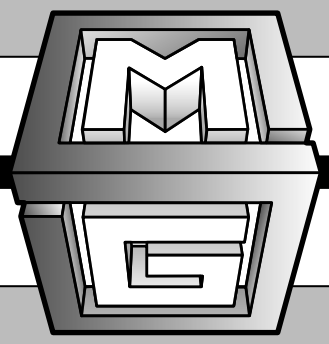


Université Libre de Bruxelles

CoDE - SMG

On the use of Valued Action Profiles for Relational Multicriteria Clustering

CoDE-SMG – Technical Report Series

Stefan Eppe, Julien Roland, Yves De Smet

CoDE-SMG – Technical Report Series

Technical Report No.
TR/SMG/2011-003

July 2011

CoDE-SMG – Technical Report Series
ISSN 2030-6296

Published by:
CoDE-SMG, CP 210/01
Université Libre de Bruxelles
Bvd du Triomphe
1050 Ixelles, Belgium

Technical report number TR/SMG/2011-003

The information provided is the sole responsibility of the authors and does not necessarily reflect the opinion of the members of CoDE-SMG. The authors take full responsability for any copyright breaches that may result from publication of this paper in the CoDE-SMG – Technical Report Series. CoDE-SMG is not responsible for any use that might be made of data appearing in this publication.

On the use of Valued Action Profiles for Relational Multicriteria Clustering

CoDE-SMG – Technical Report Series

Stefan Eppe stefan.eppe@ulb.ac.be
Julien Roland julien.roland@ulb.ac.be
Yves De Smet yves.de.smet@ulb.ac.be
CoDE-SMG, Université Libre de Bruxelles, Brussels, Belgium

July 2011

On the use of Valued Action Profiles for Relational Multicriteria Clustering

Stefan Eppe*, Julien Roland and Yves De Smet

Technical Report -- July 2012

Abstract

Data clustering techniques aim at eliciting hidden structures of a data set by partitioning it into groups of similar elements. In the case of relational clustering, the similarity is based on the relations that exist between elements rather than on intrinsic features of each element. However, applying relational clustering tools to the context of Multicriteria Decision Making requires caution due to the multicriteria nature of preference relations. This paper introduces the concept of Valued Action Profiles, a formalism for handling elements that are defined by pairwise valued outranking relations. These profiles are integrated into and studied on an adapted k-means algorithm that returns a relational partition. Experimental results on both artificial and real data sets show that the use of the proposed method leads to meaningful and robust relational partitions.

1 Introduction

Clustering, also known as unsupervised classification, is an essential data mining tool. It aims at uncovering hidden structures within large data sets by grouping similar objects into unpredefined classes. These classes are known as clusters. Most clustering methods make use of the intrinsic features of an element (such as its coordinates, weight, colour, etc.) to assess its similarity to any other element of the considered set. On the contrary, some applications require relational information (distance, preference, etc.) to be used for the similarity measure. Methods based on this second type of information are known as relational clustering techniques. They are used in various domains, such as web document clustering (Fersini et al., 2009), semantic networks extraction from text (Kok and Domingos, 2008), or pattern recognition in medical imagery (Ogiela and Tadeusiewicz, 2007). Most of these approaches are tackled efficiently by graph clustering techniques, as reviewed in Schaeffer (2007) or van Dongen (2000). Within the group of relational clustering algorithms, a further specialization consists in taking asymmetric relations into account. Typically, this happens in multicriteria decision problems where (binary or valued) preference relations are asymmetric. In this context, the sensitivity of a clustering procedure to the direction of the relation is known as *criteria-dependency* (Cailloux et al., 2007). Not taking the potential asymmetric nature of the relation into consideration would lead to loose valuable input information that may greatly influence the resulting partition. Multicriteria clustering methods have the advantage of comparing actions in an MCDA perspective, each datum being evaluated on a set of criteria. This implies that even if the evaluations on each criterion, and therefore the picture formed by the dataset in the criteria space, remains the same, the distance between two actions could be different because a criterion to minimize is replaced by another one to maximize. Hence, the solution (partition) given by clustering methods could also be different. For more motivations of multicriteria clustering and some examples of this phenomenon the interested reader may refer to Cailloux et al. (2007).

The aim of this paper is to present a method dedicated to *multicriteria relational clustering problems*. It is worth noting that the term *multicriteria clustering* can be understood in the sense of determining a partition that simultaneously optimizes several criteria (such as maximizing both the global compactness of clusters and the local connectedness of data points within a cluster as proposed in Handl and Knowles, 2005).

*Corresponding author at Computer & Decision Engineering department, Polytechnic School of Brussels, Université Libre de Bruxelles, Belgium. E-mail: stefan.eppe@ulb.ac.be

In such cases, clustering can be seen as a special case of multiobjective optimization and well performing approaches such as evolutionary multi-criterion optimization (EMO) algorithms could be successfully applied. Several methods have been proposed in the literature for tackling this type of multicriteria clustering. For additional details see for instance Ferligoj and Batagelj (1992).

This interpretation is misleading in the present context. In what follows, we will use the term *multicriteria clustering* to point out clustering methods based on asymmetric relations resulting from the application of a given multicriteria procedure.

Some recent papers have addressed the question of multicriteria clustering (De Smet and Montano Guzman, 2004; Figueira et al., 2004; Nemery and De Smet, 2005; De Smet and Eppe, 2009) in the sense we understand it here. Basically, existing methods consider three kinds of output: nominal partitions (no relations between the clusters), ordered partitions (a complete pre-order between the clusters), and relational partitions (antisymmetric relations on the clusters). In this paper, we consider the latter case. Let us note that the term *relational* applies both to the input and the output of the method. A distinctive feature of our approach is to highlight relations between the clusters and therefore to provide better insight on how they are related to each other. Moreover, we present a framework that will be applicable both on binary and valued preference relations. In this sense, this constitutes an extension of the binary approach proposed in De Smet and Eppe (2009). Moreover, we provide a detailed empirical analysis (both on artificial and real data sets) as suggested in the conclusions of that previous work.

The proposed valued approach also eliminates the need to define an additional arbitrary cutting threshold λ that is required to be able to use the binary approach for actions that are characterized by valued outranking relations. Indeed, setting a cutting threshold reduces the amount of information contained in the initial data set, while also requiring the user of the method to provide information the effects of which on the resulting relational partition are hard to foresee and can actually lead to very different results.

The sequel of the paper is organized as follows: after presenting and extending the model to valued outranking relations in section 2, the integration of our model into the k-means algorithm is described in section 3. We present the results of our validation procedures in section 4, alongside with three illustrative examples. The latter show how the method can be implemented practically and allow us to gain some further insights into the characteristics of the model we present.

2 The model

A profile associated to each single action is the key concept of the proposed approach. It allows to compare two actions, based on the preference relations each of it has with the whole set of actions. Action profiles are motivated by the goal of comparing the *relational characteristics* of any pair of actions, i.e., to define a distance measure between two actions that takes their relational aspect into consideration. One feature of this way of defining a distance is particularly noteworthy: the distance measure not only depends on the two actions to be compared, but on the whole set of actions.

More formally, we consider a set of n actions $A = \{a_1, \ldots, a_n\}$ and an outranking relation S on A. A pairwise outranking relation is usually provided as a matrix $(S_{ij}), \forall i, j \in \{1, \ldots, n\}$ and computed through an outranking method that aggregates actions on a given set of criteria $G = \{g_1, \ldots, g_q\}$. It allows to express to which degree a given action $a \in A$ *outranks* another action $b \in A$. The most prominent outranking methods are ELECTRE (Roy, 1978) and PROMETHEE (Brans, 1982); we refer to these references for further details.

