, are not selected as the best algorithm in any of the data sets used in the study.

Another interesting perspective, is to compare the algorithms regarding the real number of classes. Based on this concept, the data sets with between two and four classes are considered low class data sets and those with between six and ten are considered high class data sets. From this point of view, two data sets of Glass and Breast cancer are considered as high class and the rest are considered as low class ones. Regarding the high class data sets, the Ward algorithm outperformed the other. For all MCDM methods, the Ward algorithm was ranked first and second in the Glass and Breast cancer data sets, respectively. The PSO algorithm was ranked first and four in the aforementioned high class data sets using TOPSIS, and first and second in the aforementioned high class data sets using COPRAS and WSM. Therefore, it can be considered the second best algorithm. On the other hand, regarding the low class data sets, the PSO algorithm outranks the other algorithms.

Besides, the results for the fuzzy method are worthy of more consideration. While the fuzzy algorithm ranks first only on the Heart disease data set, in many cases it was ranked second. Using TOPSIS, the fuzzy algorithm was ranked second in six data sets. Using COPRAS and WSM, it was ranked second in five data sets. These results show the satisfactory performance of the fuzzy algorithm in experiment 1.

From the results and the explanations, it is fairly obvious that PSO and fuzzy algorithms are the only methods which perform in a satisfactory way in most cases. Using TOPSIS, COPRAS, and WSM, PSO and fuzzy algorithms were ranked amongst the two best methods in four and seven data sets, respectively. On the other hand, PSO and fuzzy algorithms were ranked amongst the two weakest methods in two and three data sets, respectively.

In addition to using the original MCDM methods, three group MCDM methods are utilized to rank the applied methods considering all the data sets simultaneously. The results of the group MCDM methods for the first experiment are presented in Table 4 and Table 5 as follows:

Table 4: Group MCDMs for experiment 1

ALGORITHM	GROUP TOPSIS			GROUP COPRAS	GROUP WSM
	$d_i^+$	$d_i^-$	$C\_index$	$Q_i$	$WSM\ Score$
<b>PSO</b>	0.1853	0.3979	0.6823	0.8644	0.1880
<b>DBSCAN</b>	0.2661	0.3163	0.5431	0.7877	0.1715
<b>LOYD</b>	0.2869	0.2961	0.5080	0.7413	0.1613
<b>WARD</b>	0.3268	0.2562	0.4395	0.8096	0.1764
<b>KMEANS++</b>	0.2369	0.3461	0.5936	0.5000	0.1077
<b>FUZZY</b>	0.1724	0.4107	0.7043	0.8973	0.1951

Table 5: The final rankings of 3 group MCDMs for experiment 1

ALGORITHM	GROUP TOPSIS	GROUP COPRAS	GROUP WSM
<b>PSO</b>	2	2	2
<b>DBSCAN</b>	4	4	4
<b>LOYD</b>	5	5	5
<b>WARD</b>	6	3	3
<b>KMEANS++</b>	3	6	6
<b>FUZZY</b>	1	1	1

The results of group MCDM methods for experiment 1 indicate that the Fuzzy clustering algorithm outperformed the other methods and the PSO algorithm is ranked second. By consideration of the  $C\_index$ , the  $Q_i$  index, and weighted sum average it can be seen that the differences between fuzzy and PSO are non-significant. The differences among these two algorithms and the other methods are significant; however, for instance, regarding group TOPSIS,  $C\_indexes$  for fuzzy, PSO and Kmeans++, which are ranked first, second and third respectively, are 0.704, 0.682, and 0.593. The difference between fuzzy and PSO is 0.022, while the difference between PSO and Kmeans++ is 0.089. Regarding group COPRAS,  $Q_i$  indexes for fuzzy, PSO and Kmeans++ are 0.897, 0.864, and 0.809, respectively. Finally, for group WSM, weighted sum averages for fuzzy, PSO and Kmeans++ are 0.195, 0.187, and 0.176, respectively. These results confirm that the fuzzy and PSO algorithms provided very close results and outperformed the other four utilized algorithms.

Among the benefits of using group MCDMs is their ability to avoid intuitive mistakes. For instance, since the PSO algorithm seems to be the best method at the first glance, due to its superior performance on four data sets out of 10. However, the utilized MCDMs proved that intuitive conclusion as specious. The three MCDM methods showed that the fuzzy algorithm, which was ranked first only on one data set, is the best algorithm due to its satisfactory results in most data

sets. This counter intuitive conclusion could readily be overlooked without utilizing the group MCDMs.

