MPOC: An Agglomerative Algorithm for Multicriteria Partially Ordered Clustering

Clara Rocha[*]
Luis C. Dias[†]

Abstract

In the field of multicriteria decision aid, considerable attention has been paid to supervised classification problems where the purpose is to assign alternatives into predefined ordered classes. In these approaches, often referred to as sorting methods, it is usually assumed that classes are either known a priori or can be identified by the decision maker. On the other hand, when the objective is to identify groups (clusters) of alternatives sharing similar characteristics, the problem is known as a clustering problem, also called an unsupervised learning problem. This paper proposes an agglomerative clustering method based on a crisp outranking relation. The method regroups alternatives into partially ordered classes, based on a quality of partition measure which reflects the percentage of pairs of alternatives that are compatible with a decision-maker's multicriteria preference model.[1]

keywords: Multi-criteria decision aiding (MCDA), Sorting problem, Clustering, Outranking relations, Agglomerative algorithm, Partial ranking

1 Introduction

The general definition of classification is the assignment of a finite set of alternatives (actions, objects, projects,...) into predefined classes (categories, groups). There are several more specific terms often used to refer to this form of decision making problem. The most common ones are Discrimination, Classification and Sorting. The term "sorting" has been established by MCDA researchers to refer to problems where the groups are defined in an ordinal way [6]. The definition of the groups does not only

[*] INESC Coimbra, R. Antero de Quental 199, 3000-033 Coimbra, Portugal. Escola Superior de Tecnologia da Saúde de Coimbra, Instituto Politécnico de Coimbra, Rua 5 de Outubro, S. Martinho do Bispo, Ap. 7006, 3040-162 Coimbra, Portugal. clarapr@estescoimbra.pt

[†] INESC Coimbra, R. Antero de Quental 199, 3000-033 Coimbra, Portugal. Faculdade de Economia, Universidade de Coimbra, Av. Dias da Silva 165, 3004-512 Coimbra, Portugal.

[1] This is a PDF file of an unedited manuscript that has been accepted for publication in 4OR. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain. The final version will be available at: https://link.springer.com/article/10.1007

provide a simple description of the alternatives, but it also incorporates additional preferential information, which could be of interest to the decision making context.

Another widely referenced technique for the resolution of several practical problems is *Clustering*. It is important to emphasize the difference between classification and clustering: in classification problems the groups are defined a priori, whereas in clustering the objective is to identify groups (clusters) of alternatives sharing similar characteristics. In other words, in classification problems the analyst knows in advance what the results of the analysis should look like, while in clustering the analyst tries to organize the knowledge embodied in a data sample in the most appropriate way according to some similarity measure [6].

Traditionally, cluster analysis algorithms can be classified as *hierarchical* (which require building a tree hierarchy) or as *non-hierarchical* (or partitional) (which do not build a tree, assigning alternatives to clusters after the number of groups to be formed is specified). Agglomerative clustering starts with all clusters with a single alternative and recursively merges two or more clusters (in contrast, divisive clustering starts with a single cluster with all the alternatives and then recursively partitions the clusters until a stopping criterion is met).

Recently, some multicriteria clustering procedures have been proposed aiming to discover data structures from a multicriteria perspective ([3], [4], [5], [7], [9], [10], [22]). Such works aim to not only detect groups with similar actions, but detecting also the preference relations between groups found. The definition of similarity, which is based on binary relations of preference between actions, makes these methods original. In this work, the central idea for definition of similarity is to characterize the quality of the model by measuring the degree of compliance with principles for multicriteria clustering. Four possible consistency principles will be discussed.

The work developed to address the multicriteria clustering problematic has mainly focused on the assignment of alternatives to totally ordered classes (with a few exceptions, e.g. [20] and [21]). But there is a variety of real world problems where some alternatives are not comparable. For instance, in diagnosing the attention-deficit hyperactivity disorder (ADHD) [29] the answers to questions about past and present problems are evaluated and we can find: A) children displaying no relevant symptoms, B) children with ADHD with predominantly inattentive type, C) children with ADHD with predominantly hyperactive-impulsive type, and D) ADHD of combined type (displaying both types of symptoms). Class A is preferable to B and C, and the latter are preferable to D, but one could consider that classes B and C are not comparable.

De Smet and Eppe [4] developed a method (an extension of the K-means algorithm) not only to detect clusters in a multicriteria context but also to identify relations between these clusters on the basis of a binary outranking matrix. We are interested in this last issue. Specifically, we are interested in classification with partially ordered classes where a class can be higher or lower ranked in comparison with some classes, whereas it can be considered incomparable with other different classes. We constrain the resulting relation to be a partial order.

We will adapt the standard agglomerative hierarchical clustering method ([14, 16]). It can be considered a heuristic approach since it does not attempt to derive an optimal partial order among all conceivable clusters of alternatives. The goal of this new method is to obtain a set of partitions, where each partition consists of a set of classes, together with relations denoting the preference order of the classes. These relations between classes are not supposed to be complete, but must be transitive and contain no cycles.

This paper is organized as follows: a brief theoretical background is given in Section 2. Four consistency principles for multicriteria clustering with partially ordered classes are discussed in Section 3, followed by the definitions of the preference structures on classes and the partition quality (Section 4). Our proposal for the agglomerative method is presented in Section 5. Section 6 presents an illustrative example and Section 7 concludes the paper.

2 Preference structures on alternatives

Let $\mathscr{A}$=$\{a_1,a_2,...,a_n\}$ denote a set of existing or fictitious alternatives represented by a vector of evaluations on m criteria. A comparison of the alternatives is the main component in any decision problem. According to Roy and Bouyssou [27], a model of preferences considers the following relations: Preference (P), Indifference (I) and Incomparability (R). Such relations are such that $\forall a_i, a_j \in \mathscr{A}$:

$$\begin{cases} a_iPa_j \Longrightarrow a_j \not{P} a_j & \text{: P is asymmetric} \\ a_iPa_j \wedge a_jPa_k \Rightarrow a_iPa_k & \text{: } P \text{ is transitive} \\ a_iIa_i & \text{: } I \text{ is reflexive} \\ a_iIa_j \Longrightarrow a_jIa_j & \text{: } I \text{ is symmetric} \\ a_iIa_j \wedge a_jIa_k \Rightarrow a_iIa_k & \text{: } I \text{ is transitive} \\ a_i \not{R} a_i & \text{: } R \text{ is not reflexive} \\ a_iRa_j \Longrightarrow a_jRa_j & \text{: } R \text{ is symmetric} \end{cases} \tag{1}$$

definition ([30]) The relations $\{P,I,R\}$ constitute a preference structure of $\mathscr{A}$ if the condition (1) holds and if, given any two alternatives a_i, a_j of $\mathscr{A}$, only one of the following properties holds: a_iPa_j, a_jPa_i, a_iIa_j or a_iRa_j.

