

Instituto de Engenharia de Sistemas e Computadores de Coimbra
Institute of Systems Engineering and Computers
INESC - Coimbra

Maria Clara Rocha
Luis C. Dias

**POC - Partially Ordered Clustering:
an agglomerative algorithm**

No. 7

2011

ISSN: 1645-2631

Instituto de Engenharia de Sistemas e Computadores de Coimbra
INESC - Coimbra
Rua Antero de Quental, 199; 3000-033 Coimbra; Portugal
www.inescc.pt

POC - Partially Ordered Clustering: an agglomerative algorithm

Maria 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 categories 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. Recently, some multicriteria clustering procedures have been proposed aiming to discover data structures with totally ordered categories from a multicriteria perspective. Here, we propose an agglomerative clustering method based on a valued outranking relation. We suggest a method for regrouping alternatives into partially ordered classes. The model is based on the quality of partition that reflects the percentage of pairs of alternatives that are compatible with a decision-maker's preferences.

Keywords: *Multi-criteria decision aiding (MCDA), Sorting problem, Clustering, ELECTRE, Agglomerative algorithm*

1 Introduction

The general definition of classification is the assignment of a finite set of alternatives (actions, objects, projects,...) into predefined ordered 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 first two terms are commonly used by statisticians as well as by scientists of the artificial intelligence field (neural networks, machine learning, etc.). The term “sorting” has been established by MCDA researchers refer to problems where the groups are defined in a ordinal way (Doupoumpos and Zopounidis, 2002). The definition of the groups does not only provide a simple description of the alternatives, but it also incorporates additional preferential information, which could be of interest to the decision making context. For simplicity reasons, the general term “classification” is used and distinction between sorting and classification is made only when necessary.

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

^{*}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, email: 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, email: lmcdias@fe.uc.pt.

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 (Doumpos and Zopounidis, 2002).

Recently, some multicriteria clustering procedures have been proposed aiming to discover data structures from a multicriteria perspective (De Smet and Montano, 2004; Figueira, De Smet and Brans, 2004; Nemery and De Smet, 2005; Fernandez et al., 2010). Such works aim to not only detect groups with similar actions, but detecting also the preference relations between groups found. It is the definition of similarity that makes these methods original, which is based on binary relations of preference between actions. In this work, we used the preferences structure defined by the decision maker to quantify the similarity measure. We propose a new measure of similarity: two groups are the more similar, the higher the quality of the partition gained from joining these groups, i.e the lower the number of violations of the relations of Preference, and Indifference Incomparability between actions classified. To evaluate the quality of the partition a principle will be used which prohibits the existence of incomparable alternatives within the same class, indifferent alternatives belonging to incomparable classes and alternatives strictly preferred to other ones belonging to worse classes.

The work developed to address the *multicriteria clustering* problematic has mainly focused, so far, on the assignment of alternatives to totally ordered categories. However, there is a variety of real world problems where many alternatives are not comparable. For example, on a medical diagnostic the patient can be better or worse regarding another patient and in agreement with his symptoms, however, they can have incomparable diseases. These cases can lead to a partial order of the categories: one category can be better or worse than other categories, but can also be incomparable to other categories.

Specifically, we are interested in multicriteria clustering problems involving preferences from a decision maker to sort a set of alternatives considering multicriteria models with *partial order structure* and not necessarily totally ordered. Our proposed method is based on valued outranking relations and an agglomerative hierarchical clustering method.

This paper is organized as follows. Section 2 briefly describes the agglomerative hierarchical clustering algorithm. In the next section will introduce the basic notation and definitions about preference structures. Section 4 introduces the notation that will be used and presents definitions of support for a preference structure. Transitivity of a partition is addressed in Section 5 and assignment rules of alternatives are studied in Section 6. Definitions of support for measures to merged classes are presented in Section 7. Section 8 presents the proposed extension of agglomerative clustering method, which is illustrated using two examples in the next section. Some concluding remarks are presented in Section 10.

2 Agglomerative hierarchical clustering algorithm

Traditionally, cluster analysis algorithms can be classified as *hierarchical* (which require the elaboration of a tree hierarchy) or as *non-hierarchical*(or partitional) (which do not require the elaboration of a tree, assigning alternatives to clusters after the number of groups to be formed is specified). In this work, we will use an hierarchical algorithm.

Hierarchical algorithms can still be divided into *agglomerative* and *divisive* (Jain and Dubes, 1988; Kaufman and Rousseeuw, 1990). Agglomerative clustering starts with all clusters with a single alternative and recursively merges two or more clusters. Divisive clustering starts with a single cluster with all the alternatives and then recursively partitions the clusters until a stopping criterion is met

(frequently when the number of clusters becomes the target that had been fixed a priori). In general, for hierarchical methods, groups are represented by a bi-dimensional diagram named dendrogram or tree diagram. In this diagram, each branch represents an alternative, while the root represents the group of all alternatives. Several hierarchical algorithms have been developed such as SLINK (Sibson,1973), COBWEB (Fisher,1987), CURE (Ghua et al.,1998) and CHAMELEON (Karypis et al., 1999). In this work, we will use Agglomerative hierarchical clustering.

In general, the agglomerative hierarchical methods using a standard algorithm, as described in Algorithm 1.

Algorithm 1 - Agglomerative hierarchical clustering scheme

s=0 (stage)

1. Initial clustering with n groups containing an element in each group and a similarity matrix D between all pairs of groups;
 2. **While** there exist at least two groups **do**
 - determine the two groups C^i and C^j such that similarity is maximum;
 - merge these groups C^i and C^j to form a new group $C^r = C^i \cup C^j$;
 - determine the similarities between C^r and the remaining groups
 - $s \leftarrow s + 1$
- end while.**
-

The difference between the methods occurs in the definition of similarity between groups which is defined according to each method. Conventional clustering measures similarity based on geometric distances or related metrics. The importance of a preference closeness measure with concern to preference similarity oriented problems, i.e, the importance of incorporating decision-maker's preferences in multicriteria cluster analysis was firstly pointed-out by De Smet and Montano Guzman (2004). Their basic idea is that all objects inside the same cluster are similar in the sense that they are preferred, indifferent and incomparable to more or less the same objects. Other proposals concerning multicriteria preference clustering are found in Figueira et al. (2004), Nemery and De Smet(2005), Fernandez et al.(2010).

In this paper, we present an extension of the agglomerative hierarchical algorithm to the multicriteria framework. We are interested in obtaining a structure that can be partially ordered and not necessarily totally ordered.

3 Preference structures

A comparison of the alternatives is the main component in any decision problem, which can be made between existing alternatives or between standards and fictitious alternatives.