It is obvious from the Table 5 that the results of the three utilized techniques are highly in concordance. Therefore, one can claim that they are robust and no further analyses are necessary.

#### 4.2. Internal validation

The internal validation evaluates the quality of clustering results without considering any external information. In this study, we used the Dunn and Davies-Bouldin measures for evaluating the clustering quality. The higher value for the former and lower value for the latter indicate the existence of a possible pattern in the clustering with regard to different numbers of clusters. So, we clustered all the data sets from two to nine clusters and for each clustering output we calculated these two measures. The Dunn index was considered as the main criterion for selecting the number of clusters and, in case of ties (equal Dunn index for two different number of clusters), we used the Davies-Bouldin one to adjust our judgment. For instance, Table 6 contains the values of the Davies-Bouldin and Dunn indexes for PSO clustering and DBSCAN on the German data set.

Table 6: Internal measures for PSO and DBSCAN

CLUSTERS	PSO CLUSTERING		DBSCAN	
	Davies_Bouldin	Dunn	Davies_Bouldin	Dunn
2	8.302	0.0141	27.700	0.0131
3	7.780	0.0141	6.518	0.0151
4	8.635	<b>0.0152</b>	19.562	<b>0.0152</b>
5	6.639	0.0108	<b>4.994</b>	<b>0.0152</b>
6	<b>6.503</b>	<b>0.0152</b>	12.962	0.0143
7	4.797	0.0134	3.994	0.0134
8	6.415	0.0143	8.846	0.0144
9	8.837	0.0108	14.057	0.0108

Based on the Dunn index for PSO clustering, 4 and 6 clusters with a Dunn value of 0.0152 are optimal numbers of clusters on this data set. Nevertheless, since opting 6 clusters provides smaller Davies Bouldin value compared to 4 clusters, we selected 6 clusters as the optimal number of clusters. Similarly, regarding the DBSCAN algorithm, the Dunn index for 4 and 5 clusters are equal. The optimal number of clusters for this method is 5 because 5 clusters have a Davies Bouldin value of 4.99 which is significantly smaller than 19.56 for 4 clusters. After selecting the optimal

number of clusters, we re-ran all the methods and we calculated the 5 evaluation indices to assess the correctness of clustering. The results are summarized in the appendix in Table 7.

Table 7: The values of 5 indices on all of 6 clustering methods over the utilized data sets (Experiment 2)