A preference structure $\{P,I,R\}$ can be the result of applying multicriteria outranking methods (Electre ([26]) and Promethee ([1]) are typical examples). Note that the outranking methods define the relation of incomparability R, unlike for instance Utility-based methods ([17]). Many of the outranking methods, as the name suggests, are based on the *outranking relation*, a binary relation on A, which traditionally corresponds to $P \cup I$: given an outranking relation S, a_i S a_j means that "a_i is at least as good as a_j", i.e., a_iPa_j or a_iIa_j. There are four different situations that can result when comparing two alternatives a_i and a_j :

$$\begin{cases} a_iSa_j \wedge a_j \not{S} a_i \Longleftrightarrow a_iPa_j \text{ } (a_i \text{ is preferable to } a_j); \\ a_jSa_i \ \wedge \ a_i \not{S} a_j \Longleftrightarrow a_jPa_i \text{ } (a_j \text{ is preferable to } a_i); \\ a_iSa_j \ \wedge \ a_jSa_i \Longleftrightarrow a_iIa_j \text{ } (a_i \text{ is indifferent to } a_j); \\ a_i \not{S} a_j \ \wedge \ a_j \not{S} a_i \Longleftrightarrow a_iRa_j \text{ } (a_i \text{ is incomparable to } a_j). \end{cases} \tag{2}$$

3 Principles for multicriteria clustering with partially ordered classes.

In this Section, we discuss four principles for multicriteria clustering with partially ordered classes.

A partition of $\mathscr{A}$ in k classes $\mathscr{P}_k$=$\{C^1, C^2, ..., C^k\}$ is defined as follows:

- $C^s \neq \emptyset, s = 1, ..., k$
- $\mathscr{A} = \cup_{s=1,...,k} C^s$
- $C^s \cap C^t = \emptyset, s \neq t.$

Given any two classes, C^s, $C^t \in \mathscr{P}_k$, let $C^s \tau C^t$ denote "the class C^s is at least as good as class C^t ". Having a relation τ and two classes $C^s, C^t \in \mathscr{P}_k$, $C^s \succ C^t$ refers to the asymmetric part of τ and denotes a strict preference of C^s over C^t, and $C^s \perp C^t$ denotes incomparability and applies to pairs (C^s, C^t) unrelated by τ.

A partial order can be defined among the classes: a class can be higher or lower ranked in comparison with some classes, whereas it can be considered incomparable with other different classes. If a partial order is sought, there are three different situations that can result when comparing two classes C^s and C^t:

$$\begin{cases} C^t \tau C^s \wedge \ C^s \not\tau \ C^t \Longleftrightarrow C^t \succ C^s \text{ (}C^t\text{ is better than }C^s\text{)} \\ C^s \tau C^t \wedge C^t \not\tau C^s \Longleftrightarrow C^s \succ C^t \text{ (}C^s\text{ is better than }C^t\text{)} \\ C^s \not\tau C^t \wedge C^t \not\tau C^s \Longleftrightarrow C^s \perp C^t \text{ (}C^s\text{ is incomparable to }C^t\text{)} \end{cases} \tag{3}$$

To cluster a set of alternatives for which S is given into a set of classes for which a partial order is to be defined, we can follow different ideas, as described next.

3.1 STrong (Sτ)-Consistency

One idea is to base the assignment of the alternatives on a **strong consistency principle** (**Sτ - Consistency**): "an alternative a_i outranks an alternative a_j if and only if the class of a_i is at least as good as the class of a_j", i.e.

$$a_i S a_j \Leftrightarrow C(a_i) \ \tau \ \mathrm{C}(a_j) \qquad (S\tau\textit{-Consistency})$$

where C(a_i) denotes the class to which alternative a_i is assigned to.

This is an appealing principle given that $a_i S a_j$ means "a_i is at least as good as (is not worse than) a_j ".

The following corollaries result from Sτ - Consistency:

(1) a_i P a_j if and only if a_i belongs to a class considered to be better than the class of a_j, i.e., $(a_i S a_j \wedge a_j \not S a_i) \Longleftrightarrow C(a_i) \succ C(a_j)$.

(2) a_i I a_j if and only if a_i belongs to the same class as a_j, i.e., $(a_i S a_j \wedge a_j S a_i) \Longleftrightarrow C(a_i) = C(a_j)$.

(3) a_i R a_j if and only if a_i belongs to a class considered to be incomparable to a_j, i.e., $(a_i \not S a_j \wedge a_j \not S a_i) \Longleftrightarrow C(a_i) \perp C(a_j)$.

An example that satisfies the Sτ - Consistency conditions is depicted in Figure 1. The only way it is possible to fully comply with these requirements is to have an S relation that is transitive. For instance, if $a_1 \not S a_3$ in Figure 1, then it would not possible to fully comply with Sτ-Consistency: we would need to say that a_1 belongs to a class better than a_4 and that a_4 belongs to the same class as a_3, but we could not conclude that the first class is better than the latter since a_1 and a_3 would be required to be in incomparable classes. Hence, we will next suggest less stringent forms of consistency.

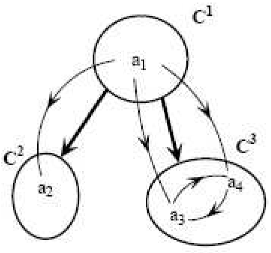


Figure 1: Illustration of Sτ - Consistency

An optimization procedure may look for solutions that minimize the number of violating pairs of alternatives or taking into account the strongest violation given a valued S relation, e.g, we may look for a least-costly violation solution. Dias and Lamboray (2010) show how to obtain the set of optimal solutions using the “min” or “lexicographical min” criteria for this problem according to a strategy of minimizing the strongest violation in the context of ranking problems. These authors also show that it is easy to find examples in which the strongest violation is indeed very strong. Hence, we will next suggest less stringent forms of consistency which might be an interesting for a clustering procedure to pursue.

3.2 Forms of Sτ-Consistency relaxed

Sτ-Consistency requires consistency from S to τ and vice-versa. Relaxing the *Sτ-Consistency*, we get two weak forms, which we call S-Consistency and τ-Consistency, resulting from splitting the two-way principle of equivalence into two one-way principles. We will first discuss these two forms of relaxation, and then we will introduce a third type of relaxation, which we call SSτ-Consistency, based on a different rationale.

A. S-Consistency

Considering **S-Consistency**, we base the assignment of the alternatives on the principle “if an alternative a_i outranks an alternative a_j then the class of a_i must be at least as good as the class of a_j", i.e.

$$a_iSa_j \Rightarrow C(a_i)\ \tau\ C(a_j) \qquad (\textit{S-Consistency})$$

The following corollaries result form S-Consistency:

(1) if a_i P a_j $(a_iSa_j \wedge a_j \not S a_i)$ then $C(a_i)\tau C(a_j)$.

(2) if a_i I a_j $(a_iSa_j \wedge a_jSa_i)$ then $C(a_i) = C(a_j)$, i.e., a_i must belong the same class as a_j.

(3) if a_i R a_j then a_i and a_j could belong to any class.

Note that, if there exists a cycle in the outranking relation $(a_iSa_jSa_lS...Sa_i)$, then S-Consistency also implies that all the alternatives involved in the cycle must be placed in the same class. This follows the philosophy of ELECTRE I and this is the principle proposed by Rocha and Dias (2008) for an interactive algorithm for ordinal classification.

The S-Consistency principle can be rephrased as stating that "alternatives belonging to a given class cannot be outranked by any alternative belonging to a lower or incomparable class, and cannot outrank any alternative belonging to a higher or incomparable class". Consequently:

(4) if C(a_i) is better than C(a_j) then we can have a_i P a_j or a_j R a_i.

