Computer Science and Information Systems 21(3):1147–1177 https://doi.org/10.2298/CSIS230302033M

MCClusteringSM: An approach for the Multicriteria Clustering Problem based on a Credibility Similarity Measure

Lugo Medrano Cesar[1] Gastelum Chavira Diego Alonso[2] Valdez Lafarga Octavio[3]
Velarde Cervantes Jose Luis[4]

[1] Doctoral Program in Business Analytics, Universidad Autonoma de Occidente
Blvd. Lola Beltran y Blvd Rolando Arjona, CP80020 Culiacan, Mexico
cesar.lugo@uadeo.mx

[2] Department of Economic-Administrative Sciences, Universidad Autonoma de Occidente
Blvd. Lola Beltran y Blvd Rolando Arjona, CP80020 Culiacan, Mexico
diego.chavira@uadeo.mx

[3] Department of Technology and Engineering, Universidad Autonoma de Occidente
Blvd. Lola Beltran y Blvd Rolando Arjona, CP80020 Culiacan, Mexico
octavio.valdezl@uadeo.mx

[4] Faculty of Computer Science, Universidad Autonoma de Sinaloa
Josefa Ortiz de Dominguez SN, Cd. Universitaria, CP80013 Culiacan, Mexico
jl.velarde17@info.uas.edu.mx

Abstract. Multicriteria clustering problem has been studied and applied scarcely. When a multicriteria clustering problem is tackled with an outranking approach, it is necessary to include preferences of decision makers on the raw dataset, e.g., weights and thresholds of the evaluation criteria. Then, it is necessary to conduct a process to obtain a comprehensive model of preferences represented in a fuzzy or crisp outranking relation. Subsequently, the model can be exploited to derive a multicriteria clustering. This work presents an exhaustive search approach using a credibility similarity measure to exploit a fuzzy outranking relation to derive a multicriteria clustering. The work includes two experimental designs to evaluate the performance of the algorithm. Results show that the proposed method has good performance exploiting fuzzy outranking relations to create the clusterings.

Keywords: MCDA, Multicriteria Clustering Problem, Similarity Measure.

1. Introduction

It is known today that one of the operational tools used to solve decision problems is Multicriteria Decision Analysis (MCDA). It is a field of Operations Research that addresses complex decision problems, some of them with multiple criteria in conflict. In MCDA, decision-makers can perform a choice, within a set of decision alternatives based on their preferences, by ranking or sorting them [35].In the choice problem, a restricted number of potential alternatives is chosen, as small as possible, which justifies eliminating the rest. Likewise, in the ranking problem, the decision maker can rank the alternatives from the best to the worst, with the possibility of ties and incompatibilities between them. Finally, in the sorting problem, each alternative is assigned to a set of pre-defined ordered categories. However, there can be an additional problem called clustering, where grouping similar objects into homogeneous groups (clusters) is necessary. These clusters are unknown a priori, and the objective is to obtain them using the information contained in the elements to be clustered [1]. The clustering problem has been addressed in the statistics and data analytics contexts. Data analytics focuses on data processing to extract and highlight useful information that would otherwise be impossible to know. Two approaches to analyzing data for grouping objects have been developed: supervised classification and unsupervised classification. Olteanu [31] refers to the first approach as a supervised grouping process that relies on a priori information regarding the groups or classes to place the objects. For the second one, there is no benefit from any knowledge of the structure of the data. The purpose of clustering is to group objects based on the natural structure of the data using measures of similarity. In the context of MCDA, the clustering problem pertains to the need to group alternatives based on the similarity measures obtained from understanding the preferences of decision makers. In that sense, Boujelben [6] defined the multicriteria clustering problem as a combination of classical clustering with MCDA, where clusters of alternatives are obtained based on criteria whose clusters are undefined a priori.

Corrente et al. [12] affirm that classification/sorting and clustering is a fundamental topic in artificial intelligence (AI), with several machine learning (ML) algorithms that are available for classification and clustering tasks, and recently these problems have adopted a constructive decision aiding perspective in contrast to the statistical pattern recognition approach typically adopted in AI/ML.

The multicriteria clustering techniques allow understanding, in a better way, the structure of a problem [18]. These authors presented an illustrative example: during the exploratory phase of a problem characterized by a lot of alternatives, one could submit to the decision-maker the representative elements of the different clusters (instead of the whole dataset) to simplify the decision process. Multicriteria clustering methods can also be used to help decision-makers build categories in a sorting context.

Though the multicriteria clustering problem has been studied and applied in specific areas, there is still room for improvement in existing clustering algorithms. One limitation of some MCDA clustering methods is the use of a cutting level λ. This cutting level is arbitrarily set to exploit a fuzzy outranking relation to create the clusters. The resulting clusters can increase or decrease due to the cutting level value. The number of clusters increases due to the increment of the incomparability between alternatives when the cutting level is near one. The number of clusters decreases because the indifference between alternatives is increased when the cutting level is near zero. The proposed method in this work avoids this situation by using a credibility similarity measure to obtain the clustering.

On the other hand, literature regarding MCDA indicates that the clustering problem has not received considerable interest, as well as in the search for other forms of clustering since a wide range of decision-making problems in various areas of engineering and management such as logistics and transport, urban planning, environmental assessment, energy efficiency analysis, and financial decision-making, require the assignment of a set of decision options (alternatives) to groups [12]. Thus, this work proposes an approach to address instances of the multicriteria clustering problem without using an arbitrary cutting level. The method is an exhaustive search approach that exploits a fuzzy outranking relation to derive clusters of alternatives in the sense of a partition, i.e., each alternative belongs to a unique cluster. The work is organized as follows: Section 2 provides a theoretical and conceptual framework of multicriteria clustering, including the foundation of outranking relations, the nature of a clustering problem and its complexity, a brief literature review, and the introduction of some important concepts about similarity measures. Section 3 presents the formulation of the multicriteria clustering problem and the proposed approach to address it. An empirical study to assess the proposed approach and a comparison with a multicriteria clustering procedure is described in Section 4. The results are presented in Section 5. Concluding general remarks and future perspectives are given in Section 6.

2. Theoretical and Conceptual Framework

2.1. Outranking Relations

In Multicriteria Decision Analysis, outranking relation theory (ORT) is a well-known area of knowledge. ORT has been widely used by MCDA researchers, mainly in Europe [17]. Bernard Roy set MCDA foundations during the late 1960s through the development of the ELECTRE (ELimination Et Choix Traduisant la REalité family of methods) [36].

An outranking relation is a binary relation that enables the decision maker to assess the strength of the outranking character of an alternative x over an alternative y. This strength increases if there are enough arguments to confirm that *"x is at least as good as y"* while there is no strong evidence to refute this statement.

Given two alternatives, x, and y, alternative x outranks y if *i*) there is a qualified majority of weighted criteria on which x is performing at least as good as y, and *ii*) there is no criterion on which y seriously outperforms x.

The outranking approaches operate in two major stages: construction and exploitation. The first involves modeling the decision-maker's preferences on criteria to construct an outranking relation among the considered alternatives. In contrast, the second stage involves the exploitation of the outranking relation to evaluating the alternatives for choice, ranking, or sorting purposes and, recently, for clustering. There are different procedures for constructing outranking relations, the most used being the ELECTRE and PROMETHEE [8] methods. Olteanu [31] also mentions the RUBIS method defined in a bipolar setting. ELECTRE I, for the choice problem, and ELECTRE III, for the ranking problem, are among the most popular outranking methods. In the construction stage, ELECTRE I builds a crisp outranking relation; meanwhile, ELECTRE III builds a fuzzy outranking [7]. Sometimes, a cutting level λ is used to transform fuzzy outranking relations into crisp ones. According to Linkov et al. [26], the cutting level is a technical parameter representing the sum of weights for the criteria that must be in concordance with the outranking relation to hold. A high cutting level means a high proportion of the criteria need to be the same or better than an alternative for the outranking relation to hold; a low cutting level means a lower proportion of the criteria are required for outranking relations to hold.

Defining the characteristics of fuzzy outranking relations and how to model decision-makers' preferences is beyond this work's scope. Still, readers can see Bouyssou [7], Fodor and Roubens [20], and Roy [37] for an explanation of them.

2.2. The Clustering Problem

There are different types of clustering, such as partition, hierarchy, pyramid, etc., the first two being the most frequent. This work is related to the partition type. In a partitional clustering, the universe of elements is divided into mutually exclusive subsets, where each element belongs to only one subset, so the pairwise intersection is empty. It is known it has an exponential complexity, where the number of partitions of a set is given by the Bell number [29]. The Stirling number of the second kind can be used to know the number of ways to divide a set of n elements within non-empty k clusters [32],

$$S(n,k) = \frac{1}{k!}\sum_{i=0}^{k} -1^{i}\binom{n}{k}(k-i)^{n} \tag{1}$$

For instance, given the set $A = \{1,2,3\}$, it can be partitioned into *i*) one cluster with the three elements: $C = \{1,2,3\}$, *ii*) two clusters in three ways to assign the elements: $C = \{\{1,2\},\{3\}\}, C = \{\{1,3\},\{2\}\}, C = \{\{2,3\},\{1\}\}$, and *iii*) three clusters in one way to assign the elements: $C = \{\{1\},\{2\},\{3\}\}$. Table 1 shows the different ways of grouping a set of three elements (n) with values of the number of clusters $k = 1$, $k = 2$, and $k = 3$.

Table 1. Different ways of grouping a set of three elements

Stirling number (n, k)	Different Ways	Groupings
$S(3,1)$	1	$C = \{1,2,3\}$
$S(3,2)$	3	$C = \{\{1,2\},\{3\}\}, C = \{\{1,3\},\{2\}\}, C = \{\{2,3\},\{1\}\}$
$S(3,3)$	1	$C = \{\{1\},\{2\},\{3\}\}$

Notes: n = The number of elements, k = The number of clusters, and C= the clustering. The first column corresponds to Stirling number with different values of n and k. The second column contains the number of ways to group a set of n elements in k clusters, obtained by the Stirling number. The third column shows the number of ways to set the three elements within non-empty k clusters.

The sum of all the possible clusters can be obtained with the Bell number formula:

$$Bell_n = \sum_{k=1}^{n} S(n,k) \tag{2}$$

The number of ways in which clusters can be constructed grows exponentially. It depends on the number of alternatives (*n*) and desired clusters (*k*). Thus, the clustering problem is non-trivial. It is shown in Table 2.

More ways to assign a set of elements in partitions are shown in Table 3, showing the complexity of the clustering problem itself.