	Methods	Rand	HA	MA	FM	Jaccard	Cluster
Australia	PSO clustering	0.608	0.216	0.218	0.614	0.443	2
	DBSCAN	0.526	0.051	0.052	0.538	0.368	2
	Lloyd's K-means	0.500	0.000	0.002	0.504	0.337	2
	Ward	0.608	0.216	0.218	0.614	0.443	2
	K-means ++	0.752	0.504	0.504	0.753	0.604	2
	Fuzzy clustering	0.605	0.216	0.217	0.491	0.287	6
Breast Cancer	PSO clustering	0.926	0.850	0.850	0.932	0.873	2
	DBSCAN	0.920	0.839	0.839	0.928	0.865	2
	Lloyd's K-means	0.920	0.839	0.839	0.928	0.865	2
	Ward	0.926	0.850	0.851	0.931	0.871	2
	K-means ++	0.920	0.839	0.839	0.928	0.865	2
	Fuzzy clustering	0.910	0.818	0.818	0.919	0.849	4
Glass	PSO clustering	0.788	0.501	0.506	0.655	0.480	3
	DBSCAN	0.757	0.423	0.429	0.597	0.420	4
	Lloyd's K-means	0.598	0.262	0.266	0.575	0.360	2
	Ward	0.607	0.261	0.265	0.565	0.356	2
	K-means ++	0.785	0.425	0.433	0.568	0.396	2
	Fuzzy clustering	0.604	0.272	0.275	0.580	0.365	2
Heart	PSO clustering	0.610	0.219	0.222	0.550	0.371	2
	DBSCAN	0.632	0.137	0.143	0.395	0.241	4
	Lloyd's K-means	0.642	0.176	0.182	0.435	0.276	4
	Ward	0.627	0.048	0.057	0.260	0.131	8
	K-means ++	0.649	0.154	0.160	0.390	0.232	5
	Fuzzy clustering	0.655	0.223	0.228	0.480	0.315	5
German	PSO clustering	0.485	0.029	0.030	0.440	0.264	6
	DBSCAN	0.461	0.005	0.006	0.373	0.204	5
	Lloyd's K-means	0.461	0.006	0.007	0.371	0.201	5
	Ward	0.540	0.044	0.045	0.616	0.444	2
	K-means ++	0.539	0.051	0.052	0.606	0.435	2
	Fuzzy clustering	0.440	-0.001	0.001	0.269	0.115	8
Iris	PSO clustering	0.776	0.568	0.571	0.771	0.595	2
	DBSCAN	0.776	0.568	0.571	0.771	0.595	2
	Lloyd's K-means	0.776	0.568	0.571	0.771	0.595	2
	Ward	0.776	0.568	0.571	0.771	0.595	2
	K-means ++	0.776	0.568	0.571	0.771	0.595	2
	Fuzzy clustering	0.764	0.540	0.544	0.750	0.572	4
TV News	PSO clustering	0.568	0.135	0.138	0.621	0.446	2
	DBSCAN	0.518	0.115	0.119	0.520	0.327	3
	Lloyd's K-means	0.475	0.046	0.051	0.463	0.276	2
	Ward	0.456	0.071	0.076	0.383	0.183	3
	K-means ++	0.518	0.115	0.119	0.520	0.327	3
	Fuzzy clustering	0.557	0.111	0.114	0.612	0.437	2
PIMA	PSO clustering	0.532	0.100	0.102	0.452	0.270	4
	DBSCAN	0.541	0.117	0.119	0.468	0.284	4
	Lloyd's K-means	0.507	0.072	0.074	0.354	0.171	9
	Ward	0.493	0.042	0.044	0.346	0.175	7
	K-means ++	0.551	0.138	0.140	0.474	0.286	5
	Fuzzy clustering	0.555	0.107	0.108	0.582	0.410	2
Mushroom	PSO clustering	0.941	0.869	0.870	0.913	0.839	19
	DBSCAN	0.895	0.751	0.754	0.831	0.700	17
	Lloyd's K-means	0.851	0.637	0.642	0.750	0.583	15
	Ward	0.784	0.434	0.443	0.602	0.372	16
	K-means ++	0.895	0.751	0.754	0.831	0.700	14
	Fuzzy clustering	0.842	0.612	0.617	0.732	0.554	15
Zoo	PSO clustering	0.882	0.630	0.644	0.712	0.540	2
	DBSCAN	0.867	0.582	0.598	0.673	0.493	2
	Lloyd's K-means	0.849	0.525	0.544	0.626	0.444	2
	Ward	0.841	0.641	0.648	0.771	0.595	4
	K-means ++	None	None	None	None	None	None
	Fuzzy clustering	0.885	0.636	0.650	0.718	0.543	2

According to Table 7, in the TV News data set which is considered as a large data set, PSO outperformed the other methods. Here, PSO achieved the 0.57 in Rand and 0.446 in Jaccard which clearly is higher than other methods. One interesting result happened on the Iris data set. All methods except Fuzzy clustering achieved the equal evaluation metrics and the same clustering pattern in the data.

Since in some cases we could not find the actual number of classes using different methods across different data sets, it might be interesting to investigate the accuracy of the utilized methods based on their ability to find the exact number of classes based on the Dunn and Davies-Bouldin index. Table 8 contains the difference of the actual number of classes in data sets and the numbers found by clustering evaluation measures.

*Table 8: The differences between actual number of classes and the number of classes determined by clustering methods*

	Australia	Breast Cancer	Glass	Heart	German	Iris	TV News	PIMA	Mush room	Zoo	STD
PSO clustering	0	0	-3	-2	4	-1	0	2	-4	-8	3.33
DBSCAN	0	0	-2	0	3	-1	1	2	-6	-8	3.45
Lloyd's K-means	0	0	-4	0	3	-1	0	7	-8	-8	4.61
Ward	0	0	-4	4	0	-1	1	5	-7	-6	3.91
K-means ++	0	0	-4	1	0	-1	1	3	-9	None	3.54
Fuzzy clustering	4	2	-4	1	6	1	0	0	-8	-8	4.70

Among all the methods, PSO clustering is the only one that could find the exact number of classes in four data sets. It seems all the algorithms except Fuzzy clustering could detect the exact number of clusters in the Australia and Breast cancer data sets. This proves the fact that PSO clustering can result in a similar pattern recognition as most of the other techniques. Also, PSO clustering has the lowest standard deviation and overall, PSO, DBSCAN, and K-means ++ algorithms have significantly less standard deviation for differences between actual and predicted classes with 3.33, 3.45 and 3.54 standard deviation, respectively. Again, here we used MCDM methods for selecting the best algorithm.

