On clustering in multiple criteria decision aid : theory and applications

Alexandru Liviu Olteanu

▶ To cite this version:

Alexandru Liviu Olteanu. On clustering in multiple criteria decision aid : theory and applications. Operations Research [math.OC]. Télécom Bretagne, Université de Bretagne Occidentale, 2013. English. NNT : . tel-00908831

N° d'ordre : 2013**telb0276**

Sous le sceau de l'Université européenne de Bretagne

Télécom Bretagne

En accréditation conjointe avec l'Ecole Doctorale Sicma

Co-tutelle avec l'Université du Luxembourg

On clustering in multiple criteria decision aid: Theory and applications

---

Thèse de Doctorat

Mention : Sciences et Technologies de l'Information et de la Communication

Présentée par **Alexandru-Liviu Olteanu**

Département : LUSSI

Laboratoire : Lab-STICC Pôle : CID, équipe DECIDE

Directeurs de thèse : Raymond Bisdorff et Philippe Lenca

Soutenue le 24 juin 2013

Jury :

M. Vincent Mousseau, Prof. Dr à l'Ecole Centrale Paris (Rapporteur)
Mme Aida Valls, Assoc. Prof. Dr. à l'Université Rovira i Virgili, Espagne (Rapporteur)
M. Raymond Bisdorff, Prof. Dr. à l'Université du Luxembourg (Directeur de thèse)
M. Philippe Lenca, Prof. Dr. à Télécom Bretagne (Directeur de thèse)
M. Patrick Meyer, Assoc. Prof. Dr. à Télécom Bretagne (Encadrant)
M. Pascal Bouvry, Prof. Dr. à l'Université du Luxembourg (Examinateur)
M. Marc Sevaux, Prof. Dr. à l'Université de Bretagne Sud (Examinateur)
M. Bernabe Dorronsoro, Dr. à l'Université de Lille (Examinateur)

Contents

List of Tables

Acknowledgements

I would like to thank, in the following, all the people that have aided me in successfully completing my Ph.D. reseach.

Foremost, I am very grateful to Raymond Bisdorff for selecting me to be his assistant and Ph.D. student at the University of Luxembourg. He has given me plenty of freedom in pursuing my research topic while steering me in the right direction when I veered off course. I've always enjoyed his company and especially his enthusiasm when discussing both about work and and other topics. In particular, I distinctly remember some of our chats over an espresso in the lounge of the Chambre de Commerce. I will definitely miss those times and hope that we would still meet again plenty more times in the future.

The second person that I owe a large amount of gratitude is Patrick Meyer, who has taken an interest in my work during the first year of the Ph.D. and has accepted to join in my supervision through an agreement with Télécom Bretagne. Patrick was always ready to give me a helping hand when I needed it and during each of my stays at Télécom Bretagne I was able to accomplish a lot with my research. I remember with pleasure my trips to Brest and especially the company of Patrick, Sophie and Noé.

I would like to thank also Philippe Lenca for accepting to become my Ph.D. director at Télécom Bretagne and also for his helpful insights into the field of Data Mining.

Furthermore, I appreciate the help of Pascal Bouvry and Bernabe Dorronsoro, who have also taken part in my Thesis Supervisory Committee and have been there at the crucial moments in the Ph.D. giving important advice on how to proceed further.

I am also grateful to Aida Valls, Vincent Mousseau and Marc Sevaux for carefully reviewing my work and for agreeing to take part in my defence as members of the Dissertation Supervisory Committee.

I would like to thank my parents and my brother as well for their support.

I am especially grateful to Roxana for being at my side for the last few years, which would have been much harder without her.

Finally, I would like to thank all my close friends, who did not forget me, despite seeing me only a few times per year.

Résumé

L'information est présente partout autour de nous, sous beaucoup des formes et en grandes quantités. Le thème de l'analyse des données se centre autour du traitement de ces données afin d'en extraire et de mettre en évidence des informations utiles qui seraient autrement impossible à extraire. Une des activités les plus fondamentales des êtres humains, lorsqu'il s'agit d'essayer de comprendre un nouvel objet ou un phénomène, est de le comparer avec d'autres éléments connus par l'identification de caractéristiques communes [Anderberg 1973, Everitt et al. 2001]. Ainsi, deux approches d'analyse de données pour grouper des objets ont été développées: la classification supervisée et la classification non supervisée (ou clustering). Cette thèse se concentre sur ce dernier aspect, mais dans un contexte où de l'information préférentielle est également disponible sur ces données.

Le domaine de l'aide multicritère à la décision (AMCD) étudie la fa con dont les gens atteignent des décisions différentes et vise à les aider dans ce processus à travers différentes méthodologies et outils. La principale différence avec l'information qui est disponible dans un contexte d'analyse des données est l'existence de différents types d'informations préférentielles sur ces données. Cette information est représentée par un large éventail de modèles, dont les paramètres doivent être réglés avec précision de sorte qu'ils reflètent la perception subjective du décideur sur les objets de décision existants, qui dans ce contexte sont appelés alternatives.

Le problème de la classification non supervisée n'a été abordé que très peu dans le domaine de l'AMCD. En outre, dans le contexte de la classification non supervisée en analyse de données, différentes mesures de similarité sont utilisées pour regrouper des objets. Nous soulevons dans ce travail la question si de telles mesures sont appropriées pour le clustering en AMCD. Est-ce que d'autres mesures sont capables de mieux discriminer entre des alternatives en tenant compte de l'information préférentielle proposée par le décideur? Si oui, y a-t-il une différence claire entre l'utilisation de ces mesures et celles de similarité?

à partir des réponses aux questions précédentes, la définition formelle du problème de la classification non supervisée en AMCD peut être correctement motivé. Par conséquent, nous définissons le clustering en AMCD en utilisant les relations qui peuvent naturellement être exprimés dans ce contexte. Suite à cela, nous nous intéressons à la modélisation et à la résolution du problème de clustering en AMCD. Le thème de la validation des algorithmes de regroupement proposés est également un point important. Alors que dans un contexte d'analyse de données, nous sommes en mesure de trouver de nombreuses ressources sur des critères de regroupement et de classification, qui nous permettent de tester de nouvelles méthodes qui envisagent un large éventail de structures possibles des données, en AMCD ce n'est pas particulièrement le cas.

Un autre sujet est centré autour de la cardinalité des ensembles de données généralement trouvés en AMCD. Beaucoup d'approches en aide à la décision se

concentrent sur les problèmes de petite cardinalité qui contiennent seulement une poignée d'alternatives de décision. Bien que certaines approches peuvent être en mesure d'examiner des ensembles de données plus importants, beaucoup d'entre elles ne s'adaptent pas bien à cause de problèmes de complexité liés aux temps d'exécution et aux exigences de mémoire. Comme le clustering est considéré comme une technique d'analyse exploratoire des données, il est naturel de l'imaginer dans un contexte de problèmes de grande taille en AMCD. Nous abordons donc également la question de trouver des problèmes d'AMCD de grande cardinalité, et nous envisageons l'extensions des approches proposées à ce contexte. En outre, nous explorons les différentes mesures qui peuvent être utilisés afin d'analyser ces jeux de données.

La thèse est divisée en trois parties, en suivant les sujets présentés dans les paragraphes précédents. Ce résumé est également structuré de cette fa con.

Partie I: Sur le problème de la classification non supervisée en AMCD

La première partie commence par introduire le problème de clustering dans le contexte original dans lequel il a été formulée, celui de l'analyse des données. Le problème de classification non supervisée est généralement défini comme la problématique du regroupement d'objets qui sont semblables tout en séparant ceux qui ne le sont pas. En raison de sa définition très générale, le clustering a été appliqué à de nombreux domaines, ayant des attentes différentes en ce qui concerne le résultat du regroupement. Par conséquent, de nombreuses mesures de similarité et de nombreuses approches de clustering ont été proposées [Jain 2010]. Le domaine de l'AMCD estime que des informations préférentieles supplémentaires sont disponibles sur les données qui sont utilisées pour décrire chaque objet, qui dans ce cas sont appelés alternatives. Cette information est habituellement donnée par un décideur et le but est de l'aider à atteindre certaines décisions sur les solutions envisagées. Deux orientations philosophiques peuvent être identifiées: la prise de décision et l'aide à la décision. La première vise à trouver une solution optimale au problème, tandis que la seconde vise à donner seulement une recommandation qui aidera le décideur à prendre sa décision. Les principaux modèles de préférence utilisés en AMCD sont les fonctions de valeur [Keeney and Raiffa 1976] et les relations surclassantes [Roy 1993]. Les fonctions de valeur réduisent chaque solution à une valeur unique, ou un score, alors que les relations de surclassement sont construits entre toutes les paires d'alternatives et peuvent être considérées comme similaires aux procédures de vote. Tous ces modèles peuvent être utilisés pour extraire les situations préférentielles suivantes : l'indifférence, la préférence stricte, la préférence faible et l'incomparabilité.

Trois principaux types de problèmes de décision peuvent aussi être identifiés [Roy 1985]: le choix, le tri et le classement. Le problème du clustering a re cu moins d'attention que les autres problèmes de décision en AMCD. Ce n'est que lors des dix dernières années que plusieurs extensions des approches de clustering classiques sont apparues dans ce domaine. Une première distinction entre les approches de clustering classiques et celles d'AMCD a été donnée dans [Cailloux et al. 2007] grâce à l'utilisation d'une propriété appelée "dépendance aux critères". Cette propriété détermine si une approche de classification utilise les informations préférentielles supplémentaires données par les critères ou non. Cependant, aucune définition formelle du clustering en AMCD n'a été donnée. Alors que de nombreuses approches de clustering en AMCD regroupent des alternatives similaires, peu d'entre-elles utilisent uniquement les relations préférentielles pouvant exister entre les alternatives. En outre, l'utilisation de la notion d'indifférence pour regrouper des alternatives au lieu de celle de similarité a été d'abord proposée dans [Fernandez et al. 2010]. C'est pourquoi nous avons considéré qu'il serait intéressant d'établir un parallèle entre les notions de similarité et celle de préférence.

Nous avons d'abord examiné plusieurs exemples pour mettre en évidence les différences entre ces notions. Nous n'avons considéré que le cas de l'utilisation des relations de surclassement pour modéliser les préférences du décideur en raison du fait que dans le cas de l'utilisation des fonctions de valeur cette analyse devient trivial. A travers ces exemples, nous avons montré que, dans certaines situations, des alternatives qui sont non-similaires peuvent cependant être considérés comme indifférentes, tandis que dans d'autres cas, des alternatives qui sont semblables peuvent être vues comme l'une strictement préférée à l'autre ou même incomparables. Afin de déterminer si ces situations sont seulement des cas isolés ou s'ils apparaissent dans une proportion importante, nous avons en outre comparé plusieurs mesures de similarité et une mesure d'indifférence. Les résultats ont montré des différences significatives entre ces mesures. Dans au moins 25

Suite à l'analyse précédente, nous définissons formellement le problème de la classification en AMCD comme le processus de regroupement d'alternatives indifférentes tout en séparant celles qui ne le sont pas. En raison du fait que l'absence d'une relation d'indifférence peut donner lieu à une relation de préférence stricte, dans un sens ou l'autre, ou d'une relation d'incomparabilité, nous identifions également différentes sous-classes de classification non-supervisée en AMCD: le clustering non-relationnel, le clustering relationnel et le clustering ordonné. Le premier est équivalent à la définition précédente que nous avons donné, le second demande à ce que les relations entre les alternatives de deux groupes soient du même type, tandis que le dernier considère que la propriété de transitivité de la relation de préférence stricte entre les groupes soit également respectée.

Partie II: Résolution du problème de clustering en AMCD

Sur la base des nouvelles définitions du problème de classification non supervisée en AMCD, qui donnent certaines intuitions sur les propriétés qu'un bon résultat de clustering devrait afficher, nous continuons par la modélisation du problème de clustering en proposant plusieurs mesures de fitness afin d'évaluer la qualité d'un résultat de classification. Nous ne nous restreignons pas à une relation de surclassement particulière en utilisant d'une fonction r de crédibilité du surclassement qui peut lui être attachée. En utilisant cette fonction, nous avons ensuite systématiquement modélisé les relations d'indifférence, de préférence stricte et d'incomparabilité entre paires d'alternatives, ainsi que celles entre des ensembles d'alternatives. Plusieurs modèles de clustering sont alors proposés. La première considère qu'un certain nombre de groupes, chacun représentés par une alternative, doivent être trouvés, tandis que le second ensemble de modèles considère que n'importe quel nombre de classes peut être proposé. Dans les deux cas, nous définissons une série de trois mesures de la qualité des résultats de clustering.