2.3. A Literature Review of Multicriteria Clustering

Multicriteria clustering can be distinguished from classical clustering because it is a preference similarity-oriented problem where clusters should be conceived in preference

Table 2. Possible partitions for six elements within non-empty k clusters

Elements	Clusterings						Partitions
	1 cluster	2 clusters	3 clusters	4 clusters	5 clusters	6 clusters	
1	1	-	-	-	-	-	1
2	1	1	-	-	-	-	2
3	1	3	1	-	-	-	5
4	1	7	6	1	-	-	15
5	1	15	25	10	1	-	52
6	1	31	90	65	15	1	203

Notes: The Element column corresponds to the number of elements (n) in the group. The Clusterings column contains the Stirling number $S(n, k)$ in function to the number of elements n and the number of clusters k. The Partition column shows the total of possible clusters obtained with the Bell number formula.

Table 3. Number of partitions for up to 18 elements

n	partitions Bell (n)	n	Partitions Bell (n)
1	1	10	115975
2	2	11	678570
3	5	12	4213597
4	15	13	27644437
5	52	14	190899322
6	203	15	1382958545
7	877	16	10480142147
8	4140	17	82864869804
9	21147	18	682076806159

Source: Own elaboration based on [40].

proximity [19]. On this basis, general references about multicriteria clustering include mainly the next four aspects: *i*) the highlight methods in MCDA clustering, *ii*) how clusters are being evaluated in preference relations, *iii*) how the set preference relations are, and *iv*) if metaheuristics are used.

The next works considered all the previous four aspects: First, De Smet and Guzman [16] proposed an extension of the well-known k-means algorithm to the multicriteria framework. This extension relies on the definition of a multicriteria distance based on the preference structure defined by the decision maker. Then, De Smet and Eppe [15] developed a method that builds clusters and relations between these clusters based on a binary outranking matrix. An extension of the k-means algorithm is presented and tested on artificial data sets. Later, Baroudi and Safia [1] proposed a new clustering approach based on the definition of a new distance that considers the problem's multicriteria nature. This distance uses the preference relations of the PROMETHEE outranking method and the Sokal and Michener index so widely used in the classification field.

In this sense, Fernandez et al [19] proposed a clustering method based on a valued indifference relation inspired by outranking methods. They suggested a method based on comparing cluster centers and clusters' average net flow scores. Baroudi and Safia [2] tackle the problem of defining relations between clusters in multicriteria decision aid clustering. Meyer and Olteanu [29] formally defined the problem of clustering in MCDA using notions that are native to this field alone and highlighted the different structures which it is tried to uncover through this process.

Baroudi and Safia [3] also Sarrazin et al [38] developed a model based on the FlowSort sorting procedure and the PROMETHEE I outranking method. Rosenfeld et al [34] presented a study to draw attention to clustering algorithms. They consider algorithms must be compared with performance indicators or criteria, leading to asymmetric preference relations.

Other contributions need to be considered too, do not include the four aspects completely, nevertheless add to the research in this field. For instance, Bisdorff [5] proposed an ELECTRE-like approach for clustering judges from their L-valued pairwise proximities in preference judgments. Cailloux et al [10] presented multicriteria clustering procedures to discover data structures from a multicriteria perspective. De Smet [14] addressed the ordered multicriteria clustering problem, detecting ordered categories. The k-means procedure and the underlying idea of the FLOWSORT method inspired the algorithm. Boujelben [6] worked on the multicriteria-ordered clustering problem too. He defined a preference profile to measure the preferential quality of the clusters and similarity and inconsistency profiles to analyze the clusters on each criterion. Chen et al. [11] proposed a total ordered clustering algorithm, which considers the preference degree between any two alternatives. Afterward, Liu et al. [27] introduced a multicriteria ordered clustering algorithm based on PROMETHEE and K-Medoids clustering algorithms. Daneshvar et al. [13] presented a multicriteria clustering method by combining k-means algorithm and PROMETHEE technique; the parameters of the problem are the cluster separator profiles which genetic algorithm (GA) is used to optimize them. Ishizaka et al. [22] proposed a new hierarchical multicriteria clustering based on PROMETHEE, where the number of clusters is not specified. On the other hand, Kandakoglu et al. [23] studied the project portfolio decision problem with uncertain multiple criteria evaluations, decision makers' preference information, and resource constraints. They proposed a new methodology based on multicriteria ordered clustering to deal with this problem. Valencia et al. [41] used clustering analysis with the multicriteria method TOPSIS [21] to group urban agricultural sites. TOPSIS was implemented directly into the clustering analysis process to generate weights for sustainability indexes before applying the k-means clustering algorithm. Later, Bashir et al. [4] motivated by the partial net outranking flow/profile of the PROMETHEE and the Fuzzy c-means clustering, presented a multicriteria-ordered profile clustering algorithm.

2.4. Similarity Measures

The concept of similarity is elemental to the clustering process. The level of similarity between two objects can be measured in many ways, which is firmly related to the nature of the elements themselves [31]. The term similarity can be related to a function used to compare elements of any type. Usually, its input is two elements, and the output is a value between 0 and 1. A value equal to zero means the elements are completely dissimilar. On the other hand, a value equal to one means the two elements are identical. The similarity is related to distance, where zero implies the two elements are identical, and one implies they are completely dissimilar, i.e., it is opposite to the similarity measure [43].

In that sense, a general way of measuring the similarity between two elements is by using proximity measures. These measures can be defined by a dissimilarity function *D*, also named a distance function, or using a similarity function S. A large value for *D* between two elements means that the objects are dissimilar. In contrast, a small value refers that they are similar. The opposite statements are made for the similarity function *S* [31].

Distance measures. Different clustering methods, e.g., k-means, use distance measures to determine the similarity or dissimilarity between any pair of elements. It is functional to indicate the distance between two instances x_i and x_j as $d(x_i, x_j)$. A valid distance measure should be symmetric and obtain a zero value in the case of identical vectors [33].

According to [1] and [42], usually, a similarity measure has the following properties:

$$\begin{aligned} \forall a_i, a_j \in A : d(a_i, a_j) \rightarrow \mathbb{R}^{+} \text{with} \\ d(a_i, a_i) = 0, \text{identity}, \\ d(a_i, a_j) \geq 0, \text{non-negativity}, \\ d(a_i, a_j) = d(a_j, a_i), \text{symmetry}, \end{aligned}$$

also, when next properties are true

$$\begin{aligned} d(a_i, a_j) = 0 \Rightarrow a_i = a_j, \text{uniqueness}, \\ \forall a_i, a_j, a_z \in A : d(a_i, a_j) \leq d(a_i, a_z) + d(a_j, a_z), \end{aligned} \tag{3}$$

then, the similarity measure is called distance. Because not all similarity measures meet the symmetry or inequality of the triangle or both, not all similarity measures are distances.

[24] establish that the distance measure can be divided into two groups. In the first group, there are the metric measures, which must have the properties above. In the second group, there is the semi-metric, which does not follow the fifth property. These measures do not make possible an appropriate ordering of points over a metric space. For three points in this space, the sum of the distances from *i* to *j* and from *j* to *z* can be shorter than the distance between *i* and *z*.

Similarity functions. An alternative concept that can be used instead of distance is the similarity function $s(x_i, x_j)$ which compares the two vectors x_i and x_j. When these vectors are somehow ”similar”, the function has a large value and the largest value for identical vectors. This function should be symmetrical (namely $s(x_i, x_j) = s(x_j, x_i)$). Sometimes, methods for calculating the ”distances” in the case of binary and nominal attributes can be considered as similarity functions rather than distances [33]

3. Formulation of a Multicriteria Clustering Problem and the Proposed Approach

3.1. Problem Formulation

Let $A = \{a_1, a_2, \ldots, a_m\}$ be a finite set of alternatives considered by a decision-maker, which is valued by a set of criteria $G = \{g_1, g_2, \ldots, g_n\}$, some of them in conflict with each other. Besides, let $S^{\sigma}_{A} \subseteq A \times A \rightarrow [0, 1]$ be a fuzzy outranking relation that integrates the preferences of that decision-maker on the multiple criteria that describe the elements of A.

The problem is to exploit S^{σ}_{A} to obtain a set of partitions P_A using a credibility similarity measure; where alternatives belonging to a specific cluster $C_i \in P_A$ are similar to each other; at the same time, they must not be similar to alternatives that belong to another cluster $C_j \in P_A$, so that P_A is a partition of A and the clusters $C_i \in P_A$ reflect the best compromise between the conflicting objectives, ”discriminate the dissimilar alternatives,” and ”group the most similar alternatives”. These partitions must be the most consistent possible to the information included in S^{σ}_{A}. Thus, this work aims to address the multicriteria clustering problem where a decision-maker’s preferences are involved.

In the following subsections, an approach to partition the set of alternatives based on the information contained in the fuzzy outranking relation is presented. This method is called *MCClusteringSM - Multicriteria Clustering based on a Credibility Similarity Measure*. Specifically, the method belongs to the family of methods that exploit outranking relations, such as the exploitation phases of ELECTRE I, ELECTRE III, and ELECTRE TRI, to tackle the choice, ranking, and sorting problems, respectively, as described in [37]. The general scheme of the MCClusteringSM is presented in Figure 1.

3.2. The Clustering Method to Exploit a Fuzzy Outranking Relation

This part introduces the proposed method to exploit a fuzzy outranking relation to solving the multicriteria clustering problem. First, the general procedure to obtain the clusters is presented. Next, an illustrative example of this procedure is exposed. Finally, the general procedure of MCClusteringSM is described.

General procedure to obtain the clusters. Here, we provide in detail the steps to obtain the cluster. Additionally, an illustrative example is described.