#### **4.2.1. Using MCDM to select the best Algorithm**

The same process described in section 4-1-1 applies for the second experiment as well. The results are illustrated in appendix B.

From the results it is clear that the PSO algorithm is far superior to the other algorithms. For all three MCDM methods, the PSO algorithm ranked first on six data sets out of 10. K-means++, Ward, Loyd, Fuzzy, and DBSCAN ranked first in data sets 3, 2, 1, 1, and 1, respectively. It should also be mentioned that in the Iris data set, all algorithms, except for the Fuzzy algorithm, reached the same results and ranked first.

Furthermore, it can be seen that regarding the data sets in which PSO has not outperformed the other algorithms, its performance is still completely satisfactory. On the other hand, Loyd, Ward, and Fuzzy clustering algorithms performed extremely poorly in some cases compared to the other algorithms. For instance, the Ward algorithm ranked last in four cases out of 10 on all MCDMs.

From high class/low class perspective, it can be seen that PSO, DBSCAN and Kmeans++ are the best three algorithms with regard to high class data sets. For high class data sets, PSO was ranked 1st for both data sets on all MCDMs. DBSCAN was ranked 2nd and 3rd on all MCDMs for Glass and Breast cancer, respectively. Finally, Kmeans++ was ranked 3rd for both data sets on all MCDMs. On the other hand, regarding low class data sets, the results are almost the same with the overall analysis provided in the above paragraphs. Furthermore, the results of the group MCDM methods for the second experiment are shown in Table 9 and Table 10 as follows:

Table 9: Group MCDMs for experiment 2

ALGORITHM	GROUP TOPSIS			GROUP COPRAS	GROUP WSM
	$d_i^+$	$d_i^-$	$C\_index$	$Q_i$	WSM Score
<b>PSO</b>	0.15994	0.49390	0.75538	0.77792	0.19137
<b>DBSCAN</b>	0.32489	0.32418	0.49945	0.63181	0.15007
<b>LOYD</b>	0.43351	0.22351	0.34019	0.56021	0.13154
<b>WARD</b>	0.36719	0.28046	0.43304	0.79365	0.19472
<b>KMEANS++</b>	0.21457	0.44105	0.67272	0.50000	0.15019
<b>FUZZY</b>	0.30874	0.34780	0.52974	0.74806	0.18210

Table 10: The final rankings of 3 group MCDMs for experiment 2

ALGORITHM	TOPSIS SCORE	COPRAS SCORE	WSM SCORE	OVERALL SCORE	OVERALL RANK
<b>PSO</b>	6	5	5	16	1st
<b>DBSCAN</b>	3	3	2	8	5th
<b>LOYD</b>	1	2	1	4	6th
<b>WARD</b>	2	6	6	14	2nd
<b>KMEANS++</b>	5	1	3	9	4th
<b>FUZZY</b>	4	4	4	12	3rd

The results of group MCDM methods for experiment 2 demonstrate the superiority of the PSO and Ward algorithms over the other methods. Although the Ward algorithm was ranked first in two out of three methods, its poor rank on group TOPSIS questioned its overall performance. PSO algorithm, on the other hand, was ranked 1st only in group TOPSIS method, but according to group COPRAS and group WSM, its performance is quite satisfactory. Considering these results, and unlike experiment 1, the results of utilized method are diverse for experiment 2. Thus, the Borda count method is used to aggregate the results into one, final ranking as shown in Table 11.

Table 9: Borda count method

ALGORITHM	TOPSIS SCORE	COPRAS SCORE	WSM SCORE	OVERALL SCORE	OVERALL RANK
PSO	6	5	5	16	1st
DBSCAN	3	3	2	8	5th
LOYD	1	2	1	4	6th
WARD	2	6	6	14	2nd
KMEANS++	5	1	3	9	4th
FUZZY	4	4	4	12	3rd

The overall score of the PSO algorithm, for example, is calculated by assigning a score based on its rank for each MCDM method. PSO is ranked 1st, 2nd and 2nd in TOPSIS, COPRAS and WSM, respectively. Therefore, its score would be six, five and five. The overall score is the sum of all three scores, which is equal to 16. The overall scores for the other algorithms are calculated in the same way.

According to the final results, the PSO algorithm outperformed the other algorithm due to its satisfactory performance on all three MCDM methods. It is worth mentioning that, running PSO clustering on very big data would be a lengthy process compared to some other traditional techniques, one can easily modify the algorithm for a parallel running in order to speed up the run time (Nouaouria, et al., 2013).