Nous continuons avec le sujet de la résolution du problème de clustering en AMCD. Compte tenu des approches de clustering existants en AMCD, seules quelques-unes d'entre elles sont compatibles avec plusieurs des modèles que nous avons proposés. C'est pourquoi nous avons étendu l'algorithme classique des K-MEANS [MACQUEEN 1967] au clustering non-relationnel, ainsi que proposé plusieurs nouvelles topologies de carte pour les cartes de KOHONEN [KOHONEN et al. 2001] qui sont bien adaptées pour le clustering relationnel et ordonnée. Nous avons en outre proposé un nouvel algorithme qui est inspiré de la théorie des graphes, qui trouve d'abord les centres des régions denses dans le graphe d'indifférence et ensuite construit les clusters autour d'eux.

Les approches algorithmiques proposées sont ensuite validées en les exécutant sur un large ensemble d'instances. Ces données ont été créés afin que des alternatives de plus en plus contrastées soient générées à l'intérieur de chaque instance de problème. Nous avons également testé l'approche classique des K-MEANS qui utilise des mesures de similarité pour la construction de ses résultats, mais aussi une extension existante de cette approche de regroupement en AMCD [DE SMET and EPPE 2009]. Nous avons montré que les approches que nous proposons fournissent constamment des résultats de meilleure qualité que les deux autres approches.

Partie III: Exploration de grands ensembles de données préférentielles

Dans la troisième partie de la thèse, nous tournons notre attention vers l'application pratique du problème de clustering en AMCD. Dans l'introduction, nous avons soulevé la question de savoir si de grands ensembles de données, peuvent également être trouvés dans le domaine de l'AMCD. Bien que n'étant pas fixé pour le moment à un problème de décision réel, nous avons été en mesure de trouver un grand ensemble de données au sujet duquel de l'information préférentielle peut facilement être exprimée. Toutefois, afin d'explorer cette base de données, plusieurs questions doivent être abordées de manière préléminaire.

La première est liée à l'exploitation du résultat du clustering. Bien que certaines mesures existent déjà pour décrire les classes d'alternatives dans le contexte du problème de tri [PERNY 1998, BELACEL 2000, YU 1992, ROY and BOUYSSOU 1993], ils sont donnés soit directement par le décideur ou construits à partir de séries d'exemples d'affectation. Nous étendons ces mesures, tout en en proposant plusieurs nouvelles, qui permettent de décrire des ensembles d'alternatives. Il s'agit des profils centraux, délimitants et séparants. Un profil central peut être utilisé pour réduire un ensemble d'alternatives à une seule alternative, les profils délimitants peuvent être utilisés pour délimiter l'ensemble par rapport aux préférences du décideur, tandis qu'un profil de séparation peut être utilisé pour séparer deux ensembles d'alternatives qui sont ordonnés. Nous avons exploré plusieurs approches algorithmiques pour la construction de ces profils, que nous avons ensuite validées et comparées en utilisant les ensembles de données générées dans la deuxième partie de la thèse. Nous concluons que la sélection d'un représentant comme profil est dans la plupart des cas, une bonne stratégie. Les résultats de cette approche peuvent être considérablement améliorées grâce à l'utilisation d'approches méta-heuristiques.

Nous continuons en mettant l'accent sur l'extension des approches de clustering afin d'être en mesure de gérer de grands ensembles de données. Nous considérons trois grandes orientations pour la manipulation de grands ensembles de données [Jain 2010], ce qui réduit le problème d'origine à de plus petits problèmes, en mettant en équilibre la qualité de la solution, les temps d'exécution et les besoins de mémoire. Ce sont l'approche de diviser et conquérir, celle basée sur la progression et celle par échantillonnage. Plusieurs algorithmes sont proposés, tout en considérant pour l'instant seulement le problème de clustering non-relationnel. Les approches ont été testées sur des ensembles de données plus importants que ceux de la deuxième partie de la thèse. Les résultats ont montré une bonne performance pour toutes les approches proposées. Nous concluons que des recherches supplémentaires sur ce sujet seront à effectuer, bien que les approches proposées sont suffisantes pour explorer la grande base de données de l'étude de cas.

Dans le dernier chapitre de la thèse nous considérons comme étude de cas des données relatives aux pratiques de rejet de produits chimiques toxiques par les installations aux états-Unis d'Amérique [USEPAa]. Nous avons considéré un point de vue fictif sur la qualité de ces pratiques, compte tenu du fait que nous préfèrerons naturellement que les produits chimiques manipulés soient les moins toxiques possibles, que les quantités libérées soient minimales, ainsi que l'amélioration des pratiques d'atténuation. En raison de la taille de l'ensemble de données (plus de 20.000 alternatives), nous avons utilisé les approches de clustering qui peuvent gérer de grands ensembles de données. Nous avons ensuite illustré l'un de ces résultats, mettant en évidence un potentiel d'utilisation des profils centraux et limitant, à savoir celle de la classification des pratiques de libération sans connaître à l'avance les classes.

Nous concluons que ce travail a ouvert une nouvelle voie de recherche en AMCD: l'exploration des grands ensembles de données préférentielles en utilisant des approches de clustering. Nous avons montré qu'il existe des différences significatives entre les notions de similarité et d'indifférence, qui peuvent être utilisées pour grouper des objets ou des alternatives qui ne peuvent pas être distinguées. Sur cette base, nous avons défini formellement le problème de la classification non supervisée en AMCD, et nous avons proposé plusieurs modèles et algorithmes pour le résoudre, que nous avons testé et validé sur un large éventail de données générées artificiellement. Les résultats ont montré que ces approches sont en mesure de trouver de meilleurs résultats que les approches de clustering classiques. Enfin, nous avons envisagé d'appliquer notre travail au clustering de grands ensembles de données en AMCD, et nous avons montré que ce type de problèmes existe dans la vie réelle. En étendant les algorithmes de clustering pour la manipulation de grands ensembles de données, mais aussi en proposant plusieurs mesures qui peuvent être utilisés pour décrire les résultats de clustering, nous avons mis en évidence une application potentielle du problème de clustering pour explorer des grands ensembles de données. Néanmoins, de nombreuses possibilités pour la poursuite des travaux autour de ce sujet subsistent, qui seront abordé dans le futur.

Introduction

We cannot deny the fact that increasing amounts of information, or data, are rapidly becoming available to us in more and more diverse forms. The topic of data analysis is centred around the processing of this data in order to extract and highlight useful information.

Although many data analysis techniques exist, one natural approach of extracting this information is to place it into groups that display certain properties. These properties may be given directly (classification), or derived from grouping the objects using measures of similarity (clustering). Many grouping approaches have been developed, in a wide range of fields, where different expectations with respect to the resulting groups exist.

The field of Multiple Criteria Decision Aid (MCDA) studies the way in which people reach different decisions, and aims at helping in this process, through different methodologies and tools. The main difference from the information that is available in a data analysis context, is the existence of different types of preferential information on the data. This information is represented through a wide array of models, whose parameters need to be tuned precisely so that they reflect the subjective perception of a real person on the existing data objects, which in this context are called alternatives. Furthermore, it follows from the existence of this additional preferential information and the different models used to reflect the perspective of a person on the alternatives, that more diverse types of relations can be expressed between them, as opposed to those from a data analysis perspective.

The problem of clustering has also been tackled in the field of MCDA, however it has received little interest. Furthermore, while many MCDA clustering approaches use measures of similarity for constructing their results, only a few efforts have advocated the use of the relations that exist in a MCDA context. The reason behind grouping similar objects together is that we cannot discriminate between them and therefore do not have sufficient reasons for separating them. Is this also the case in MCDA? Would we be able to find certain situations where measures of similarity are not sufficient for grouping alternatives that cannot be distinguished, and if so, are these situations frequent? The answers to these questions will be among the first to be explored in this work, following an analysis of the differences between measures of similarity and those that are generally expressed in an MCDA context.

Based on the answers from the previous question, we would like to define the problem of clustering in MCDA using only the notions that are native to this field. Based on how the existing clustering approaches from MCDA fit with respect to this definition of clustering, we would also like to consider the extension of several seminal clustering approaches to the MCDA case. The question of validating these approaches is also an important one. While in a data analysis context we are able to find many resources on benchmarks for clustering and classification, which allow us to test new methods considering a wide range of potential structures of the data, in MCDA this is not particularly the case. Nevertheless, we may find several problem instances that illustrate the way in which certain MCDA approaches work, however these problem instances cannot be considered to characterise all potential problems that may be encountered in other cases. Although a few efforts have been made in this direction, we would like to additionally consider the topic of constructing MCDA problem instances that contain a large range of structures between the alternatives inside them, so that an empirical validation of the proposed algorithmic clustering approaches may be performed.

Another topic is centred around the cardinality of the datasets generally found in MCDA. Many decision aiding approaches focus on problems of small cardinality, which can be explored exhaustively by a real person. In many cases, this in a natural outcome of the real-life MCDA applications, which generally do not consider a very large set of decision alternatives. Nevertheless, although certain approaches may be able to consider larger datasets, many of them do not scale well due to complexity issues related to execution times and memory requirements. This is especially the case for those approaches that involve the use of mathematical programs in order to output an optimal result with respect to their considered fitness measures. Since clustering is seen as an exploratory data analysis technique, it may also appear to be useful when considering larger MCDA problem instances. Clustering may be used in order to reduce the larger datasets to a more manageable size, by extracting and condensing the information that is present in the original data. We will therefore like to address the question of finding MCDA problem instances of large cardinality, but also the extension of the proposed clustering approaches to them. Furthermore, we will explore different measures that may be used in order to analyse these datasets, considering the additional preferential information that is present in an MCDA context. The application of heuristic and meta-heuristic approaches will further be tackled as an alternative to using mathematical programs.

The thesis is divided into three parts, following the topics discussed in the previous three paragraphs. Chapter 1 contains an introduction to the field of Data analysis and a detailed overview of the problem of clustering in this field. In Chapter 2, the field of MCDA is introduced, along with the main concepts, constructions and decision problems. Finally, Chapter 3 presents a thorough state-of-the-art of clustering approaches in the field of MCDA, followed by a comparison between the measures used when clustering in a data analysis context, and those that naturally exist in MCDA. Lastly, we define the problem of clustering in MCDA based on the relations that exist in this field alone, compare it to the other classical decision problems and motivate its use further.

The second part follows with Chapter 4 illustrating several ways of modelling the clustering problem in MCDA, while only considering the outranking paradigm when constructing the relations between the alternatives. Comparing sets of alternatives as single entities is not as trivial as in the case of only pairs of alternatives, and so this topic is also addressed here. This is then followed by modelling of the different types of structures that may be sought when clustering in MCDA and the properties of these structures. In Chapter 5 the topic of solving the clustering problem is addressed, with the proposal of several extensions of the K-MEANS algorithm, and Kohonen's self-organizing maps. Additionally, another algorithm inspired from Graph Theory is proposed. Finally, in Chapter 6 the proposed algorithms are validated empirically over a large set of benchmarks. These benchmarks correspond to fictive decision problems and are constructed so that many different structures over the alternatives are built. In this way the simulations that are performed are representative for real applications.

In the third part of the thesis we turn towards the practical application of the studied problem. We begin by defining, in Chapter 7, several measures that may be used to characterise a clustering result, along with several approaches for extracting them from the clustering result. Furthermore, these approaches are validated empirically over the previously defined set of benchmarks. In the following chapter, Chapter 8, we tackle the topic of scalability, proposing several extensions of the proposed algorithms for handling large datasets. These approaches are also validated in the end of the chapter through tests over a set of benchmarks containing large problem instances. Finally, we apply all of these extensions to a case study containing a large number of alternatives, which we explore in Chapter 9.

A summary of the main achievements, perspectives for future work and concluding remarks follow in the end of the thesis.