The general steps to obtain the partitions by using MCClusteringSM are described below:

Step 1: Obtaining the Degrees of Credibility Similarity (DCS).
First, according to each credibility level that corresponds to the alternatives, a difference between each pairwise alternative is calculated over the intensity that outranks one alternative to another; this is the intensity which i outranks j is c_{ij} and the intensity with which j outranks i is c_{ji} , the subtraction needed is over these two values, considering the result in absolute value, this difference d can be noted as d_{ij} or d_{ji}:

$$d_{ij} = |c_{ij} - c_{ji}| \quad d_{ji} = |c_{ji} - c_{ij}| \quad d_{ij} = d_{ji} \tag{4}$$

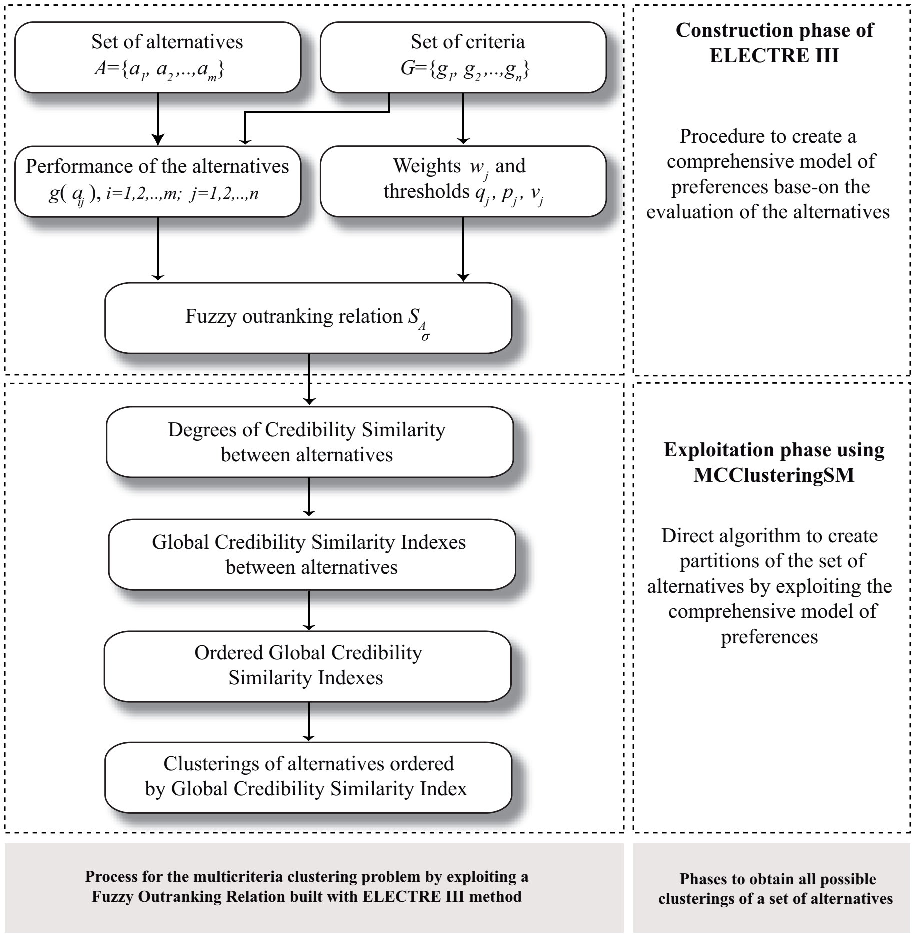
Set of alternatives
$A=\{a_1, a_2,..,a_m\}$
Set of criteria
$G=\{g_1, g_2,..,g_n\}$
Construction phase of ELECTRE III
Performance of the alternatives
$g(a_{ij})$, $i=1,2,..,m;\ j=1,2,..,n$
Weights w_j and thresholds q_j, p_j, v_j
Procedure to create a comprehensive model of preferences base-on the evaluation of the alternatives
Fuzzy outranking relation S^A_σ
Degrees of Credibility Similarity between alternatives
Exploitation phase using MCClusteringSM
Global Credibility Similarity Indexes between alternatives
Direct algorithm to create partitions of the set of alternatives by exploiting the comprehensive model of preferences
Ordered Global Credibility Similarity Indexes
Clusterings of alternatives ordered by Global Credibility Similarity Index
Process for the multicriteria clustering problem by exploiting a Fuzzy Outranking Relation built with ELECTRE III method
Phases to obtain all possible clusterings of a set of alternatives

Fig. 1. The general approach for the multicriteria clustering problem Using ELECTRE III and MCClusteringSM

Source: Own elaboration based on [25]

Then, depending on the degree of credibility to calculate will be the differences to use in Equation 4; for a simple relation between two alternatives, the equation to calculate the degree of credibility similarity S is:

$$S_{ij} = 1 - \frac{d_{ij}}{c_{ij} + c_{ji}} \tag{5}$$

Equation 5 is the degree of credibility similarity of a cluster composed of two alternatives. Now, if a cluster of three alternatives or more can be formed, the equation would change in this way:

For the $M = \binom{N}{2}$ possible pairwise comparisons over the N alternatives of a subset n. Setting d_m as the m^{th} level of credibility difference between each pairwise i, j of alternatives, where $m = 1, 2, \ldots, M$, then, the DCS is:

$$S_n = 1 - \frac{\sum_{m=1}^{M} |c_{ijm} - c_{jim}| \in d_m}{\sum_{m=1}^{M} |c_{ijm} + c_{jim}| \in d_m} \tag{6}$$

For instance, the degree of credibility similarity of a cluster composed of the alternatives a, b, and c is:

$$S_{abc} = 1 - \frac{|c_{ab} - c_{ba}| + |c_{ac} - c_{ca}| + |c_{bc} - c_{cb}|}{|c_{ab} + c_{ba}| + |c_{ac} + c_{ca}| + |c_{bc} + c_{cb}|} \tag{7}$$

From Equation 6, a degree of one means the highest intensity of similarity between the alternatives, which suggests a similarity between the alternatives. However, a degree of zero means the lowest level of similarity, and this suggests that these alternatives should not be in the same cluster.

Some limitations have been found in this proposed similarity measure, which is because it does not fully follow the properties of a metric. In strict order, it should be called a semi-metric; one of these limitations is that it cannot be fully relied upon to represent dissimilarities in an Euclidean space without appropriate transformation [9]. Therefore, a transformation is proposed with good results for the research objectives. The new values of the transformed metric could still represent the data but will be more amenable to analysis or comparison. As an initial proposal, for this study, the metric was squared for it to represent dissimilarities in a Euclidean space better:

$$S_{ij} = 1 - \left[\frac{d_{ij}}{c_{ij} + c_{ji}}\right]^2 \tag{8}$$

This same power transformation is applied for three or more alternatives:

$$S_n = 1 - \left[\frac{\sum_{m=1}^{M} |c_{ijm} - c_{jim}| \in d_m}{\sum_{m=1}^{M} |c_{ijm} + c_{jim}| \in d_m}\right]^2 \tag{9}$$

Step 2: Compute the Global Credibility Similarity Index (GCSI).
Once the credibility similarity degrees are calculated, many scenarios can be presented depending on the number of clusters k required; the number of scenarios is equal to the number of ways to divide a set of alternatives within non-empty clusters. To obtain this, we can use Equation 1. Then we average the DCS of the partitions for each scenario to obtain a global credibility similarity index so that the best solution, including the clusters, will be the one with the highest index.

To set this, let's call W the number of ways to divide the set of alternatives A within non-empty clusters k, which is the Stirling number of the second kind. Then, we set n as the subset of alternatives that conforms to the k cluster in each scenario, where w is the w^{th} way to divide the set of alternatives A within non-empty clusters k and $w = 1, 2, \ldots, W$, the GCSI is:

$$S_w = \sum_{n \in k} \frac{S_{nw}}{k} \tag{10}$$

Step 3: Ordering the Global Credibility Similarity Indexes.
To get the best clustering solution, according to the number of clusters required, ordering the global credibility similarity index is necessary to start the solution assignment from the best to the worst according to the index value.

An illustrative Example. Unlike classical methods, such as k-means, the alternatives are grouped based on the preference relations between alternatives given by the decision makers on the raw data set as thresholds and weights of the evaluation criteria. However, the cluster allocations are made directly from the data manifested in the model of preferences represented in a fuzzy outranking relation based on a credibility similarity measure calculated for all the possible combinations of grouped alternatives.

Let us now consider an illustrative example to see how the MCClusteringSM algorithm performs the calculation steps for the clustering process. The goal is to partition the set of alternatives $A = \{a, b, c, d\}$ into a given number of clusters in the best way.

Given the fuzzy outranking relation S_A^σ an input:

Fuzzy Outranking Relation

	a	**b**	**c**	**d**
a	1.00	0.90	0.70	1.00
b	0.80	1.00	0.70	0.80
c	0.30	0.30	1.00	0.30
d	0.00	0.30	0.30	1.00

Fig. 2. Fuzzy outranking relation of four alternatives
Source: Own elaboration

Step 1: Obtaining the Degrees of Credibility Similarity (DCS).

$$
\begin{aligned}
S_a &= 1\\
S_b &= 1\\
S_c &= 1\\
S_d &= 1\\
S_{ab} &= 1 - \left[\frac{|c_{ab} - c_{ba}|}{c_{ab} + c_{ba}}\right]^2\\
&= 1 - \left[\frac{|0.9 - 0.8|}{0.9 + 0.8}\right]^2\\
&= 0.9965\\
S_{ac} &= 1 - \left[\frac{|0.7 - 0.3|}{0.7 + 0.3}\right]^2\\
&= 0.84\\
S_{ad} &= 1 - \left[\frac{|1 - 0|}{1 + 0}\right]^2\\
&= 0\\
S_{bc} &= 0.84\\
S_{bd} &= 0.7933\\
S_{cd} &= 1\\
S_{abc} &= 1 - \left[\frac{|0.9 - 0.8| + |0.7 - 0.3| + |0.7 - 0.3|}{0.9 + 0.8 + 0.7 + 0.3 + 0.7 + 0.3}\right]^2\\
&= 1 - (0.243)^2\\
&= 0.9409\\
S_{bcd} &= 0.8888\\
S_{acd} &= 0.71\\
S_{abd} &= 0.8227\\
S_{abcd} &= 1 - \left[\frac{|0.9 - 0.8| + |0.7 - 0.3| + |1 - 0| + |0.7 - 0.3| + |0.8 - 0.3| + |0.3 - 0.3|}{0.9 + 0.8 + 0.7 + 0.3 + 1 + 0 + 0.7 + 0.3 + 0.8 + 0.3 + 0.3 + 0.3}\right]^2\\
&= 1 - \left[\frac{2.4}{6.4}\right]^2\\
&= 0.8594
\end{aligned}
$$

In an ordered way is:

$$
\begin{array}{lllll}
S_a = 1, & S_b = 1, & S_c = 1, & S_d = 1, & S_{cd} = 1,\\
S_{ab} = 0.9965, & S_{abc} = 0.9409, & S_{bcd} = 0.8888, & S_{abcd} = 0.8594, & S_{ac} = 0.84,\\
S_{bc} = 0.84, & S_{abd} = 0.8227, & S_{bd} = 0.7933, & S_{acd} = 0.71, & S_{ad} = 0
\end{array}
$$

Step 2: Compute the Global Credibility Similarity Index (GCSI).
Using Equation 10 to compute the Global Credibility Similarity Index, in this example, the possible numbers of clusters are from 2 to 3; using all the alternatives in one cluster or each alternative separated, this is $k = 2$ and $k = 3$.