On the other hand, though the Ward algorithm was recognized as the best method in two cases, due to its poor rank in one case, it was ranked second by the Borda count method. The Fuzzy algorithm, which was ranked 3rd, was selected as the 3rd method by all MCDM methods as well. Regardless of the final ranking, it should be mentioned that two algorithms, namely Ward and Kmeans++, experienced significant fluctuations in their performance which negatively affects their reliability.



The main advantage of using the Borda count method is that it can unify different, and sometimes inconsistent, results of multiple methods. Since the scores assigned to each algorithm are commensurate with their ranks, the Borda count method is not biased toward one method or another.

To recapitulate, two experiments were conducted in this study. One with predetermined cluster numbers, and one without. The former is useful for understanding how the methods work under controlled conditions. The latter, on the other hand, evaluates the performance of algorithms under uncontrolled situations. Regarding the first experiment, fuzzy and PSO algorithms were selected as the best methods, while in the second experiment, PSO and Ward were chosen as the superior ones. Though one might claim that the second experiment is more comprehensive, and thus more reliable, none of the two experiments are able to eliminate the need for the other. That is because the difference between the two experiments is the distinction between supervised and unsupervised learning.

## **5. Conclusion**

The task of evaluation and selection of models has been an important and challenging issue in numerous disciplines, such as operations research, data mining, and machine learning. Given the multitude of clustering algorithms and the nature of cluster analysis, this issue is even more demanding when it comes to clustering.

While there exist numerous studies regarding prioritizing, ranking, and selecting clustering methods, the lack of a comprehensive, objective, and robust framework for evaluating and ranking various clustering algorithms has been a major deficiency.

This paper proposed a novel evaluation framework that utilizes MCDM methods to assess the quality of numerous clustering algorithms. The proposed approach first utilizes three MCDM and group MCDM method to aggregate the results of a collection of both external and internal evaluation measures on a number of data sets. Afterwards, the results of group MCDM methods are unified using the Borda count method.

Two experiments were designed to validate the proposed framework. Six clustering algorithms, seven performance criteria including five external and two internal measures, three MCDM and

group MCDM methods and one final aggregating, namely the Borda count method, and ten data sets were analyzed in the experiments.

The results indicate that no single algorithm can achieve the best performance on all measures for any data set. Besides, it is necessary to utilize more than one single performance measure to evaluate clustering methods. The experimental studies also indicate that the group MCDM methods may disagree with each other. In such cases, the Borda count method can unify the results and provide a final solution. This is the case for experiment two. The proposed group MCDM compromises our decision on selection of clustering algorithms in different group of datasets. It selects one clustering model instead of analysing all the models one by one, which has more probability to be the best clustering algorithm based on different internal and external measures. Therefore, this approach is a filter methodology and valuable in evaluating large numbers of data sets for decreasing the computation time.

For future work, multi objective optimization and evaluation of clustering algorithms with more objective functions could be considered. Also, there are many different clustering evaluation measures (Figueiredo, et al., 2019; Khanmohammadi, Adibeig, & Shanehbandy, 2017; Wiwie, et al., 2015), and future studies can explore the effectiveness of these in overall performance of model selection. Applying the proposed evaluation approach to time series data instead of standard data sets can be another future research direction. Moreover, calculating meta-features for clustering evaluation instead of applying internal and external measures can exploit the power of meta-learning in the area of clustering.

## References:

- Aggarwal, C. C., & Reddy, C. K. (2013). *Data clustering: algorithms and applications*: CRC press.
- Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 1027-1035): Society for Industrial and Applied Mathematics.
- Bache, K., & Lichman, M. (2013). UCI machine learning repository. In.
- Baraldi, A., & Blonda, P. (1999). A survey of fuzzy clustering algorithms for pattern recognition. I. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29, 778-785.
- Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data* (pp. 25-71): Springer.
- Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial-temporal data. *Data & Knowledge Engineering*, 60, 208-221.