Part I

On the problem of clustering in MCDA

Chapter 1

On clustering

Contents

Abstract

The work that will be presented in this thesis is centred around the topic of clustering in the field of MCDA. While this problem has received a limited amount of interest in this context, clustering in data analysis has been thoroughly studied. Hence, we begin by defining clustering as a data analysis technique, highlighting also the relation it has with a closely related problem, that of classification. Its role, its numerous applications and the diverse fields in which it has been used then follow. Due to this diversity, we then look at the different forms of information that can be used to describe an object and the means to compare them together. Finally, we briefly overview the many clustering approaches which have been developed so far and highlight some of the more commonly used ones.

1.1 An introduction to cluster analysis

1.1.1 Clustering and Classification

There is no doubt around the fact that we are living in an information age. Data related to almost every aspect of our lives is present and easily accessible in rapidly growing quantities. Only in 2011 the amount of digital information in the world was estimated to 1.8 zettabytes (1 zettabyte $= \sim 10^{21}$ bytes) [Gantz and Reinsel 2011]. This data is not only vast in quantity but also very diverse in nature. It can exist in the form of measurements and observations, describing different types of objects, it can show the dynamic of different processes or it can simply be in the form of text, images or videos. Due to the amount and diversity of this information, the need to automatically process and analyse it arises. For this reason the field of data analysis emerged.

Data analysis techniques are divided in two major directions [Tukey 1977]: exploratory or descriptive, where no prior information regarding the structure of the data is known and the aim lies in uncovering and understanding this structure and confirmatory or inferential, where certain hypotheses or models are assumed to exist and need to be validated. There are many methods of analysing data such as analysis of variance, linear regression, canonical correlation analysis, multidimensional scaling, principal component analysis and cluster analysis to mention a few. For more details on this topic, the interested reader is referred to [Tabachnick and Fidell 2006].

One of the most common techniques in data analysis is to group together objects that display similar properties, based on certain criteria. This is a very natural process and plays a very important role in our lives. In fact, one of the most basic activities of human beings, when trying to understand a new object or phenomenon, is to compare it with other known ones by identifying common characteristics [Anderberg 1973, Everitt et al. 2001]. From a methodological point of view, the process of grouping objects together in data analysis is split in two: classification and clustering.

Classification is a supervised grouping process that relies on *a priori* information regarding the groups in which to place the objects, which are called classes. This information may, for instance, be provided in the form of a subset of previously labelled objects [Duda et al. 2001]. Therefore we know the number of groups that need to be created and some of their characteristics, based on the labelled objects. The aim of classification is thus to construct a mechanism from this information in order to group the entire dataset, which is called the test set. Following the above division of data analysis techniques, classification is considered a confirmatory approach.

Clustering, on the other hand, does not benefit from any knowledge on the structure of the data. The purpose of clustering is to group objects together, based on the natural structure of the data, using measures of similarity. Hence, objects that are similar should be placed in the same cluster while objects that are dissimilar should be placed in different ones. The study of methods and algorithms that cluster data is called cluster analysis [Jain 2010]. The evident question of what is the natural structure of the data, or what can be considered as a good measure of similarity, is not easy to answer. It depends heavily on the nature of the data, the application domain and the expectations of the ones who wish to undergo this process of grouping it. Due to the fewer assumptions that are made on the structure of the data, clustering is considered to be an exploratory data analysis technique.

Another grouping process related both to classification and clustering, on which more and more focus is placed recently, may also be mentioned here. Semi-supervised clustering [Chapelle et al. 2006], like classification, uses a certain amount of information provided by an expert on the data, although this information is much more condensed, in the form of pair-wise constraints. These constraints correspond to judgements regarding the fact that two objects should be placed in the same cluster or not, but no information is given on the number of clusters or their characteristics.

The three previously defined data grouping processes are illustrated in Figure 1.1.

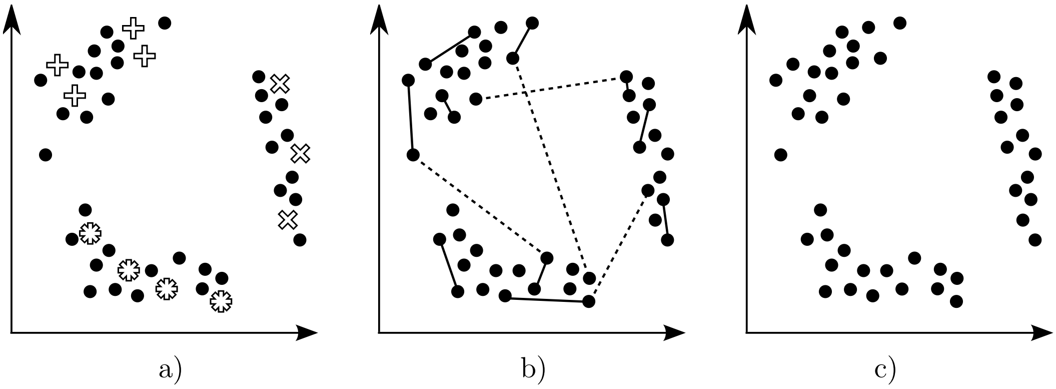


Figure 1.1: Illustrations of the information available on a two-dimensional dataset for grouping processes that are: a) supervised (classification), b) semi-supervised (constrained clustering) and c) unsupervised (clustering);

Each image depicts the information that is available when considering the same two-dimensional dataset for each of the presented grouping processes. In the first image we highlight the previously labelled objects that are available when classifying the data. The different shapes of the considered points denote the class in which they need to be included. It follows that three groups of objects will be extracted in the end. In the middle image constraints between different pairs of objects are illustrated, using continuous lines data points that should be placed in the same group and dashed ones for those that should not. Finally, in the last image no external information is given on the existing data points, therefore we are in an unsupervised clustering context.

1.1.2 Fields of application

Clustering has been applied to a large number of problems originating from a wide range of fields. A few of them have been mentioned in [Xu and Wunsch II 2008].

Engineering. There are many clustering applications in the field of engineering, ranging from biometric and speech recognition to signal processing, compression and noise removal. Some sub-fields include machine learning, pattern recognition, mechanical and electrical engineering, to name a few.

Computer sciences. This is the central field in which clustering is applied. From the existing applications we mention web mining, database analysis, information retrieval image segmentation *etc.*.

Life and medical sciences. We mention in this case applications to genetics, biology, palaeontology, psychiatry *etc.*. Clustering has received the most attention from these areas since its beginning and continues to do so even now.

Astronomy and earth sciences. Clustering can be used here to group, for instance, stars and planets, to partition different land regions and cities, but also to analyse different river systems. One of the very first clustering applications can be traced to the field of astronomy and more specifically to Hertzsprungwho, over a century ago, developed a diagram for grouping a series of observed stars using four types of information available on them [Hertszprung 1911].

Social sciences. We identify in this context applications in behaviour pattern analysis, the study of the history of languages, social networks, archaeological findings classification, criminal psychology, *etc.*.

Economics. In this case, clustering has been used to identify customer characteristics and purchasing patterns, to group firms together or identifying stock trends.

1.1.3 Clustering challenges

Due to the fact that clustering has been used in many different areas, which contain different types of information and have different expectations over what clustering should do, a number of difficulties also arise.

In [Jain and Dubes 1988] several important questions are mentioned, which need to be raised by those that wish to undergo a clustering process over the data they have at their disposal.

Data representation. From the very start, the way in which the data is represented is a fundamental factor that determines the output of the clustering process. In this context, the choice regarding the information that describes the objects needs to be a good one. If information that is not relevant is used, besides the fact that this increases the computational complexity of any clustering algorithm, the clusters will not reflect well the structure that exists in the data. There is no ideal way of selecting these features though and this needs to be guided by the domain knowledge.

Purpose of clustering. The selection of the information that characterises the objects is also tied to the purpose of the clustering process. In [Pampalk et al. 2003] we have an example highlighting this issue, where a dataset of 16 animals represented using 13 boolean indicators is used. These indicators are grouped into appearance indicators and behaviour indicators, therefore, shifting the relative importance between them leads to the extraction of very different sets of clusters. In the case where more emphasis was placed on the appearance indicators, the animals were grouped into mammals and birds. When more emphasis was placed on the behaviour indicators the animals were split into predators and non-predators. Both these results are meaningful and show valid ways of grouping these animals.

Similarity measure. Since clustering is defined as a process of grouping similar object and separating dissimilar ones, the choice of the similarity measure that is used directly affects the clustering result. Certain measures are more suited for different types of data and the choice of the similarity measure that is used should be carefully analysed.

Number of clusters. The issue regarding the number of clusters that need to be found is one of the most difficult ones in cluster analysis. Most methods that determine this number automatically have strong dependencies on the model that is used when clustering. In the general case, several runs of the selected clustering algorithm are done, using different values for this parameter and then the best value is selected, based on certain criteria. There are many other measures that give an estimation of the number of clusters that should be looked for in a particular dataset, from which we mention here the Bayesian Information Criterion [Schwarz 1978].

Cluster validity. Another important aspect when clustering is determining if the generated set of clusters is valid. Clustering algorithms will generally find a set of clusters even if there is no such structure in the data. In [Smith and Jain 1984] it is suggested that the user should first determine if the data has a clustering tendency before undertaking the process of clustering it. Cluster validity indices are split into three categories: internal, relative and external [Jain and Dubes 1988]. The internal indices determine how well the structure given by the clustering result fits with the data, using only the data itself. The indices based on the relative criteria compare different structures together and select the best one. External indices compare the clustering structure with a priori information in the form of labels, however if such information is available then classification would be better suited.

1.2 Measuring similarity

As previously mentioned, the notion of similarity is fundamental to the clustering process. There are many ways of measuring the level of similarity between two objects, which is closely tied to the nature of the objects themselves. We will begin, therefore, by describing the possible types of data that can form up an object and then move on to presenting very generally several frequently used similarity measures.

1.2.1 Data characteristics

A data object is described by a set of features, also called attributes or variables. Let's denote with $X = \{x, y, z, ...\}$ a set of n objects and with $F = \{1, 2, ..., m\}$ a set of m features. If $x \in X$ then the evaluation of x on the i^{th} feature is denoted with x_i.

An attribute can be continuous, discrete, or binary [Anderberg 1973, Jain and Dubes 1988]. A continuous attribute can take an uncountably infinite number of values, such as for example the measure of the distance between two cities. A discrete attribute can take only a finite number of values, like for instance the number of colors that can be represented in a digital image, while binary attributes are a special case of discrete attributes which have only two values, showing the presence or absence of that particular feature.

Each attribute also has a level of measurement, or scale [Stevens 1946], attached to it, which gives the relative significance of the evaluations on it. There are four types of scales: nominal, ordinal, interval and ratio. The first two are also considered qualitative scales, while the latter quantitative [Anderberg 1973, Jain and Dubes 1988].

Nominal attributes are normally represented by labels. Even if they have a numerical representation, no meaningful operation other then that of equality can be made (*e.g.* the names of people, colors of cars, *etc.*).

Ordinal attributes are also represented by labels, but an order between them exists. The difference operation between the values on this scale can still not be properly defined.

Interval attributes are represented by numbers and the difference operator has meaning in this case, although the ratio operator does not, due to the absence of a true zero (*e.g.* the Celsius temperature scale).

Finally, the ratio scale is the richest out of all the measurement scales and both the difference and ratio operators are well-defined on it (*e.g.* the quantity of a certain material).

The general way of measuring the level of similarity between two objects, which are defined by attributes on certain scales, is to use proximity measures [Xu and Wunsch II 2008]. These measures can be defined either by using a dissimilarity function D, also called a distance function, or using a similarity function S. These functions encode both notions of similarity and dissimilarity. A large value for D between two objects corresponds to the fact that the objects are dissimilar while a small values means they are similar. The reverse judgements are made for the similarity function S.

In real-life applications an important issue, which may arise, is that we may have objects with missing evaluations on certain attributes. Several strategies to overcome this difficulty exist, although none of them can be considered the best approach, the performance of each depending on the actual application.

The simplest approach, for dealing with missing data, is to discard the records containing missing data altogether, however this will be very ineffective on datasets containing many missing evaluations. A second approach, when comparing two objects, only uses those attributes on which both are evaluated, while a last one replaces any missing evaluation on a particular attribute with the median evaluation from all the objects in the dataset. Dixon [Dixon 1979] suggested that the second of the above approaches was the best among them. For more details on the topic of missing data the interested reader is referred to [Little and Rubin 2002].