For $k = 2$

$S_{cd\_ab} = \frac{1+0.9965}{2} = 0.9985$

$S_{ac\_bd} = \frac{0.84+0.7933}{2} = 0.81665$

$S_{bc\_ad} = \frac{0.84+0}{2} = 0.42$

$S_{abc\_d} = \frac{0.9409+1}{2} = 0.97045$

$S_{abd\_c} = \frac{0.8227+1}{2} = 0.91135$

$S_{acd\_b} = \frac{0.71+1}{2} = 0.855$

$S_{bcd\_a} = \frac{0.8888+1}{2} = 0.9444$

For $k = 3$

$S_{a\_b\_cd} = \frac{1+1+1}{3} = 1$

$S_{a\_c\_bd} = \frac{1+1+0.7933}{3} = 0.9311$

$S_{a\_d\_bc} = \frac{1+1+0.84}{3} = 0.9466$

$S_{b\_c\_ad} = \frac{1+1+0}{3} = 0.66$

$S_{b\_d\_ac} = \frac{1+1+0.84}{3} = 0.9466$

$S_{c\_d\_ab} = \frac{1+1+0.9965}{3} = 0.9988$

Step 3: Ordering the Global Credibility Similarity Indexes.

For k=2, rank the clustering from highest to lowest:

1.$S_{cd\_ab} = 0.99825$
2.$S_{abc\_d} = 0.97045$
3.$S_{bcd\_a} = 0.9444$
4.$S_{abd\_c} = 0.91135$
5.$S_{acd\_b} = 0.855$
6.$S_{ac\_bd} = 0.81665$
7.$S_{bc\_ad} = 0.42$

Thus, the best solution for$k$=2 is the cluster $C = \{\{c, d\}, \{a, b\}\}$.
For k=3, rank the clustering from highest to lowest:

1.$S_{a\_b\_cd} = 1$
2.$S_{c\_d\_ab} = 0.9988$
3.$S_{a\_d\_bc} = 0.9466$
4.$S_{b\_d\_ac} = 0.9466$
5.$S_{a\_c\_bd} = 0.9311$
6.$S_{b\_c\_ad} = 0.66$

The best solution for k=3 is the clustering $C = \{\{a\}, \{b\}, \{c, d\}\}$.

The General procedure of MCClusteringSM is presented in Algorithm 1. The input of MCClusteringSM is a fuzzy outranking relation, which will be partitioned according to a given number of clusters and alternatives. Each clustering obtained results from a partitional clustering process that considers preference relations based on the global credibility similarity index presented in Equation (10). MCClusteringSM was implemented in C language.

Algorithm 1: The MCClusteringSM procedure

```
Input  : Fuzzy Outranking Relation $\boldsymbol{S_A^\sigma}$
Output: Partitional Clustering based on the credibility similarity index
Begin.
Set the number of clusters $N_C$ and the number of alternatives $N_A$.
Compute the degrees of credibility similarity $DCS$ between all possible combinations of
  the $N_A$ alternatives.
Compute the Global Credibility Similarity Index $GCSI$ using $DCS$ calculations from 1
  to $N_C$.
Rank the $GCSI$ indexes in decreasing order to get all possible ordered clustering.
End.
```

4. Performance of MCClusteringSM

This section includes an empirical evaluation of MCClusteringSM and a comparison with a multicriteria clustering procedure.

This empirical evaluation aims to analyze how the direct method proposed performs when solving clustering problems with different structures and sizes. We intended to capture essential characteristics and analyze observations from the experiment execution. For this, with the help of an instance generator, we created a set of simulated reference sets, each of which is composed of classes of alternatives with different sizes and structures, and an associated fuzzy outranking relation that represents a model of preferences. This relation is constructed such that it can be utilized to generate cluster allocations of alternatives without inconsistencies according to its associated classes of alternatives. MCClusteringSM was executed to exploit each of the generated fuzzy outranking relations and evaluated if it successfully found the best solutions according to the fuzzy outranking relations supplied.

Besides, we attempted to evaluate the performance of MCClusteringSM compared to mccClusters [31]. It is a multicriteria clustering procedure implemented in the software Diviz [28]. This procedure computes clusters of alternatives based on the clustering typology: non-relational, relational, ordered, exclusive relational, and exclusive ordered multicriteria clustering. These evaluations were done to have a comparison point to analyze the performance of MCClusteringSM and to discuss its advantages and disadvantages.

4.1. Empirical evaluation of MCClusteringSM

Test criterion for evaluating the MCClusteringSM method. When a fuzzy outranking relation without cycles among alternatives is provided to a good clustering procedure, it should exploit this relation, preserving the cluster allocation of the alternatives according to preference relations. Thus, to evaluate the performance of MCClusteringSM, a fuzzy outranking relation without inconsistencies was provided to it. The best clustering solutions should be those that reflect the information contained in the fuzzy outranking relation. In MCClusteringSM, the best clustering solutions have Global Credibility Similarity Indexes of 1.0 or near 1.0, obtained with Equation 10.

Let us define $A = \{a_1, a_2, \ldots, a_m\}$ as a set of alternatives, $S_A^\sigma \subseteq A \times A \to [0,1]$ as a fuzzy outranking relation, and C_A as an optimal clustered set of alternatives A, obtained with MCClusteringSM. Thus, according to this test criterion, the clustering of $(a_i, a_j) \in A \times A$ in S_A^σ should be reflected in C_A.

Experimental design. The experimental design for evaluating MCClusteringSM included a set of variables and a set of fuzzy outranking relations with different structures and sizes. This experimental design is described below.

The variables considered in this study were related to the number of clusters (V_1) and the number of alternatives (V_2). We selected small numbers for both variables due to the combinatorial nature of the problem. V_1 was defined with four values, and V_2 was defined with nine values. The first value of V_2 started with a value of six because of the maximum value of V_1, and it finished with a value of fourteen because MCClusteringSM is an exhaustive method. It last implies using high computing resources, in this case, a large capacity of RAM (Random Access Memory). The variables of the experimental design are shown in Table 4.

Table 4. Variables of the experimental design

Variable	Value
V_1	2, 3, 4, 5
V_2	6, 7, 8, 9, 10, 11, 12, 13, 14

By assessing the performance of MCClusteringSM with the four values of V_1 and the nine values of V_2, we could achieve conclusions about the performance of our purpose in a multicriteria clustering context.

On the other hand, the general procedure to generate the fuzzy outranking relations of this test is presented in Algorithm 2. The output of it is a matrix that represents the fuzzy outranking relation. The sizes of the matrix come from six to fourteen alternatives. Besides, a vector is randomly generated to indicate which alternative belongs to each cluster. Thus, the structure of the fuzzy outranking relation is based on such vector that represents a clustering. In this test, we wanted to know if the value of the clustering represented in the vector and implicitly in the fuzzy outranking relation is one of the best possible solutions, preferably, with a value of 1.0.

Algorithm 2: Instance generator

```
Input  : Number of clusters $N_C$, Number of Alternatives $N_A$
Output: Fuzzy Outranking Relation $\boldsymbol{S}^{\boldsymbol{\sigma}}_{\boldsymbol{A}}$
Begin.
Form a vector $V$ of alternatives of size $N_A$ and randomly assign the $N_C$ clusters to $N_A$
  alternatives, to ensure each cluster has at least an alternative.
Randomly complete $V$ with the $N_C$ clusters.
Create a matrix $\boldsymbol{S}^{\boldsymbol{\sigma}}_{\boldsymbol{A}}$ of size $N_A \times N_A$ and fill it with values $[0, 1]$ based on $V$, i.e., the
  clusters have the alternatives assigned on that basis.
End.
```

For each combination of the values of variables V_1 and V_2, Algorithm 2 was run 100 times. Thus, 3600 sets of references were created. This data generation was implemented in C language. The program was executed on an Asus computer with an Intel Core i5-10300H (2.5 GHz) processor, 16GB in RAM (DDR4 SDRAM), and a hard disk 250 GB 3G SAT 7.2K rpm LFF3. Figure 3 shows how a fuzzy outranking relation is generated with Algorithm 2.

Meanwhile, Algorithm 1 was implemented in C language and executed 3600 times. Instances of V_2 = 6, 7, 8, 10, 11, and 12 alternatives were executed in the Asus computer. Instances of V_2 = 13, 14 alternatives were run on a Lenovo server with processor Intel Xeon Gold 5218 (2.3 GHz) 8 cores, 128GB in RAM, and 764GB of hard disk.

In this experimental design, a response variable was defined to analyze the proposed method for the multicriteria clustering problem. It corresponds to the number of solu-