- Breaban, M. E., & Luchian, H. (2011). PSO aided k-means clustering: introducing connectivity in k-means. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation* (pp. 1227-1234): ACM.
- Carrico, J., Silva-Costa, C., Melo-Cristino, J., Pinto, F., De Lencastre, H., Almeida, J., & Ramirez, M. (2006). Illustration of a common framework for relating multiple typing methods by application to macrolide-resistant *Streptococcus pyogenes*. *Journal of Clinical Microbiology*, *44*, 2524-2532.
- Chen, C.-Y., & Ye, F. (2012). Particle swarm optimization algorithm and its application to clustering analysis. In *Electrical Power Distribution Networks (EPDC), 2012 Proceedings of 17th Conference on* (pp. 789-794): IEEE.
- Chen, R.-M. (2011). Particle swarm optimization with justification and designed mechanisms for resource-constrained project scheduling problem. *Expert Systems with Applications*, *38*, 7102-7111.
- Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation*, *6*, 58-73.
- Das, S., Abraham, A., & Konar, A. (2008). Automatic clustering using an improved differential evolution algorithm. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans*, *38*, 218-237.
- Ding, Y., & Fu, X. (2016). Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm. *Neurocomputing*, *188*, 233-238.
- Duò, A., Robinson, M. D., & Soneson, C. (2018). A systematic performance evaluation of clustering methods for single-cell RNA-seq data. *F1000Research*, *7*.
- Esmin, A. A., Coelho, R. A., & Matwin, S. (2015). A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data. *Artificial Intelligence Review*, *44*, 23-45.
- Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (Vol. 96, pp. 226-231).
- Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., Foufou, S., & Bouras, A. (2014). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. *IEEE transactions on emerging topics in computing*, *2*, 267-279.
- Figueiredo, E., Macedo, M., Siqueira, H. V., Santana Jr, C. J., Gokhale, A., & Bastos-Filho, C. J. (2019). Swarm intelligence for clustering—A systematic review with new perspectives on data mining. *Engineering Applications of Artificial Intelligence*, *82*, 313-329.
- Fowlkes, E. B., & Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American statistical association*, *78*, 553-569.
- Fulcher, B. D., Little, M. A., & Jones, N. S. (2013). Highly comparative time-series analysis: the empirical structure of time series and their methods. *Journal of The Royal Society Interface*, *10*, 20130048.
- Hamerly, G., & Drake, J. (2015). Accelerating Lloyd's algorithm for k-means clustering. In *Partitional clustering algorithms* (pp. 41-78): Springer.
- Hennig, C. (2008). Dissolution point and isolation robustness: robustness criteria for general cluster analysis methods. *Journal of multivariate analysis*, *99*, 1154-1176.
- Hsu, D. (2015). Comparison of integrated clustering methods for accurate and stable prediction of building energy consumption data. *Applied energy*, *160*, 153-163.
- Huang, A. (2008). Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand* (pp. 49-56).
- Hwang, C.-L., & Yoon, K. (1981). Methods for multiple attribute decision making. In *Multiple attribute decision making* (pp. 58-191): Springer.
- Jain, A. K., & Dubes, R. C. (1988). Algorithms for clustering data.
- Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*, 264-323.
- Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, *32*, 241-254.