1.2.2 Measures on quantitative scales

For comparing objects defined by attributes on quantitative scales, the Euclidean distance is the most commonly used. This distance is a particular case of the Minkowski distance, depicted in Equation (1.1), where we fix p to the value of 2.

$$D(x,y) = \left(\sum_{i \in F} |x_i - y_i|^{\frac{1}{p}} \right)^p, \forall x, y \in X. \tag{1.1}$$

Considering the space defined by n features on continuous scales, the Euclidean distance is used for finding hyper-spherical clusters. Duda *et al.*[Duda et al. 2001] have shown that the clusters found using this distance are invariant to translations and rotations in the feature space, although this tends to give more importance to attributes with higher range or variance. One way to overcome this difficulty is to scale each attribute so that it has a zero mean and unit variance.

Other commonly used particular cases of the Minkowski distance are the Manhattan distance ($p = 1$) and the Supremum distance ($p \to \infty$).

For finding hyper-ellipsoidal clusters, the Mahalanobis distance, which is presented in Equation (1.2), can be used.

$$D(x,y) = (x-y)^T \cdot C^{-1} \cdot (x-y), \forall x, y \in X. \tag{1.2}$$

$(x-y)^T$ represents, in the formula above, the transposed vector containing the difference between the evaluations of x and y on each attribute, while C is the matrix containing on its diagonal the covariance within each cluster. Computing this matrix makes the use of the Mahalanobis distance impractical for large datasets.

Other distance measures shift the emphasis from measuring the magnitude of the differences between two objects on each attribute towards measuring the number of attributes on which these objects have different evaluations. In this case, for instance, we find a distance measure that is derived from the Pearson correlation coefficient [Kaufman and Rousseeuw 1990]. This type of distance is used very often in clustering micro-array gene expression data [Eisen et al. 1998]. In this category of distance measures falls also the cosine similarity, presented in Equation (1.3), which is the most popular measure used for document clustering.

$$S(x,y) = \frac{x^T y}{\|x\|\|y\|}, \forall x,y \in X. \tag{1.3}$$

In the above equation, $\|.\|$ is used in order to represent the Euclidean norm, therefore $\|x\|$ represents the Euclidian distance between the origin of the feature space and x. The vector of x is transposed, in the numerator of this equation, so that the multiplication with the vector of y leads to a single value.

1.2.3 Measures on qualitative scales

When considering qualitative scales, the difference between the values, that encode the levels on these scales, has no meaning and so different measures need to be used. A further distinction is made between binary scales and scales with more than two levels.

Binary attributes show the presence or absence of a particular attribute, or feature, usually denoted through the 1 and 0 values respectively. Kaufman and Rousseeuw [Kaufman and Rousseeuw 1990] further classify the binary attributes in two: symmetric and asymmetric. If a binary attribute is symmetric then both values are equally important, while if this attribute is asymmetric then only one value, usually the 1, is considered to be informative.

Let's consider $n_{00}(x,y)$ to be the number of attributes on which objects $x,y \in X$ have a value of 0, $n_{11}(x,y)$ the number of attributes on which they both have a value of 1 and $n_{10}(x,y)$ and $n_{01}(x,y)$, the numbers of attributes on which one of them has a value of 0 and the other a value of 1. We present the general formula for the similarity measures between objects defined on binary attributes as:

$$S(x,y) = \frac{n_{11}(x,y) + \alpha n_{00}(x,y)}{n_{11}(x,y) + \alpha n_{00}(x,y) + \beta(n_{10}(x,y) + n_{01}(x,y))}, \forall x,y \in X. \tag{1.4}$$

The formula above contains two additional variables: α and β. The first is used to denote the fact that the binary attributes are symmetric ($\alpha = 1$) or asymmetric ($\alpha = 0$). For $\alpha = 1$, we mention several known similarity measures for β equal to 1

[Zubin 1938], 2 [Rogers and Tanimoto 1960] and 1/2 [Sokal and Sneath 1963]. In the case of asymmetric binary attributes there have also been several measures defined for β equal to 1 [Jaccard 1908], 2 [Sokal and Sneath 1963] and 1/2 [Dice 1945].

When dealing with qualitative scales, which have more than two levels, two approaches for measuring similarity can be identified. The first involves transforming each attribute into several binary attributes, one for each level in the original scale [Anderberg 1973, Kaufman and Rousseeuw 1990]. This approach has the potential of creating a very large number of attributes and so a second approach is to use a matching coefficient [Everitt et al. 2001, Kaufman and Rousseeuw 1990], as defined further:

$$S(x,y) = \frac{1}{m}\sum_{i\in F} S_i(x,y), \forall x,y \in X \tag{1.5}$$

where

$$S_i(x,y) = \begin{cases} 0 & \text{, if } x_i \neq y_i \\ w & \text{, if } x_i = y_i \end{cases} \tag{1.6}$$

The parameter w is usually fixed to the value of 1, although it can also be given a different value in some cases. For example, w can be greater than 1 if the attribute has a large number of possible values, therefore a matching is very unlikely to occur.

When dealing with objects defined on both qualitative and quantitative scales, the above measure can also be used to compute the similarity between them, by scaling S_i, for those $i \in F$ that are defined on quantitative scales, to the $[0,1]$ interval [Gower 1971].

1.3 Overview of clustering approaches

As previously mentioned, many clustering approaches have been developed over the years, each tailored specifically in the context of the application field and the type of available data.

Clustering approaches can be split in two major groups: partitioning and hierarchical. These groups are not disjoint, certain algorithms belonging to both, nor complete, as other groups of clustering algorithms may further be identified. Nevertheless, most clustering approaches may be covered by the two mentioned classes of algorithms, hence the overview that follows will be structured around them.

1.3.1 Partitioning clustering approaches

In the case of partitioning clustering approaches, the original set of objects X is split completely into several disjoint sets, K, *i.e.* a partition.

Each problem can be characterised using the number of clusters that need to be retrieved and also through a fitness function which reflects the quality of the clustering result. There are many fitness functions that can characterise a set of clusters, although the most common one is the squared Euclidian distance between the objects in each cluster and the centroid of that cluster. A centroid is an object that is placed in the centre of a cluster, either in the form of an existing object, or an artificially constructed one. This function is used by one of the most well-known partitioning clustering algorithms, which will be descried a little further.

A presentation on the complexity of clustering problems can be found in [Gonzalez 1982]. There, it is shown that most partitioning clustering problems are NP-hard, *i.e.* they cannot be solved in a polynomial time with respect to the size of the problem, nor have a solution validated within this time. Therefore, using exact algorithms to solve these problems is impractical, especially considering larger problem instances. For a broader presentation of the issues related to computational complexity the reader should refer to [Garey and Johnson 1979].

One of the most well-known partitioning clustering algorithms is the K-means algorithm, which was independently discovered in different fields by Steinhaus [Steinhaus 1956], Lloyd [Lloyd 1982], Ball and Hall [Ball and Hall 1965] and MacQueen [MacQueen 1967].

The algorithm can be summarised in the following way: a set of centroids is generated, usually randomly; the objects in X are assigned to the centroid that is closest to them, therefore constructing a partition; a new set of centroids is then generated from the average evaluations of the objects from each set inside the partition; all these steps, except the first, are then repeated until a certain amount of time, or a certain number of iterations have passed, or if the centroids no longer change significantly across consecutive iterations. The general structure of the algorithm is also presented in Algorithm 1.

Algorithm 1 K-means algorithm;

```
$C \leftarrow$ InitializeCentroids()
while not BreakCondition() do
    $K \leftarrow$ GenerateNewPartition($X, C$)
    $C \leftarrow$ GenerateNewCentroids($K$)
return K
```

Since more than 50 years ago, when K-means was first proposed, many variations have been created, in order to deal with some of the shortcomings of this method. Some of the more important ones pertain to the choice of the number of clusters the algorithm should retrieve, the convergence towards locally optimal solutions and the fact that the resulting clusters are hyper-spherical in shape.

In order to deal with the first issue, a number of heuristics have been discussed in [Tibshirani et al. 2001]. The most general approach is to run the algorithm several times, looking for a different number of clusters each time and then selecting the result which is the best with respect to certain criteria.

The second issue focuses on the convergence towards the globally optimal solution. Although the algorithm was shown to converge [Selim and Ismail 1984], it will most of the time only find locally optimal solutions. For escaping local optimality, the same approach of running the algorithm several times can be used. As the algorithm is very dependant on the initialization step, different strategies for creating the initial centroids can be used. The method of Forgy [Forgy 1965] selects the centroids randomly from the existing data points while MacQueen [MacQueen 1967] uses an on-line version of this approach. Kaufman and Rousseeuw [Kaufman and Rousseeuw 1990] employ a different strategy. The first centroid is selected as the most centrally located data point, while the following centroids are selected as the data points that decrease most the cost, or fitness function. These methods have been tested on three datasets [Peña et al. 1999] from the Machine Learning Repository [Asuncion and Newman 2007], concluding that the approach of Kaufman and Rousseeuw is the most efficient one when considering the quality of the solution, while that of MacQueen gives a faster convergence of the algorithm.

Since K-means most commonly makes use of the the Euclidean distance, the clusters are hyper-spherical in shape, however other extensions using different distance measures have also been developed. The Mahalanobis distance was used in order to find hyper-ellipsoidal clusters in [Mao and Jain 1996], while the Itakura-Saito distance was used for speech processing in [Linde et al. 1980]. The Manhattan distance has also been used in [Kashima et al. 2003], while the family of Bregman distances have been applied in [Banerjee et al. 2005].

K-means is also known to be sensitive to outliers, which represent objects that are far away from any cluster centroid and should normally not be included in any cluster. Due to fact that the distances between them and the centroids are large, adding such objects to a cluster, or removing them, will have a significant effect in the construction of the centroid of that cluster.

Other extensions of K-means add heuristics regarding the size of the clusters, or with respect to operations of merging and splitting of clusters. ISODATA [Ball and Hall 1965] and FORGY [Forgy 1965] have been used in pattern recognition. Eschrich has increased the speed of the algorithm by replacing group examples with their centroids [Eschrich et al. 2003]. Dunn proposed Fuzzy C-means [Dunn 1973], which can assign an object to several clusters at once, though this approach does not provide a partition as the end result, but clusters that overlap. A variant of K-means has also been proposed by [Steinbach et al. 2000], called Bisecting K-means, which builds a hierarchy of nested clusters. A fast version of K-means that requires only a single pass and does not need all the data to be loaded into memory was developed by Bradley in [Bradley et al. 1998]. X-means [Pelleg and Moore 2000] finds automatically the number of clusters, while in [Kaufman and Rousseeuw 2005] a variant of K-means, which can deal with data on qualitative scales, K-medoids, has been proposed. Finally Kernel K-means [Schölkopf et al. 1998] can be used to detect arbitrarily shaped clusters by using a kernel similarity function.

1.3.2 Hierarchical clustering approaches

Hierarchical clustering approaches also construct partitions on the set of objects, however they do not restrict themselves to a certain number of clusters and thus build a hierarchy of nested partitions [Everitt et al. 2001, Jain and Dubes 1988].

This structure is normally represented using a binary tree, or dendrogram. The leaves of the tree represent the data objects, while the root has all the objects grouped into a single cluster. The nodes in-between show how the objects have been grouped together at each level of the hierarchy. The height at which each level in the dendrogram is placed may also highlight a certain feature related to the fitness of that particular partition. Finally, any of the constructed partitions may be retrieved by cutting the dendrogram at a certain level. Figure 1.3.2 depicts one such example of a dendrogram, as well as its Venn diagram representation [Venn 1880].

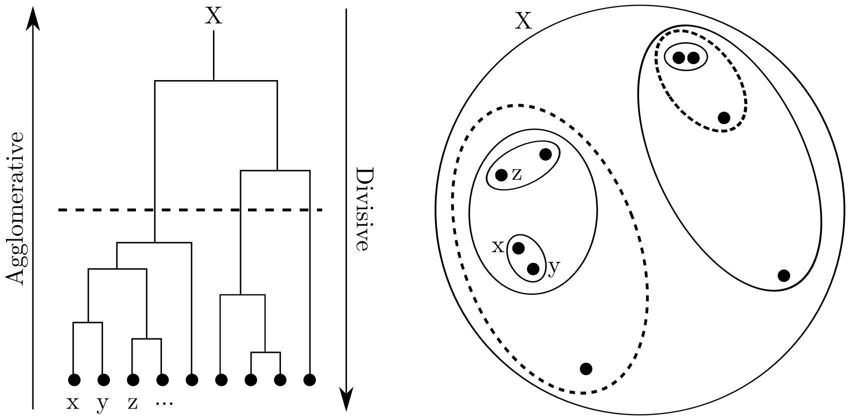


Figure 1.2: A dendrogram (left) and Venn diagram (right) of a 9 objects dataset;

Two main approaches to build the hierarchy of nested partitions exist: agglomerative and divisive. Agglomerative hierarchical clustering approaches build the dendrogram from the bottom up, while divisive hierarchical clustering approaches construct it from the top down.