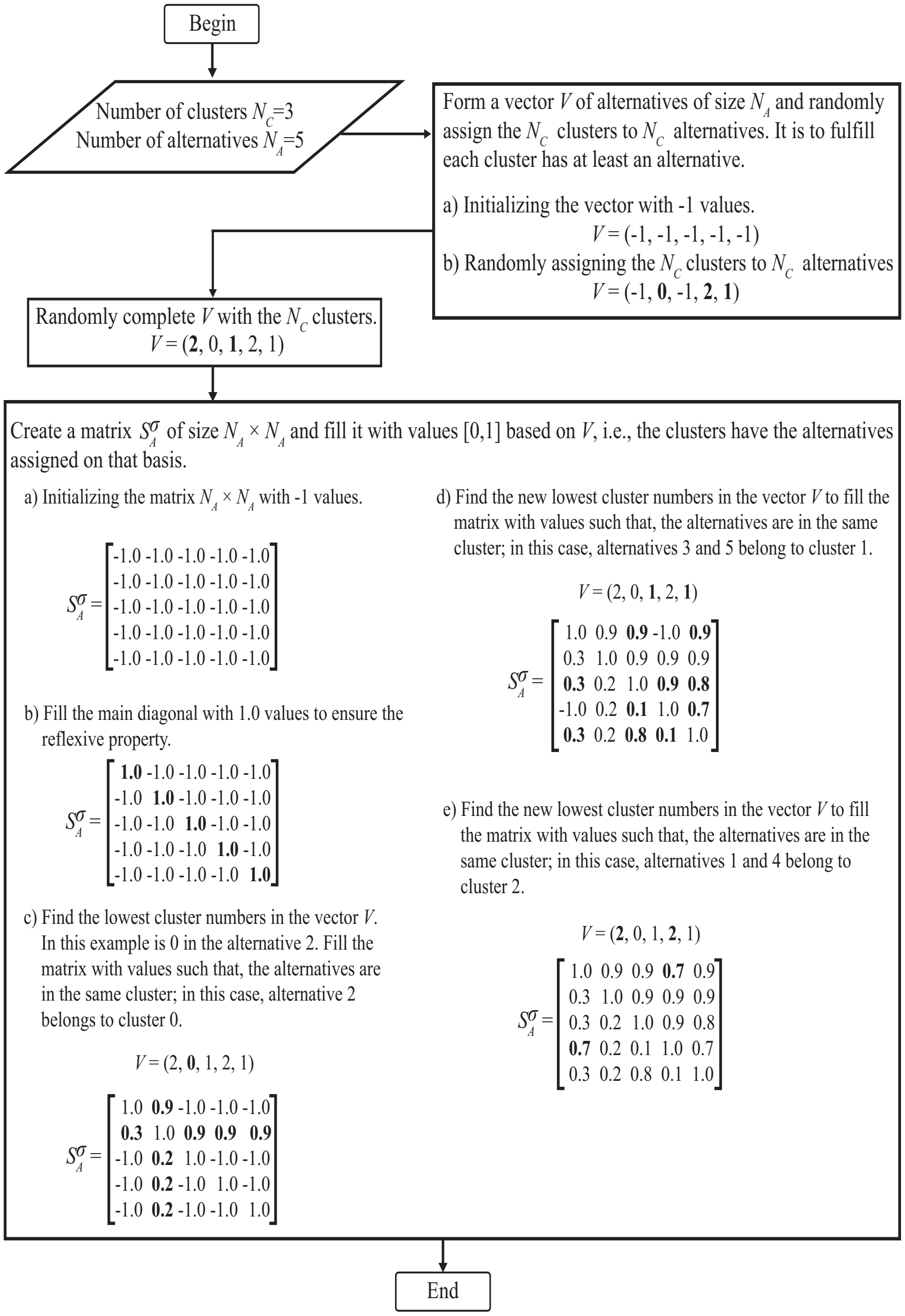
Begin
Number of clusters N_C=3
Number of alternatives N_A=5
Form a vector V of alternatives of size N_A and randomly assign the N_C clusters to N_C alternatives. It is to fulfill each cluster has at least an alternative.
a) Initializing the vector with -1 values.
V = (-1, -1, -1, -1, -1)
b) Randomly assigning the N_C clusters to N_C alternatives
V = (-1, **0**, -1, **2**, **1**)
Randomly complete V with the N_C clusters.
V = (**2**, 0, **1**, 2, 1)
Create a matrix S_A^σ of size $N_A \times N_A$ and fill it with values [0,1] based on V, i.e., the clusters have the alternatives assigned on that basis.
a) Initializing the matrix $N_A \times N_A$ with -1 values.
S_A^σ =
-1.0 -1.0 -1.0 -1.0 -1.0
-1.0 -1.0 -1.0 -1.0 -1.0
-1.0 -1.0 -1.0 -1.0 -1.0
-1.0 -1.0 -1.0 -1.0 -1.0
-1.0 -1.0 -1.0 -1.0 -1.0
b) Fill the main diagonal with 1.0 values to ensure the reflexive property.
S_A^σ =
1.0 -1.0 -1.0 -1.0 -1.0
-1.0 **1.0** -1.0 -1.0 -1.0
-1.0 -1.0 **1.0** -1.0 -1.0
-1.0 -1.0 -1.0 **1.0** -1.0
-1.0 -1.0 -1.0 -1.0 **1.0**
c) Find the lowest cluster numbers in the vector V. In this example is 0 in the alternative 2. Fill the matrix with values such that, the alternatives are in the same cluster; in this case, alternative 2 belongs to cluster 0.
V = (2, **0**, 1, 2, 1)
S_A^σ =
1.0 **0.9** -1.0 -1.0 -1.0
0.3 1.0 **0.9** **0.9** **0.9**
-1.0 **0.2** 1.0 -1.0 -1.0
-1.0 **0.2** -1.0 1.0 -1.0
-1.0 **0.2** -1.0 -1.0 1.0
d) Find the new lowest cluster numbers in the vector V to fill the matrix with values such that, the alternatives are in the same cluster; in this case, alternatives 3 and 5 belong to cluster 1.
V = (2, 0, **1**, 2, **1**)
S_A^σ =
1.0 0.9 **0.9** -1.0 **0.9**
0.3 1.0 0.9 0.9 0.9
0.3 0.2 1.0 **0.9** **0.8**
-1.0 0.2 **0.1** 1.0 **0.7**
0.3 0.2 **0.8** **0.1** 1.0
e) Find the new lowest cluster numbers in the vector V to fill the matrix with values such that, the alternatives are in the same cluster; in this case, alternatives 1 and 4 belong to cluster 2.
V = (**2**, 0, 1, **2**, 1)
S_A^σ =
1.0 0.9 0.9 **0.7** 0.9
0.3 1.0 0.9 0.9 0.9
0.3 0.2 1.0 0.9 0.8
0.7 0.2 0.1 1.0 0.7
0.3 0.2 0.8 0.1 1.0
End

Fig. 3. A fuzzy outranking relation generated with Algorithm 2
Source: Own elaboration

tions without inconsistencies found per combination of variable 1 and variable 2 (V_1, V_2). Each combination (V_1, V_2) has 100 different fuzzy outranking relations of the 3600. For each run i-th of Algorithm 1 on a single fuzzy outranking relation, an auxiliary binary variable $y_i(V_1, V_2)$ is defined, which takes a value of one if Algorithm 1 finds the best solution without any inconsistency or zero otherwise. This last according to the source clustering generated in steps 3 and 4 of Algorithm 1, represented in the vector V. Thus, the response variable can be defined as $Y(V_1, V_2) = \Sigma_{i=1}^{100} y_i(V_1, V_2)$, which is the sum of all the 100 auxiliary binary variables per combination. The evaluation of the results of MCClusteringSM was done using an implementation of Algorithm 3 in C language.

Algorithm 3: Results evaluator

```
Input  : Source clustering $V$, results of MCClusteringSM $F$
Output: Global Credibility Similarity Index of $V$ in $F$
begin
    set $pos$ = 1
    repeat
        if $V = F[pos]$ then
            return Global Credibility Similarity Index of $V$ ;
        else
            $pos = pos + 1$;
        end
    until the end of $F$ is met;
    return -1
end
```

4.2. Comparison of MCClusteringSM and Diviz mccClusters method

The study included a comparison with a multicriteria clustering method presented by Olteanu in [31]. This method is implemented in Diviz software [28] in the mccClusters component. This method was selected because it had good performance when exploiting fuzzy outranking relations to generate clusterings and by its availability. To use the mccClusters component, a workflow was designed in Diviz by adding the necessary components and parameters for the clustering generation.

Sample size and selection. Evaluating the mccClusters module with the 3,600 fuzzy outranking relations was impractical. We decided to use a sample whose size was calculated using the formula presented in [39] and defined as follows:

$$n = \frac{{Z_{\alpha/2}}^2 \cdot N \cdot p \cdot q}{E^2 \cdot (N-1) + {Z_{\alpha/2}}^2 \cdot p \cdot q} \quad (11)$$

where :

$$\begin{aligned} n &= \text{sample size,} \\ N &= \text{population size,} \\ Z_{\alpha/2} &= \text{critical value according to the confidence level selected,} \\ p &= \text{expected probability of the parameter to be evaluated,} \\ q &= 1 - p, \\ E &= \text{margin of error or imprecision allowed.} \end{aligned}$$

Thus, for a population of 3,600 fuzzy outranking relations, a 95% confidence level, a margin of error of 5.65%, and a probability p=0.50, the sample size was 277.14 fuzzy outranking relations:

$$n = \frac{1.96^2 \cdot 3600 \cdot 0.5 \cdot 0.5}{0.0565^2 \cdot (3600 - 1) + 1.96^2 \cdot 0.5 \cdot 0.5} = 277.14$$

Because the set of 3,600 fuzzy outranking relations was made up of 100 relations per each scenario of V_1 = 2, 3, 4, 5 clusters with V_2 = 6, 7, 8, 9, 10, 11, 12, 13, 14 alternatives; we decided to use a stratified sampling, composed of 36 strata, selecting randomly for practicality and convenience in each of them eight fuzzy outranking relations. Thus we selected 288 fuzzy outranking relations as a sample, mildly exceeding the minimum requirement established by the formula.

Clusters generation with Diviz mccClusters component. The mccClusters component computes clusters of alternatives based on the clustering typology: non-relational, relational, ordered, exclusive relational, and exclusive ordered multicriteria clustering [30]. In our case, the non-relational type was used because it groups a set of alternatives that are indifferent to each other, separating those that are not indifferent. In Data Analysis, each of these groups includes only one complementary relation (similarity and dissimilarity) [29]. Additionally, a fuzzy outranking relation is provided to the mccPreferenceRelation component. It creates a preference relation which is an input of the mccClusters component. It provides an option to define a cutting level from 0 to 1. If the cutting level is not used in an execution, this last is abruptly finished, and no result is produced. In this regard, Diviz issues an error message that this is probably due to a bug in the service. For this reason, the cutting level's default level (0.5) was used for this procedure.

Therefore, we exploited simulated fuzzy outranking relations to evaluate mccClusters, in the same way as MCClusteringSM, i.e., the success in finding the best solutions. The test included the same test criterion, variables, and response variable. For this test, a Diviz workflow was designed. It was composed of two file components for the alternatives and the fuzzy outranking relation, respectively. Also, it includes a mccPreferenceRelations component for the preference relations and a mccClusters for the final clustering. The Diviz workflow is shown in Figure 4.

The clusters generation with the mccClusters component was made considering a conversion. It was because the fuzzy outranking relations generated to test being constructed in plain text files. The conversion was made from plain text to the XMCDA format that Diviz uses. XMCDA is based on XML language for interoperability between different computer programs [28].

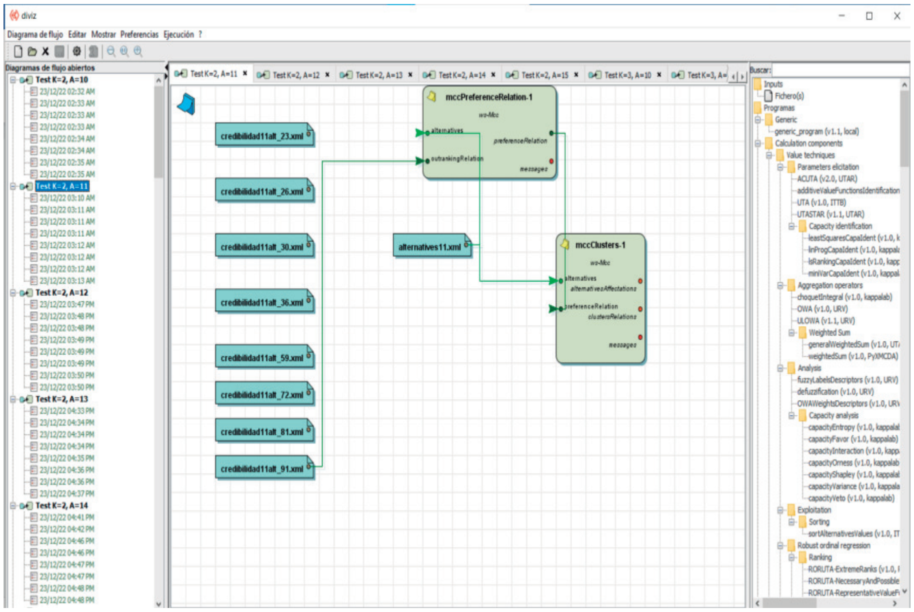

Fig. 4. Diviz workflow used for clusters generation with mccClusters component
Source: Own elaboration

Test criterion for evaluating the mccClusters component. In the same way, as in the test with MCClusteringSM, a solution without inconsistencies was considered just when the obtained clustering with mccClusters was the same as the reference clustering. This reference clustering is the vector V of alternatives of size N_A defined in step 5 of Algorithm 2. So, for our purpose, we can use the same logic for the response variable that in the previous test, considering that here the mccClusters module finds the best solution without inconsistency or zero otherwise. Then, the response variable can be defined as $Y(V_1, V_2) = \Sigma_{i=1}^{8} y_i(V_1, V_2)$, which is the sum of all the eight auxiliary binary variables per combination cause of the sampling used.