(5) if C(a_i) is incomparable to C(a_j) then we must have a_i R a_j.

(6) if C(a_i) = C(a_j) (i.e., $a_i$and a_j are in the same class) then we can have a_i P a_j, a_i I a_j, or even a_i R a_j.

Note that placing all alternatives in the same class we obtain a trivial weak S-consistent solution. Hence, this relaxation of S-Consistency is not fully satisfactory.

B. τ - Consistency

Considering τ - Consistency, we base the assignment of the alternatives on the principle " if the class of a_i is at least as good as the class of a_j then the alternative a_i outranks the alternative a_j ":

C(a_i) τ C(a_j) $\Rightarrow$ a_i S a_j (τ - *Consistency*)

The following corollaries result form τ-Consistency:

(1) if C(a_i) is better than C(a_j) then we can have a_i P a_j or a_i I a_j.

(2) if C(a_i) is incomparable to C(a_j) we can have a_i P a_j, a_j P a_i, a_i R a_j, or even a_i I a_j.

(3) if C(a_i) = C(a_j) then we must have a_i I a_j.

Note that, by corollary 2, if each class has only one alternative and the classes are declared all incomparable then this would be a τ-Consistent solution. Again, this relaxation of Sτ-Consistency is not fully satisfactory.

Table 1: Combinations that are forbidden by S-consistency (S) and combinations that are forbidden by τ-consistency (τ)).

		a_iPa_j		a_jPa_i		a_iIa_j		a_iRa_j	
		a_iSa_j	$a_j \not S a_i$	$a_i \not S a_j$	a_jSa_i	a_iSa_j	a_jSa_i	$a_i \not S a_j$	$a_j \not S a_i$
C(a_i) $\succ$ C(a_j)	C(a_i) τ C(a_j)			τ				τ	
	C(a_j) $\not\tau$ C(a_i)				S		S		
C(a_i) = C(a_j)	C(a_i) τ C(a_j)			τ				τ	
	C(a_j) τ C(a_i)		τ						τ
C(a_i) $\perp$ C(a_j)	C(a_i) $\not\tau$ C(a_j)	S				S			
	C(a_j) $\not\tau$ C(a_i)				S		S		

C. Semi-STrong (SSτ) - Consistency

Table 1 summarizes what has been discussed so far. Given two alternatives a_i and a_j belonging to classes C(a_i) and C(a_j), it shows the possible combinations that are inconsistent according to S-consistency or τ-consistency. It is clear from this table that S-consistency does not pose any constraints to placing two alternatives in the same class, and that τ-consistency does not pose any constraints to placing two alternatives in two incomparable classes.

Combinations forbidden by S-consistency or τ-consistency are also forbidden by strong consistency (Sτ-Consistency). In accordance with the principle Sτ-Consistency, the following inconsistencies should not occur:

(1) $C(a_i) \succ C(a_j)$, but $a_i \not P a_j$ (i.e, $a_i \not S a_j$ or a_jSa_i).

(2) $C(a_i) = C(a_j)$, but $a_i \not I a_j$ (i.e, $a_i \not S a_j$ or $a_j \not S a_i$).

(3) $C(a_i) \perp C(a_j)$, but $a_i \not R a_j$ (i.e, a_iSa_j or a_jSa_i).

Given a relation among two classes, strong consistency imposes a condition on a_iSa_j plus a condition on a_jSa_i. A different principle that can be used to relax strong consistency (besides S and τ-consistency) is to accept that one of these two conditions on S is violated. We will denote this relaxation of strong consistency by Semi-STrong consistency, or (SSτ)-Consistency. According to this new principle, the following inconsistencies should not occur:

(1) $C(a_i) \succ C(a_j)$ when a_jPa_i ($a_i \not S a_j$ and a_jSa_i).

(2) $C(a_i) = C(a_j)$ when a_iRa_j ($a_i \not S a_j$ and $a_j \not S a_i$).

(3) $C(a_i) \perp C(a_j)$ when a_iIa_j (a_iSa_j and a_jSa_i).

Table 1 also shows which combinations are now forbidden by semi-strong consistency. All the combinations that are forbidden either by S-consistency or τ-consistency but affect only one ordered pair, i.e., one direction, from a_i to a_j or from a_j to a_i are now accepted: if $C(a_i) \succ C(a_j)$ this principle accepts that they are indifferent or incomparable (besides accepting a_iPa_j); if $C(a_i) = C(a_j)$ or $C(a_i) \perp C(a_j)$ this principle accepts that one of the alternatives is preferred over the other (besides accepting a_iIa_j or a_iRa_j, respectively). On the other hand, this principle forbids shaded cells in Table 1, i.e, rejects combinations in which the two ordered pairs (i.e., direction from a_i to a_j and from a_j to a_i) are wrong, although such situations would be accepted by S-consistency or τ-consistency: it does not accept indifferent alternatives in incomparable classes (which τ-consistency accepts), and it does not accept incomparable alternatives the same class (which S-consistency accepts).

In the next section, we present an extension of the agglomerative hierarchical algorithm to the multicriteria framework, based on the (SSτ)-Consistency, given a possibly intransitive outranking relation S. The same ideas can be applied using Sτ-Consistency instead. For the reasons already mentioned, S-consistency and τ-consistency principles are less interesting. In fact, experiments that we performed with the principles S-Consistency and τ-Consistency showed that the S-Consistency principle leads to partitions with one of the categories with almost all the alternatives and the τ-Consistency principle leads to partitions with many incomparable categories.

4 Preference structures on classes and quality of the partition

Once the objects in $\mathscr{A}$ have been clustered, a partition of partially ordered classes of $\mathscr{A}$ can be defined. In the process of ranking clusters, we need arguments to establish when a cluster should be ranked higher than another. Such arguments can be obtained from comparing all alternatives that are grouped in different clusters. For this purpose we will define the outranking relation τ between classes as follows: for each pair (C^s, C^t), a class outranking degree θ_{st} (Definition 2) is computed indicating the degree to which C^s outranks C^t. The outranking relation is considered to hold if the outranking degree is at least 50%, i.e., a weak majority of the pairs agree with the outranking assertion (Definition 3).

definition Let $C^s, C^t \in \mathscr{P}_k$, and let n_s and n_t denote the number of alternatives of C^s and C^t respectively. The **outranking degree** of C^s on C^t represents the proportion of pairs of alternatives $(a_i, a_j) \in (C^s, C^t)$ that indicate a_i outranks a_j:

$$\theta_{st} = \frac{\sum\limits_{a_i \in C^s} \sum\limits_{a_j \in C^t} \theta(a_i, a_j)}{n_s \times n_t}, \text{ with } \theta(a_i, a_j) = \begin{cases} 1 & if \;\; a_iSa_j \\ 0 & \text{otherwise} \end{cases}.$$

definition Let $(C^s, C^t) \in \mathscr{P}_k \times \mathscr{P}_k$. We say that C^s **outranks** C^t $(C^s \tau C^t)$ iff the outranking degree of C^s on C^t is at least 0.5, i.e, $\theta_{st} \geq 0.5$.