- Kaklauskas, A., Zavadskas, E. K., Raslanas, S., Ginevicius, R., Komka, A., & Malinauskas, P. (2006). Selection of low-e windows in retrofit of public buildings by applying multiple criteria method COPRAS: A Lithuanian case. *Energy and Buildings*, 38, 454-462.
- Karaboga, D., & Ozturk, C. (2011). A novel clustering approach: Artificial Bee Colony (ABC) algorithm. *Applied soft computing*, 11, 652-657.
- Karmitsa, N., Bagirov, A. M., & Taheri, S. (2018). Clustering in large data sets with the limited memory bundle method. *Pattern Recognition*, 83, 245-259.
- Khanmohammadi, S., Adibeig, N., & Shanehbandy, S. (2017). An improved overlapping k-means clustering method for medical applications. *Expert Systems with Applications*, 67, 12-18.
- Kou, G., Peng, Y., & Wang, G. (2014). Evaluation of clustering algorithms for financial risk analysis using MCDM methods. *Information Sciences*, 275, 1-12.
- Krohling, R. A., & Campanharo, V. C. (2011). Fuzzy TOPSIS for group decision making: A case study for accidents with oil spill in the sea. *Expert Systems with Applications*, 38, 4190-4197.
- Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern recognition*, 38, 1857-1874.
- Liu, C., Li, Y., Zhao, Q., & Liu, C. (2019). Reference vector-based multi-objective clustering for high-dimensional data. *Applied Soft Computing*.
- Lv, Y., Ma, T., Tang, M., Cao, J., Tian, Y., Al-Dhelaan, A., & Al-Rodhaan, M. (2016). An efficient and scalable density-based clustering algorithm for datasets with complex structures. *Neurocomputing*, 171, 9-22.
- Márquez, D. G., Otero, A., Félix, P., & García, C. A. (2018). A novel and simple strategy for evolving prototype based clustering. *Pattern Recognition*, 82, 16-30.
- Meila, M., & Heckerman, D. (2013). An experimental comparison of several clustering and initialization methods. *arXiv preprint arXiv:1301.7401*.
- Michalski, R. S., & Stepp, R. E. (1983). Learning from observation: Conceptual clustering. In *Machine Learning, Volume I* (pp. 331-363): Elsevier.
- Morey, L. C., & Agresti, A. (1984). The measurement of classification agreement: An adjustment to the Rand statistic for chance agreement. *Educational and Psychological Measurement*, 44, 33-37.
- Murtagh, F., & Legendre, P. (2014). Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion? *Journal of classification*, 31, 274-295.
- Nanda, S. J., & Panda, G. (2014). A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary computation*, 16, 1-18.
- Nouaouria, N., Boukadoum, M., & Proulx, R. (2013). Particle swarm classification: A survey and positioning. *Pattern Recognition*, 46, 2028-2044.
- Oehler, K. L., & Gray, R. M. (1995). Combining image compression and classification using vector quantization. *IEEE transactions on pattern analysis and machine intelligence*, 17, 461-473.
- Opricovic, S., & Tzeng, G.-H. (2004). Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. *European journal of operational research*, 156, 445-455.
- Petrovsky, A. B. (2014). Group Multiple Criteria Decision Making: Multiset Approach. In *Recent Developments and New Directions in Soft Computing* (pp. 19-33): Springer.
- Pham, T. X., Siarry, P., & Oulhadj, H. (2018). Integrating fuzzy entropy clustering with an improved PSO for MRI brain image segmentation. *Applied Soft Computing*, 65, 230-242.
- Pohekar, S., & Ramachandran, M. (2004). Application of multi-criteria decision making to sustainable energy planning—a review. *Renewable and sustainable energy reviews*, 8, 365-381.
- Ray, S., & Turi, R. H. (1999). Determination of number of clusters in k-means clustering and application in colour image segmentation. In *Proceedings of the 4th international conference on advances in pattern recognition and digital techniques* (pp. 137-143): Calcutta, India.
- Reyes-Sierra, M., & Coello, C. C. (2006). Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research*, 2, 287-308.
- Ros, F., & Guillaume, S. (2019). A hierarchical clustering algorithm and an improvement of the single linkage criterion to deal with noise. *Expert Systems with Applications*.

- Sanchez, M. A., Castillo, O., Castro, J. R., & Melin, P. (2014). Fuzzy granular gravitational clustering algorithm for multivariate data. *Information Sciences*, 279, 498-511.
- Schlimmer, J. (1981). Mushroom records drawn from The Audubon Society field guide to north American mushrooms. *GH Lincoff (Pres)*, New York.
- Shang, R., Zhang, W., Li, F., Jiao, L., & Stolkin, R. (2019). Multi-objective artificial immune algorithm for fuzzy clustering based on multiple kernels. *Swarm and Evolutionary Computation*.
- Shih, H.-S. (2008). Incremental analysis for MCDM with an application to group TOPSIS. *European journal of operational research*, 186, 720-734.
- Sneath, P. H., & Sokal, R. R. (1973). *Numerical taxonomy. The principles and practice of numerical classification*.
- Triantaphyllou, E., & Mann, S. H. (1989). An examination of the effectiveness of multi-dimensional decision-making methods: a decision-making paradox. *Decision Support Systems*, 5, 303-312.
- Tzortzis, G., & Likas, A. (2014). The MinMax k-Means clustering algorithm. *Pattern recognition*, 47, 2505-2516.
- Ünlü, R., & Xanthopoulos, P. (2019). Estimating the number of clusters in a dataset via consensus clustering. *Expert Systems with Applications*, 125, 33-39.
- Van der Merwe, D., & Engelbrecht, A. P. (2003). Data clustering using particle swarm optimization. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on* (Vol. 1, pp. 215-220): IEEE.
- Van Erp, M., & Schomaker, L. (2000). Variants of the borda count method for combining ranked classifier hypotheses. In *IN THE SEVENTH INTERNATIONAL WORKSHOP ON FRONTIERS IN HANDWRITING RECOGNITION. 2000. AMSTERDAM LEARNING METHODOLOGY INSPIRED BY HUMAN'S INTELLIGENCE BO ZHANG, DAYONG DING, AND LING ZHANG*: Citeseer.
- Vyas, A., Kannao, R., Bhargava, V., & Guha, P. (2014). Commercial block detection in broadcast news videos. In *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing* (pp. 63): ACM.
- Wei-guo, F., & Hong, Z. (2007). A multi-attribute group decision-making method approaching to group ideal solution. In *Grey Systems and Intelligent Services, 2007. GSIS 2007. IEEE International Conference on* (pp. 815-819): IEEE.
- Wiwie, C., Baumbach, J., & Röttger, R. (2015). Comparing the performance of biomedical clustering methods. *Nature methods*, 12, 1033.
- Xu, D., & Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2, 165-193.
- Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16, 645-678.
- Yang, S., Li, K., Liang, Z., Li, W., & Xue, Y. (2018). A novel cluster validity index for fuzzy c-means algorithm. *Soft Computing*, 22, 1921-1931.
- Yeung, K. Y., & Ruzzo, W. L. (2001). Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data. *Bioinformatics*, 17, 763-774.