We will consider the outranking matrix S as being given; its generation will hence not be considered in this section. However, the illustrative example presented in section 4.2 will briefly describe a possible way of eliciting S, based on the ELECTRE III method (Roy, 1978).

The outranking degree of an action $a_i \in A$ over another action $a_j \in A$ will be denoted by $S_{ij} = S(a_i, a_j)$ in the following. The outranking relation S_{ij} can be expressed in two ways: when considered as a binary relation, S_{ij} can yield either value 0 or 1, numerically representing the Boolean assertion that action a_i outranks a_j ($S_{ij} = 1$), or not ($S_{ij} = 0$). As the relation is not symmetrical, not having a_i outranking a_j does not imply that a_j outranks a_i. The case of binary outranking matrices has been studied in De Smet and Montano Guzman (2004) as well as in De Smet and Eppe (2009); the present work is an extension of (De Smet and Eppe, 2009) to the case of valued outranking matrices. When looking at valued outranking degrees from the viewpoint of fuzzy logic, S_{ij} can be interpreted as the credibility − expressed in the interval

$[0,1]$ − that action a_i outranks another action a_j. Using the above outranking degree $S(a_i, a_j)$, the following two subsets of A can therefore be associated to any given action $a_i \in A$.

[Outranking & outranked subsets]
Let $A = \{a_1, \ldots, a_n\}$ be a set of n actions and S be a valued outranking matrix defined on A. The *outranking subset* $S^+_{a_i}$ and *outranked subset* $S^-_{a_i}$ associated with action $a_i \in A$ are defined by their respective membership functions:

$$\begin{aligned} \mu_{S^+_{a_i}}(a_j) &= S(a_i, a_j), \quad \forall\, (a_i, a_j) \in A \times A \\ \mu_{S^-_{a_i}}(a_j) &= S(a_j, a_i), \quad \forall\, (a_i, a_j) \in A \times A. \end{aligned}$$

Based on the provided information (A and S) and on the definition proposed in De Smet and Montano Guzman (2004), we define the valued action profile $Q(a)$ associated with action $a \in A$ as follows.

[Action Profile] Let $A = \{a_1, \ldots, a_n\}$ be a set of n actions and S be a valued outranking matrix defined on A. The profile $Q(a_i)$ associated with action $a_i \in A$ is a quadruple defined as follows.

$$\begin{aligned} Q(a_i) &= \langle\, Q_1(a_i),\, Q_2(a_i),\, Q_3(a_i),\, Q_4(a_i)\,\rangle \\ &= \langle\, J(a_i),\, P^+(a_i),\, P^-(a_i),\, I(a_i)\,\rangle. \end{aligned}$$

An outranking relation induces a preference structure that can be elicited on the basis of subsets $S^+_{a_i}$ and $S^-_{a_i}$ as a partition of the set A. For any action $a_i \in A$, let us define

$$\begin{aligned} J(a_i) &= \bar{S}^+_{a_i} \cap \bar{S}^-_{a_i} \\ P^+(a_i) &= S^+_{a_i} \cap \bar{S}^-_{a_i} \\ P^-(a_i) &= \bar{S}^+_{a_i} \cap S^-_{a_i} \\ I(a_i) &= S^+_{a_i} \cap S^-_{a_i}, \end{aligned}$$

where $\bar{X} = A \setminus X$ represents the complementary of a subset X with respect to set A. For fuzzy sets, the membership function of the complementary set can be defined by $\mu_{\bar{X}}(a) = 1 - \mu_X(a), \ \ \forall a \in A$.

The above subsets represent, with respect to action a_i, the set of incomparable actions ($J(a_i)$), the actions a_i is preferred to ($P^+(a_i)$), the actions that are preferred to a_i ($P^-(a_i)$), and the actions that are indifferent to a_i ($I(a_i)$).

In terms of membership functions, the last set of equations yields the following formulations:

$$\begin{cases} \mu_{J(a_i)}(a_j) = (1 - S_{ij}) \cdot (1 - S_{ji}) \\ \mu_{P^+(a_i)}(a_j) = S_{ij} \cdot (1 - S_{ji}) \\ \mu_{P^-(a_i)}(a_j) = (1 - S_{ij}) \cdot S_{ji} \\ \mu_{I(a_i)}(a_j) = S_{ij} \cdot S_{ji}. \end{cases}$$

The probabilistic T-norm $T(x, y) = x \cdot y$ has been chosen as inclusion operator. Given two subsets $A_1, A_2 \subseteq A$, and an element $a \in A$, $\mu_{A_1 \cap A_2}(a) = \mu_{A_1}(a) \cdot \mu_{A_2}(a)$. It has been chosen because it has the following desirable property.

Given any pair of actions $(a_i, a_j) \in A \times A$, we have

$$\sum_{l=1}^{4} \mu_{Q_l(a_i)}(a_j) = 1.$$

$\sum_{l=1}^{4} \mu_{Q_l(a_i)}(a_j) = \mu_{J(a_i)}(a_j) + \mu_{P^-(a_i)}(a_j) + \mu_{P^+(a_i)}(a_j) + \mu_{I(a_i)}(a_j) = (1 - S_{ij})(1 - S_{ji}) + S_{ij} \cdot (1 - S_{ji}) + (1 - S_{ij}) \cdot S_{ji} + S_{ij} \cdot S_{ji} = 1$.

Indeed, this property allows an intuitive interpretation of the profile sets $Q_l(a_i)$, where $l = 1, \ldots, 4$, as a fuzzy partition, for action a_i, of the set A into 4 fuzzy subsets: $J(a)$, $P^+(a)$, $P^-(a)$, and $I(a)$. This interpretation is consistent with what has been proposed in the binary case (De Smet and Eppe (2009)), the latter becoming a particular case of the former generalized frame.

Table 1: Possible ways of expressing the resulting relational partition $(P_k, S_{P_k}.)$

S_{P_k} / P_k	crisp	fuzzy
crisp	binary outranking relations between crisp clusters	binary outranking relations between fuzzy clusters
fuzzy	valued outranking relations between crisp clusters	valued outranking relations between fuzzy clusters

On the basis of the proposed relational characterization, two actions can consequently be compared by using their respective profiles. One intuitive way of defining a distance is straightforward: two actions will be considered as being close if they relate similarly to all actions of the set A. The similarity of two actions can be quantitatively measured by cardinalities of the intersections of the profile's subsets J, P^+, P^-, and I.

[Distance] Let $Q(a_i)$ be the valued profile of a_i, then the distance between two actions $a_i, a_j \in A$ is defined by:

$$\begin{aligned} d(a_i, a_j) &= 1 - \frac{1}{n}\sum_{l=1}^{4} |Q_l(a_i) \cap Q_l(a_j)| \\ &= 1 - \frac{1}{n}\sum_{l=1}^{4}\sum_{a\in A} \mu_{Q_l(a_i)\cap Q_l(a_j)}(a) \\ &= 1 - \frac{1}{n}\sum_{l=1}^{4}\sum_{a\in A} \min\left(\mu_{Q_l(a_i)}(a) \, , \, \mu_{Q_l(a_j)}(a) \right). \end{aligned}$$

where the set intersection operator is defined by the Zadeh operators. The proof of the above definition being a distance is provided in Appendix A.