Agglomerative hierarchical clustering approaches start with a partition containing only singleton clusters and repeatedly merge the pair containing the closest ones. The choice of the distance measure used greatly influences the clustering results. In addition, there are also a number of ways of computing a distance between groups of objects, min, max, or average being a only a few of them. In order to deal with complexity issues that arise from recomputing the distance between a newly formed cluster and the rest, several heuristics have been proposed. These heuristics, also called cluster linkages are single linkage, complete linkage, group average linkage, weighted average linkage, centroid linkage, median linkage and Ward's linkage. More details on them can be found in [Lance and Williams 1967].

One drawback associated with agglomerative hierarchical clustering is that poorly made decisions to merge clusters during the early steps cannot be undone later and so the clustering results that are higher in the dendrogram tend to reflect with less accuracy the actual structure of the data.

Divisive hierarchical clustering approaches start with a single cluster containing all the objects and repeatedly divide the most heterogeneous cluster (the one with the largest diameter) into two new clusters. These approaches are less popular than the agglomerative ones, due to certain complexity issues. While agglomerative hierarchical clustering approaches needs to consider $\frac{n(n-1)}{2}$ possible operations in the beginning, divisive approaches have $2^n - 1$ such possible operations to consider.

One advantage of using the divisive approach lies in the fact that the clustering results that are higher in the dendrogram reflect the actual structure of the data better than those from agglomerative approaches, due to the fact that they are generated at the early steps of this process [Kaufman and Rousseeuw 1990].

In order to improve on the complexity issues, DIANA was proposed in [Kaufman and Rousseeuw 1990]. This algorithm uses a heuristic approach in order to consider only a subset of possible partitions that derive from the current one at each step. Having selected the cluster that will be split, the algorithm places all the objects inside this cluster into the first newly formed cluster, while the second one is set as empty. Then a series of steps are performed, which iteratively take an object from the first cluster and place it in the second, based on the fact that it is farthest away from the other objects in the first cluster and closest to the objects in the second cluster. This series of steps ends when the object that is selected to be moved from the first cluster to the second is in fact closer to the remaining objects in the first cluster rather than towards those in the second.

Other hierarchical clustering approaches try to address the scalability issues that come with the classical approaches. BIRCH [Zhang et al. 1996] summarizes the data through so called "clustering features" (CF), which capture the important information of the original data, but reduce storage requirements. An agglomerative hierarchical clustering algorithm is then used directly on these CFs. BRICH also uses a pre-processing step to remove noise and outliers from the data.

CURE [Guha et al. 1998] is an approach in-between single link and centroid link agglomerative clustering. It uses a set of well-scattered objects as representatives for a cluster. They are further brought towards the centroid of the cluster using certain parameters, in order to deal with the problem of outliers. The algorithm initially samples the data and then splits it into a number of partitions of equal size. Each partition is then pre-clustered, in order to reduce the computation time and then an agglomerative hierarchical clustering approach is used. Small clusters are considered to be outliers and then the original data is split using the clusters representatives. Another approach very similar to CURE is ROCK [Guha et al. 2000], where the main difference lies in the fact that the objects that are clustered are defined on qualitative attributes and not quantitative.

CHAMELEON [Karypis et al. 1999] is another hierarchical clustering algorithm which has two steps. The first uses the k-nearest-neighbour graph and splits it into initial clusters using a minimal edge cutting algorithm. The second step consists in an agglomerative clustering of these initial clusters into the final ones. Another graph representation used in hierarchical clustering is the DEALUNAY triangulation, the dual of the Voronoi diagram [Okabe et al. 2000]. Cherng and Lo [Cherng and Lo 2001] used this representation in order to construct a hyper-graph that was then used inside a clustering algorithm similar to CHAMELEON.

In [Leung et al. 2000] a parallel to visual systems is made, where each point is considered as a light source and a cluster as a blob. This image is blurred until everything is contained in one single blob, therefore the approach is an agglomerative hierarchical one. In [Li and Biswas 2002] the agglomerative hierarchical clustering algorithm was extended to work on both qualitative and quantitative data, by using the Goodall similarity [Goodall 1966].

Another approach uses an unsupervised decision tree in order to perform divisive hierarchical clustering, by splitting the clusters according to a single feature, which is selected based on several criteria from information theory [Basak and Krishnapuram 2005]. Other methods worth mentioning are the Agglomerative Likelihood Tree and the Markov Chain Monte Carlo-based method [Castro et al. 2004]. Also parallel hierarchical clustering algorithms have been proposed in [Li 1990, Olson 1995, Dahlhaus 2000, Rajasekaran 2005].

1.3.3 Other approaches

Along the two major classes of clustering methods described so far, several other points of view regarding the notion of clustering exist.

Density based clustering

Clustering can also be defined as the process of finding high density regions in the feature space, which are separated by regions of low density. In this definition the notions of neighbourhood and connectivity play an important role.

We mention in this category the JARVIS-PATRICK algorithm which evaluates the similarity of two objects based on the number of common neighbours they have, where the neighbourhood of an objects is given by an area of a certain radius around that object [Frank and Todeschini 1994]. A similar algorithm is DBSCAN [Ester et al. 1996], which estimates the density of a region through the kernel density estimation method. The performance of these two methods depends on the parameters related to the size of the neighbourhood and its density in order to be included in a cluster.

Other methods use probabilistic mixture models to estimate the density function. The Expectation Maximization algorithm [Dempster et al. 1977] is often used to infer the parameters of the mixture models. In order to improve the mixture models, several Bayesian approaches have been created such as the Latent Dirichlet Allocation [Blei et al. 2003], Pachinko Allocation model [Li and McCallum 2006] and undirected graphical model for clustering [Welling et al. 2004].

Density-based methods have a number of limitations when it comes to clustering high dimensional data. In such cases the probability of having the data points close to each other is lower, therefore distinguishing between high and low density regions becomes more difficult. The answer to this issue comes from the sub-field of subspace clustering, namely the CLIQUE algorithm [Agrawal et al. 1998]. This algorithm selects a subset of representative dimensions, according to some criteria and finds high density regions in this subspace.

Graph theoretic clustering

Graph theoretic clustering, also called spectral clustering, models the similarity between objects using a weighted graph and therefore these approaches draw inspiration from Graph Theory. Clustering in this context splits the graph in such a way that the sum of the weights of the edges that are removed is minimized.

The minimum cut algorithm can be used in this context, though it often leads to imbalanced cluster sizes. A constraint regarding the size of the clusters was adopted in the Ratio cut algorithm [Hagen and Kahng 1992], while one of the fastest minimum cut algorithms was developed by Karger in [Karger 1993]. An efficient approximative algorithm for graph cutting, called the Normalized Cut algorithm, which includes also a constraint related to the size of the clusters, was proposed in [Shi and Malik 2000]. Meila and Shi used a Markov Random Walk approach in the Modified Normalized Cut algorithm [Meila and Shi 2001] which is able to find automatically the number of clusters.

Ng proposed another approach that uses data representation derived from the normalized eigenvectors of a kernel matrix [Ng et al. 2001]. Finally, in [Belkin and Niyogi 2001] Laplacian Eigenmap is proposed, which derives the data representation from the Laplacian graph.

Summary

We conclude the introduction to the problem of clustering, in the original context in which it was formulated, that of data analysis.

In this context, clustering is seen as a process of grouping similar objects together and separating those that are not. It is related to the problem of classification, however, it uses less a priori information and is more exploratory in nature, while classification tries to validate a certain model over the existing data.

As an exploratory data analysis technique, clustering can and has been used in many fields, where different types of information are available. Therefore, many ways of defining similarity between these objects have been constructed and also many different clustering approaches. Some of the most common similarity measure and clustering algorithms have been herewith overviewed.

Further on, we will focus on the field of Multiple Criteria Decision Aid and on defining clustering in this context. Due to the fact that other types of measures than similarity are used in this field, clustering will also be defined in relation to them. We then later take inspiration from several classical approaches to clustering in data analysis, namely K-MEANS from partitioning clustering and a few methods from graph theoretic clustering and extend them to the MCDA context.

Chapter 2

Multiple criteria aid for decisions

Contents

Abstract

We now turn our attention to the field of Multiple Criteria Decision Aid, the context in which we will study the problem of clustering. We begin by presenting the motivations for which this field has emerged, followed by focusing on the topic of decision aiding and the way in which it can be modelled. We continue by highlighting the major approaches of modelling and extracting the preferences of one or several persons. Finally, we present the main types of decision problems that can be stated, along, with an overview of the specific methods that have been developed in order to solve them.

2.1 Multiple criteria decision aiding

2.1.1 A brief introduction

We are faced on a daily basis with a multitude of situations which require us to make a decision, from choosing what to wear based on what we have planed to do during a particular day, to deciding on a way to manage our savings. It is quite clear that these situations can be very diverse and as such we go about solving them in different ways. Not only the differences between situations leads us to treat them differently, but also our personality, knowledge and values.

Most of the times, these situations are trivial, in which case we may even resort to making a random decision. This potentially irrational approach is completely motivated by the possible low emotional implication from our part, the low impact of this decision and the simplicity with which we can approach it.

In certain cases, the need to analyse the possible choices and their impact is very important. Thus, we need to make a rational decision, the process of making it should be transparent and the result well-motivated. For instance, this may be the case when considering an international company that plans to expand by constructing a new headquarters. The decision needs to be made considering different available sites from a number of countries. In this case, the decision to select one of those sites needs to be very well motivated, based on the positive and negative aspects of each site with respect to the aims of that particular company. Additionally, this decision might need to be made not by a single person, but by multiple stakeholders [Bana e Costa et al. 2001]. In cases such as the one previously described, the need to use scientific methods to describe, analyse and propose possible solutions arises.

In general, there are two actors in a decision aiding process: the decision maker (DM) and the analyst. The first represents the person who has to make a decision. This person may not be familiar with the methodology that can be used in decision aiding and so the second actor, the analyst, is tasked with helping the DM in reaching a decision, by making use of different formal methods and tools. The final decision is up to the DM and should therefore reflect his preferences only. It is worth mentioning that the DM may be a group of people with possibly conflicting interests between which a consensus needs to be made. Finally, the DM may already have extensive knowledge of the methods used for aiding him in making a decision and so he can play also the role of the analyst.

One of the main difficulties in decision analysis is the fact that most problems are multidimensional in nature [Roy 1988]. Let's consider the following example:

Example 1 *Thierry wants to buy a car. His decision comes down to car A and car B. On the one hand, Thierry prefers car A because it is more powerful than car B, but, on the other hand, car B is cheaper than car A. As Thierry would also prefer a cheaper car, he is faced with a decision problem.*

It is quite clear that making a decision between the two cars is not trivial and depends on what is more important to Thierry, the performance of the car or its price. In order to solve this problem, a multiple criteria approach needs to be used, considering the multidimensional character of the potential decision.

Multiple Criteria Decision Aid (MCDA) therefore aims at helping a DM to make a decision and studies decision problems where several points of view have to be considered. In no case will it try to force the DM to make a decision and it can range from simply structuring the decision problem to proposing a recommendation.

Multiple Criteria Decision Aiding can be said to have begun as far back as the 18th century, where the Marquis DE CONDORCET started applying mathematics in social sciences. In 1785, he wrote on decision making in presence of multiple voters [CONDORCET 1785]. The foundation of this field was set however later, in the 20th century, with SAMUELSON's theory of revealed preferences [SAMUELSON 1938], the beginning of game theory [VON NEUMANN and MORGENSTERN 1944] and social choice [ARROW 1951].

An important step forward came with SIMON's theory of bounded rationality [SIMON 1957], which states that real decision problems contain different factors that limit the capability of the DM to make a completely rational decision. The DM has a bounded rationality and will make a decision based only on his limited knowledge and cognitive capacity. From a mathematical point of view, the result of the decision process may not be the optimal one but one with which the DM is satisfied.