4.3. Empirical evaluation of MCClusteringSM considering a cutting level

In this section, we present another empirical evaluation to analyze the performance of MCClusteringSM. It included fuzzy outranking relations with different structures and sizes based on random cutting levels. Here, we attempted to capture the performance of MCClusteringSM with this new dataset and accomplish a comparison with the Diviz mcClusters method [30]. For this empirical evaluation, the fuzzy outranking relations were randomly generated so that there is a clustering of alternatives without inconsistencies for a given cutting level. The cutting level value was randomly obtained in a range of 0.5-0.75. The fuzzy outranking relations were created with a new version of the instance generator.

The test included the same test criterion, variables, values, and response variable as the empirical evaluation of section 4.2. However, this last differs in the number of replications. For each combination of the values of variables $V_1 = 2, 3, 4, 5$ clusters with

$V_2 = 6, 7, 8, 9, 10, 11, 12, 13, 14$ alternatives, the instance generator created 10 fuzzy outranking relations. Thus, 360 relations were created. This number is supported by the sample size calculated in subsection 4.2.1. The response variable was defined as $Y(V_1, V_2) = \Sigma_{i=1}^{10} y_i(V_1, V_2)$, which is the sum of all the ten binary variables per combination caused by the sampling used.

The MCClusteringSM was executed to exploit each of the generated fuzzy outranking relations. Then, we evaluated whether it successfully found the best solutions concerning the simulated fuzzy outranking relations. The valuations were done to get a reference point for the cutting level's inclusion in the input data generation. A solution without inconsistencies was considered just when the resulting clustering was the same as the one created by the instance generator.

Data generation using a cutting level $\lambda - cut$ The general procedure to generate the fuzzy outranking relations of this empirical evaluation is presented in Algorithm 4.

```
Algorithm 4: Instance generator considering a cutting level
Input  : Number of clusters $N_C$, Number of Alternatives $N_A$
Output: Fuzzy Outranking Relation $S_A^\sigma$, vector $V$, and a cutting level $\lambda - cut$
Begin.
Form a vector $V$ of alternatives of size $N_A$ and randomly assign the $N_C$ clusters to $N_A$
  alternatives, to ensure each cluster has at least an alternative.
Randomly complete $V$ with the $N_C$ clusters.
Randomly generate a cutting level $\lambda - cut$ $(0.5 \leqslant \lambda - cut \leqslant 0.75)$ with the $N_C$ clusters.
Create a matrix $S_A^\sigma$ of size $N_A \times N_A$ and fill it with values $[0, 1]$ based on $V$ and $\lambda - cut$,
  i.e., the clusters have the alternatives assigned on that basis.
End.
```

The outputs of Algorithm 4 are a matrix, a vector, and a cutting level $\lambda - cut$. The first represents the fuzzy outranking relation according to the given cutting level $\lambda - cut$. The second one is an array indicating which alternative belongs to each cluster. The third one is a random cutting level between 0.5 and 0.75. The matrix and vector sizes depend on the number of alternatives (6-14), as in the previous test. The instance generator was implemented in C language and executed under the same conditions described for the previous test.

4.4. Comparison of MCClusteringSM and Diviz mccClusters method using a cutting level

As with MCClusteringSM, a cutting level was considered for the execution in the Diviz mccClusters component. It was done to compare the performance of both methods with this addition.

Like in the test with MCClusteringSM, a solution without inconsistencies was considered just when the resulting clustering with mccClusters was the same as the one created by the instance generator. For our purpose, the same test criterion, variables, and values, and for the response variable, the same logic was applied as in the previous test for the Diviz mccClusters method. Still, the replication number was 10 instead of 8, considering what was done in section 4.3.1. Thus, the response variable was defined as $Y(V_1, V_2) = \Sigma_{i=1}^{10} y_i(V_1, V_2)$. This response variable is the sum of all the ten binary variables per combination caused by the sampling used.

Equally, as described in subsection 4.2.2., a fuzzy outranking relation is provided to the mccPreferenceRelation component, creating a preference relation. This relation is an input of the mccClusters component. Whereas in the last test with the Diviz mccClusters method, a cutting level was not used, we provided a cutting level generated by the instance generator for this new test. This cutting level needs to be supplied in the mccPreferenceRelation component shown in Figure 4.

5. Results

The carried-out experiments allowed us to portray the behavior of the proposed method. This section presents the analysis results of MCClusteringSM with the fuzzy outranking relations simulated to evaluate the inconsistencies regarding reference clusterings. Furthermore, a second analysis is presented between the multicriteria clustering method mccClusters with a sample of induced instances for comparison purposes with ours.

5.1. Results of MCClusteringSM vs. Induced Clusters

This experiment allowed us to observe the performance of the proposed method in a controlled environment. Specifically, according to the reference clustering, its accuracy in obtaining the best solutions for each simulated fuzzy outranking relation. Table 5 shows the results of our response variable against each possible combination of the independent variables, i.e., variables V_1: number of clusters and V_2: number of alternatives.

In all sets, MCClusteringSM found all the best solutions without inconsistencies in each one of the 100 fuzzy outranking relations evaluated per scenario. So, we can infer that the method was effective in 100% of the experiment scenarios.

For all the instances from six to fourteen alternatives, the credibility similarity metric obtained in the best solution was equal to one. A value of one means the highest intensity of similarity between the alternatives, which suggests a similarity between the alternatives. It also indicates the best solution with the highest intensity, indicating that the clusters were consistent with the fuzzy outranking relation provided.

Additionally, the results suggest that there is evidence to conclude that the size of the clusters and alternatives does not affect the performance of the similarity metric. However, the computational complexity of the method is a weakness in time and computational resources due to its combinatorial nature. Instances of V_2 = 6 to 12 alternatives, i.e., fuzzy outranking relations of size 6×6 to 12×12, required from 1GB up to 12GB of RAM. The required time in seconds to exploit each instance from 8 to 12 alternatives on the Asus computer was less than 0.5, 1, 2, 5, and 25, respectively. On the other hand, instances of V_2 = 13, 14 alternatives required around 26GB and 127GB of RAM, respectively. The required time for these cases on the Lenovo server was 47 seconds for the first one and 5 minutes for the second one.

Table 5. Results of MCClusteringSM vs. induced clusterings

Clusters	Alternatives	Solutions	Clusters	Alternatives	Solutions
2	6	100	4	6	100
	7	100		7	100
	8	100		8	100
	9	100		9	100
	10	100		10	100
	11	100		11	100
	12	100		12	100
	13	100		13	100
	14	100		14	100
3	6	100	5	6	100
	7	100		7	100
	8	100		8	100
	9	100		9	100
	10	100		10	100
	11	100		11	100
	12	100		12	100
	13	100		13	100
	14	100		14	100

Note: Solutions columns show the number of results without inconsistencies found per combination using MCClusteringSM.

5.2. Results of mccClusters module vs. Induced Clusters

The results of the mccClusters component executions, with the 288 fuzzy outranking relations, are presented in Table 6. As can be observed, this mccClusters only generated nineteen outcomes without inconsistencies in this experiment.

The cited nineteen outcomes without inconsistencies correspond to multicriteria clusterings with 6, 7, 8, and 9 alternatives in the four numbers of clusters tested. 63.15% of these solutions correspond to clusterings with 4 and 5 clusters. Therefore, the best performance of mccClusters was for clusterings with 4 and 5 clusters. For clustering with 4 clusters, it obtained 62.5%, 12.5%, 0%, 0%, 0%, 0%, 0%, 0%, and 0% of solutions without inconsistencies for 6, 7, 8, 9, 10, 11, 12, 13, and 14 alternatives respectively. For clustering with 5 clusters, the results were 25%, 37.5%, 12.5%, 0%, 0%, 0%, 0%, 0%, and 0% of solutions without inconsistencies for 6, 7, 8, 9, 10, 11, 12, 13 and 14 alternatives respectively. On the other hand, the worst performance of mccClusters was for clustering with 2 clusters. In this case, the results were 25%, 12.5%, 0%, 0%, 0%, 0%, 0%, 0%, and 0% of solutions without inconsistencies for 6, 7, 8, 9, 10, 11, 12, 13, and 14 alternatives respectively. Figure 5 illustrates these results.

In a global sense, the effectiveness of mccClusters was 6.59% over the 288 fuzzy outranking relations of the experiment. Unlike MCClusteringSM, the computational complexity in mccClusters method is not properly a weakness in time and computational resources due to the time duration to exploit each instance being less than 3 seconds, and Diviz software was easily executed on the Asus computer.

Table 6. Results of mccClusters module vs. Induced Clusters

Clusters	Alternatives	FOR [1]								Clusters	Alternatives	FOR							
		1	2	3	4	5	6	7	8 [2]			1	2	3	4	5	6	7	8
2	6	0	0	0	0	1	0	1	0	4	6	0	0	1	1	1	1	0	1
	7	0	0	0	0	0	0	0	1		7	0	0	0	0	1	0	0	0
	8	0	0	0	0	0	0	0	0		8	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0		9	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0		10	0	0	0	0	0	0	0	0
	11	0	0	0	0	0	0	0	0		11	0	0	0	0	0	0	0	0
	12	0	0	0	0	0	0	0	0		12	0	0	0	0	0	0	0	0
	13	0	0	0	0	0	0	0	0		13	0	0	0	0	0	0	0	0
	14	0	0	0	0	0	0	0	0		14	0	0	0	0	0	0	0	0
3	6	0	0	0	1	0	0	0	0	5	6	0	0	1	0	0	0	1	0
	7	0	0	0	0	1	0	0	0		7	0	0	0	0	1	1	0	1
	8	0	0	0	0	0	0	0	0		8	1	0	0	0	0	0	0	0
	9	0	0	0	0	1	1	0	0		9	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0		10	0	0	0	0	0	0	0	0
	11	0	0	0	0	0	0	0	0		11	0	0	0	0	0	0	0	0
	12	0	0	0	0	0	0	0	0		12	0	0	0	0	0	0	0	0
	13	0	0	0	0	0	0	0	0		13	0	0	0	0	0	0	0	0
	14	0	0	0	0	0	0	0	0		14	0	0	0	0	0	0	0	0

[1] Fuzzy Outranking Relation.