In opposition to the probabilistic T-norm operator that has been chosen for defining the 4 subsets $J(a_i)$, $P^+(a_i)$, $P^-(a_i)$, and $I(a_i)$ associated with action a_i, the Zadeh T-norm, $T(x, y) = \min(x, y)$, is chosen as inclusion operator for these subsets. The practical motivation for this choice can be illustrated by a simple example: consider an element x and a set$A$ such that $\mu_A(x) = \frac{1}{3}$. Using the probabilistic T-norm, $T(x, y) = x \cdot y$, would yield that the membership of x to the intersection of A with itself is smaller than to the set A, i.e., $\mu_{A\cap A}(x) = \frac{1}{3} \cdot \frac{1}{3} = \frac{1}{9} < \frac{1}{3} = \mu_A(x)$. The Zadeh T-norm does not expose us to this counter-intuitive kind of results.

Having finally a way of assessing the similarity of two actions, we can now go back to our main concern. The aim of the proposed approach is twofold: first, we want to partition the set of actions into k clusters; second, based solely on the initially provided information, we want to elicit the outranking relations between the determined clusters. The particular advantage of the proposed method is to integrate both aspects (partition and relations between clusters) into the optimization process that the k-means algorithm performs. It is opposed to a 2-phase approach that would first elicit a partition with a conventional clustering method, and only then apply another method for determining the relations between the clusters.

The result of our method is thus given by a partition $P_k(A)$ of A into a set of clusters $C = \{C_1, \ldots, C_k\}$, and a $k \times k$ binary antisymmetric outranking matrix S_{P_k} on that partition.

In opposition to the binary case, when considering valued outranking relations, one has to decide "where" the fuzziness is expressed. Table 1 shows the possible combinations of considering fuzziness at the partition level, expressing partial membership of an action to a cluster, or at relational level, expressing valued outranking degrees between clusters. In the present approach, we have chosen to impose binary outranking relations on crisply defined clusters. Although the results may arguably be considered as over-simplified, our choice is motivated by the ease of interpretation for the decision maker.

Applying a k-means algorithm to partition a set of elements into clusters is an optimization process intended at maximizing the quality of its result. It will numerically be represented by means of a fitness measure f. We use the same definition for fitness as in De Smet and Eppe (2009) since it is implicitly extended to valued outranking relations through the adapted definition of the distance.

$$f(A, S, P_k, S_{P_k}) = 1 - \frac{1}{n} \sum_{i=1}^{k} \sum_{a \in C_i} d(a, c_i),$$

where c_i is the centroid associated with cluster C_i.

3 Algorithm

As already mentioned, the proposed algorithm is an extension of the k-means algorithm: starting from a random partition, the centroids of the clusters are computed. Each action is then associated to the closest cluster, i.e., with the minimum distance to the cluster's centroid. This provides a new partition, leading to new centroids, etc. The procedure is repeated until the partition stabilizes or a maximum number of iterations is reached.

In order to be able to apply the k-means algorithm, we thus still need to define procedures to (1) build a cluster's centroid, and (2) determine the outranking relations between clusters (S_{P_k}). Since the first is needed to define the second, and vice-versa, we will begin with the definition of the clusters centroids, considering that we are given a binary outranking matrix S_{P_k}. This order is in accordance to what is done in the k-means algorithm, where the relational partition (P_k, S_{P_k}) is initially randomly generated.

[Centroid's profile] Let (P_k, S_{P_k}) be a relational partition. The centroid's profile $Q(c_i)$ associated to cluster $C_i \in C$ is defined by :

$$\mu_{P^+(c_i)}(a) = \begin{cases} 1, & if\ \exists m \in \{1, \ldots, k\} : a \in C_m \wedge (C_i S_{P_k} C_m) \wedge (C_m \neg S_{P_k} C_i) \\ 0, & otherwise \end{cases}$$

$$\mu_{P^-(c_i)}(a) = \begin{cases} 1, & if\ \exists m \in \{1, \ldots, k\} : a \in C_m \wedge (C_m S_{P_k} C_i) \wedge (C_i \neg S_{P_k} C_m) \\ 0, & otherwise \end{cases}$$

$$\mu_{J(c_i)}(a) = \begin{cases} 1, & if\ \exists m \in \{1, \ldots, k\} : a \in C_m \wedge (C_i \neg S_{P_k} C_m) \wedge (C_m \neg S_{P_k} C_i) \\ 0, & otherwise \end{cases}$$

$$\mu_{I(c_i)}(a) = \begin{cases} 1, & if\ a \in C_i \\ 0, & otherwise. \end{cases}$$

Although it is based on the formalism of fuzzy logic, it is to be noted that S_{P_k} is a binary relation and that the membership functions above define crisp profiles, i.e., their membership function only takes values 0 or 1.

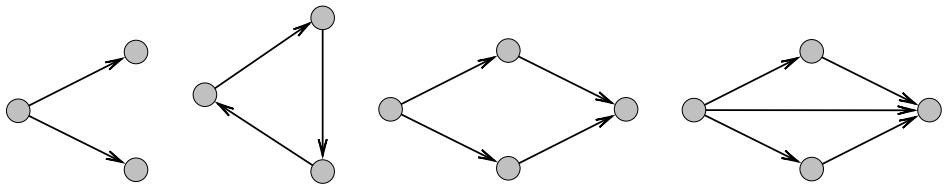
Having determined the centroid's profiles, the binary outranking relations between the clusters can be computed. The intuitive way of optimizing S_{P_k} by maximizing the overall fitness f is computationally prohibitive since its complexity is $O(3^{\frac{k(k-1)}{2}})$ and that it has to be computed at each single iteration of the main algorithm. Fortunately, optimizing the fitness of a relational partition is equivalent to applying the faster voting procedure described hereafter.

$$l^*_{rq} = \begin{cases} \underset{l_{rq} \in \{1,2,3\}}{\arg\max} \left(\sum_{a_i \in C_r} \sum_{a_j \in C_q} \left| Q_{lrq}(a_j) \cap Q_{l_{rq}}(a_i) \right| \right) & \Leftrightarrow\ r \neq q \\ 4 & \Leftrightarrow\ r = q, \end{cases}$$

where l^*_{rq} represents the outranking relation between clusters r and q. It has been shown in De Smet and Eppe (2009) that this procedure leads to the optimal relation S_{P_k}. The extension of this proposition to the valued case is not given here, since it can be deduced directly from the proof for the binary case.

Consequently, the centroid's profile $Q(c_r)$ can be build as follows:

$$Q_l(c_r) = \bigcup_{C_q \in P_k} \{a_j \in C_q | l = l^*_{rq}\}, \forall l \in \{1, 2, 3, 4\}.$$



(a) Model 1 (b) Model 2 (c) Model 3 (d) Model 4

Figure 1: This figure shows 4 reference relational models that have been considered for validation of the proposed method (De Smet and Eppe, 2009). Each relational model represents the underlying outranking relations between the clusters that we wish to be able to elicit with our method.

We can finally establish the following algorithm:

Algorithm 1: Relational Multicriteria Clustering Algorithm

input : $A = \{a_1, ..., a_n\}, S, k$
foreach $a_i \in A$ **do**
  Compute $Q(a_i)$;
end
Randomly initialize the relational partition (P_k, S_{P_k}), such that $\forall C_h \in P_k$, $|C_h| > 0$;
repeat
  foreach $C_j \in P_k$ **do**
    Compute $Q(c_j)$;
  end
  foreach $a_i \in A$ **do**
    Assign a_i to the cluster C_j such that $d(a_i, c_j)$ is minimized;
  end
  Compute S_{P_k} maximizing f;
until *stopping criteria is met* ;

This algorithm is a heuristic of complexity $O(tn^2k^2)$ where t is the number of iterations of the main loop.