According to Roy and Bouyssou (1993), a model of preferences considers the following relations: Preference (P), Indifference (I) and Incomparability (R), resulting from the comparison between two alternatives a_i and a_j . Such relations satisfy the following conditions:

$$\forall a_i, a_j \in A \begin{cases} a_i P a_j \implies a_j \not P a_j & : P \text{ is asymmetric} \\ a_i P a_j \wedge a_j P a_k \implies a_i P a_k & : P \text{ is transitive} \\ a_i I a_i & : I \text{ is reflexive} \\ a_i I a_j \implies a_j I a_i & : I \text{ is symmetric} \\ a_i I a_j \wedge a_j I a_k \implies a_i I a_k & : I \text{ is transitive} \\ a_i \not R a_i & : R \text{ is not reflexive} \\ a_i R a_j \implies a_j R a_i & : R \text{ is symmetric} \end{cases} \quad (1)$$

Definition 3.1. (Vincke, 1992) *The relations $\{P, I, R\}$ constitute a preference structure of \mathcal{A} if it satisfies the condition (1) and if, given any two alternatives a_i, a_j of \mathcal{A} , only one of the following properties holds: $a_i P a_j, a_j P a_i, a_i I a_j$ or $a_i R a_j$.*

The application of multicriteria tools, including outranking methods (Electre family of methods (Roy, 1991) and Promethee (Brans and Vincke, 1985) are typical examples), and the concepts of modeling preferences define relations $\{P, I, R\}$ between any two alternatives of \mathcal{A} . Note that the outranking methods define the relation of incomparability R . When we apply other methods, AHP (Saaty, 1980,1996) or Utility Theory (Keeney and Raiffa,1993) for example, the relation R remains empty and the comparison between pairs of alternatives is restricted to relations P and I .

Many of the outranking methods, as the name suggests, are based on the *outranking relation* between pairs of alternatives (a_i, a_j) , which corresponds to meeting the Preference and Indifference relations. Be given the outranking relation S . For all pairs of alternatives $(a_i, a_j) \in A$, $a_i S a_j$ means that “ a_i is at least as good as a_j ”, i.e., $a_i P a_j$ or $a_i I a_j$.

There are four different situations that can result when comparing two alternatives a_i and a_j :

$$\begin{aligned} a_i S a_j \wedge a_j \not S a_i &\iff a_i P a_j \text{ (} a_i \text{ is preferable to } a_j \text{);} \\ a_j S a_i \wedge a_i \not S a_j &\iff a_j P a_i \text{ (} a_j \text{ is preferable to } a_i \text{);} \\ a_i S a_j \wedge a_j S a_i &\iff a_i I a_j \text{ (} a_i \text{ is indifferent to } a_j \text{);} \\ a_i \not S a_j \wedge a_j \not S a_i &\iff a_i R a_j \text{ (} a_i \text{ is incomparable to } a_j \text{).} \end{aligned} \quad (2)$$

The outranking relation used by methods such as ELECTRE III (Roy, 1978) and ELECTRE TRI (Roy, 1993; Yu, 1992), is here defined as 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 $S(a_i, a_t)$ is computed, indicating the degree to which a_i outranks a_t . To do this, one may start by computing of single-criterion concordance indices $c_j(a_i, a_t)$ and single-criterion discordance indices $d_j(a_i, a_t)$. Let $\Delta_j(a_i, a_t)$ denote the advantage of a_i over a_t :

$$\Delta_j(a_i, a_t) = \begin{cases} g_j(a_i) - g_j(a_t) & \text{if } g_j \text{ is to maximize} \\ g_j(a_t) - g_j(a_i) & \text{if } g_j \text{ is to minimize} \end{cases} \quad (3)$$

$c_j(a_i, a_t)$ can be written as follows:

$$c_j(a_i, a_t) = \begin{cases} 1 & \text{if } \Delta_j \geq -q_j \\ 0 & \text{if } \Delta_j < -p_j \\ \frac{p_j + \Delta_j}{p_j - q_j} & \text{otherwise} \end{cases} \quad (4)$$

The concordance is maximum, when the alternative a_i is better than a_t or worse but for a small difference (up to q_j). It is minimum, when this difference becomes less than p_j . Finally, when a_i is worse than a_t , the concordance begins to decrease gradually between 1 and 0, as the difference in favor of a_t is becoming increasingly greater than q_j and closer to p_j .

A global concordance index $c(a_i, a_t)$ is computed by aggregating the n single-criterion concordance indices obtained before. It represents the level of majority among the criteria in favor of the conclusion that a_i outranks a_t . The computation $c(a_i, a_t)$ takes into account a vector of criteria weights. Each of these weights k_s ($s = 1, \dots, n$) can be interpreted as the voting power of the respective criterion. $c(a_i, a_t)$ can be written as follows:

$$C(a_i, a_t) = \frac{\sum_{s=1}^n k_s c_s(a_i, a_t)}{\sum_{s=1}^n k_s} \quad (5)$$

Moreover, the single-criterion discordance index $d_j(a_i, a_t)$ indicates the degree to which the j -th criterion ($j = 1, \dots, n$) disagrees with the conclusion that a_i outranks a_t . This index is computed taking into account the difference of performances on the criterion considered, as well as two thresholds: discordance u_j and veto v_j ($p_j \leq u_j \leq v_j$) (Mousseau and Dias, 2004).

$$d_j(a_i, a_t) = \begin{cases} 0 & \text{if } -\Delta_j \leq u_j \\ \frac{-\Delta_j - u_j}{v_j - u_j} & \text{if } u_j < -\Delta_j \leq v_j \\ 1 & \text{if } -\Delta_j > v_j \end{cases} \quad (6)$$

The computation of the discordance index requires the specification of an additional parameter, the veto threshold v_j . Conceptually, the veto threshold represents the smallest difference between the performance of an alternative a_i and the performance of an alternative a_t on criterion g_j , above which the criterion vetoes the outranking character of the alternative a_i over the a_t .

Once the concordance and discordance indices are set as described above, the next stage of the process is to combine the two indices to compute an overall outranking degree of an alternative a_i over an alternative a_t . One possibility proposed by (Mousseau and Dias, 2004) is:

$$S(a_i, a_t) = C(a_i, a_t) \cdot [1 - d^{\max}(a_i, a_t)]. \quad (7)$$

with

$$d^{\max}(a_i, a_t) = \max_{j \in \{1, \dots, n\}} d_j(a_i, a_t). \quad (8)$$

The credibility index provides the means to decide whether a_i outranks a_t ($a_i S a_t$) or not. The outranking relation is considered to hold if $S(a_i, a_t) \geq \lambda$. The cut-off point λ is defined by a decision maker, such that it ranges between 0.5 and 1.

4 Preference Structure proposed

Let $\mathcal{A} = \{a_1, \dots, a_m\}$ denote a set of alternatives represented by a vector of evaluations on n criteria. Let $\mathcal{G} = \{g_1(\cdot), \dots, g_n(\cdot)\}$ denote the set of criteria functions, such that $g_t(a_i)$ indicates the evaluation (performance) of the i -th alternative according to the t -th criterion. A partition of \mathcal{A} in k categories $\mathcal{P} = \{C^1, C^2, \dots, C^k\}$ is defined as follows:

- $C^i \neq \emptyset, i = 1, \dots, k$
- $\mathcal{A} = \cup_{i=1, \dots, k} C^i$
- $C^i \cap C^j = \emptyset, i \neq j$.