The first methods started appearing a bit later, with the creation of outranking methods by ROY in 1968 [ROY 1968]. Later still, in 1976, KEENEY and RAIFFA extended value theory to the multidimensional case [KEENEY and RAIFFA 1976].

Two main philosophical directions can further be identified, from the point of view of the output of the decision process. These are decision aiding and decision making. The first aims at giving only a recommendation, while the latter focuses on finding the optimal solution. We will concentrate in what follows on decision aiding.

2.1.2 Decision aiding

The alternatives, or potential decisions, are the fundamental objects in the MCDA field. These may be real entities or concepts that can be a part of the final recommendation or they can be fictitious, only aiding in the process of constructing this recommendation.

The alternatives are defined on several attributes, which need to reflect all the characteristics that are important to the DM when making his decision.. Each attribute may be defined according to any of the four measurement scales, which have been mentioned in the previous chapter.

The way in which two evaluations on one attribute are compared needs to be in accordance with the perspective of the DM. Therefore, a criterion [BOUYSSOU 1990] is constructed from each attribute. In [Bouyssou et al. 2006] a criterion is defined as a dimension on which a preference model can be associated. In other words, the evaluations of the alternatives on the attributes that define them is not sufficient, additional preferential information from the DM being necessary in order to properly compare them.

We keep the notations from the previous chapter and denote the set of alternatives with X, its cardinality with $n > 1$, the set of criteria, also called a family of criteria, with F and its cardinality with $m > 1$.

The most common decision problems have been identified in [Roy 1985] as: choice, sorting and ranking.

The choice problem, denoted with P_α, aims at finding one or several of the alternatives which can be considered as the best ones, based on the preferences of the DM.

The sorting problem, P_β, tries to place all the alternatives into predefined categories, which may be ordered or not. This is done through a series of rules which determine the assignment and therefore can be easily compared to the process of classification from data analysis.

The ranking problem, or P_γ, aims at ordering the alternatives based on the preferences of the DM, either completely or partially, also allowing for ties between them to occur.

Among other types of problems, less explored however, we find clustering, which is the topic on which we will focus in this work. A more detailed resource on the most common decision problems in MCDA can be found in [Bana e Costa 1996].

2.1.3 Modelling a decision aiding process

As part of the decision aiding process, several steps can be identified. These are the problem situation, the problem formulation, the selection of an evaluation model and the extraction of a final recommendation [Tsoukiàs 2007]. They may not be all explored during each decision aiding process, nor in any particular order,

The problem situation

In general, one of the first steps that need to be taken, when considering a decision problem, is to situate the problem.

Some of the main questions that should be answered during this step are [Tsoukiàs 2007]:

- who has a problem?
- why is this a problem?
- who decides on this problem?
- what are the stakes?
- *etc.*

Although the DM may already consider some of these questions before committing to making a decision, the answers to these questions should also be explored when employing the aid of an analyst. During this step, the set of actors which take part in the decision process, the stakes each of them brings and the resources they commit on their stakes need to be identified. Exploring this step is also beneficial in helping to eliminate any misunderstandings between the DM and the analyst.

Formulating the problem

While the previous step has a more descriptive purpose, the step of formulating the problem has a more constructive character. At this point the analyst needs to reduce the real problem to a formal one. If the model is accepted by the DM, then different methods can be used afterwards in order to solve the decision problem.

At this point, the alternatives, or potential decisions, need to be identified, along with the set of points of view on them, *i.e.* how the alternatives are evaluated, and finally the objective, or what the DM expects to find at the end of this process.

Several decision aiding approaches will stop after this step, as they consider that understanding the problem is equivalent to solving it. Other approaches consider the formulation of the problem as given and will only consider the next steps. Regardless of these approaches, the step of proposing a problem formulation is very important in the decision aid process and is used further to construct the evaluation model.

The evaluation model

This step represents the part that most classical decision aiding approaches focus on. With the alternatives defined and the criteria, which express the features of the alternatives that are important to the DM, constructed, different models and approaches can be employed in order to find a formal solution.

A certain model, which is able to capture the preferences of the DM on the set of alternatives, needs to be first selected and tuned accordingly. This step is very important as the final result will be strongly reflected by this choice. The model is then used to construct a global relation between the alternatives which is then exploited by a selected method in order to provide a solution.

An important aspect when selecting the evaluation model is to have this model emerge naturally from the problem formulation and not by first choosing a specific method and then formulating the problem around it [Tsoukiàs 2007].

The final recommendation

The final recommendation is constructed by translating the formal solution from the previous step into a clear and easily understandable recommendation from the point of view of the DM.

Several aspects need to be validated when constructing this result:

- the analyst needs to be sure that the model is formally correct;
- the DM needs to be sure the he understands the model and that it reflects his preferences;
- the recommendation needs to be accepted or not by the DM, following which the reasons for this decision should be properly understood.

2.2 Modelling preferences

Finding a preference model and tuning it so that it is able to capture the point of view of the DM is a crucial step in the decision aiding process. We will focus in this section on the two major approaches to modelling preferences: utility function and outranking relations.

Before detailing them, it is important to state that regardless of the chosen model, one of the following binary relations may be extracted for any pair of alternatives [Roy 1996, Greco et al. 2010]:

- Indifference (I): a reflexive and symmetric relation that appears between alternatives that are considered equivalent by the DM;
- Strict preference (P): a non-reflexive and asymmetric relation between an alternatives that is clearly favoured by the DM instead of the other;
- Weak preference (Q): a non-reflexive and asymmetric relation between an alternative which is either strictly preferred or indifferent to another;
- Incomparability (R): a non-reflexive and symmetric relation between two alternatives that cannot be placed in either of the previous relations.

2.2.1 Value functions

The Multi-attribute Value Theory (MAVT) [Keeney and Raiffa 1976] avoids the difficulty of the multidimensional character of the alternatives by creating a unique criterion that aggregates all the criteria together. In this way all the dimensions that define an alternative are reduced to a single one, the vector of evaluations of the alternatives to a single value, or score.

This approach assumes that the preferences of the decision maker can be modelled as a weak order over the set of alternatives X and models the complete and transitive binary relation $\succeq$ on X using an overall value function U [Roy 1971b, Keeney and Raiffa 1976] in such a way that:

$$x \succeq y \iff U(x) \geqslant U(y), \quad \forall x, y \in X.$$

The most common overall value function is the additive form [Fishburn 1964, Fishburn 1965, Fishburn 1970], which is equal to the sum of all the marginal value functions u_i:

$$U(x) = \sum_{i \in I} u_i(x_i), \quad \forall x \in X,$$

The marginal value function u_i determines the score given to an alternative on criterion $i \in F$.

One particular and well studied form of the overall value function is the weighted sum, where we associate with every criterion $i \in F$ a weight w_i:

$$U(x) = \sum_{i \in F} w_i.x_i, \quad \forall x \in X.$$

The weights, in this case, have the role of trade-offs between criteria, *e.g.* a low value on one criterion may be compensated by a high value on another, if first criterion has a lower weight than that of the second.

Let us notice that the additive form makes an important assumption on the independence of the criteria. If the criteria are not independent, we may use different models, like for instance the Choquet integral [Choquet 1953] or the Sugeno integral [Sugeno 1974], which also consider possible interactions between criteria. These interactions can be positive, when, for instance, the impact of one criterion is reinforced by the impact of another, or negative, when two criteria are partially redundant.

In comparison to the additive function, the Choquet integral and the Sugeno integral have more parameters in the form of weights, which are called capacities for the first and fuzzy measures for the second.

Several common issues exist when expressing preferences using value functions. The first lies in the selection of the value function and its form, *i.e.* its parameters. This is a difficult task and different ways to elicit these parameters have been proposed. We will mention several of them later in this section. A second issue revolves around the conversion of qualitative scales into quantitative ones. This conversion is needed due to the way in which value functions are defined. For example, transforming the scale $\{good, medium, bad\}$ into $\{3, 2, 1\}$ would impose the fact that a *good* evaluation is considered three times better than a *bad* evaluation, which may not be the case.

One advantage of representing the preferences of the DM in this way is the fact that it produces a complete weak order on the set of alternatives. This allows us to compare the alternatives together with ease. We will always be able to construct either a relation of preference (P) between two alternatives, if one has a higher score than the other, or one of indifference (I), if their scores are equal, but never a relation of incomparability (R).

2.2.2 Outranking relations

Instead of reducing the evaluations of each alternative to a single score, another approach is to compare the evaluations between pairs of alternatives systematically and aggregate these comparisons into a binary relation called an outranking relation.

There are several methods of constructing these relations, among the most commonly used being the ELECTRE methods [Keeney and Raiffa 1976, Roy 1993] and the PROMETHEE methods [Brans and Mareschal 2002]. We mention also the RUBIS method [Bisdorff et al. 2008], which is defined in a bipolar setting. We will detail below the principle on which all these methods build the outranking relation.

An alternative x is said to outrank another alternative y if [Roy 1993]:

1. there is a qualified majority of weighted criteria on which x is performing at least as good as y;

2. there is no criterion on which y seriously outperforms x.

The first statement is modelled through local concordance degrees, which measure for each criterion $i \in F$ if the evaluation of x on i is at least as good as that of y. Generally, these degrees are valued on a scale from 0 to 1. If x is not at least as good as y on criterion i, then the local concordance degree is 0, if x is at least as good as y then this degree is 1. If there is a measure of uncertainty with respect to the two statements, a value between 0 and 1 may be used, depending on the particular type of outranking relation begin used.

The criteria on which x has an evaluation at least as good as y are then put into balance, by additionally using a set of criteria importance weights and constructing a global concordance degree. The importance weights have, in this case, a different meaning than the weights used in value functions. As the outranking methods are inspired from the voting procedures in social choice theory, each criterion can be seen as a group of voters having the same opinion, while the weight assigned to them represents their strength, as in number. On the other hand, the weights using in value functions are interpreted as trade-offs between criteria. The condition of having the criteria independent is also required in the case of outranking relations.

Unlike in MAVT, dealing with both qualitative and quantitative scales is much simpler in this case. We no longer need to convert a qualitative scale into a numerical representation, but only to define an operator for assessing if an evaluation is at least as good as another or not. In order to deal with potential imprecision in the evaluations of the alternatives on the set of criteria, some of these operators take into account discrimination thresholds. In most cases, two such thresholds are defined on each criterion: an indifference threshold and a preference threshold, which are used to define intervals inside which a difference between evaluations can be considered as indifferent, or clearly significant respectively. Discrimination thresholds are also used to deal with the subjective perception of the DM with respect to the criteria evaluations. For instance, if a DM considers the problem of buying a car, focusing equally on only two criteria, the price and the fuel consumptions, if one of the cars is more expensive but has a fuel consumption lower than the second, then he is faced with a dilemma. If the first car is more expensive with only a little amount, which the DM considers negligible, then the initial dilemma is removed. He is therefore able to select the slightly more expensive car.

The second part of the definition of the outranking relation can be easily verified by determining if, for any criterion $i \in F$, x has a much lower performance than y. In certain cases, if such a situation occurs, the judgements that can be extracted from the global concordance degree are invalidated and x is considered to not outrank y. For this reason this part of the definition of an outranking relation is also called a veto. For other outranking models, the presence of such a situation has the role of weakening the judgements given by the global concordance degree. In the case where no veto situation is present between x and y, the outranking relation is simply extracted from the global concordance degree. Usually, if the global concordance degree is above 0.5, which is the median level, then x is considered to outrank y, although, higher thresholds could be required, when criteria importance weights are not considered to precisely reflect the perspective of the DM. These thresholds are also called **cut levels**.

Another property that sets outranking relations apart from values functions lies in the fact that all the binary relations mentioned in the beginning of this section, namely indifference, weak preference, strict preference and incomparability, can be constructed from them. Two alternatives x and y are indifferent if x outranks y, but also if y outranks x. One alternative, x, is said to be strictly preferred to another alternative, y, when x outranks y and y does not outrank x. These alternatives are incomparable when neither of them outranks the other [Roubens and Vincke 1985].

The addition of the situation of incomparability is directly generated through the second part of the definition of an outranking relation and can only manifest itself in this context. This situation, according to Roy [Roy 1990], represents the DM hesitations, which may result from uncertainty, conflicts and/or contradictions. Generally, this situation results when two alternatives have very contrasting advantages. This means that each alternative has a very good performance on at least one criterion, when compared to the other.