4 Validation

In this section the proposed approach will be applied both on artificial and 3 real-life data sets. The goal is to validate and to gain further insight into its behaviour, as well as to give an example of how it is practically applied.

4.1 Artificial data sets

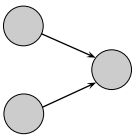
We will use 4 relational models (introduced in De Smet and Eppe (2009) and represented in figure 1) to construct artificial datasets, apply the proposed algorithm on them, and finally compare the obtained results with the original, underlying data structure. We will quantitatively study the stability of the algorithm by using a measure of entropy, analyse the convergence behaviour during execution, and deduce further qualitative observations.

In a second stage, we will study the evolution of performance and convergence of the algorithm during its execution. The algorithm has been implemented in the R programming language.

The validation approach is structured as follows:

1. On the basis of a chosen relational model R, generate a perturbed outranking matrix $S_R^\star$ and a reference partition P_R (Fig. 2)

2. Apply the modified k-means algorithm based on our model; the result is a relational partition (P_k, S_P)



$$R = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

(a) Relational model R

$$P_R = \{\{1,2\},\{3,4\},\{5,6,7\}\} \qquad S_R = \left[\begin{array}{cc|cc|ccc} 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ \hline 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right]$$

(b) Reference partition P_R and extended outranking matrix S_R

$$S_R^\star = \left[\begin{array}{cc|cc|ccc} 1 & \mathbf{0} & \mathbf{1} & 0 & 1 & \mathbf{0} & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ \hline 0 & 0 & 1 & 1 & 1 & \mathbf{0} & 1 \\ \mathbf{1} & 0 & 1 & 1 & 1 & 1 & \mathbf{0} \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & \mathbf{1} & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{array}\right]$$

(c) Perturbated outranking matrix $S_R^\star$ and its graphical representation where zero values are shown in white and unitary values are shown in black

Figure 2: First steps of the validation process: creating artificial datasets. After having chosen a relational model *(a)*, the corresponding matrix R is extended to produce a reference outranking matrix S_R *(b)*. Perturbations are finally applied and yield $S_R^\star$ *(c)*.

3. Compare the result (P, S_P) with the reference relational partition (P_R, R)

The first step is to generate an intermediate outranking matrix $S_R = (S_{R,ij})$. Let k_R be the number of clusters of R. We build $S_R = f(R,t)$ as a function of the underlying relational model R and the cluster size vector $t = (t_1, \ldots, t_{k_R})$. This latter vector gives, for each cluster, the number of actions it has to hold. Figure 2 gives an example of this extension process for $t = \{2,2,3\}$, meaning that cluster 1 and 2 each contain 2 actions while cluster 3 contains 3 actions. Once an extended outranking matrix $S_R = (S_{R,ij})$ has been derived from the relational model R, a perturbation is applied to the former. Based on S_R, we build a new matrix $S_R^\star = (S_{R,ij}^\star)$ by:

$$S_{R,ij}^\star = \begin{cases} S_{R,ij} - X, & \text{if } S_{R,ij} = 1 \\ S_{R,ij} + X, & \text{if } S_{R,ij} = 0, \end{cases}$$

where X is a Beta distribution that acts as perturbation (noise) factor. The Beta distribution is characterized by two parameters: α and β. In the following we will keep $\alpha = 1$ and let only the parameter β continuously vary in the domain $\beta \in [1,4]$. As can be seen on Fig. 3, the closer β is to value 1, the more perturbed the resulting outranking matrix $S_R^\star$.

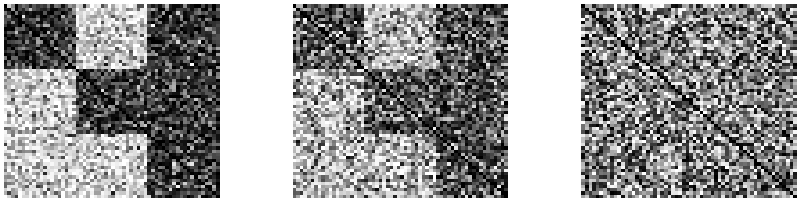
From a practical point of view we start by choosing a relational model R. For each pair (R, β) that allows to generate perturbed outranking matrices $S_R^\star$, we collect the following information:

- average $(\hat{\mu}_f)$ and standard deviation $(\hat{\sigma}_f)$ of the fitness,
- confidence interval of $\hat{\mu}_f$ (CI),
- average number of iterations $(\hat{\mu}_i)$ and average execution time $(\hat{\mu}_t)$
- average entropy $(\hat{\mu}_e)$ between the reference partition and the resulting partition

These statistics are based on 450 executions of the algorithm. For each value of β, 30 perturbed outranking matrices are generated and, for each of those 30 matrices, the algorithm is executed 15 times.

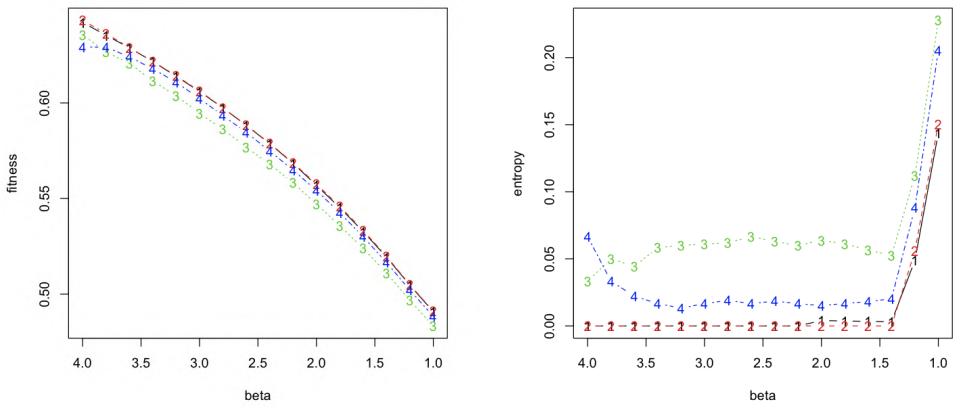
After each iteration of the algorithm we check the following stopping criteria:

- the relational partition is stabilized,



(a) $\alpha = 1,\ \beta = 3$ (b) $\alpha = 1,\ \beta = 2$ (c) $\alpha = 1,\ \beta = 1$

Figure 3: Examples of perturbed outranking matrices for different parameter values. The darker a field, the closer its value is to one; white fields represent a zero value. It can be seen that increasing values of β parameter progressively restore the underlying data structure.



(a) (b)

Figure 4: The 4 considered relational models behave very similarly for different values of the β perturbation parameter. It is shown through the evolution of the average fitness (a), and average entropy (b) with regard to an increasing value of β. *(clusters with 50 actions; the algorithm has been executed 15 times for each value of β)*

- the number of iterations is bigger than 40,
- the fitness did not increase during the 10 last iterations

We have applied this methodology to several relational models. Figure 4 shows that similar results were obtained for all considered relational models. In the following we will thus focus on one of them.
Table 2 presents the summarized results for the relational model presented in Figure 1 with 3 equally sized clusters.