Formally, a *classification with partially ordered categories* is defined among the categories: a category can be higher or lower ranked in comparison with some categories, whereas it can be considered incomparable with other different categories.

The structure of preferences of the partition obtained in each stage of Algorithm 1 is (\succ, \perp, \approx) where " $C^i \succ C^j$ " denotes that class C^i is better than class C^j , " $C^i \perp C^j$ " denotes that class C^i is incomparable to C^j and " $C^i \approx C^j$ " denotes those classes C^i and C^j are indifferent.

The key idea to build the preference structure in each stage of Algorithm 1, is to evaluate the outranking relation τ between two classes. Given any two classes, $C^i, C^j \in \mathcal{P}$, let $C^i \tau C^j$ denote "the class C^i is at least as good as class C^j ", i.e, $C^i \succ C^j$ or $C^i \approx C^j$.

There are four different situations that can result when comparing two classes C^i and C^j :

$$\left\{ \begin{array}{l} C^j \tau C^i \wedge C^i \not\tau C^j \iff C^j \succ C^i \text{ (} C^j \text{ is better than } C^i \text{)} \\ C^i \tau C^j \wedge C^j \not\tau C^i \iff C^i \succ C^j \text{ (} C^i \text{ is better than } C^j \text{)} \\ C^i \tau C^j \wedge C^j \tau C^i \iff C^i \approx C^j \text{ (} C^i \text{ is indifferent to } C^j \text{)} \\ C^i \not\tau C^j \wedge C^j \not\tau C^i \iff C^i \perp C^j \text{ (} C^i \text{ is incomparable to } C^j \text{)} \end{array} \right. \quad (9)$$

The outranking relation between classes is defined as follows : classes are compared as pairs, and for each pair (C^i, C^j) , a credibility degree s_{ij} (definition 4.1) is computed indicating the degree to which C^i outranks C^j . The outranking relation is considered to hold if the outrank degree is at least 50% (definition 4.2).

Definition 4.1. Let $C^i, C^j \in \mathcal{P}_s$, with n_i and n_j the number of alternatives of C^i and C^j respectively.

The *outranking degree* of C^i on C^j is

$$s_{ij} = \frac{\sum_{a_s \in C^i} \sum_{a_t \in C^j} s(a_s, a_t)}{n_i \times n_j}, \text{ with } s(a_s, a_t) = \begin{cases} 1 & \text{if } a_s S a_t \\ 0 & \text{otherwise} \end{cases}.$$

The outranking degree s_{ij} between pair of classes (C^i, C^j) indicates the proportion of pairs of alternatives $(a_s, a_t) \in (C^i, C^j)$ that indicate a_s outranks a_t .

Definition 4.2. Let $(C^i, C^j) \in \mathcal{P}_s \times \mathcal{P}_s$. We say that C^i **outranks** C^j ($C^i \tau C^j$) iff the outranking degree of C^i on C^j is at least 0.5, i.e, $s_{ij} \geq 0.5$.

Example 4.1. Let $\mathcal{P} = \{C^1, C^2, C^3, C^4, C^5\}$ be a partition whose outranking degrees are presented in Table1.

	C^1	C^2	C^3	C^4	C^5
C^1	0	0.3333	0.2000	0.1333	0.1000
C^2	0.3333	0	0.6	0	0.1250
C^3	0.4000	0.5833	0	0.3333	0.3750
C^4	0.4667	0.4222	0	0	0.5833
C^5	0.4000	0.9167	0	0	0

Table 1: Outrank degree s_{ij} of C^i on C^j , $i, j=1, \dots, 5$

The resulting preference structure is shown in Figure1.

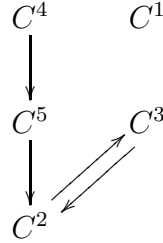


Figure 1: Final Partition

The purpose of this model is to obtain a final partition such that the structure of preferences does not have the relation “ \approx ”, i.e., there are no indifferent classes. This is achieved by a re-evaluation of the partition obtained, as shown in Section 5. A *final partially ordered partition*, with a structure of preferences (\succ, \perp) , can be defined as follows:

- $C^i \succ C^j \Rightarrow C^j \not\succeq C^i$: \succ is asymmetric
- $C^i \succ C^j \wedge C^j \succ C^k \Rightarrow C^i \succ C^k$: \succ is transitive
- $C^i \perp C^j \Rightarrow C^j \perp C^i$: \perp is symmetric
- $C^i \perp C^j \vee C^i \succ C^j \vee C^j \succ C^i$: \approx is empty

5 Transitivity of partition

An outranking relation enables modeling situations where transitivity does not hold. A well-known example is the one presented by Luce(1956)(see also Roy e Vincke, 1981): obviously we can tell the difference between a cup of coffee with αgr of sugar and a cup of coffee with $\alpha + 0.01gr$; therefore, there is a indifference relation between these two situations. Similarly, there is a indifference relation between $\alpha + 0.01gr$ and $\alpha + 0.02gr$ of sugar. If the indifference relation is transitive, then αgr e $\alpha + 0.02gr$ of sugar should be considered as indifferent. Following the same line of inference, it can deduced that there is no difference between a cup of coffee with αgr of sugar and a cup of coffee that is full of sugar, irrespective of α . Obviously, this is an incorrect conclusion, indicating that there are situations where the transitivity is not valid. However, to obtain a structure with a partial order relation, we need to have a transitive outranking relation.

Considering three alternatives a_i, a_j and a_k , the transitivity property is formally expressed as follows:

$$a_i \mathfrak{S} a_j \wedge a_j \mathfrak{S} a_k \implies a_i \mathfrak{S} a_k \quad (10)$$

For example, in a partition \mathcal{P}_s , since the outranking relation S is not transitive, by (10), the following situation can occur:

$$\exists \{C^i, C^j, C^k\} \in \mathcal{P}_s: C^i \tau C^j \wedge C^j \tau C^k \wedge C^i \not\tau C^k$$

Schematically:

$$C^i \longrightarrow C^j \longrightarrow C^k$$

where each arc $C^i \longrightarrow C^j$ represents the existence of the relation $C^i \tau C^j$.

To make S a transitive relation to three classes $\{C^i, C^j, C^k\}$, there are three solutions:

1. Require the outranking relation $C^i \tau C^k$: $C^i \xrightarrow{\quad} C^j \xrightarrow{\quad} C^k$
2. Eliminate the outranking relation $C^i \tau C^j$: $C^i \quad C^j \longrightarrow C^k$
3. Eliminate the outranking relation $C^j \tau C^k$: $C^i \longrightarrow C^j \quad C^k$

If the partition is transitive, we should evaluate the quality of the three possible partitions and the one that leads to better quality will be the final partition. If the partition is still intransitive, it should be resolved in the same way (recursive process).