According to the $SS\tau$-consistency principle, a partition $\mathscr{P}_k$ should be as close as possible to fully respecting of the following three principles of Preference, Indifference and Incomparability. First, the order of preference on classes should not contradict the order of preference of alternatives. The number of contradictions of this principle is given by v_P. Second, a class should contain comparable alternatives, leading to the definition of the number of contradictions v_I. Third, incomparable classes should not contain indifferent alternatives, leading to the definition of the number of contradictions v_R. The v_P, v_I and v_R indicators are defined as follows.

definition Let $\Gamma_P(a_i,a_j) = \begin{cases} 1 & iff \;\; a_jPa_i \\ 0 & \text{otherwise} \end{cases}$. The v_P index is defined as follows:

$$v_P = \sum_{(a_i,a_j)\in\mathscr{A}^2:C(a_i)\succ C(a_j)} \Gamma_P(a_i,a_j) \tag{4}$$

Let $\Gamma_I(a_i,a_j) = \begin{cases} 1 & iff \;\; a_iRa_j \\ 0 & \text{otherwise} \end{cases}$. The v_I index is defined as follows:

$$v_I = \sum_{(a_i,a_j)\in\mathscr{A}^2:C(a_i)=C(a_j)} \Gamma_I(a_i,a_j) \tag{5}$$

Let $\Gamma_R(a_i,a_j) = \begin{cases} 1 & iff \;\; a_iIa_j \\ 0 & \text{otherwise} \end{cases}$. The v_R index is defined as follows:

$$v_R = \sum_{(a_i,a_j)\in\mathscr{A}^2:C(a_i)\perp C(a_j)} \Gamma_R(a_i,a_j) \tag{6}$$

Fernandez et al. [8] proposed a multiobjective optimization problem, more flexible than a single objective optimization approach because it allows to model preferences on different objectives.

Like Fernandez et al. [8], we propose a partition quality defined on the basis of a consistency vector of the relation τ on the classes and the relation S on the alternatives. We define the quality of the partition through a single objective, assigning weights to inconsistencies of Preference, Indifference and Incomparability in accordance with the requirements of the decision maker. A decision maker can "play" with these weights to study trade-offs between the number of pairs violating each condition.

Therefore, let (v_P, v_I, v_R) be an *inconsistencies vector* associated to a partition $\mathscr{P}_k$ that will contain the pairs of alternatives that are not compatible with preference, indifference and incomparability conditions.

Let α_P, α_I and α_R denote the weights respectively assigned by the decision maker to conditions of Preference, Indifference and Incomparability. The *Quality* $Q(\mathscr{P}_k)$ of partition $\mathscr{P}_k$ is defined as follows

$$Q(\mathscr{P}_k) = 1 - 2 \times \frac{\alpha_P v_P + \alpha_I v_I + \alpha_R v_R}{n(n-1)} \tag{7}$$

with $\alpha_P, \alpha_I, \alpha_R \in$ [0, 1].

5 Agglomerative method for a transitive partially ordered clustering

The goal of this new method is to obtain a set of n partitions $\mathscr{P}_k$ of $\mathscr{A}$, k ∈ [1, n], where a partition $\mathscr{P}_k$ contains k classes, together with relations on $\mathscr{P}_k$ denoting the preference order of the classes composing the partition. These relations are not supposed to be complete, but must be transitive and contain no cycles. A partially ordered partition, with a structure of preferences $(\succ, \perp)$, can be defined as follows:

- $C^s \succ C^t \Rightarrow C^t \not\succ C^s$: $\succ$ is asymmetric
- $C^s \succ C^t \wedge C^t \succ C^r \Rightarrow C^s \succ C^r$: $\succ$ is transitive
- $C^s \perp C^t \Rightarrow C^t \perp C^s$: $\perp$ is symmetric

Following the principle of agglomerative clustering, the proposed method starts with $\mathscr{P}_n$, the partition of n classes each containing a single alternative. Then it iteratively builds $\mathscr{P}_{k-1}$ from $\mathscr{P}_k$ by choosing two classes to merge, until the n partitions $\mathscr{P}_1, \ldots, \mathscr{P}_n$ are built. The two classes to merge are the ones that yield the highest partition quality after merging.

Algorithm 1 - Agglomerative Method - extension

```
1. Enter outranking relation $S$ as an input
2. $C^i = \{a_i\}$ with $a_i \in \mathscr{A}, \forall i = 1, \ldots n$ ($n$ initial classes)
3. k=n (stage)
4. $\mathscr{P}_k = \{C^1, \ldots, C^n\}$
   While $k > 1$ do
5.     Determine the pair of neighbors $(C^s, C^t) \in \mathscr{P}_k \times \mathscr{P}_k$ such that $Q(\mathscr{P}_{k-1})$ is minimum
       when merging $C^s$ and $C^t$ to form a new class $C^r = C^s \bigcup C^t$
6.     Merge $C^s$ and $C^t$ to form a new class $C^r = C^s \bigcup C^t$
7.     Update preference structure of $\mathscr{P}_{k-1}$
8.     k=k-1
   end while
Check the partitions that have the number of clusters indicated by the decision-maker
for the transitivity property, and make corrections.
```

Note that in this proposed method, the decision maker is not required to specify the number of classes initially, which can be done in the end with the help of the method's results. Indeed, the decision maker may eventually decide the number of classes based on the violations of conditions of Preference, Indifference and Incomparability or even the relative size of the classes.

Another good point of this proposed method is that does not lead to major changes between relations τ of $\mathscr{P}_{k-1}$ and $\mathscr{P}_k$. In fact, the only changes are in relations concerning two classes. Thus it is easier for the decision maker to interpret and choose a good partition according to his preferences.

The aim of this method is to obtain a *transitive partially ordered partition* of a given set of alternatives. Thus, to guarantee the transitivity of the final partition, it is necessary to reassess the quality of the partitions that are not transitive and/or containing indifferent classes after taking the necessary corrective measures. This can be made only at the end or along of the algorithm. However, this last option makes the algorithm much slower which in our experiments was not compensated by better results.

A partition $\mathscr{P}_k$ derived from a nontransitive outranking relation (e.g., a relation built according to an Electre method), might include classes such that:

$\exists\{C^s,C^t,C^r\} \in \mathscr{P}_k$: $C^s\tau C^t \wedge C^t\tau C^r \wedge C^s \not\tau C^r$

Schematically:

$$\begin{tikzcd} C^s \arrow[r] & C^t \arrow[r] & C^r \end{tikzcd}$$

where each arc $\begin{tikzcd} C^s \arrow[r] & C^t \end{tikzcd}$ represents the existence of the relation $C^s\tau C^t$.

To enforce transitivity on the triplet $\{C^s,C^t,C^r\}$ there are three solutions:

1. To require the arc $C^s\tau C^r$:

$$\begin{tikzcd} C^s \arrow[r] \arrow[rr, bend left] & C^t \arrow[r] & C^r \end{tikzcd}$$

2. To eliminate the arc $C^s\tau C^t$:

$$\begin{tikzcd} C^s & C^t \arrow[r] & C^r \end{tikzcd}$$

3. To eliminate the arc $C^t\tau C^r$:

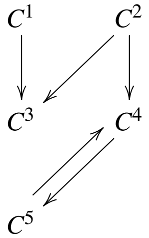
$$\begin{tikzcd} C^s \arrow[r] & C^t & C^r \end{tikzcd}$$

If a partition is intransitive, one should evaluate the quality of the possible solutions to restore transitivity (obtained by recursively considering the three ways of adjusting each intransitive triplet) and the one that leads to better quality will be the final partition. If the outranking relation between classes has a cycle ($C^s\tau C^t\tau ...\tau C^s$) then all partitions obtained by removing one of the relations of the cycle should be evaluated to choose the one that leads to better final quality. For indifferent classes ($C^s\tau C^t \wedge C^t\tau C^s$) a further possibility is to merge the two classes.