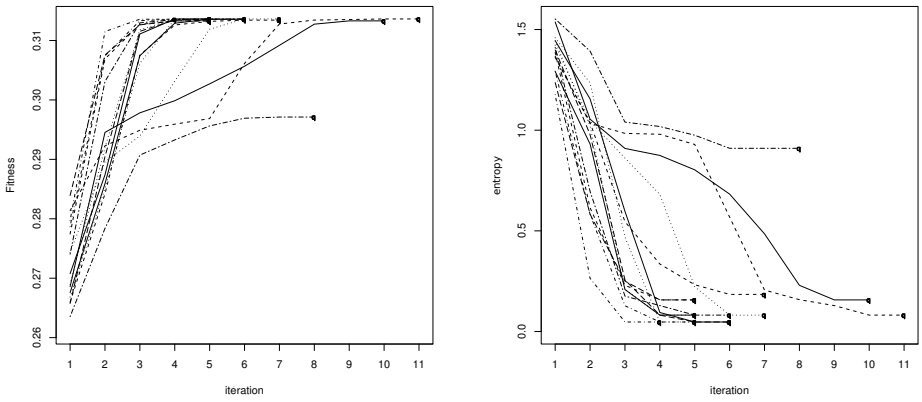
For the considered relational model we observe a linear relation between the parameter β and the average fitness $\hat{\mu}_f$ (they have a correlation equal to 0.996). The entropy remaining very low for β values as low as 0.1 confirms the robustness observed qualitatively in Figure 4.

Figure 5 shows the convergence of the algorithm through the evolution of fitness and entropy measures during the execution of the algorithm.

As a further validation step we consider the stability of the algorithm's outcome, i.e., the preservation of the resulting relational partition when the modified k-means is applied to a randomly chosen fraction of the original action set A. Figure **??** shows for several subset sizes (indicated as fraction of the original set of actions) the evolution of the average entropy $\hat{\mu}_e$ for increasing perturbations, i.e. decreasing values of the

β	$\hat{\mu}_f$	σ_f	CI	Median	$\hat{\mu}_i$	$\hat{\mu}_t$	$\hat{\mu}_e$
4	0.64	0.0015	[0.642;0.643]	0.64	3.7	0.5	0
3.75	0.63	0.0021	[0.625;0.626]	0.63	3.3	0.44	0
3.5	0.61	0.0019	[0.606;0.608]	0.61	3.2	0.44	0
3.25	0.59	0.0019	[0.587;0.589]	0.59	3.2	0.45	0
3	0.57	0.002	[0.564;0.566]	0.57	3.1	0.42	0
2.75	0.54	0.0017	[0.54;0.542]	0.54	3.1	0.42	0
2.5	0.51	0.0022	[0.513;0.515]	0.51	3.1	0.43	0
2.25	0.48	0.0023	[0.481;0.483]	0.48	3.2	0.44	0
2	0.45	0.0016	[0.449;0.45]	0.45	3.3	0.46	0
1.75	0.41	0.0018	[0.408;0.41]	0.41	3.5	0.49	0
1.5	0.36	0.0025	[0.363;0.365]	0.36	4	0.55	0
1.25	0.31	0.0016	[0.313;0.315]	0.31	7.5	1	0.12
1	0.29	0.00057	[0.286;0.287]	0.29	15	2.1	1.6

Table 2: Statistical results obtained on model 1 after 450 executions of the algorithm for different values of the β perturbation parameter. (3 clusters of 50 actions)



(a) Fitness (b) Entropy

Figure 5: Evolution of fitness and entropy during iterations shown for 15 executions of the modified k-means algorithm on the relational model of Figure 1 (3 clusters of 50 actions, $\beta = 1.25$).

β parameter. The low values of the entropy for high levels of noise validates the stability of the proposed model.

4.2 Real data sets

We will now apply our modified k-means algorithm to 3 different real-life data sets:

1. Mortality vs. natality rates (from the CIA World Fact Book [1])
2. Life expectancy, Education, and GDP, from the Human Development Index (HDI)[2]
3. Environmental Performance Index (EPI)[3]

The first set will provide some additional insight into the algorithms behaviour. We will then apply the same procedure to investigate the data for the second and third example and only focus on the results that will confirm the findings of the first one, rather than repeating how these results have been computed.

Mortality vs. natality In this data set, 223 countries are characterized by their respective natality and mortality rates. It has already been used in several papers (Mucha and Bartel, 2004; Hartigan, 1975) and is proposed in the R package *fEcofin*[4]. This example has been chosen for illustrative purpose as it is easier to map it in two dimensions.

The proposed example will also be used to study some general features (mainly the conditions of apparition of cycles in the relational structure and cluster transitivity) of the resulting partitions, offering a further validation of our approach.

The natality rate is assumed to be maximized and the mortality to be minimized. In order to build the valued outranking matrix for this data set, we use the ELECTRE III method.[5] This method requires to define, for each considered criterion $j \in \{1, \ldots, q\}$, a weight w_j, a preference threshold p_j, an indifference threshold q_j, and a veto threshold v_j. Since our goal is to illustrate the algorithm, we will not discuss the choice of the parameters into more details. Instead, we arbitrarily define their value as follows:

$$w_j = \frac{1}{2}, \qquad q_j = 0, \qquad v_j = IQR_j, \qquad p_j = \frac{v_j}{2},$$

where IQR_j is the interquartile range on the set $\{g_j(b) - g_j(a), \forall (a,b) \in A \times A\}$, and $g_i(a)$ is the evaluation of action a on criterion j. The veto threshold v_j is defined as the size of the interquartile range of the evaluation of each action on criterion j. In other words, we reject the outranking relation aSb as soon as one criteria j satisfies the following: $g_j(b) - g_j(a) > IQR_j$.

The number of clusters k has to be chosen beforehand as it is an input parameter for the modified k-means algorithm. In order to determine the best value of k we evaluate relational partitions for $k = 2, \ldots, 10$ using common intern criteria (Dunn, Silhouette, and fitness). Respective evolution of each of those criteria are presented in figure 6. As we can see from this figure, these criteria lead us to choose $k = 3$.

Using $k = 3$, figure 7 and 8 give the resulting outranking matrix and partition. Clusters 1, 2 and 3 are noted C_1, C_2 and C_3, respectively. Hence, we obtain two clusters with a preference relation ($C_1 P C_2$), the third cluster being incomparable to the other two ($C_2 J C_3 \wedge C_1 J C_3$). This result is in concordance with the nature of the data set. Indeed, cluster C_1 corresponds to the countries having a mean rate of natality and a low rate of mortality, cluster C_2 correspond to the countries having a mean natality rate and a mean mortality rate, and cluster C_3 corresponds to the countries having a low natality rate and a high mortality rate. This leads us to conclude that we have obtained a natural relational structure underlying the data set. Hence, from our point of view, this example gives a good motivation for our clustering method, since it confirms relations between clusters that could have been brought out *a priori* by other means.

[1] Data of 2009, available on `https://www.cia.gov/library/publications/the-world-factbook`

[2] Data of 2009 from the "R" package. See also `http://hdr.undp.org/en/statistics/hdi`

[3] Data of 2009 from the "R" package. See also `http://epi.yale.edu`

[4] `http://cran.r-project.org/web/packages/fEcofin/index.html`

[5] As it is not our purpose to describe the method here, the interested reader may refer to Roy (1978) for a detailed presentation of ELECTRE III.