[2] In columns 1-8, a value of 1 means that the mccClusters component generated a solution without inconsistencies against the reference clustering. A value of 0 indicates the opposite.

5.3. Test Results of the Proposed Method vs. Induced Clusters considering a cutting level

This test allowed us to observe the behavior of the proposed method with the addition of a cutting level for constructing the fuzzy outranking relation. Particularly if clustering generated by the proposed method as the best solution corresponds to the induced clustering. Table 7 shows the results of our response variable defined in 4.3. against each possible combination of the independent variables, i.e., variables V_1: number of clusters and V_2: number of alternatives.

MCClusteringSM found 218 outcomes without inconsistencies and with the highest GCSI in each possible combination of the 360 fuzzy outranking relations evaluated. Thus, the method was strictly effective in 60.55% of the experiment scenarios.

About the mentioned 218 outcomes without inconsistencies, the distribution by cluster was 75, 65, 42, and 36 solutions for 2, 3, 4, and 5 clusters, respectively. For clustering with 2 clusters, MCClusteringSM found 83.33% of solutions without inconsistencies. The result of clustering with 3 clusters was 72.22% solutions without inconsistencies. For clustering with 4 clusters, it obtained 46.67% solutions without inconsistencies, and finally, for clustering with 5 clusters, the result was 40%. These results can be seen in detail corresponding to the solutions for each alternative value in Figure 6.

From the previous results, 342 solutions without inconsistencies given by MCClusteringSM were in the top 5 highest GCSI. Moreover, each one of these solutions was the

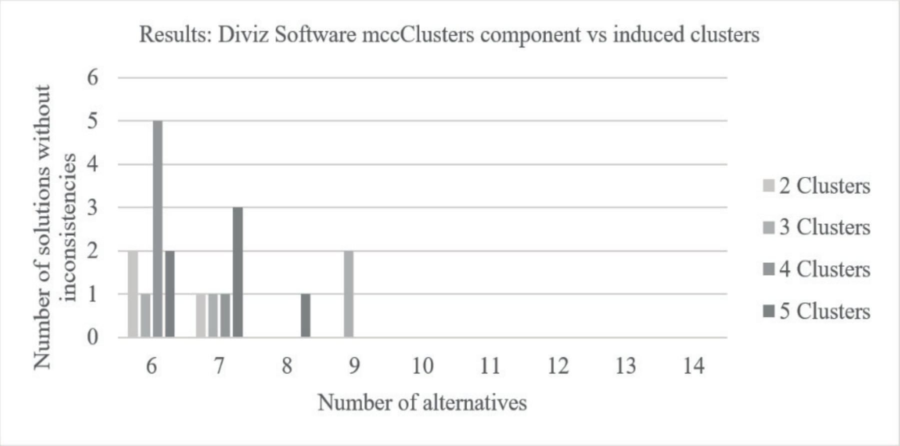

Fig. 5. Results of mccClusters for 288 fuzzy outranking relations

Source: Own elaboration

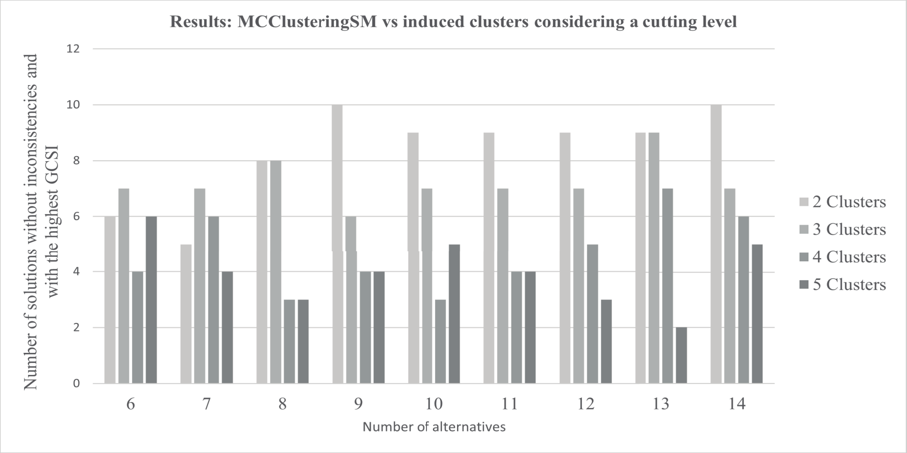

Fig. 6. Results of MCClusteringSM vs induced clusters for 360 fuzzy outranking relations

Source: Own elaboration

same as the reference clustering. Under this consideration, the method was effective in 95% of the scenarios, besides with values of credibility similarity metric from 0.988 to 1. Figure 7 illustrates these results.

Regarding the mentioned 342 outcomes, the distribution by cluster was 90, 84, 81, and 87 solutions for 2, 3, 4, and 5 clusters, respectively. In terms of percentage, it is 100%, 93.33%, 90%, and 96.66% of solutions without inconsistencies and within the top 5 highest GCSI for each cluster, respectively.

Table 7. MCClusteringSM vs. mccClusters using cut levels

Clusters	Alternatives	MCClusteringSM [1]										mccClusters [2]									
		1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
2	6	1	0	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1
	7	1	1	0	1	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
	8	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	10	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	11	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	12	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	13	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	6	1	0	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
	7	1	1	0	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
	8	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
	9	0	1	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1
	10	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1
	11	0	1	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1
	12	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
	13	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
	14	1	1	0	1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1
4	6	0	1	1	0	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1
	7	1	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	8	0	0	1	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1
	9	0	0	0	1	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	10	0	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
	11	1	1	0	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1
	12	1	1	1	0	0	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1
	13	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
	14	0	0	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1
5	6	0	1	0	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1
	7	1	0	0	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1
	8	0	0	0	0	1	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1
	9	1	0	1	0	0	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1
	10	1	1	0	0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1
	11	1	0	1	0	0	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1
	12	1	0	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1
	13	0	0	0	1	0	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1
	14	0	1	1	0	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1

[1] In columns labeled from 1-10, a value of 1 means that MCClusteringSM generated a solution without inconsistencies against the reference clustering. A value of 0 indicates the opposite.

[2] In columns labeled from 1-10, a value of 1 means that mccClusters generated a solution without inconsistencies against the reference clustering. A value of 0 indicates the opposite.

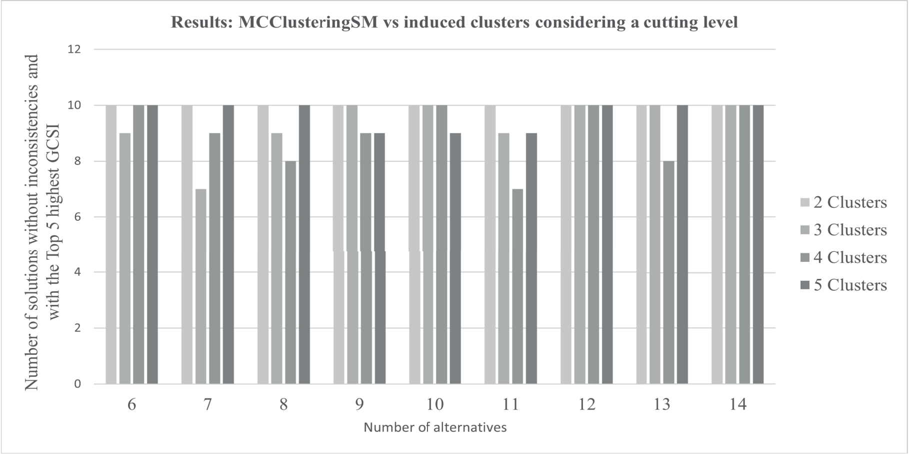

Fig. 7. Effectiveness of 95% of MCClusteringSM with 360 fuzzy outranking relations
Source: Own elaboration

5.4. Test mccClusters component results considering a cutting level

The results of the mccClusters component executions, with the 360 fuzzy outranking relations, are presented in Table 7. As can be observed, in all sets, this method found all the best solutions without inconsistencies in each of the 10 fuzzy outranking relations evaluated per scenario; namely, the clustering given by the Diviz mccClusters component as the best solution matches the induced clustering.

Furthermore, the results suggest that there is evidence to conclude that the size of the clusters and alternatives does not affect the performance of the mccClusters component. In a global sense, the effectiveness of mccClusters was 100% over the 360 fuzzy outranking relations of the experiment.

6. Concluding remarks and future perspectives

This paper addressed the problem of multicriteria clustering. We tackled this problem by using a method that includes a similarity measure to evaluate preferential information of the decision maker in a Multicriteria Decision Analysis context. The proposed method was designed to tackle the multicriteria clustering of other ones presented in the literature review of multicriteria clustering in section 2.3. There are several potential applications of multicriteria clustering, as many as there are in the choice, ranking, and sorting multicriteria problems. This problem has not received enough attention, but as methods are developed, this problem will be addressed more frequently.

Exploiting a preferences model to obtain clustering is a complex task. Nevertheless, based on the results, the proposed clustering procedure achieves good performances on the set of fuzzy outranking relations. Importantly, it outperforms the compared algorithm in finding the reference solutions. This approach can effectively exploit fuzzy outranking relations with up to 14 alternatives due to the computational complexity of deriving a clustering. Due to MCClusteringSM being an exhaustive search method, it evaluates all possible forms of clustering and can obtain the best partitioning by generating clusters with the measure of similarity.

When the cutting level was used for the generation of the fuzzy outranking relations to be evaluated, outperformance with the compared algorithm was not achieved as in the first comparison made. However, 95% of the time, the reference solution was found in the top 5 best outcomes of our method, which indicates good effectiveness taking into consideration that MCClusteringSM does not use a cutting level for clustering construction whereas mccClusters component does. This $\lambda - cut$ value needs to be supplied and must be defined by the researcher from 0 to 1.

By using the similarity metric in the multicriteria clustering problem, an effort is being made to contribute to avoiding the cutting level λ, which sometimes is questionable and hard to assign it. In our approach, a metric is used directly to the input data, i.e., the credibility levels of the fuzzy outranking relation. It means that the clusterings are obtained using only the intrinsic information in the relation that already contains the decision maker's preferences. It avoids including a subjective decision when the cutting level is elicited to construct the clusters. The results indicate MCClusteringSM has good effectiveness even if it does not use a cutting level. On the other hand, mccClusters component also has good success but uses the subjective cutting level.

For future work, the proposed method should be improved regarding time and computational resources. Also, we consider that more validation tests are required using real and simulated datasets to highlight the efficiency of the MCClusteringSM algorithm. It will allow us to explore the method's limits by finding the maximum size within alternatives that can be solved with acceptable performance. Moreover, we will incorporate a cutting level in the simulated set generation to analyze any interesting characteristics from results against multicriteria clustering methods that use that cut level. We know the exhaustive characteristic of MCClusteringSM is its weakness because of the combinatorial nature of the problem. Thus, the proposed method will be extended using a metaheuristic based on evolutionary algorithms. Finally, applying the proposed method to a real problem is planned. The last would be interesting to value its behavior against other multicriteria clustering methods empirically.