If the outranking relation between classes has a cycle ($C^i \tau C^j \tau \dots \tau C^i$) then all partitions obtained by removing one of the relations of the cycle should be evaluated and choose the one that leads to better final quality. In particular, for indifferent classes ($C^i \tau C^j \tau C^i$) it must be added the case of the merging of classes to partitions obtained by removing one of the two relations of the cycle, and then

choose the best partition in terms of quality obtained.

Example

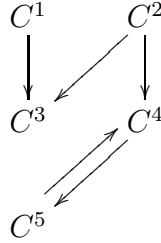


Figure 2: Preferred structure of \mathcal{P}

Consider the partition $\mathcal{P}=\{C^1, C^2, C^3, C^4, C^5\}$ and the preference structure presented in Figure 2.

In this structure there are two problems: the intransitivity of (C^2, C^4, C^5) and a indifference of pair (C^4, C^5) . To solve the problem, and starting with indifference, there are three possible solutions: S1, S2 and S3 (Figure 3).

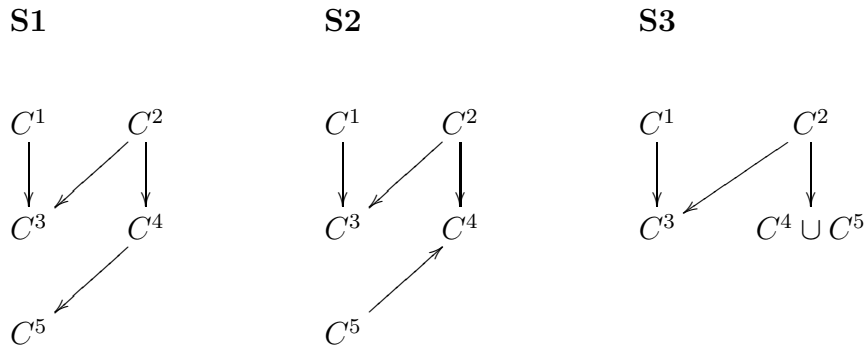


Figure 3: Possible solutions for intransitive partition \mathcal{P}_s

S2 and S3 are transitive partitions, but S1 does not verify the transitivity property. Now, applying the same rule to S1 obtain transitive partitions S4, S5 and S6 (Figure 4).

Finally, we should evaluate the quality of the seven possible transitive partitions and the one that leads to better quality will be the final partition.

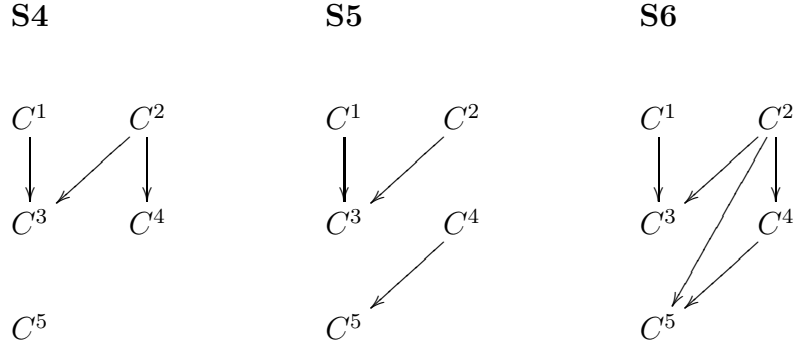


Figure 4: Possible solutions for intransitive partition S1

6 Principles for multicriteria clustering with partially ordered classes.

Let $C(a_i)$ denote a class to which alternative a_i is assigned to. 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.

6.1 Strong ($S\tau$)-Consistency

One idea is to base the assignment of the alternatives on a **strong consistency principle** ($S\tau$ - **Consistency**): “an alternative a_i outranks an alternative a_j if and only if the category of a_i is at least as good as the category of a_j ”, i.e.

$$a_i S a_j \Leftrightarrow C(a_i) \tau C(a_j) \quad (S\tau\text{-Consistency})$$

Although this requirement is not respected by ELECTRE TRI (Roy and Bouyssou, 1993), it is an appealing principle given that $a_i S a_j$ means “ a_i is at least as good as (or is not worse than) a_j ”.

The following corollaries result from $S\tau$ - Consistency:

- (1) $a_i P a_j$ if and only if a_i belongs to a class better than the class of a_j , i.e. $(a_i S a_j \wedge a_j \not S a_i) \Leftrightarrow C(a_i) \succ C(a_j)$.
- (2) $a_i I a_j$ if and only if a_i belongs to the same class as a_j , i.e. $(a_i S a_j \wedge a_j S a_i) \Leftrightarrow C(a_i) = C(a_j)$.
- (3) $a_i R a_j$ if and only if a_i belongs to a class incomparable to a_j , i.e. $(a_i \not S a_j \wedge a_j \not S a_i) \Leftrightarrow C(a_i) \perp C(a_j)$.

The second corollary stems from the fact that $a_i I a_j$ if and only if a_i and a_j outrank each other. The latter corollary stems from the fact that $a_i R a_j$ if and only if not $(C(a_i) \tau C(a_j))$ and not $(C(a_j) \tau C(a_i))$. An example that satisfies the $S\tau$ - Consistency conditions is depicted in Figure 5.

The only way it is possible to fully comply with these requirements is to have an S relation that is transitive. For instance, for the outranking relation in Figure 6 it is not possible to fully comply with $S\tau$ -Consistency: we would need to say that a_1 belongs to a class better than a_2 and that a_2 belongs to a class better than a_3 , but we could not conclude that the first class is better than the

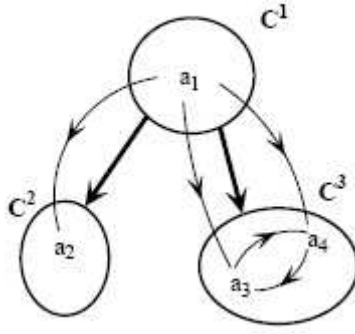


Figure 5: Illustration of $S\tau$ - Consistency

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.

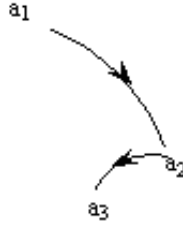


Figure 6: Impossible to comply with $S\tau$ - Consistency

6.2 Forms of $S\tau$ -Consistency relaxed

$S\tau$ -Consistency requires consistency from S to τ and vice-versa. Relaxing the $S\tau$ -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 $SS\tau$ -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 category of a_i must be at least as good as the category of a_j ”, i.e.

$$a_i S a_j \Rightarrow C(a_i) \tau C(a_j) \quad (S\text{-Consistency})$$

The following corollaries result from S-Consistency (proof omitted):

- (1) if $a_i P a_j$ then a_i must belong to a class at least as better than the class of a_j , i.e. $(a_i S a_j \wedge a_j \not S a_i) \Rightarrow C(a_i) \tau C(a_j)$.

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

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

The third corollary stems from the fact that if none of the alternatives outranks the other, then nothing can be concluded.