The construction of an outranking relation is generally considered to be simpler than that of a value function, although it is more difficult to exploit. The reason for this lies in the fact that outranking relations may not output a complete weak order between the alternatives. The relation they output may not be complete nor transitive [Roy 1990]. In this case, constructing a final recommendation is considerably more difficult.

2.2.3 Eliciting the model parameters

As we have seen so far, the two main approaches of modelling the preferences of a DM rely on certain parameters that make them reflect these preferences with fidelity. They can be in the form of criteria weights, discrimination thresholds, cut levels, capacities or fuzzy measures. These parameters need to be elicited first so that the relations between the alternatives are well-modelled.

Mousseau defines preference elicitation as the "process that goes through an interaction between the decision-maker and the analyst (or a software) and leads the decision-maker to express preference information within the framework of a selected multiple criteria aggregation procedure" [Mousseau 2005].

The elicitation of the model parameters can be input-oriented, also called an aggregation procedure or direct parameter elicitation, or output-oriented, also called disaggregation procedure, indirect parameter elicitation procedure, ordinal regression analysis or inverse analysis.

The input-oriented preferential information can be in the form of:

- an exact value for a parameter (*e.g.* the weight for criterion i is exactly 0.5);
- an interval for a parameter (*e.g.* the weight for criterion i should be between 0.2 and 0.3);
- a ratio or a trade-off between two criteria weights (*e.g.* criterion i is twice as important as criterion j);
- a relative importance between two criteria or two coalitions of criteria (*e.g.* criteria i and j are more important together than criterion k);
- *etc.*

Asking for the model parameters directly implies that the DM is aware of and can express with great precision his preferences, but also that he is completely familiar with the preference model being used and knows what each parameter means and how it affects the relation between the alternatives. For instance, the AHP method [Saaty 1980] determines the parameters by asking the DM the relative importance between criteria.

The output-oriented preferential information can be in the form of:

- indifference between two alternatives;
- strict preference of one alternative over another;
- weak preference of one alternative over another;
- incomparability between two alternatives.

In the majority of cases, these methods use linear programming techniques [Jacquet-Lagrèze and Siskos 1982, Siskos and Yanacopoulos 1985, Siskos 1985] and concentrate on a single solution that maximizes an objective function, instead of exploring the entire set of potential parameters. Even so, the number of optimal or suboptimal solutions can be very large and an exhaustive search method, like for instance the reverse simplex method [Dantzig and Orchard-Hays 1953], can be impractical to use even on small datasets. In such cases, sensitivity analysis approaches need to be used in order to ensure the stability of the retrieved parameters.

The first of these methods is the UTA method [Jacquet-Lagrèze and Siskos 1982, Siskos et al. 2005], which uses linear programming for extracting the parameters of an additive value function from a partial ranking of the alternatives. Many methods have derived from it, out of which we mention UTADIS [Devaud et al. 1980], UTA* [Siskos and Yanacopoulos 1985] and also ADELAIS [Siskos and Despotis 1989], which uses the UTA method in an interactive manner. For more on the family of UTA methods the reader is refered to [Despotis et al. 1990]. Non-additive models have been used in [Angilella et al. 2010], while an overview of methods that extract the capacities of Choquet integrals can be found in [Grabisch et al. 2008].

When considering disaggregation approaches for outranking relations we mention the ELECCALC system [Kiss et al. 1994], which is used for inferring the parameters of the ELECTRE II method. A considerable amount of work has been done in the context of the ELECTRE TRI method. Interactive approaches for assessing the weights of the criteria from the assignment of alternatives to predefined categories have been proposed in [Mousseau and Slowinski 1998] and [Mousseau et al. 2001], which use mixed integer programs. In order to decrease the complexity of these programs, an adaptation of the valued outranking relation used in ELECTRE II and ELECTRE TRI has been proposed in [Mousseau and Dias 2004]. Mathematical programs have also been proposed for inferring the veto thresholds in [Dias and Mousseau 2006], while the criteria weights and cutting level parameters can be inferred without having predefined categories in [Rocha and Dias 2008]. We mention also the IRIS software [Dias and Mousseau 2003], which allows the DM to provide assignment examples and constraints on criteria weights and cutting level in the context of the ELECTRE TRI method. In [Meyer et al. 2008], a more general approach that uses a mixed integer linear program is proposed in order to determine the criteria weights, discrimination thresholds and also the evaluations of the alternatives from a median-cut outranking relation.

2.3 The classical decision problems

In this section, we will present the most common types of decision problems in MCDA (choice, sorting and ranking) and overview some of the most known approaches that are used to solve them. We will mostly focus on methods based on outranking relations, as these problems become very easy to solve when value functions are used.

Furthermore, we will not focus at this point on the problem of clustering, which has received much less attention than the three aforementioned decision problems. We will however dedicate the following chapter to it.

2.3.1 The choice problem

The problem of choice is defined as: adopting a selection procedure, more modest but more realistic than optimization, oriented to highlighting a subset [...] of satisfactory alternatives. [...] This subset will be as small as possible in order to justify the largest number possible of "non-choices".[1].

Among the first methods used to solve this problem in the outranking paradigm is the ELECTRE I method [Roy 1968]. As with most of the methods from the ELECTRE family, this one can be divided in two steps: the construction of an outranking relation and its exploitation. ELECTRE I builds an outranking relation on alternatives that are defined on criteria with the same numerical scale. The exploitation of this relation consists in finding a kernel in the graph derived from this relation, which is then recommended as the set of alternatives most likely to be chosen. A kernel is unique only in the case where no cycles exist in this graph. If this is not the case, then the maximal directed cycles are reduced to singleton elements prior to the main exploitation step. An extension of this method, ELECTRE IV, was proposed in [Maystre et al. 1994], which added the use of veto thresholds in order to account for large differences on each of the criteria scales. The ELECTRE IS method [Roy and Skalka 1984] further added the use of indifference and preference thresholds on each criterion, in order to account for imprecision in the evaluations of the alternatives on them.

Another approach that finds a best choice recommendation is the RUBIS method [Bisdorff et al. 2008], which also uses the notion of kernel, however it is extracted from a valued directed graph induced by a bipolar-valued outranking relation.

2.3.2 The sorting problem

The problem of sorting in MCDA is defined as: helping to sort the alternatives [...] through norms or to elaborate an assignment procedure.[2].

A clear parallel can be made in this case to the classification problem from data analysis. A fundamental distinction, however, lies in the structure of the results of these problems. While in a data analysis context, classification defines the groups of objects in a nominal way, in MCDA sorting very often defines them in an ordinal way, though not exclusively.

The most common sorting method, which uses an outranking relation to compare the alternatives, is ELECTRE TRI [Yu 1992, Roy and Bouyssou 1993]. This method assigns the alternatives to ordered categories defined by delimiting profiles using two heuristics: the pessimistic and the optimistic assignment procedures. In both cases

[1] "Elle consiste à adopter une procédure de sélection, plus modeste mais plus réaliste que l'optimisation, orientée vers la mise en évidence d'un sous-ensemble [...] d'actions satisfaisantes. [...] Ce sous-ensemble sera aussi restreint que possible en vue de justifier le non-choix du plus grand nombre possible d'actions."[Roy 1985].

[2] "aider à trier les actions [...] d'après des normes ou à élaborer une procédure d'affectation"[Roy 1985].

the assignment is based on the outranking relations between the alternatives and the profiles of the categories. Perny [Perny 1998] and Belacel [Belacel 2000] have extended this method to categories that are defined in a nominal way. In these cases, the categories are defined through a representative alternative and the alternatives are assigned to the category who's representative each alternative is most similar to. Other similar sorting methods that use outranking relations can be found in [Moscarola and Roy 1977, Massaglia and Ostanello 1991].

When value functions are used, the problem of sorting becomes very trivial, as the categories need to be defined as intervals on the scale on which the alternatives are given scores. This approach can be found in the utadis method [Jacquet-Lagrèze 1995, Zopounidis and Doumpos 1999] and its variants [Doumpos and Zopounidis 1998]. In [Zopounidis and Doumpos 2000] an alternative approach, which uses several utility functions to produce a sorting of the alternatives gradually, is proposed. Initially, two utility functions are used to split the alternatives into the ones belonging to the best category, *i.e.* the category with alternatives that are considered the best with respect to the preferences of the dm, and the alternatives that belong to the rest of the categories. Other two utility functions are then used to separate the second best category from the others. This process is repeated until all desired categories have been created. Non-additive value functions can also be used in conjunction with the problem of sorting [Keeney and Raiffa 1976], though the complexity of these models make their use in practical applications limited.

The interested reader may look at a more thorough review on sorting approaches in [Zopounidis and Doumpos 2002].

2.3.3 The ranking problem

The problem of ranking orders the alternatives [...] by comparing their relative merits. [...] The procedure [...] is of classification type, the resulting successive classes of ex-aequo not having, a priori, any absolute meaning [...] . The alternatives inside a class (rank) are considered equivalent and the alternatives of rank r are considered "more satisfactory" than those of rank $r^{'}$ $(r^{'} > r)$. [3].

As in mcda we find two large families of methods, electre [Figueira et al. 2005] and promethee [Brans and Mareschal 2002], approaches for ranking the alternatives have been proposed in both contexts.

From the electre family we first mention the electre ii method [Roy 1971a]. This method uses two outranking relations, a strong one and a weak one, where the first is used to partition the set of alternatives such that maximal cycles that emerge from this relation are considered as a single entity. Two partial orders are then generated from this partition. The first is constructed by building an initial

[3] La problematique en cause est une problématique du rangement [...] des actions [...] en comparant leurs mérites relatifs. [...] La procédure sera du type classement, les classes d'ex-aequo successives qui en résultent n'ayant, a priori, aucune signification absolue [...] . Les actions contenues dans une même classe (rang) seront jugées équivalentes et les actions de rang r seront jugées «plus satisfaisantes» que les actions de rang $r^{'}$ $(r^{'} > r)$. [Roy 1985]

order through successively removing the elements from the initial partition that are not preferred by any others and then refining this result using the weak outranking relation. Similarly, the second partial order is generated with the distinction that it is first built by successively removing the elements from the initial partition that are not preferred to any others. These two partial orders are then used to construct the final one. The ELECTRE III method [Roy 1978] improves ELECTRE II by being able to deal with inaccurate, imprecise, uncertain or ill-determined data. It also builds two complete pre-orders through processes called distillations, in a way similar to the ELECTRE II method, which are then merged into a single complete pre-order. Another extension of these methods, the ELECTRE IV method, arose from a subway network problem in Paris [Hugonnard and Roy 1982] and uses 5 different outranking relations in order to build a complete pre-order on the alternatives, the process of doing so being similar to the two methods described so far.

The PROMETHEE family of methods generally construct two indices for each alternative. These indices quantify the number of alternatives that are preferred by the considered alternative (the positive outranking flow) and the number of alternatives that are preferred to it (the negative outranking flow). Using these indices, the PROMETHEE I method [Brans 1982] builds a partial order, by placing an alternative above another if both the flows of the first alternative are better than those of the second, or if only one flow is better and the other is equal to that of the second alternative. The reader should note that the positive outranking flow is considered better than another if its value is larger, while the negative outranking flow should be lower for it to be considered better. The PROMETHEE II method proceeds to build a complete order on the set of alternatives by constructing a net outranking flow from the difference of the positive and negative outranking flows, therefore giving an unique score to each alternative. Other related methods are PROMETHEE III (ranking on intervals) and PROMETHEE IV (continuous case). In 1988, the GAIA visual representation tool [Mareschal and Brans 1988] for supporting the PROMETHEE methodology was developed. Other extensions can be found in [Brans and Mareschal 1992, Brans and Mareschal 1994].

We may also consider a stricter definition of ranking, which aims at constructing a linear order between the set of alternatives. In this context, we turn to Kemeny's problem [Kemeny 1959], in which, considering m sets of preferences in the form of linear orders on a set of objects, the aim is to propose a linear order on them which reflects the initial preferences as accurately as possible. Another related problem is that of Slater [Slater 1961]. In this case, given a relation which forms a tournament on a set of objects, a minimal transformation of this relation is desired so that it becomes a linear order. Such a tournament may be easily constructed using an outranking relation, therefore this definition of a ranking problem can be easily adopted also in MCDA. A multitude of methods related to these problems, also called ranking rules, have been developed. A thorough review can be found in [Charon and Hudry 2010]. Prudent ranking rules could also be considered for proposing a prudent order on the set of alternatives. This order can be seen as the "optimal trade-off between the absence of cycles and the existence of a linear order" [Lamboray 2007b]. A comparison of different prudent ranking rules and the ranking they obtain can be found in [Lamboray 2007a].