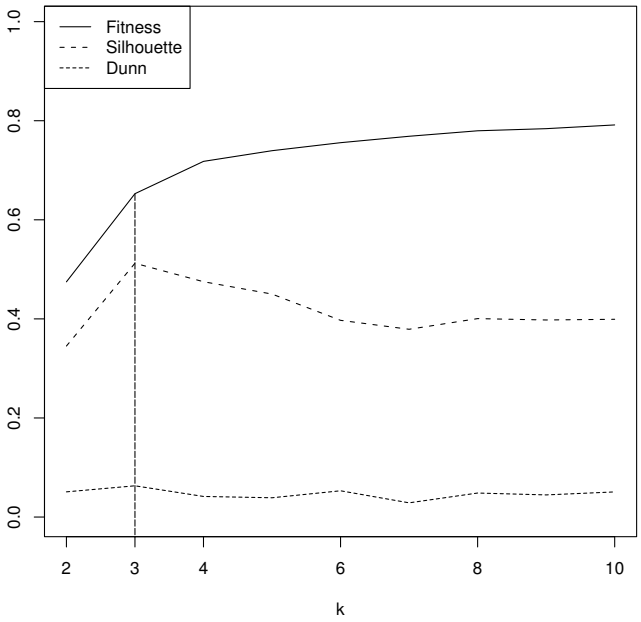


Figure 6: Before executing the algorithm, the number of clusters k has to be determined. This plot shows the respective evolution of fitness, Silhouette, and Dunn criteria with respect to different values of k. Mainly based on the inflexion of the Silhoutte criterion, we have selected $k = 3$ as being the best possible number of clusters.

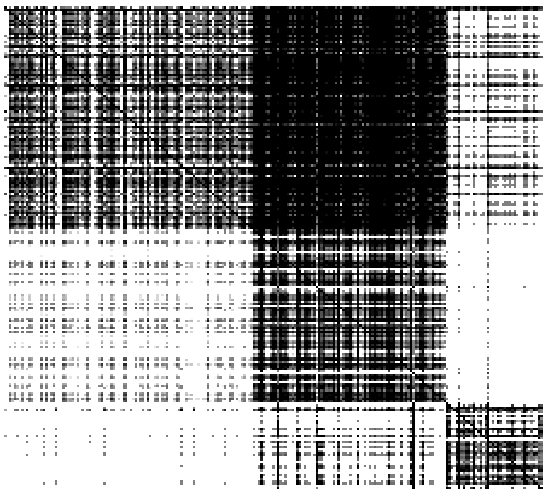


Figure 7: Outranking matrix organized with $k = 3$

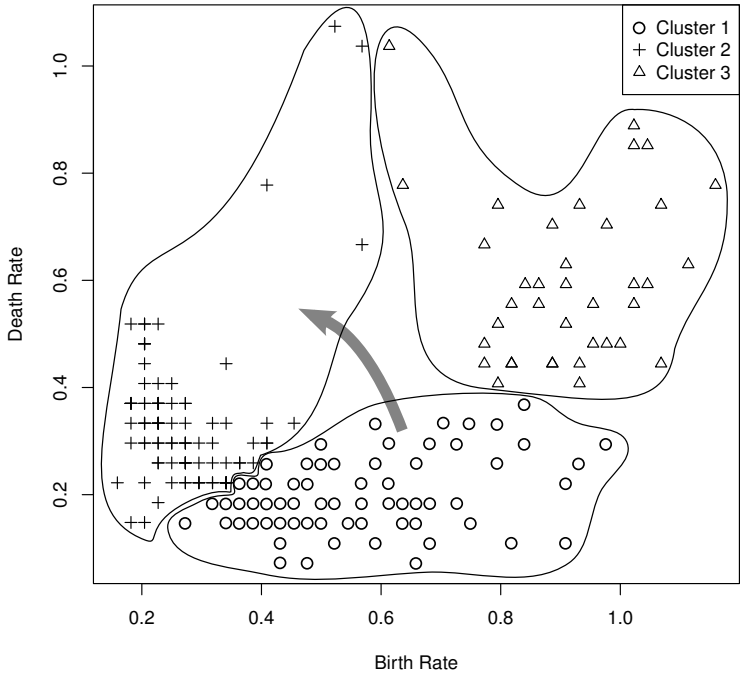


Figure 8: Partition obtained using our clustering method with $k = 3$

Human Development Index and Environmental Performance Index The Human Development Index (HDI) characterizes a set of 182 countries by three indicators: life expectancy, education, and GDP. The Environmental Performance Index (EPI) characterize 149 countries by the use of two indicators: environmental health, and ecosystem vitality. In both cases, the respective indicators are used as criteria on which the ELECTRE III method will be applied.

On each example, the algorithm is applied to a range of k (number of clusters). For each value of k, the resulting relational partition is checked on the following criteria: totality, transitivity, and appearance of cycles. We also compute the size of the smallest cluster to measure the acceptability of the output.

In practice, totally ordered partitions appear in two cases: Either the data set is defined by correlated criteria, or the number of clusters is too low for describing the underlying structure in a satisfying way. Usually, in the former case, a cluster can be generally split up into two incomparable clusters by increasing the number of clusters.

Non-transitive relations seem to never appear with a low number of clusters. For instance in the case of EPI, the relations are transitive up to four clusters.

Cycles only appear when there is over-fitting, i.e., when the number of clusters is close to the number of actions. However, this is a local phenomenon inside some regions of the outranking graph and such actions usually have the same relation with other actions. Hence, this should not appear in the clustering since it is used to simplify the underlying structure of the data set. In the case of EPI cycles, for instance, one has to set the number of clusters to an artificial value as high as 78 to see cycles appear.

The analysis of the algorithm's behaviour reveals that the solution may get trapped in a non-transitive local optimum while the optimal partition is transitive. In such cases the optimal solution can be hard to reach for an increasing number of clusters.

The interpretation of the relational partition may become very difficult in some cases. This is for instance the case when the EPI data set is partitioned into 9 clusters as illustrated in figure 10a. By removing the relations that can be obtained by transitive closure, one obtains the solution pictured in figure 10b that is

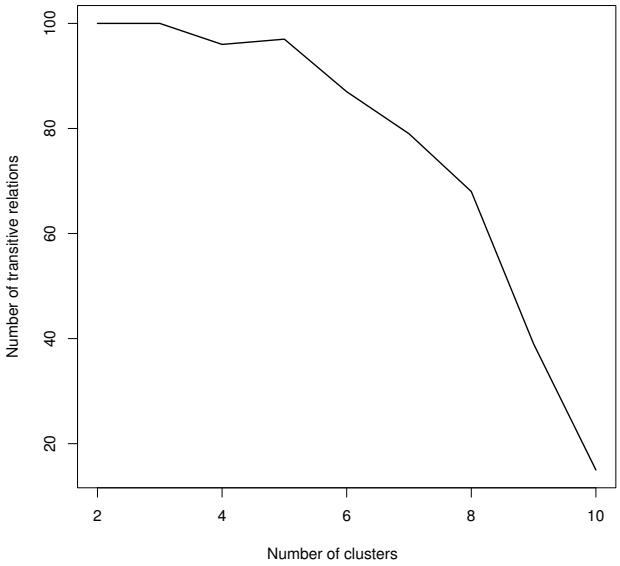


Figure 9: This figure shows the number of transitive relations for an increasing number k of clusters for the data set EPI.

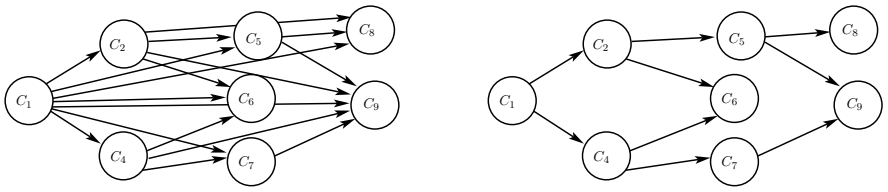


Figure 10: Relational partition obtained for the EPI dataset with the number of clusters set to $k = 9$. In (a), the raw solution returned by the algorithm is shown. Figure (b) is the cleaned-up version; the arcs that could be obtained by transitive closure are have been removed.

easier to understand.