Given a partial order of classes, the S-Consistency principle can be rephrased as stating that “alternatives belonging to a given category cannot be outranked by any alternative belonging to a lower or incomparable category, and cannot outrank any alternative belonging to a higher or incomparable category”, i.e.,

$$(C(a_i) \not\prec C(a_j) \implies a_i \not S a_j) \iff [(C(a_j) \succ C(a_i) \vee C(a_i) \perp C(a_j)) \implies a_i \not S a_j]$$

As a consequence, we obtain the three following corollaries:

(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 must have $a_i R a_j$.

(6) if $C(a_i) = C(a_j)$ then we can have $a_i P a_j$, $a_i I a_j$, or even $a_i R a_j$.

In accordance with the principle S-Consistency, the following inconsistencies should not occur:

(1) $a_i P a_j \wedge (C(a_j) \succ C(a_i) \vee C(a_j) \perp C(a_i))$ (*S-Inconsistencies*)

(2) $a_i I a_j \wedge (C(a_i) \succ C(a_j) \vee C(a_j) \succ C(a_i) \vee C(a_j) \perp C(a_i))$

Note that placing all alternatives in the same class, by the last corollary, we obtain a trivial weak S-consistent solution (Figure 7). Hence, this relaxation of S-Consistency is not fully satisfactory, even though it allows for a greater freedom in finding a clustering result that does not violate too much the initial S relation.

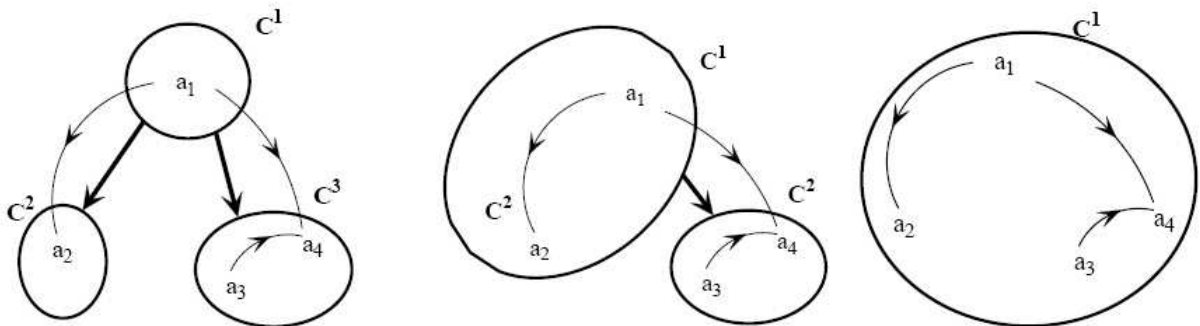


Figure 7: Illustration of S-Consistency: three solutions that satisfy the conditions

Note that, if there exists a cycle in the outranking relation $(a_i S a_j S a_k S \dots S a_i)$, then S-Consistency also implies that all the alternatives involved in the cycle must be placed in the same category. 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.

B. τ - Consistency

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

$$C(a_i) \tau C(a_j) \Rightarrow a_i S a_j \quad (\tau - \text{Consistency})$$

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 $a_i I a_j$.

The τ -Consistency principle can be rephrased as stating that “alternatives that do not outrank another alternatives, belong to a lower or incomparable category than that of the latter”, i.e.,

$$(a_i \not S a_j \Rightarrow C(a_i) \not C(a_j)) \iff [(a_i \not S a_j) \Rightarrow (C(a_j) \succ C(a_i) \vee C(a_i) \perp C(a_j))]$$

As a consequence, we obtain the three following corollaries:

- (4) if $a_i P a_j$ then a_i must belong to a class either incomparable or better than the class of a_j .
- (5) if $a_i I a_j$ then a_i and a_j can belong to any class.
- (6) if $a_i R a_j$ then a_i and a_j must belong to incomparable classes.

In accordance with the principle τ -Consistency, the following inconsistencies can occur:

- $a_i P a_j \wedge (C(a_j) \succ C(a_i) \vee C(a_j) = C(a_i))$ (τ -Inconsistencies)
- $a_i R a_j \wedge (C(a_j) = C(a_i) \vee C(a_i) \succ C(a_j) \vee C(a_j) \succ C(a_i))$

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 $S\tau$ -Consistency is not fully satisfactory, even though it allows for a greater freedom in finding a clustering result that does not violate too much the initial S relation.

C. Semi-Strong (SS τ) - Consistency

Table 2 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

		a_iPa_j	a_jPa_i	a_iIa_j	a_iRa_j
		a_iSa_j $a_j \not\mathcal{S}a_i$	$a_i \not\mathcal{S}a_j$ a_jSa_i	a_iSa_j a_jSa_i	$a_i \not\mathcal{S}a_j$ $a_j \not\mathcal{S}a_i$
$C(a_i) \succ C(a_j)$	$C(a_i) \tau C(a_j)$ $C(a_j) \not\tau C(a_i)$		$\nearrow \tau$ $\swarrow S$	 $\swarrow S$	$\nearrow \tau$
$C(a_i) = C(a_j)$	$C(a_i) \tau C(a_j)$ $C(a_j) \tau C(a_i)$		$\nearrow \tau$		$\nearrow \tau$ $\nearrow \tau$
$C(a_i) \perp C(a_j)$	$C(a_i) \not\tau C(a_j)$ $C(a_j) \not\tau C(a_i)$	$\swarrow S$		$\swarrow S$ $\swarrow S$	

Table 2: Combinations that are forbidden by S-consistency ($\swarrow S$) and combinations that are forbidden by τ -consistency ($\nearrow \tau$).

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 (S τ -Consistency). In accordance with the principle S τ -Consistency, the following inconsistencies should not occur:

- (1) $C(a_i) \succ C(a_j)$, but $a_i \not Pa_j$ (i.e., $a_i \not\mathcal{S}a_j$ or a_jSa_i).
- (2) $C(a_i) = C(a_j)$, but $a_i \not Ia_j$ (i.e., $a_i \not\mathcal{S}a_j$ or $a_j \not\mathcal{S}a_i$). (S τ -Inconsistencies)
- (3) $C(a_i) \perp C(a_j)$, but $a_i \not Ra_j$ (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-SStrong consistency, or (SS τ)-Consistency. According to this new principle, the following inconsistencies should not occur:

- (1) $C(a_i) \succ C(a_j)$ and a_jPa_i ($a_i \not\mathcal{S}a_j$ and a_jSa_i).
- (2) $C(a_i) = C(a_j)$ and a_iRa_j ($a_i \not\mathcal{S}a_j$ and $a_j \not\mathcal{S}a_i$). (SS τ Inconsistencies)
- (3) $C(a_i) \perp C(a_j)$ and a_iIa_j (a_iSa_j and a_jSa_i).

In Table 2 we can also see 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 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 this paper, we present an extension of the agglomerative hierarchical algorithm to the multicriteria framework, based on the (SS τ)-Consistency, which seems to be the best principle for grouping alternatives into partially ordered classes, given a possibly intransitive outranking relation S.