We propose an algorithm to restore transitivity (see Algorithm 2). The relation matrix W is an $k \times k$ Boolean (zero-one) matrix defined by $w_{st} = \begin{cases} 1 & \text{iff } C^s\tau C^t \\ 0 & \text{otherwise} \end{cases}$. Let

[W,s,t,r] denote the sub-matrix of W with the s-th, t-th and r-th rows and columns (dimension 3×3). For the algorithm to compute the best transitive and acyclic partition, we proposed a FIFO (First-In-First-Out) queue data structure[18]. At the beginning, the queue has matrix W. Every time that a new matrix W is removed from the queue, we test if there exists a triplet (s,t,r) such that the [W,s,t,r] sub-matrix is intransitive ($w_{st} = 1$ and $w_{tr} = 1$ and $w_{sr} = 0$) or has at least one cycle (($w_{st} = 1$ and $w_{ts} = 1$) or ($w_{rt} = 1$ and $w_{tr} = 1$)). If this occurs, new matrices are constructed replating W but with the sub-matrix corrected to be transitive and without cycles, according to the solutions presented above. If the new considered matrix has no cycles nor intransitivities and has the best quality of partitions studied, then the algorithm updates the best partition as well as its quality.

As an example, let us consider the outranking degrees between classes presented in Table 2. The corresponding relation between classes is depicted in Figure 2 and matrix W. In this structure there are two problems: the intransitivity of (C^2, C^4, C^5) and the cycle in (C^4, C^5).



$$W = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Figure 2: Preferred structure of $\mathscr{P}$

To enforce transitivity, and starting with cycle, there are five possible solutions, corresponding to the matrices given by Algorithm 2 (Figure3). The successive iterations of procedure Algorithm 2 would proceed as follows:

Algorithm 2 - Transitive and acyclic partitions

```
Input: W is an adjacency matrix corresponding to binary relations on $k$ elements.
Output: W*, Q(W*).                                   % adjacency matrix and its quality

queue:=W;
W*:=W;
Q(W*):=Q(W):
While NotEmpty(queue) do
    W := dequeue(queue);                % remove first element from queue
    for each triplet (s,t,r) do
      if $w_{st} = 1$ and $w_{ts} = 1$ then                        % Cycle
          W1:=W; merge $C^s$ and $C^t$; update W1; W2:=W; $w2_{st}$=0; W3:=W; $w3_{ts}$=0;
          insert (queue, W1, W2, W3);
          exit this for loop;
      end if
      if $w_{tr} = 1$ and $w_{rt} = 1$ then                        % Cycle
          W1:=W; merge $C^t$ and $C^r$; update W1; W2:=W; $w2_{rt}$=0; W3:=W; $w3_{tr}$=0;
          insert (queue, W1, W2, W3);
          exit this for loop;
      end if
      if $w_{st} = 1$ and $w_{tr} = 1$ and $W_{sr} = 0$ then              % Intransitivity
          W1:=W; $w1_{sr}$=1; W2:=W; $w2_{st}$=0; W3:=W; $w3_{tr}$=0;
          insert (queue, W1, W2, W3);
          exit this for loop;
      end if
    end for
    if $Q(W) > Q(W*)$ then W*:=W; Q(W*):=Q(W) end if
end while
```

Table 2: Outranking degree θ_{st} of C^s on C^t, s,t=1,...,5

	C^1	C^2	C^3	C^4	C^5
C^1		0.333	0.6	0.133	0.100
C^2	0.333		0.6	0.583	0.125
C^3	0.400	0.183		0.333	0.375
C^4	0.167	0.333	0		0.583
C^5	0.4	0.183	0	0.917	

Iteration 1 : w_{54}=1 and w_{45}=1	

W1:=W; merge C^4 and C^5; update W1;
W2:=W; $w2_{54}$=0;
W3:=W; $w3_{45}$=0;
(queue = $\{$D,I,C$\}$)
	Solution D
	Solution I (not displayed)
	Solution C

Iteration 2 :	
W:=Solution D update (W*, Q(W*))	
Iteration 3: w_{24}=1 and w_{45}=1 and w_{25}=0	

W:=Solution I;
W1:=W; $w1_{25}$=1;
W2:=W; $w2_{24}$=0;
W3:=W; $w3_{45}$=0;
(queue = $\{$C,H,G,B$\}$)
	Solution H
	Solution G
	Solution B

Iteration 4 :	
W:=Solution C (possibly) update (W*, Q(W*))	
Iteration 5 :	
W:=Solution H (possibly) update (W*, Q(W*))	
Iteration 6 :	
W:=Solution G (possibly) update (W*, Q(W*))	
Iteration 7 :	

W:=Solution B
(possibly) update (W*, Q(W*))
END
W* will be relation with highest quality Q(W*)	

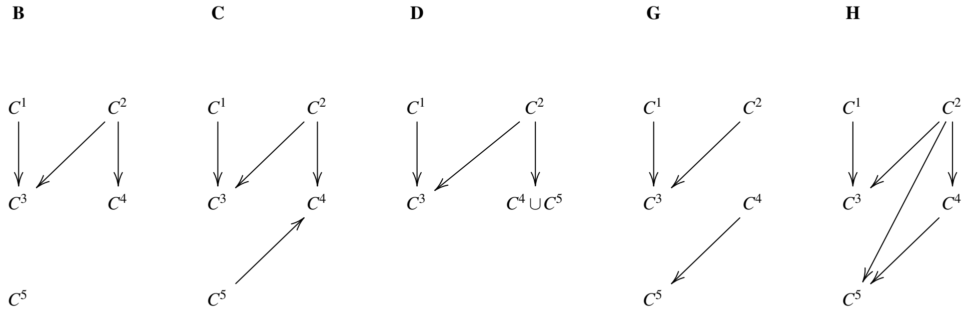


Figure 3: Possible solutions for intransitive partition $\mathscr{P}_s$

6 Illustrative Example

We will illustrate the use of the MPOC algorithm using data from an application for sorting stocks listed in the Athens Stock Exchange ([13]), namely 20 alternatives from the commercial sector, which were evaluated on 6 criteria (Table 3, where $g_t(a_i)$ indicates the performance of the i-th alternative according to the t-th criterion). The criteria names are not relevant here, therefore we will note only that all criteria are to be maximized, except $g_3(.)$, where the lower the values, the better.

To obtain an outranking relation we computed the credibility degrees of ELECTRE III ([25]) and ELECTRE TRI ([27], [31]), as defined in the variant proposed by Mousseau and Dias (2004). Alternatives are compared as pairs, and for each ordered pair (a_i, a_t), a credibility degree $\mathrm{S}(a_i, a_t)$ is computed, indicating the degree to which a_i outranks a_t, taking into account the weights of the criteria that are in concordance with the outranking assertion and the possible veto effect by criteria that are in discordance with that assertion (for detailed formulas see Mousseau and Dias, 2004). The outranking relation is considered to hold if $\mathrm{S}(a_i, a_t) \geq \lambda$. The cut-off point λ is defined by a decision maker, such that it ranges between 0.5 and 1. In this illustration we considered $\lambda = 0.6$.