## Appendix A: The detailed tables for MCDM methods for experiment 1

Table A 1: the utilized algorithms' performance on each data set (Experiment 1) according to TOPSIS methods

ALGORITHM	1	2	3	4	5	6	7	8	9	10
PSO	3	5	3	4	5	1	1	1	1	4
DBSCAN	4	4	3	3	4	4	3	5	4	3
LOYD	4	3	6	5	3	3	5	6	4	5
WARD	6	2	1	1	6	6	6	3	2	1
KMEANS++	1	1	2	6	2	4	4	4	3	6
FUZZY	2	6	5	2	1	2	2	2	6	2

Table A 2: the utilized algorithms' performance on each data set (Experiment 1) according to COPRAS methods

ALGORITHM	1	2	3	4	5	6	7	8	9	10
PSO	3	5	3	2	5	1	1	1	1	4
DBSCAN	4	4	3	5	4	4	3	5	4	3
LOYD	4	3	6	6	3	3	5	6	4	5
WARD	6	2	1	1	6	6	6	3	2	1
KMEANS++	1	1	2	4	2	4	4	4	3	6
FUZZY	2	6	5	3	1	2	2	2	6	2

Table A 3: the utilized algorithms' performance on each data set (Experiment 1) according to WSM methods

ALGORITHM	1	2	3	4	5	6	7	8	9	10
PSO	3	5	3	2	5	1	1	1	1	4
DBSCAN	4	4	3	5	4	4	3	5	4	3
LOYD	4	3	6	6	3	3	5	6	4	5
WARD	6	2	1	1	6	6	6	3	2	1
KMEANS++	1	1	2	4	2	4	4	4	3	6
FUZZY	2	6	5	3	1	2	2	2	6	2

## Appendix B: The detailed tables for MCDM methods for experiment 2

Table B 1: The utilized algorithms' performance on each data set (Experiment 2) according to TOPSIS methods

ALGORITHM	1	2	3	4	5	6	7	8	9	10
PSO	2	3	1	1	1	4	1	1	1	3
DBSCAN	5	5	2	2	5	3	3	1	3	4
LOYD	6	4	4	5	3	5	6	1	3	5
WARD	3	2	6	6	6	6	5	1	2	1
KMEANS++	1	1	2	3	4	2	3	1	3	6
FUZZY	4	6	5	4	2	1	2	6	6	2

Table B 2: The utilized algorithms' performance on each data set (Experiment 2) according to COPRAS methods

<b>ALGORITHM</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>PSO</b>	2	3	1	1	1	4	1	1	1	3
<b>DBSCAN</b>	5	5	2	2	5	3	3	1	3	4
<b>LOYD</b>	6	4	4	5	3	5	6	1	3	5
<b>WARD</b>	2	2	6	6	6	6	5	1	2	1
<b>KMEANS++</b>	1	1	2	3	4	2	3	1	3	6
<b>FUZZY</b>	4	6	5	4	2	1	2	6	6	2

Table B 3: The utilized algorithms' performance on each data set (Experiment 2) according to WSM methods

<b>ALGORITHM</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>PSO</b>	2	3	1	1	1	4	1	1	1	3
<b>DBSCAN</b>	5	5	2	2	5	3	3	1	3	4
<b>LOYD</b>	6	4	4	5	3	5	6	1	3	5
<b>WARD</b>	2	2	6	6	6	6	5	1	2	1
<b>KMEANS++</b>	1	1	2	3	4	2	3	1	3	6
<b>FUZZY</b>	4	6	5	4	2	1	2	6	6	2