7 Measure of similarity between classes

We propose that the measure of similarity between classes is established for the quality of the partition obtained, that is, seeking to minimize the number of violations of the Preference, Indifference and Incomparability conditions of (SS τ)-Principle (Table 3).

Conditions	Condition violations
Preference: $C(a_i) \succ C(a_j) \Rightarrow a_j \not P a_i$	$C(a_i) \succ C(a_j) \wedge (a_j P a_i) \Leftrightarrow C(a_i) \succ C(a_j) \wedge (a_j S a_i \wedge a_i \not S a_j)$
Indifference: $C(a_i) \perp C(a_j) \Rightarrow a_i \not I a_j$	$C(a_i) \perp C(a_j) \wedge a_i I a_j \Leftrightarrow C(a_i) \perp C(a_j) \wedge (a_i S a_j \wedge a_j S a_i)$
Incomparability: $C(a_i) = C(a_j) \Rightarrow a_i \not R a_j$	$C(a_i) = C(a_j) \wedge a_i R a_j \Leftrightarrow C(a_i) = C(a_j) \wedge (a_i \not S a_j \wedge a_j \not S a_i)$

Table 3: Preference, Indifference and Incomparability violations.

Therefore, the final partition obtained should verify the following:

- alternatives strictly preferred to other ones do not belong to worse categories;
- incomparable alternatives belong to different categories;
- indifferent alternatives do not belong to incomparable categories.

This will be accounted for by three distinct indicators: v_R (proportion of pairs that violate the incomparability condition), v_I (proportion of pairs that violate the indifference condition) and v_P (proportion of pairs that violate the preference condition).

Definition 7.1. Let $\mathcal{P}=\{C^1, C^2, \dots, C^k\}$ be a partition of $\mathcal{A}=\{a_1, a_2, \dots, a_n\}$. Let $a_i, a_j \in \mathcal{A}$ and

$$R_{ij} = \begin{cases} 1 & \text{if } a_i R a_j \\ 0 & \text{otherwise} \end{cases} \quad \text{the value that indicates the existence of a incomparability relation between } a_i$$

and a_j . The v_R index is defined as follows:

$$v_R = \frac{\sum_{\forall C^t \in \mathcal{P}} \sum_{(i,j): a_i, a_j \in C^t, i < j} R_{ij}}{n(n-1)/2} \quad (11)$$

Definition 7.2. Let $\mathcal{P}=\{C^1, C^2, \dots, C^k\}$ be a partition of $\mathcal{A}=\{a_1, a_2, \dots, a_n\}$. Let $a_i \in C^p$ and $a_j \in C^q$ and $I_{ij} = \begin{cases} 1 & \text{if } a_i I a_j \\ 0 & \text{otherwise} \end{cases}$ the value that indicates the existence of an indifference relation between a_i and a_j . The index v_I is defined as follows:

$$v_I = \frac{\sum_{\forall C^p \perp C^q} \sum_{(i,j): a_i \in C^p, a_j \in C^q} I_{ij}}{n(n-1)/2} \quad (12)$$

Definition 7.3. Let $\mathcal{P}=\{C^1, C^2, \dots, C^k\}$ be a partition of $\mathcal{A}=\{a_1, a_2, \dots, a_n\}$. Let C^p and $C^q : C^q \succ C^p$ and let $a_i \in C^p$ and $a_j \in C^q$. Let $P_{ij} = \begin{cases} 1 & \text{if } a_i P a_j \\ 0 & \text{otherwise} \end{cases}$ the value that indicates the existence of a preference relation between a_i and a_j . The index v_P is defined as follows:

$$v_P = \frac{\sum_{C^q \succ C^p} \sum_{(i,j): a_i \in C^p, a_j \in C^q} P_{ij}}{n(n-1)/2} \quad (13)$$

Therefore, let $ID_{\mathcal{P}_s}$ be an *inconsistencies vector* associated to a partition \mathcal{P}_s that will contain the ratio of pairs of alternatives that are not compatible with preference, indifference and incomparability conditions (14).

$$ID_{\mathcal{P}_s} = (v_P, v_I, v_R) \quad (14)$$

Definition 7.4. Let α_P, α_I and α_R denote the weights respectively assigned by the decision maker to conditions of Preference (v_P), Indifference (v_I) and Incomparability (v_R). The Quality $Q(\mathcal{P}_s)$ of partition \mathcal{P}_s is defined as follows

$$Q(\mathcal{P}_s) = 1 - \alpha_P v_P - \alpha_I v_I - \alpha_R v_R \quad (15)$$

The higher the value of $Q(\mathcal{P}_s)$, better the partition \mathcal{P}_s . The hierarchical clustering algorithm needs to find, in each stage s of Algorithm 1, the partially ordered partition \mathcal{P}_s^* such that:

$$Q(\mathcal{P}_s^*) > Q(\mathcal{P}_t), \forall \mathcal{P}_t \text{ formed in stage } s$$

8 Agglomerative Method for Partially Ordered Clustering

Initially, like Algorithm 1, we form as many classes as there are alternatives of \mathcal{A} , where each class is formed by one alternative. Then the preference structure of the first partition is built. In each

1. $C^i = \{a_i\}$ with $a_i \in A, \forall i = 1, \dots, m$ (m initial classes)
2. Compute matrix S ($S_{ij} = S(a_i, a_j)$)
3. $s=0$ (stage)
4. $\mathcal{P}_s = \{C^1, \dots, C^m\}$

While there exist at least two classes in \mathcal{P}_s **do**

5. Determine the pair of neighbors $(C^q, C^p) \in \mathcal{P}_s^2$ such that $Q(\mathcal{P}_{s+1})$ is minimum when merge C^q and C^p to form a new class $C^r = C^q \cup C^p$
6. Merge C^q and C^p to form a new class $C^r = C^q \cup C^p$
7. Update preference structure of \mathcal{P}_{s+1}
8. $s=s+1$

end while

Check the partitions that have the number of clusters indicated by the decision-maker for the transitivity property, and make corrections.

stage s of the proposed method we choose the pair of classes that leads to the greatest improvement in the quality of the partition \mathcal{P}_{s+1} when merged, making the partition more compatible with the preferences given by the decision maker. Next, we update the structure of preferences of the new partition obtained and reiterate (Algorithm 2).

Note that in this proposed method, the decision maker is not forced 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 categories.

The aim of this paper is to obtain a *transitive partially ordered partition* of a given set of alternatives. Thus, to guarantee the transitivity of the final partition, we reassess the quality of the partitions that are not transitive and/or containing indifferent classes after taking the necessary corrective measures.

9 Illustrative Examples

We will illustrate the use of the Partially Ordered Clustering using two sets of data. In both cases, the cutting level is unchanged ($\lambda = 0.6$) and it is considered $\alpha_P = \alpha_I = \alpha_R = 1$.

9.1 First example

Let us consider in a first example the data from an application for sorting stocks listed in the Athens Stock Exchange (Hurson and Zopounidis, 1997), namely 20 alternatives from the commercial sector, which were evaluated on 6 criteria (Table 1). The criteria names are not relevant here, therefore we will note only that all criteria are to be maximized, except $g_3(\cdot)$, where the lower the values, the better.