This is why insuring the transitivity of the returned partition might help reducing the complexity of the solution, as well as making the optimal solution easier and faster to find.

5 Conclusions and future research

In this paper, we have addressed the problem of relational multicriteria clustering. The main added value of this contribution is to propose a formalism that can both be applied on a binary and on a valued outranking matrix. As a consequence, this constitutes an extension of previous works (De Smet and Montano Guzman, 2004; De Smet and Eppe, 2009).

A distinctive feature of the method relies on the fact that it takes the multicriteria nature of the problem into account. Put in other words, it constitutes a suitable approach to exploit asymmetric relations between the objects (as it usually happens in multicriteria contexts). Finally, it helps the decision maker to identify both a partition and potential relations between the clusters. In this sense, it differs from classical approaches of relational clustering.

Validation tests have been conducted on both artificial and real data sets. We have been able to show

k	Transitivity	Completeness	Cycles	$\bar{x}$	s	Min	Max
2	1	1	0	81.50	13.44	72	91
3	1	0	0	54.33	16.29	43	73
4	1	0	0	40.75	16.24	26	63
5	1	0	0	32.60	21.24	9	67
6	1	0	0	27.17	20.00	5	52
7	1	0	0	23.29	20.58	2	52
8	1	0	0	20.38	20.24	1	51
9	1	0	0	18.11	19.84	1	52
10	0	0	0	16.30	19.49	1	52
11	0	0	0	14.82	18.95	1	52
12	0	0	0	13.58	18.74	1	52
13	0	0	0	12.54	18.33	1	52
14	0	0	0	11.64	17.88	1	52
15	0	0	0	10.87	17.69	1	53
16	0	0	0	10.19	16.92	1	50
17	0	0	0	9.59	16.74	1	52
18	0	0	0	9.06	16.39	1	52
19	0	0	0	8.58	16.29	1	61
20	0	0	0	8.15	15.68	1	55

Table 3: Data Set EPI on the best solution obtained by Algorithm 1 for 30 executions.

that the resulting relational partitions are robust and meaningful. From our point of view, the elicitation of crisp relations between the clusters offers additional information to the decision maker that can help him have a better insight into the multicriteria structure of the data sets. It has also been shown experimentally that cycling relations only appear when setting the parameter k, i.e. the number of clusters, to a very high value. The advantage of forcing transitive cluster relations has finally also been highlighted, offering one possible path to further improve the proposed method.

Acknowledgement

Stefan Eppe acknowledges support from the META-X project, an *Action de Recherche Concertée* funded by the Scientific Research Directorate of the French Community of Belgium.

A Proof that Definition 2 is a distance

- $d(a,b) = d(b,a)$ by definition.
- $d(a,b) \geq 0$

$$d(a,b) \geq 0 \iff \sum_{l=1}^{4} |Q_l(a) \cap Q_l(b)| \leq n \tag{1}$$

Using Zadeh operators,

$$\sum_{l=1}^{4} |Q_l(a) \cap Q_l(b)| = \sum_{l=1}^{4} \sum_{x \in A} \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x)) \tag{2}$$

Since we have $\sum_{l=1}^{4} \mu_{Q_l(a_i)}(a_j) = 1, \forall (a_i, a_j) \in A \times A$ we obtain, by summing on the set of alternatives A,

$$\sum_{x \in A} \sum_{l=1}^{4} \mu_{Q_l(a)}(x) = n \tag{3}$$

As $\mu_{Q_l(a)}(x) \geq \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x))$, this implies that

$$\sum_{l=1}^{4} \sum_{x \in A} \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x)) \leq n \tag{4}$$

k	Transitivity	Completeness	Cycles	$\bar{x}$	s	Min	Max
2	1	1	0	111.50	19.09	98	125
3	1	0	0	74.33	39.55	30	106
4	1	0	0	55.75	45.46	14	113
5	1	0	0	44.60	42.58	10	117
6	1	0	0	37.17	37.12	6	106
7	1	0	0	31.86	37.31	2	108
8	0	0	0	27.88	35.80	1	106
9	0	0	0	24.78	35.14	1	108
10	0	0	0	22.30	33.61	1	106
11	0	0	0	20.27	33.38	1	110
12	0	0	0	18.58	32.34	1	110
13	0	0	0	17.15	30.96	1	113
14	0	0	0	15.93	30.75	1	111
15	0	0	0	14.87	29.80	1	111
16	0	0	0	13.94	28.52	1	114
17	0	0	0	13.12	27.38	1	102
18	0	0	0	12.39	25.78	1	96
19	0	0	0	11.74	25.00	1	92
20	0	0	0	11.15	26.06	1	114

Table 4: Data Set Birth/Death Rates on the best solution obtained by Algorithm 1 for 30 executions.

- $d(a,b) = 0 \Longleftrightarrow a = b$
  - $(a = b) \Rightarrow (d(a,b) = 0)$ Equation (3) implies

$$n = \sum_{x \in A} \sum_{l=1}^{4} \min(\mu_{Q_l(a)}(x), \mu_{Q_l(a)}(x)) \tag{5}$$

    as $a = b$ (i.e. for all $l \in \{1, \cdots, 4\}$ and $x \in A$: $\mu_{Q_l(a)}(x) = \mu_{Q_l(b)}(x)$) we can deduce :

$$n = \sum_{x \in A} \sum_{l=1}^{4} \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x)) \tag{6}$$

    hence, we can conclude that

$$d(a,b) = 1 - \frac{1}{n} \sum_{x \in A} \sum_{l=1}^{4} \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x)) = 1 - \frac{n}{n} = 0 \tag{7}$$

  - $(d(a,b) = 0) \Rightarrow (a = b)$ This is equivalent to showing that $(a \neq b) \Rightarrow (d(a,b) > 0)$. As $a \neq b$, for at least one $l' \in \{1, \ldots, 4\}$ and one $x' \in A$:

$$\mu_{Q_{l'}(a)}(x') \neq \mu_{Q_{l'}(b)}(x') \tag{8}$$

    Let us assume without loss of generality that

$$\mu_{Q_{l'}(a)}(x') < \mu_{Q_{l'}(b)}(x') \tag{9}$$

    This leads us to

$$\min(\mu_{Q_{l'}(a)}(x'), \mu_{Q_{l'}(b)}(x')) < \mu_{Q_{l'}(b)}(x') \tag{10}$$

    Furthermore, $\forall l \in \{1, \ldots, 4\}$ and $x \in A$,

$$\min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x)) \leq \mu_{Q_l(b)}(x) \tag{11}$$

    Combining equations (3), (10) and (11), we obtain

$$n = \sum_{l=1}^{4} \sum_{x \in A} \mu_{Q_l(b)}(x) > \sum_{l=1}^{4} \sum_{x \in A} \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x)) \tag{12}$$

    Replacing the above equation in the distance formula finally yields $d(a,b) > 0$.

k	Transitivity	Completeness	Cycles	$\bar{x}$	s	Min	Max
2	1	1	0	90.00	45.25	58	122
3	1	1	0	60.00	17.32	50	80
4	1	1	0	45.00	28.91	18	86
5	1	1	0	36.00	31.36	6	86
6	1	1	0	30.00	31.13	1	85
7	1	1	0	25.71	31.00	1	86
8	1	1	0	22.50	30.49	1	87
9	0	1	1	20.00	29.12	1	86
10	1	1	0	18.00	28.17	1	86
11	1	1	0	16.36	27.17	1	86
12	1	1	0	15.00	24.54	1	86
13	1	1	0	13.85	24.72	1	87
14	1	1	0	12.86	23.29	1	86
15	1	0	0	12.00	22.59	1	81
16	1	1	0	11.25	21.30	1	75
17	0	1	1	10.59	21.42	1	89
18	1	1	0	10.00	19.67	1	65
19	1	1	0	9.47	20.55	1	87
20	1	0	0	9.00	19.48	1	86

Table 5: Data Set HDI on the best solution obtained by Algorithm 1 for 30 executions.