Table 3: Evaluations on six criteria for 20 stocks.

	g_1	g_2	g_3	g_4	g_5	g_6		g_1	g_2	g_3	g_4	g_5	g_6
a_0	0,82	0,45	0,26	-4,7	-100	0,45	a_{10}	0,8	0,58	0,62	13,7	34,6	1,54
a_1	0,41	0,63	0,03	2,28	-20	2,04	a_{11}	1,23	0,37	0,64	8,97	45,9	0,96
a_2	0,57	0,2	0,1	6,08	-33,3	1,08	a_{12}	0,24	0,28	0,73	-1,75	0	0,72
a_3	0,24	0,02	0,08	2,41	-53,5	0,62	a_{13}	0,26	0,65	0,58	4,88	7,14	0,9
a_4	0	0,46	0,62	5,04	-76,5	3,02	a_{14}	1,1	0,76	0,54	0,29	0	0,73
a_5	0,93	0,02	0,14	2,82	6,38	0,72	a_{15}	1,79	0,55	0,73	5,88	-100	2,69
a_6	0,01	0,69	0,77	7,55	-40	3,23	a_{16}	1,02	1,06	0,82	5,5	6,38	0,73
a_7	0,86	0,86	0,86	4,28	3,71	0,57	a_{17}	1,32	1,12	0,94	12,06	-61	2,69
a_8	2,16	0,6	0,12	2,11	56,3	0,51	a_{18}	1,36	0,04	1,02	1,79	110	2,31
a_9	1,24	0,12	0,62	11,65	12,5	1,17	a_{19}	0,57	0,17	0,23	-11,5	0	0,52

We used the original values ([13]) for the indifference and preference thresholds (Table 4), but we will use veto thresholds that are not as tight as the original ones. This occurs, in this particular example, because the original values for the veto thresholds caused a high number of incomparability situations. Since the original data set ([13]) does not indicate any information about criteria importance, all criteria are considered to have the same weight, i.e., $k_i = 1/6, i \in \{1,...,6\}$.

Table 4: Indifference, preference and veto thresholds.

	g_1	g_2	g_3	g_4	g_5	g_6
q_j	0,05	0,05	0	0,1	8,72	0,05
p_j	0,25	0,2	0,2	0,5	10	0,25
v_j	20	10	10	100	180	2,75

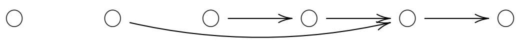
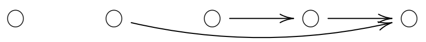
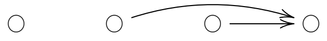
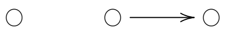
Evaluating the set of these 20 alternatives based on their performance, criteria thresholds and weights, leads to the credibility matrix presented in Table 5. For $\lambda = 0.6$, 121 (63.7%) pairs of alternatives have a relation of Preference, 8 (4.2%) have a relation of Indifference and 61 (32.1%) have a relation of Incomparability.

The results of the clustering procedure were computed considering $\alpha_P = \alpha_I = \alpha_R = 1$. Only for partitions with at least 12 classes we get a final quality Q*=100%. Considering that the decision maker is only interested in solutions with less than 7 classes

Table 5: Credibility degrees S(a_i, a_t), i,j=1,...,20.

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}	a_{20}
a_1	0	0.19	0.37	0.42	0.07	0.35	0	0.44	0.14	0.33	0.27	0.2	0.47	0.33	0.17	0.2	0.21	0.17	0	0.47
a_2	**0.83**	0	**0.74**	**0.99**	**0.67**	0.5	**0.62**	0.33	**0.61**	0.5	0.5	0.5	**0.83**	**0.67**	0.58	0.5	0.33	0.33	0.29	**0.74**
a_3	**0.67**	0.44	0	**0.98**	0.32	**0.67**	0.24	0.5	0.5	0.47	0.18	0.37	**0.8**	**0.67**	0.5	0.46	0.5	0.33	0.22	**0.83**
a_4	**0.67**	0.36	0.19	0	0.14	0.5	0.06	0.33	0.41	0.28	0.17	0.17	**0.63**	0.33	0.45	0.27	0.28	0.27	0.1	0.56
a_5	**0.67**	0.37	0.33	0.51	0	0.5	0.37	0.5	0.28	0.5	0.41	0.34	**0.61**	0.48	0.43	**0.62**	0.35	0.33	0	0.5
a_6	**0.83**	0.57	0.49	**0.95**	0.18	0	0.1	**0.67**	0.48	0.44	0.33	0.18	**0.83**	0.56	**0.73**	0.31	**0.63**	0.31	0.45	**0.89**
a_7	**0.67**	0.5	**0.67**	**0.68**	**0.88**	0.5	0	0.53	0.49	0.38	0.38	0.39	**0.65**	0.51	0.48	**0.8**	0.5	0.5	0.18	0.5
a_8	**0.83**	0.51	0.5	**0.83**	0.12	**0.73**	0.04	0	0.5	0.32	0.33	0.17	**0.81**	0.5	0.58	0.25	0.45	0.25	0.4	**0.83**
a_9	**1**	0.49	**0.65**	**0.83**	0.1	**0.7**	0.01	**0.66**	0	**0.67**	**0.67**	**0.67**	**0.87**	**0.67**	**0.74**	0.23	0.53	0.23	0.38	**1**
a_{10}	**0.67**	0.5	**0.8**	**0.83**	0.36	**0.83**	0.28	**0.83**	0.33	0	0.33	**0.67**	**0.88**	**0.8**	**0.77**	0.49	**0.83**	0.49	0.49	**0.83**
a_{11}	**0.83**	**0.67**	**0.83**	**0.83**	0.51	**0.77**	0.42	**0.83**	0.5	**0.83**	0	**0.67**	**1**	**0.94**	**0.62**	**0.64**	**0.69**	0.5	0.5	**0.83**
a_{12}	**0.8**	0.5	**0.78**	**0.83**	0.28	**0.83**	0.19	**0.83**	0.33	**0.68**	0.48	0	**1**	**0.78**	**0.75**	0.41	**0.83**	0.41	0.56	**0.83**
a_{13}	0.53	0.23	0.33	**0.67**	0.18	0.5	0.1	0.5	0.17	0.24	0.08	0.22	0	0.43	0.34	0.31	0.5	0.31	0.33	**0.67**
a_{14}	**0.67**	0.58	0.39	**0.83**	0.25	**0.67**	0.17	**0.67**	0.5	0.5	0.33	0.49	**1**	0	**0.73**	0.38	0.5	0.33	0.45	**0.67**
a_{15}	**0.83**	0.5	0.5	**0.67**	0.18	**0.67**	0.1	**0.78**	0.33	0.43	0.5	0.45	**1**	**0.73**	0	0.32	**0.67**	0.32	0.33	**0.83**
a_{16}	**0.83**	0.59	**0.63**	**0.67**	0.58	0.43	0.4	0.45	0.14	0.4	0.27	0.2	0.47	0.43	0.47	0	0.43	0.5	0	0.47
a_{17}	**0.83**	0.58	0.5	**0.83**	0.18	**0.83**	0.1	**1**	0.5	0.36	0.33	0.23	**0.93**	**0.73**	**0.81**	0.32	0	0.32	0.45	**0.83**
a_{18}	**0.83**	**0.67**	**0.67**	**0.83**	**0.67**	**0.66**	0.53	**0.68**	0.37	**0.63**	0.5	0.43	**0.67**	**0.66**	**0.67**	**0.67**	**0.66**	0	0.05	**0.67**
a_{19}	**0.67**	0.5	0.54	**0.67**	0.33	**0.67**	0.33	0.53	0.41	**0.63**	0.5	0.5	**0.67**	0.5	**0.67**	0.17	0.5	0.43	0	**0.74**
a_{20}	0.5	0.33	0.56	**0.67**	0.1	0.47	0.02	0.5	0.24	0.33	0.18	0.17	**0.64**	0.5	0.37	0.23	0.37	0.23	0.33	0