	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

Table 4: Evaluations on six criteria for 20 stocks.

We will use the original values (Hurson and Zopounidis, 1997) for the indifference and preference thresholds (Table 3), 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 led to many veto situations, which would imply having in the results either a large number of violations or a large number of categories. We used $\lambda = 0.6$ and $u_j = p_j$.

	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

Table 5: Indifference, preference and veto thresholds.

Since the original data set (Hurson and Zopounidis, 1997) does not indicate any information about criteria importance, all criteria are considered to have the same weight, i.e., $k_i = 1/6, i \in \{1, \dots, 6\}$.

Evaluating the set of these 20 alternatives based on their performance, criteria thresholds and weights, of the 190 pairs of alternatives studied ($\frac{n(n-1)}{2}, n = 20$), 121 (63.7%) verified the relation of Preference, 8 (4.2%) verified the relation of Indifference and 61 (32.1%) verified the relation of Incomparability.

As we can see in Figure 8, only for partitions with at least 12 classes we get a final quality $Q^* = 100\%$, which, given the number of alternatives, may be not a satisfactory solution. Considering that the decision maker is only interested in solutions with less than 7 classes (which is reasonable taking into account that $|\mathcal{A}| = 20$), the results obtained by applying Algorithm 2, are shown in Table 6.

In this example, all partitions are transitive. If the decision maker chooses $k=5$ classes as a good solution (the quality does not improve significantly from 5 classes), we obtain a partition with a quality of 97.37% resulting from 0% of violations of the Preference conditions, 0.53% of Indifference and 1.58% of Incomparability, with

- $C^1 = \{a_1, a_2, a_4, a_7, a_{19}\}$,

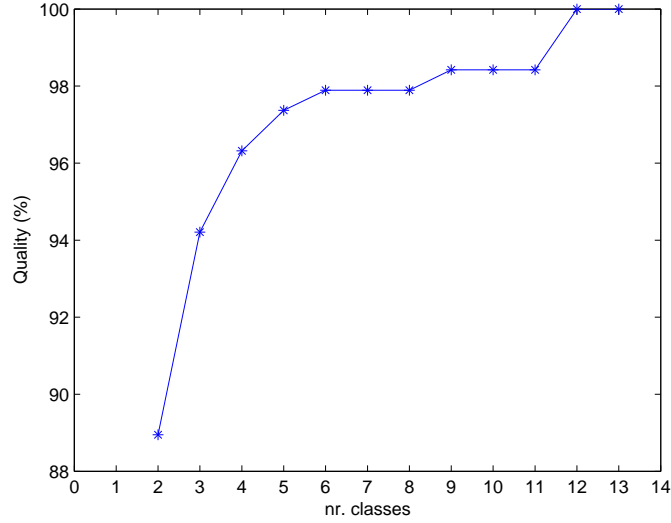


Figure 8: Partition quality as a function of the number of classes (Example 1).

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)	

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

- $C^2 = \{a_3, a_6, a_8, a_{13}, a_{14}, a_{20}\}$,
- $C^3 = \{a_5, a_{18}\}$,
- $C^4 = \{a_9, a_{11}, a_{16}\}$ and
- $C^5 = \{a_{10}, a_{12}, a_{15}, a_{17}\}$.

From the outranking degrees between classes (Table 7) we obtained the partially ordered and transitive partition presented in Figure 9.

	C^1	C^2	C^3	C^4	C^5
C^1	0	0.3333	0.2000	0.1333	0.1000
C^2	0.3333	0	0	0	0.1250
C^3	0.4000	0.5833	0	0.3333	0.3750
C^4	0.4667	0.7222	0	0	0.5833
C^5	0.4000	0.9167	0	0	0

Table 7: Outranking degree s_{ij} within (C^i, C^j) , $i=1..5$

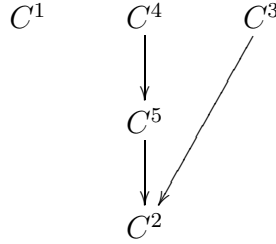


Figure 9: Final Partition

9.2 Second example

We will now present a second example with more alternatives to sort. We will use an example from Yu (1992), referring to the evaluation of 100 alternatives to be sorted, based on their performances on 7 criteria to be minimized.

We will consider as fixed the weights, indifference, preference, discordance, and veto thresholds associated with each criterion, indicated in Table 5. We allow for discordance to occur, although we have chosen values for u_j and v_j that do not allow veto situations to occur frequently when comparing the alternatives.

	g_1	g_2	g_3	g_4	g_5	g_6	g_7
q_j	0.65	1.7	1.7	1.7	1.7	1.7	2.3
p_j	1.31	3.45	3.45	3.45	3.45	3.45	4.7
u_j	1.95	10.4	10.4	10.4	10.4	10.5	18.7
v_j	1.95	100	100	100	100	100	18.7
k_j	0.24	0.12	0.12	0.12	0.12	0.12	0.16

Table 8: Thresholds and weights associated with the criteria.

Evaluating the set of alternatives based on their performance, weights, criteria thresholds (with v_j and $\lambda = 0.6$), of the 4950 pairs of alternatives studied ($\frac{n(n-1)}{2}$, $n = 100$), 2967 (59.9%) verify the relation of Preference, 364 (7.4%) verify the relation of Indifference, and 1619 (32.7%) verify the relation of Incomparability.

Considering that the decision maker is interested only in solutions with less than 8 classes, the results of the merged classes that cause the better quality of the partition are presented in Table 9. Figure 10 depicts how the number of classes impacts on partition quality. As we can see, a good solution can be $k = 6$ because the quality of the partitions with more than 6 classes is not significantly better .

In this example, all partitions are transitive. If the decision maker chooses $k=6$ classes as a good solution, we obtain a partition with a quality of 99.23% resulting from 0% of violations of the

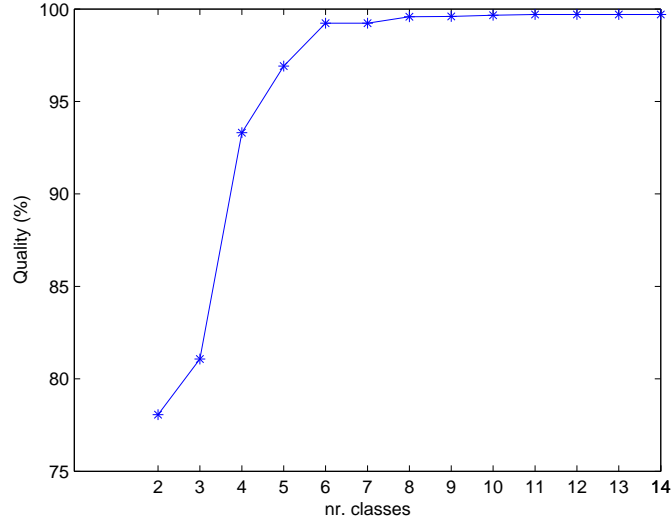


Figure 10: Partition quality as a function of the number of classes (Example 2).