References

1. Baroudi, R., Safia, N.B.: Towards multicriteria analysis: A new clustering approach. In: 2010 International Conference on Machine and Web Intelligence, ICMWI 2010 - Proceedings (2010)
2. Baroudi, R., Safia, N.B.: Towards the definition of relations between clusters in multicriteria decision aid clustering. In: Procedia Computer Science. vol. 17, pp. 134–140. Elsevier B.V. (2013)
3. Baroudi, R., Safia, N.B.: A multicriteria clustering approach based on similarity indices and clustering ensemble techniques. International Journal of Information Technology and Decision Making 13(4), 811–837 (2014)
4. Bashir, M.A., Muhiuddin, G., Rashid, T., Sardar, M.S.: Multicriteria Ordered the Profile Clustering Algorithm Based on PROMETHEE and Fuzzy c-Means. Mathematical Problems in Engineering 2023, 5268340 (2023), https://doi.org/10.1155/2023/5268340
5. Bisdorff, R.: Electre-like clustering from a pairwise fuzzy proximity index. European Journal of Operational Research 138(2), 320–331 (2002)

6. Boujelben, M.A.: A unicriterion analysis based on the promethee principles for multicriteria ordered clustering. Omega 69, 126–140 (2017), https://www.sciencedirect.com/science/article/pii/S0305048316305126
7. Bouyssou, D.: Outranking relations: Do they have special properties? Journal of Multi-Criteria Decision Analysis 5(2), 99–111 (1996)
8. Brans, J.P., Vincke, P., Vincke, P.H.: A Preference Ranking Organisation Method: (The PROMETHEE Method for Multiple Criteria Decision-Making) A PREFERENCE RANKING ORGANISATION METHODt (The PROMETHEE Method for Multiple Criteria Decision-Making). Tech. rep. (1985)
9. Buttigieg, P.L., Ramette, A.: A guide to statistical analysis in microbial ecology: A community-focused, living review of multivariate data analyses. FEMS Microbiology Ecology 90(3), 543–550 (2014)
10. Cailloux, O., Lamboray, C., Nemery, P.: A taxonomy of clustering procedures (01 2007)
11. Chen, L., Xu, Z., Wang, H., Liu, S.: An ordered clustering algorithm based on K-means and the PROMETHEE method. International Journal of Machine Learning and Cybernetics 9(6), 917–926 (2018)
12. Corrente, S., De Smet, Y., Doumpos, M., Greco, S., Zopounidis, C.: Classification, sorting and clustering methods based on multiple criteria: recent trends. Annals of Operations Research pp. 767–770 (2023)
13. Daneshvar, A., Homayounfar, M., Farahmandnejad, A.: Developing an Intelligent Multi Criteria Clustering Method Based on PROMETHEE. Journal of Industrial Management Perspective 9(Issue 4, Winter 2020), 41–61 (2020), https://jimp.sbu.ac.ir/article_87470.html https://jimp.sbu.ac.ir/article_87470_6bb05ceeda7f456d6992e185c421b7d9.pdf
14. De Smet, Y.: P2CLUST: An extension of PROMETHEE II for multicriteria ordered clustering. IEEE International Conference on Industrial Engineering and Engineering Management pp. 848–851 (2014)
15. De Smet, Y., Eppe, S.: Evolutionary Multi-Criterion Optimization. Evolutionary Multi-Criterion Optimization 5467(April), 380–392 (2009), http://www.springerlink.com/content/x79325217n618v74
16. De Smet, Y., Guzmán, L.M.: Towards multicriteria clustering: An extension of the k-means algorithm. European Journal of Operational Research 158(2), 390–398 (oct 2004)
17. Doumpos, M., Zopounidis, C.: Multicriteria Decision Aid Classification Methods. Kluwer Academic Publishiers (2004)
18. Doumpos, M., Figueira, J.R., Greco, S., Zopounidis, C. (eds.): New perspectives in multiple criteria decision making. Multiple Criteria Decision Making, Springer Nature, Cham, Switzerland, 1 edn. (2019)
19. Fernandez, E., Navarro, J., Bernal, S.: Handling multicriteria preferences in cluster analysis. European Journal of Operational Research 202(3), 819–827 (may 2010)
20. Fodor, J.C., Roubens, M.: Fuzzy preference modelling and multicriteria decision support. In: Theory and Decision Library A: (1994)
21. Hwang, Ching-Lai Yoon, K.: Multiple Attribute Decision Making. Lecture Notes in Economics and Mathematical Systems, Springer Berlin-Heidelberg, Berlin, Switzerland, 1 edn. (1981)
22. Ishizaka, A., Lokman, B., Tasiou, M.: A Stochastic Multi-criteria divisive hierarchical clustering algorithm. Omega (United Kingdom) 103, 102370 (2021), https://doi.org/10.1016/j.omega.2020.102370
23. Kandakoglu, M., Walther, G., Amor, S.B.: A robust multicriteria clustering methodology for portfolio decision analysis. Computers & Industrial Engineering 174, 108803 (2022), https://www.sciencedirect.com/science/article/pii/S0360835222007914
24. Legendre, L., Legendre, P.: Numerical ecology. Developments in Environmental Modelling, 3.– (01 1983)

25. Leyva López, J.C., Solano Noriega, J.J., Figueira, J.R., Liu, J., Gastélum Chavira, D.A.: Non-dominated sorting genetic-based algorithm for exploiting a large-sized fuzzy outranking relation. European Journal of Operational Research 293(2), 615–631 (2021), https://www.sciencedirect.com/science/article/pii/S0377221720310699
26. Linkov, I., Moberg, E., Trump, B.D., Yatsalo, B., Keisler, J.M.: Multi-Criteria Decision Analysis Case Studies in Engineering and the Environment. CRC Press, second edn. (2021)
27. Liu, X., Yu, H., Wang, G., Guo, L.: A multi-criteria ordered clustering algorithm based on PROMETHEE (2020)
28. Meyer, P., Bigaret, S.: Diviz: A software for modeling, processing and sharing algorithmic workflows in MCDA. Intelligent Decision Technologies 6(4), 283–296 (2012)
29. Meyer, P., Olteanu, A.L.: Formalizing and solving the problem of clustering in MCDA. European Journal of Operational Research 227(3), 494–502 (2013)
30. Olteanu, A.L.: Decision Deck XMCDA Web Services mccClusters (2013), https://www.decision-deck.org/ws/webservices_xmcda-v4/wsd-mccClusters-ws-Mcc.html#myPage
31. Olteanu, A.L.: On clustering in multiple criteria decision aid : theory and applications. Ph.D. thesis (06 2013)
32. Qi, F.: An Explicit Formula for the Bell Numbers in Terms of the Lah and Stirling Numbers. Mediterranean Journal of Mathematics 13(5), 2795–2800 (2016)
33. Rokach, L., Maimon, O.: Clustering Methods, pp. 321–352. Springer US, Boston, MA (2005), https://doi.org/10.1007/0-387-25465-X_15
34. Rosenfeld, J., Smet, Y.D., Debeir, O., Decaestecker, C.: Assessing partially ordered clustering in a multicriteria comparative context. Pattern Recognition 114, 107850 (2021), https://doi.org/10.1016/j.patcog.2021.107850
35. Roy, B.: Méthodologie multicritère d'aide à la décision. Politiques et Management Public (1986)
36. Roy, B.: Classement et choix en presence de points de vue multiples (La méthode Electre). Revue française d'informatique et de recherche opérationnelle (1968)
37. Roy, B.: The outranking approach and the foundations of electre methods. Theory and Decision (1991)
38. Sarrazin, R., De Smet, Y., Rosenfeld, J.: An extension of PROMETHEE to interval clustering. Omega (United Kingdom) 80, 12–21 (2018), https://doi.org/10.1016/j.omega.2017.09.001
39. Scheaffer, R.L.: Elementos de muestreo. Thomson Editores Spain Paraninfo, Madrid [etc, sixth ed. edn. (2006)
40. Sloane, N.: Bell or exponential numbers: number of ways to partition a set of n labeled elements. (2014), https://oeis.org/search?q=a000110&sort=&language=&go=Search.
41. Valencia, A., Qiu, J., Chang, N.B.: Integrating sustainability indicators and governance structures via clustering analysis and multicriteria decision making for an urban agriculture network. Ecological Indicators 142, 109237 (2022), https://www.sciencedirect.com/science/article/pii/S1470160X22007099
42. Veltkamp, R.C., Latecki, L.J.: Properties and performance of shape similarity measures. In: Batagelj, V., Bock, H.H., Ferligoj, A., Žiberna, A. (eds.) Data Science and Classification. pp. 47–56. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
43. Vlachos, M.: Encyclopedia of Machine Learning and Data Mining. In: Sammut, C., Webb, G. (eds.) Similarity Measures, vol. 3, pp. 1163–1666. Springer, second edn. (2017)

Cesar Lugo Medrano is a PhD student in Business Analytics from Universidad Autonoma de Occidente, Mexico, and holds an M.S. in Engineering with a specialty in Quality and Productivity Systems from Instituto Tecnologico y de Estudios Superiores de Monterrey, Universidad Virtual. He holds a B. S. in Industrial and Systems Engineering from Instituto Tecnologico y de Estudios Superiores de Monterrey, Campus Sinaloa, Mexico, and is professor of the undergraduate degree at Tecnologico de Monterrey, Campus Sinaloa, and collaborates as a tutor-professor for postgraduate studies in the Online Education modality in the Master's Degree in Quality and Productivity Systems at Tecnologico de Monterrey.

Diego Alonso Gastelum Chavira is a full-time Economics and Management Sciences Department professor at Universidad Autonoma de Occidente, Mexico. He holds a PhD in Management Sciences from Universidad de Occidente, Mexico, an M.S. in Applied Informatics, and a B.S. in Informatics from Universidad Autonoma de Sinaloa, Mexico. He is a member of the National Researchers System of Mexico.

Octavio Valdez Lafarga is a full-time Engineering and Technology Department professor at Universidad Autonoma de Occidente, Mexico, and coordinator for the Business Analytics Master of Science program at the same university. He holds a PhD in Business Administration (concentration in Agribusiness) from Arizona State University and holds an M.S. in Agribusiness from Arizona State University. He holds a B. S. in Industrial and System Engineering from Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Sinaloa, Mexico.

Jose Luis Velarde Cervantes is a B. S. student in Informatics from Universidad Autonoma de Sinaloa, Mexico.

Received: March 02, 2023; Accepted: March 27, 2024.