Table 6: Partitions Quality with $SS\tau$-Consistency

k	Transitivity	Quality (%)	Inconsistences vector	Structure
6	√	97.89	(0 0 0.0211)	
5	√	97.37	(0 0.0053 0.0211)	
4	√	96.32	(0 0.0053 0.0316)	
3	√	94.21	(0 0.0053 0.0526)	
2	√	88.95	(0 0.0053 0.1053)	
1	√	67.89	(0 0 0.3211)	

(which is reasonable taking into account that $|\mathscr{A}|$ =20), the results obtained by applying Algorithm 2, are shown in Table 6. In this example, all partitions are transitive.

The decision maker may need some help in order to determine the number of classes. For this purpose we determine the inconsistencies vector for different values of k (see Table 6) and the corresponding quality. From the Figure 4 we can see the gain in quality that is achieved by increasing the number of clusters from k=2 to k=13. Above k=4, the gain of quality is not substantial. Therefore, if the decision maker chooses k=4 classes as a good solution, we obtain a partition with a quality of 96.32% resulting from 0% of violations of the Preference conditions, 0.53% of Indifference and 3.16% of Incomparability, with

- $C^1 = \{a_1, a_2, a_4, a_7, a_{19}\}$,
- $C^2 = \{a_3, a_6, a_8, a_{13}, a_{14}, a_{20}\}$,
- $C^3 = \{a_5, a_{18}\}$, and
- $C^4 = \{a_9, a_{10}, a_{11}, a_{12}, a_{15}, a_{16}, a_{17}\}$.

The application of a K-means algorithm leads to a partition without any relation between the clusters. On the contrary, our method provides not only clusters but also a

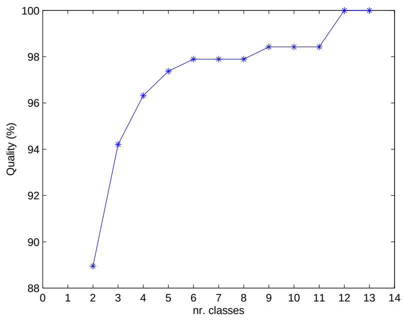


Figure 4: Partition quality as a function of the number of classes (Example 1).

partial order on these clusters. As a comparison, let us consider the strategy suggested in Cailloux et al. (2007) and Rocha et al. (2012): first, the K-means algorithm is applied on the data; in a second step, a partial order is sought on the set of the categories obtained on the first stage. In the second step, the centroids of the k categories obtained on the first stage (ρ_t, t=1,...,k) are compared using an outranking relation (see §2). For each ordered pair of centroids (ρ_s,ρ_t), the method decides if the first one outranks the second one (ρ_s S ρ_t) or not. The preference structure of the partition is defined through the centroid relations $\{P, I, R\}$. Thus, a category C^s is preferred to C^t ($C^s \succ C^t$) if the centroid ρ_s of category C^s is preferred to centroid ρ_t of category C^t, they are indifferent ($C^s = C^t$) if their centroids are indifferent and they are incomparable ($C^s \perp C^t$) if their centroids are incomparable.

Applying the classical K-means clustering method on the data set constituted by the 20 evaluations and considering k=4, the *K-Means* algorithm yields four homogeneous groups - C^1,C^2,C^3 and C^4:

- $C^1 = \{a_1, a_2, a_3, a_4, a_6, a_{13}, a_{20}\}$,
- $C^2 = \{a_{10}, a_{11}, a_{12}, a_{19}\}$,
- $C^3 = \{a_5, a_7, a_{16}, a_{18}\}$, and
- $C^4 = \{a_8, a_9, a_{14}, a_{15}, a_{17}\}$.

On the second stage, we will use the same indifference, preference and veto thresholds used in MPOC and indicated in Table 4, the same weights and threshold cut-off λ (0.6). For each of the obtained groups, we computed its centroid (ρ_1, ρ_2, ρ_3 and ρ_4), and then ranked the resulting centroids based on the outranking relation in Table 7, originating a structure of partially ordered categories (Figure 5).

After obtaining the partitions we analyzed the inconsistencies due to Indifference, Incomparability and Preference conditions. The results are presented on Table 8 as well as the results obtained with the MPOC algorithm. As we can see in Table 8, there exist many more inconsistencies when applying the classical K-means clustering method in this particular example.

Table 7: Credibility degree between centroids.

centroid	ρ_1	ρ_2	ρ_3	ρ_4
ρ_1	-	0.3333	0.2884	0.3333
ρ_2	0.8333	-	0.5348	0.7158
ρ_3	0.6667	0.3522	-	0.4814
ρ_4	0.7158	0.4767	0.2120	-

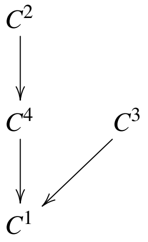


Figure 5: Structure obtained with K-means and sorting centroids.

Condition	MPOC	K-means
Preference	0 (0.0%)	8 (4.2%)
Indifference	1(0.53%)	0 (0.0%)
Incomparability	6 (3.2%)	7(3.7%)
Total	7 (3.68%)	15(7.9%)
$Q(\mathscr{P})$	96.32%	92.10%

Table 8: Inconsistencies obtained for k=4.

7 Conclusions

In this work, we proposed an approach for the classification of a set A of alternatives, based on multiple criteria, to a set of partially ordered classes with an unknown structure a priori. Formally, clustering with partially ordered classes consists in finding a partition of A where the partial order relation is defined by the classes: one class can be better or worse compared with other classes, but can also be incomparable to other classes. This is an innovative feature in a literature where the prevailing problems addressed are those of (complete) ordered clustering and clustering without preferences among classes.

This paper presents several ideas and contributions. One of the proposals is to define preference relations between classes on the basis of weak majorities of pairs supporting that a class should be seen as at least as good as another class. This provided consistently better results that considering another natural option, which would be to say a class outranks another class when the centroid of the former class outranks the centroid of the latter one (results for this modelling option are not presented in this paper).