Preference conditions, 0.16% of Indifference and 0.61% of Incomparability. We obtained the partially ordered partition presented in Fig.11.

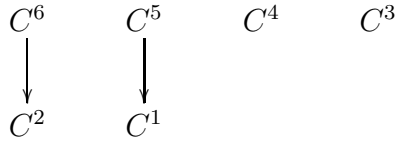


Figure 11: Final partition

10 Conclusions

In this work, we proposed an approach for the classification of a set \mathcal{A} of alternatives, based on multiple criteria, to a set of partially ordered categories with an unknown structure a priori. Formally, the *ranking with partially ordered categories* consists in finding a partition of \mathcal{A} where the partial order relation is defined by the categories: one category can be better or worse comparing with other categories, but can also be incomparable to other categories.

An heuristic method has been proposed on the basis on valued outranking relation and agglomerative hierarchical clustering method (PCO - Partially Ordered Clustering). For the assignment of the alternatives we started by considering the Strong consistency principle. These principle is desirable but there are too many outranking relations in which it is impossible to comply with this requirement, due to the intransitivity of S . Therefore, two new principles are presented: S-consistency and τ -consistency. This principles relax strong consistency by replacing the equivalence $S \Leftrightarrow \tau$ into an implication: $S \Rightarrow \tau$ (S-consistency) or $\tau \Rightarrow S$ (τ -consistency). The problem is 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. Therefore it is trivial to find a (poor) solution in which all alternatives are in a single class (S-consistency) or each alternative is a different class incomparable with the other ones (τ -consistency). We also present

k	Transitivity	Quality	Inconsistences	Structure
7	✓	99.23	(0 0.0016 0.0061)	
6	✓	99.23	(0 0.0016 0.0061)	
5	✓	96.91	(0 0.0016 0.0293)	
4	✓	93.31	(0 0.001 0.0659)	
3	✓	81.07	(0 0.0004 0.1889)	
2	✓	78.06	(0 0.0002 0.2192)	
1	-	67.29	(0 0 0.3271)	

Table 9: Partitions Quality with $SS\tau$ -Consistency

the concept of semi-strong consistency, which seems to overcome the above problems. This principle characterizes the following conditions: similar alternatives should not be placed into incomparable categories (Indifference condition); if an alternative is better than another alternative then the first one should not belong to a worse category (Preference condition); incomparable alternatives should not remain in the same category (Incomparability condition).

The objective of the proposed model has minimize the ratio of the number of misclassified pairs of alternatives to the total number of pairs of alternatives, i.e, the ratio of pairs of alternatives 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. Furthermore, the decision maker may also have some preferences regarding the number of categories or even the relative size of the categories. The PCO method is able to model these different aspects of the preferences of a decision maker.

The key idea to build the preference structure in each stage of the algorithm proposed, is to evaluate the outranking degree between two classes: the class C^i is better than C^j if the outranking degree of C^i on C^j is at least 50% and the outranking degree of C^j on C^i is less than 50%; the classes C^i and C^j are indifferent if the outranking degree of C^i on C^j and C^j on C^i are at least to 50%; and finally, the classes C^i and C^j are incomparable when both outranking degrees are less than 50%.

For each triplet of classes $\{C^i, C^j, C^k\}$, we propose a “negation” of one of its arcs belonging to the intransitive classes set: we should evaluate the quality of the possible partitions obtained by removing one of the relations and the one that leads to better quality will be the final partition.

If the outranking relation between classes has a cycle ($C^i SC^j S...SC^i$) then all partitions obtained by removing one of the relations of the cycle should be evaluated in order to choose the one that leads to better final quality. In particular, for classes such that $C^i SC^j SC^i$ we must add the possibility of merging the classes when choosing best partition in terms of quality obtained.

An acknowledged limitation of the PCO approach is that does not guarantee an “optimal” partition minimizing the violations. 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] 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
- [3] Dias, L.C., C. Lamboray (2010), Extensions of the prudence principle to exploit a valued outranking relation, European Journal of Operational Research, Vol. 201, No. 3, 828-837.
- [4] Doumpos, M., Zopounidis, C. (2002), Multicriteria Decision Aid Classification Methods, Kluwer Academic Publishers, Dordrecht.
- [5] Fernandez E., Navarro J., Bernal S. (2010). Handling multicriteria preferences in cluster analysis. European Journal of Operational Research 202 : 819-827.
- [6] 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.
- [7] Fisher, D. (1987). Knowledge acquisition via incremental conceptual clustering. Machine Learning, 2, 139-172.
- [8] 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.
- [9] Jain, A.K., Dubes, R.C. (1988). Algorithms for Clustering Data. Prentice-Hall advanced reference series. Prentice-Hall, Inc., Upper Saddle River, NJ.
- [10] Karypis, G., Han, E.H., Kumar, V. (1999). CHAMELEON: A hierarchical clustering algorithm using dynamic modeling, COMPUTER, 32, 68-75.
- [11] Kaufman, L., Rousseeuw, P. (1990). Finding Groups in Data : An introduction to Cluster Analysis. John Wiley and Sons, New York.
- [12] Keeney, R. L., Raiffa, H. (1993), Decisions with multiple objectives: preferences and value tradeoff, Cambridge University Press, Cambridge.
- [13] Mousseau, V., Dias, L. (2004), Valued outranking relations in ELECTRE providing manageable disaggregation procedures. *European Journal of Operational Research*, 156(2), 467-482.
- [14] Mousseau, V., Slowinski, R. (1998), Inferring an ELECTRE-TRI model from assignment examples. *Journal of Global Optimization* 12(2), 157-174.
- [15] Nemery, P.; De Smet, Y. (2005). *Multicriteria Ordered Clustering*. Technical Report TR/SMG/2005-003, SMG, Universit Libre de Bruxelles, 2005.
- [16] 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.

- [17] Roy B., Bouyssou D. (1993), Aide multicritère à la décision : Méthodes et cas. Economica, Paris.
- [18] Roy B.(1991), The outranking approach and the foundations of ELECTRE methods. Theory and Decision 31, 49-73.
- [19] Roy B., Bouyssou D. (1993), Aide multicritre la dcision : Mthodes et cas. Economica, Paris.
- [20] Saaty, T.L.(1980), The analytic hierarchy process, McGraw-Hill, New York.
- [21] Saaty, T.L.(1996), Decision making with dependence and feedback: the analytic network process. RWS Publications, Pittsburg.
- [22] Sibson, R. (1973). SLINK: An optimally efficient algorithm for the single link cluster method. Computer Journal, 16, 30-34.
- [23] Vincke, P.(1992). Multicriteria Decision-Aid. New York: J.Wiley.
- [24] Yu, W.(1992), Aide multicritre la dcision dans le cadre de la problmatique du tri: concepts, mthodes et applications. Doctoral dissertation, Universit Paris-Dauphine.