- $d(a,c) + d(c,b) \geq d(a,b)$ Let us start by showing the following proposition: For all $a, b, c, x \in A$:

$$\begin{aligned} \sum_{l=1}^{4} \Big(\min(\mu_{Q_l(a)}(x), \mu_{Q_l(c)}(x)) + \min(\mu_{Q_l(c)}(x), \mu_{Q_l(b)}(x)) \Big) \\ \leq 1 + \sum_{l=1}^{4} \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x)) \end{aligned} \tag{13}$$

Let $\alpha_l = \min(\mu_{Q_l(a)}(x), \mu_{Q_l(c)}(x))$ and $\beta_l = \min(\mu_{Q_l(c)}(x), \mu_{Q_l(b)}(x))$. Hence for all $l \in L$: $\alpha_l \leq \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x))$ or $\beta_l \leq \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x))$, with $L = \{1, 2, 3, 4\}$. Indeed, if $\mu_{Q_l(a)}(x) \leq \mu_{Q_l(b)}(x)$ then $\alpha_l \leq \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x))$ by definition of α_l, otherwise $\mu_{Q_l(a)}(x) > \mu_{Q_l(b)}(x)$ and then $\beta_l \leq \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x))$ by definition of β_l. We now define two sets:

$$\begin{cases} L_\alpha = \{l \in L \mid \alpha_l \leq \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x))\} \\ L_\beta = \{l \in L \mid \beta_l \leq \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x)) \wedge \alpha_l > \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x))\} \end{cases}$$

Note that $L_\alpha \cap L_\beta = \emptyset$ and $L_\alpha \cup L_\beta = L$. Using these notations, we see that

$$\sum_{l \in L_\alpha} \alpha_l + \sum_{l \in L_\beta} \beta_l \leq \sum_{l=1}^{4} \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x)) \tag{14}$$

By definition, for all $l \in L$: $\alpha_l \leq \mu_{Q_l(c)}(x)$ and $\beta_l \leq \mu_{Q_l(c)}(x)$. Then,

$$\sum_{l \in \bar{L}_\alpha} \alpha_l + \sum_{l \in \bar{L}_\beta} \beta_l \leq \sum_{l=1}^{4} \mu_{Q_l(c)}(x) = 1 \tag{15}$$

where $\bar{L}_\alpha$ and $\bar{L}_\beta$ are complements on L of L_α and L_β, respectively. Combining equations (14) and (15),

$$\sum_{l=1}^{4} \alpha_l + \sum_{l=1}^{4} \beta_l \leq 1 + \sum_{l=1}^{4} \min(\mu_{Q_l(a)}(x), \mu_{Q_l(b)}(x)) \tag{16}$$

This shows equation (13). The triangular inequality is proven by the fllowing.

$$
\begin{aligned}
d(a,b) &= 1 - \frac{1}{n}\sum_{x\in A}\sum_{l=1}^{4}\min(\mu_{Q_l(a)}(x),\mu_{Q_l(b)}(x)) \\
&\leq 1 - \frac{1}{n}\sum_{x\in A}\left\{\Big(\sum_{l=1}^{4}(\alpha_l+\beta_l)\Big) - 1\right\} \\
&\leq 2 - \frac{1}{n}\sum_{x\in A}\sum_{l=1}^{4}(\alpha_l+\beta_l) \\
&\leq 1 - \frac{1}{n}\sum_{x\in A}\sum_{l=1}^{4}\alpha_l + 1 - \frac{1}{n}\sum_{x\in A}\sum_{l=1}^{4}\beta_l \\
&\leq 1 - \frac{1}{n}\sum_{x\in A}\sum_{l=1}^{4}\min(\mu_{Q_l(a)}(x),\mu_{Q_l(c)}(x)) + 1 - \frac{1}{n}\sum_{x\in A}\sum_{l=1}^{4}\min(\mu_{Q_l(c)}(x),\mu_{Q_l(b)}(x)) \\
&\leq d(a,c) + d(c,b)
\end{aligned}
\tag{17}
$$

References

Brans, J., 1982. L'ingénièrie de la décision; Elaboration d'instruments d'aide à la décision. La méthode PROMETHEE. In: Nadeau, R., Landry, M. (Eds.), L'aide à la décision: Nature, Instruments et Perspectives d'Avenir. Presses de l'Université Laval, Québec, Canada, pp. 183–213.

Cailloux, O., Lamboray, C., Nemery, P., 2007. A taxonomy of clustering procedures. In: Proceedings of the 66th Meeting of the EWG on MCDA, Marrakech, Maroc.

De Smet, Y., Eppe, S., 2009. Multicriteria relational clustering: The case of binary outranking matrices. In: Ehrgott, M., et al. (Eds.), Conference on Evolutionary Multi-Criterion Optimization (EMO 2009). Vol. 5467 of LNCS. Springer, pp. 380–392.

De Smet, Y., Montano Guzman, L., October 2004. Towards multicriteria clustering: An extension of the k-means algorithm. European Journal of Operational Research 127 (2), 390–398.

Ferligoj, A., Batagelj, V., 1992. Direct multicriteria clustering algorithms. Journal of Classification 9 (1), 43–61.

Fersini, E., Messina, E., Archetti, F., 2009. A probabilistic relational approach for web document clustering. Information Processing and Management.

Figueira, J., De Smet, Y., Brans, J.-P., 2004. MCDA methods for sorting and clustering problems: Promethee TRI and promethee CLUSTER. Tech. Rep. TR/SMG/2004-002, SMG, Universit Libre de Bruxelles.

Handl, J., Knowles, J. D., 2005. Multiobjective clustering around medoids. In: Congress on Evolutionary Computation. IEEE, pp. 632–639.

Hartigan, J., 1975. Clustering Algorithms. John Wiley and Sons, New York.

Kok, S., Domingos, P., 2008. Extracting semantic networks from text via relational clustering. In: Daelemans, W., Goethals, B., Morik, K. (Eds.), ECML/PKDD (1). Vol. 5211 of Lecture Notes in Computer Science. Springer, pp. 624–639.

Mucha, H.-J., Bartel, H.-G., 2004. Cluscorr98 – adaptive clustering, multivariate visualization, and validation of results. In: Proceedings of the 27th Annual Conference of the GfKl. Springer.

Nemery, P., De Smet, Y., 2005. Multicriteria ordered clustering. Tech. Rep. TR/SMG/2005-003, SMG, Universit Libre de Bruxelles.

Ogiela, M., Tadeusiewicz, R., 2007. Pattern recognition, clustering and classification applied to selected medical images. Vol. 84 of Studies in Computational Intelligence (SCI). Springer Berlin, Ch. 6, pp. 117–151.

Roy, B., 1978. ELECTRE III : Un algorithme de classements fondé sur une représentation floue des préférences en présence de critères multiples. Cahiers du CERO 20 (1), 3–24.

Schaeffer, S. E., 2007. Graph clustering. Computer Science Review 1 (1), 27–64.

van Dongen, S., May 2000. Graph clustering by flow simulation. Ph.D. thesis, University of Utrecht, Utrecht.