Another contribution is the discussion of what principles should be pursued when judging the quality of candidate partitions. The Strong Consistency principle is appealing: it intends to respect the outranking relation concerning both directions (a_i, a_j) and (a_j, a_i) for each pair of alternatives. However, when S is intransitive it is never possible to fully comply with it. The Semi Strong Consistency principle is also appealing and easy to understand: more modestly, it intends to avoid that the outranking relation is respected in at least one of the directions (a_i, a_j) or (a_j, a_i) for each pair of alternatives. S-consistency and τ-consistency are not recommended: S-consistency does not pose any constraints to placing two alternatives in the same class, and that τ-consistency does not pose any constraints to placing two alternatives in two incomparable classes. Therefore it is trivial to find a (poor) solution.

To obtain candidate partitions, this paper proposes an adaptation of the agglomerative hierarchical clustering algorithm. The notion of similarity between two classes is replaced by an assessment of how the quality improves when the classes are merged. Successive iterations yield possible partitions with a decreasing number of classes, which avoids specifying at the outset the number of classes sought. These candidate partitions can be evaluated by a decision maker based on the quality of the partition, the respect of transitivity, and the way the partition matches her intuition. The quality of a partition is defined by the ratio of pairs of alternatives that do not check the conditions of Preference, Indifference and Incomparability. However, the decision maker might not want simply to minimize the number of total violations. Indeed, the decision maker may reasonably decide that some violations are more serious than others by placing more weight on those violations. The paper also discusses what might be done to adjust a posteriori an intransitive partition to become a transitive one.

An acknowledged limitation of the MPOC approach is that does not guarantee an "optimal" partition minimizing the violations, and so it can be regarded as a heuristic approach (this is also true for iterative clustering methods in general, such as K-means or hierarchical methods). Related with this topic, one field for future research would be the development of optimization approaches to match the preferences of a decision maker. The comparisons with other ways for multicriteria clustering is another field for future research.

References

[1] Brans, J.P, Vincke, Ph (1985). A preference ranking organization method. Management Science 31(6):647-656.

[2] Cailloux, O., Lamboray, Cl. and Nemery, Ph. 2007. A taxonomy of clustering procedures. Proceedings of the 66th Meeting of the EWG on MCDA, Marrakech, Maroc.

[3] De Smet, Y., Montano, G.L. (2004). Towards multicriteria clustering: an extension of the K-means algorithm. European Journal of Operational Research 158(2) : 390-398.

[4] De Smet Y., Eppe S. (2009) Multicriteria Relational Clustering: The case of binary outranking matrices. Evolutionary multi-criterion optimization Lecture Notes in Computer Science, volume 5467/2009, 380-392, DOI: 10.1007/ 978-3-642-01020-0-31.

[5] De Smet Y., Nemery Ph., Selvaraj R. (2012). An exact algorithm for the multicriteria ordered clustering problem, Omega.

[6] Doumpos, M., Zopounidis, C. (2002). Multicriteria Decision Aid Classification Methods, Kluwer Academic Publishers, Dordrecht.

[7] Ferligoj, A. (1997). Recent developments in cluster analysis,ISSN 1330-1012, 19th International Conference on INFORMATION TECHNOLOGY INTERFACES ITI '97, pp 253.

[8] Fernandez E., Navarro J., Bernal S. (2009). Multicriteria sorting using a valued indifference relation under a preference-disaggregation paradigm. European Journal of Operational Research 198:602-609.

[9] Fernandez E., Navarro J., Bernal S. (2010). Handling multicriteria preferences in cluster analysis. European Journal of Operational Research 202:819-827.

[10] Figueira, J., De Smet, Y., Brans, JP (2004). Promethee for MCDA Classification and Sorting problems: Promethee TRI and Promethee CLUSTER. Technical Report TR/SMG/2004-02. Universite Libre de Bruxelles.

[11] Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2, 139-172.

[12] Guha, S., Rastogi, R., and Shim, K. (1998). CURE: An efficient clustering algorithm for large databases. In Proceedings of the ACM SIGMOD Conference, 73-84, Seattle, WA.

[13] Hurson,Ch., Zopounidis, C.(1997). On the use of multicriteria decision aid methods to portfolio selection, in Clmaco, J.(ed), Multi-Criteria Analysis, Springer, 496-507.

[14] Jain, A.K., Dubes, R.C. (1988). Algorithms for Clustering Data. Prentice-Hall advanced reference series. Prentice-Hall,Inc., Upper Saddle River, NJ.

[15] Karypis, G., Han, E.H., Kumar, V. (1999). CHAMELEON: A hierarchical clustering algorithm using dynamic modeling, COMPUTER, 32, 68-75.

[16] Kaufman, L., Rousseeuw, P. (1990). Finding Groups in Data : An introduction to Cluster Analysis. John Wiley and Sons, New York.

[17] Keeney, R. L., Raiffa, H. (1993). Decisions with multiple objectives: preferences and value tradeoff, Cambridge University Press, Cambridge.

[18] Knuth, D. (1997). The Art of Computer Programming, Volume 1: Fundamental Algorithms, Third Edition. Addison-Wesley. ISBN 0-201-89683-4. Section 2.2.1: Stacks, Queues, and Deques, pp. 238243.

[19] Mousseau, V., Dias, L. (2004). Valued outranking relations in ELECTRE providing manageable disaggregation procedures. *European Journal of Operational Research*,156(2),467-482.

[20] Nemery Ph. (2008). An outranking-based sorting method for partially ordered categories, DIMACS, Workshop and Meeting of the COST Action ICO602, Paris, Universit Paris Dauphine, 28-31 October.

[21] Nemery Ph. (2008). A multicriteria sorting method for partially ordered categories, Proceedings of the doctoral work shop of EUROMOT 2008 - The Third European Conference on Management of Technology, ”Industry-University Collaborations in Techno Parks”, Nice.

[22] Nemery, P., De Smet, Y.(2005). Multicriteria Ordered Clustering, Technical Report TR/SMG/2005-003, SMG, Universit Libre de Bruxelles, 2005.

[23] Rocha, C., Dias, L.(2008). An Algorithm for Ordinal Sorting Based on ELECTRE with Categories Defined by Examples. Journal of Global Optimization, Vol. 42, No. 2, 255-277.

[24] Rocha, C., Dias, L., Dimas, I. (2012). Multicriteria classification with unknown categories: a clustering-sorting approach and an application to conflict management. Journal of Multi Criteria Decision Analysis. (To apear)

[25] Roy B. (1978). ELECTRE III: Un algorithme de classements fondé sur une représentation floue des préférences de critères multiples. *Cahiers du* CERO, 20(1):3-24.

[26] Roy B.(1991). The outranking approach and the foundations of ELECTRE methods. Theory and Decision 31, 49-73.

[27] Roy B., Bouyssou D. (1993). Aide multicritère à la décision : Méthodes et cas. Economica, Paris.

[28] Sibson, R. (1973). SLINK: An optimally efficient algorithm for the single link cluster method. Computer Journal, 16, 30-34.

[29] Tannock, R. (1998). Attention Deficit Hyperactivity Disorder: Advances in cognitive, neurobiological, and genetic research. Journal of the Child Psychology and Psychiatry, 39, 65-99.

[30] Vincke, P.(1992). Multicriteria Decision-Aid. New York: J.Wiley.

[31] Yu, W.(1992), Aide multicritère la décision dans le cadre de la problématique du tri: concepts, méthodes et applications. Doctoral dissertation, Université Paris-Dauphine.