Summary

In the above, a detailed overview of the field of Multiple Criteria Decision Aid has been given, in order define the context in which we will address the problem of clustering, the topic of this thesis.

Following the introduction and the motivations for the emergence of this field, we have focused on the main approaches for modelling the preferences of one or several people over the set of alternatives on which they wish to make certain decisions. The outranking paradigm leads to the creation of several and complex relations between the alternatives, which although can be easier to construct, are more difficult to exploit, as opposed to value functions.

Finally, we have also presented an overview of the classical decision problems and the most commonly used methods that have been created for solving them. We have not yet considered the problem of clustering, both due to the fact that it has received less attention than the others, but also due to the fact that we will explore it in detail in the following chapter. We will furthermore confront these classical decision problems with the clustering problem, in order to characterise and separate the latter from them.

Chapter 3

On the MCDA clustering problem

Contents

Abstract

Having previously defined the problem of clustering in its original context, followed by a general presentation of the field of MCDA, we now focus on studying the clustering problem in this field. The question whether the use of similarity measures, which are generally applied when clustering, is well-motivated when clustering alternatives on which additional preferential information can be expressed will be addressed within.

We begin by first presenting an overview of the literature around clustering in MCDA, and then highlighting two of the earliest approaches that have been proposed in order to solve this problem. We then illustrate the main differences between the measures of similarity, and those that are naturally expressed in MCDA. Finally, we propose several definitions of clustering in MCDA, compare this problem to both classical clustering and the three most studied decision problems in MCDA, in order to set it apart and motivate its use further.

3.1 State-of-the-art of clustering in MCDA

3.1.1 Overview

A first taxonomy of existing MCDA clustering methods was presented in [Cailloux et al. 2007], where the clustering approaches are grouped in two, based on a property the authors call "criteria-dependency". This property is met by those clustering approaches that use the additional information of the DM, reflected through the criteria which define the alternatives. In other words, if a method is not "criteria-dependent", then it is a classical clustering approach that considers the criteria as attributes. The "criteria-dependent" clustering approaches take into account the additional information given by the criteria and are further divided in two: relational and ordered. Relational clustering approaches propose preferential relations between the clusters, at a local level, while ordered clustering approaches propose sets of clusters that are ordered from the best one to the worst.

Among the classical clustering approaches applied to the field of MCDA we mention here the efforts of Bisdorff [Bisdorff 2002] who actually proceeds to cluster the criteria and not the alternatives. This approach makes use of a similarity based proximity index for comparing the criteria together and extracts the clusters from the kernels in the graph constructed from this index, which is cut at a median level.

de Smet and Guzman have extended the classical K-means algorithm to the MCDA context in [de Smet and Guzman 2004], using a distance measure constructed from a given crisp preference relation between the alternatives. In this first work the authors do not propose a way to construct the relations between the different clusters, an issue which was later addressed in [de Smet and Eppe 2009].

Figuera *et al.* also extended the K-means algorithm in a multiple criteria framework [Figueira et al. 2004], while a more recent effort on extending this classical algorithm can be found in [Baroudi and Safia 2010].

Nemery and de Smet also proposed a clustering approach that finds a set of ordered clusters in [Nemery and de Smet 2005]. The authors work with given preference degrees between alternatives, which they try to minimize between the alternatives placed in the same cluster and to maximized between those from different clusters. A more recent work on the topic of ordered clustering was undertaken by Fernandez *et al.* [Fernandez et al. 2010], who propose to use an indifference measure in place of the similarity one generally used in classical clustering. In these approaches, the order between the clusters is complete, however Rocha *et al.* have worked very recently on a method that is able to find sets of partially ordered clusters [Rocha et al. 2012]. However, in this last work, the authors initially cluster the alternatives using measures of similarity.

We mention also additional work around this topic in [Valls and Torra 2000] and [Valls et al. 2009], but also several parts of the work that will be presented further, which has been disseminated in [Bisdorff et al. 2011, Meyer and Olteanu 2012, Meyer and Olteanu 2013].

3.1.2 Multicriteria Relational Clustering

We focus on a particular clustering approach from MCDA, the Multicriteria Relational Clustering approach from [DE SMET and EPPE 2009]. This is, to our knowledge, one of the first contribution to clustering in MCDA, which additionally takes into account the preferential information available in this context.

In this work, for a given set X of n alternatives, it is considered that initially one out of the three possible types of relations, indifference (I), strict preference (P) and incomparability (R), is given between any pair of alternatives in a crisp form. Additionally, a number of $k \in \{1, ..., n\}$ clusters is desired.

The authors define a proximity measure which reflects how much two alternatives compare in the same way to the entire dataset. A profile is defined for each alternative, $Q(x), \forall x \in X$, containing the set of alternatives to which x is indifferent, the set of alternatives to which x is preferred, the set of alternatives that are preferred to x and finally the set containing the alternatives that are incomparable with x:

$$Q(x) = \big\{\{y \in X : x \,\mathrm{I}\, y\}, \{y \in X : x \,\mathrm{P}\, y\}, \{y \in X : y \,\mathrm{P}\, x\}, \{y \in X : x \,\mathrm{R}\, y\}\big\}, \forall x \in X. \quad (3.1)$$

Each of the four sets in $Q(x)$ is referenced using an index from 1 to 4.

The proximity measure between any pair of alternatives x and y from X is then defined as:

$$d(x, y) = 1 - \frac{1}{n} \sum_{l \in 1..4} |Q_l(x) \cap Q_l(y)|, \forall x, y \in X. \quad (3.2)$$

The method follows the general lines of the K-MEANS approach, with a slight change. Unlike in classical K-MEANS, an initial set of clusters is generated first and not the set of centroids. The authors have used an approach of randomly constructing this initial partition.

The centroids are then generated from this initial set of clusters, through their profiles, in the following way:

$$Q_l(A) = \Big\{x \in X\colon\ l = \operatorname*{arg\,max}_{q \in 1..4} \big|\{y \in A\colon\ y \in Q_q(x)\}\big|\Big\}, \forall l \in 1..4, \forall A \subseteq X. \quad (3.3)$$

Hence, considering a centroid of a cluster, an alternative from the dataset will be placed in one of the four sets of the profiles of this centroid with the condition that it also appears in the same set for a majority of the alternatives of that cluster.

A new set of clusters is built from the centroids by assigning each alternative to the cluster whose centroid it is closest to, using the distance measure defined before.

In case several possibilities of assigning an alternative to a cluster exist, the one that occurred the first time is used. Furthermore, it may happen that certain centroids do not gather any alternatives inside their clusters. In this case, the authors place in this cluster the alternative from X that is closest to its centroid. The process is repeated if there are more empty clusters, however without considering the alternative that has been previously moved, in order to eliminate a potential infinite loop.

The described sequence of steps is repeated until no changes are made to the set of clusters.

Finally, a relation between any two clusters A and B is proposed through a simple majority rule:

$$l_{AB} = \begin{cases} \underset{l \in \{2,3,4\}}{\arg\max} \sum\limits_{x \in A} \sum\limits_{y \in B} |Q_{l_{AB}}(x) \cap \{y\}| & \text{, if } A \neq B, \\ 1 & \text{, otherwise.} \end{cases} \tag{3.4}$$

Therefore, if $l_{AB} = 1$ then $A \,\mathrm{I}\, B$, if $l_{AB} = 2$ then $A \,\mathrm{P}\, B$, if $l_{AB} = 3$ then $B \,\mathrm{P}\, A$, and if $l_{AB} = 4$ then $A \,\mathrm{R}\, B$. In case l_{AB} may be assigned more than one value, following the above formula, one value among them is selected at random.

It is clear that this approach follows the general lines of the K-MEANS algorithm. However, like the original, it suffers from the same shortcomings, requiring a fixed number of clusters to be extracted and being strongly dependent on the initialization step.

In addition, the use of the distance measure defined by the authors may be criticised, as it considers all the relations between two alternatives and the rest of the dataset, instead of considering the alternatives independently of the rest. By adding or removing alternatives from the dataset, the distance between two existing alternatives is altered, although the alternatives themselves do not change.

3.1.3 Multicriteria Ordered Clustering

We continue with another seminal work on clustering in MCDA, the Multicriteria Ordered Clustering approach from [NEMERY and DE SMET 2005], which constructs a set of ordered clusters.

Given the same set X of n alternatives, the authors assume that a preference matrix $\Pi = \left(\pi_{lm}\right)$, containing preference degrees between all pairs of alternatives, is given. Their aim is to propose and ordered partition $\mathcal{P}$, which is in most accordance to the original preferential information in Π.

For assessing the accordance between the clustering result and Π, two performance measures are defined: a homogeneity indicator $\mathcal{HI}_{\mathcal{P}}$ and a coherence indicator $\mathcal{CI}_{\mathcal{P}}$.

Considering that the ordered partition $\mathcal{P}$ contains k clusters, $C_i, \forall i \in 1..k$, where $C_i \succ C_j$ means that cluster C_j has a lower rank than cluster C_i, a local homogeneity indicator is defined for each cluster as:

$$\mathcal{HI}(C_i) = \mathcal{HI}_i = \max_{\forall l,m:\ x_l, x_m \in C_i} \pi_{lm} \tag{3.5}$$

It is clear that having a low homogeneity indicator is desired. The global homogeneity indicator of the ordered partition $\mathcal{P}$ is defined as:

$$\mathcal{HI}_{\mathcal{P}} = \max_{\forall i \in 1..k} \mathcal{HI}_i \tag{3.6}$$

The coherence indicator measures the discordances between the original preference degrees in Π and the order between the clusters in $\mathcal{P}$. A local indicator is defined between any two clusters from neighbouring ranks in the order, $C_i, C_j \in \mathcal{P}, C_i \succ C_j$, as:

$$\mathcal{CI}(C_i, C_j) = \mathcal{CI}_{ij} = \max_{\forall l,m:\ x_l \in C_i, x_m \in C_j} \pi_{ml} \tag{3.7}$$

A low coherence indicator denotes the fact that the alternatives in the cluster ranked higher are preferred by those in a cluster ranked below with a low degree, therefore showing a coherence with the relation between the two clusters. A global coherence indicator is constructed as:

$$\mathcal{CI}_{\mathcal{P}} = \max_{\forall i \in 1..k-1, j=i+1} \mathcal{CI}_{ij} \tag{3.8}$$

The overall fitness of the ordered partition $\mathcal{P}$ is then expressed as $\mathcal{FI}_{\mathcal{P}} = \max\left(\mathcal{HI}_{\mathcal{P}}, \mathcal{CI}_{\mathcal{P}}\right)$. The authors express the problem of clustering in this context as an optimization problem where $\mathcal{FI}_{\mathcal{P}}$ needs to be minimized.

The proposed approach to solving this problem is a mix between divisive hierarchical clustering and the Tabu Search meta-heuristic [Glover 1989, Glover 1990].

The outline of the algorithm is presented below:

1. generate the initial partition $\mathcal{P}$;
2. select C_i with the largest $\mathcal{HI}_i$ and split it in SC_{1i} and SC_{2i};
3. merge SC_{1i} or SC_{2i} with C_j so that $\mathcal{HI}(C_j \cup SC_{1i})$ or $\mathcal{HI}(C_j \cup SC_{2i})$ is minimized;
4. C_i becomes the sub-cluster SC_{1i} or SC_{2i} which was not previously merged;
5. construct a new $\mathcal{P}$ as the permutation of clusters that minimizes $\mathcal{CI}$;
6. go to step 2 until $\mathcal{FI}_{\mathcal{P}}$ does not improve for a certain number of iterations.

The method starts with an initial partition which is generated randomly and then successively splits the cluster with the largest homogeneity indicator and partially merges it with another. The split is done by initially identifying the two alternatives with the largest preference degree between them, which are placed in the two new sub-clusters. Following this, the other alternatives are assigned to the cluster to which they increase the homogeneity indicator the least amount. Then one of these sub-clusters is merged with another, following the same rule of minimizing the increase in homogeneity. After this step, all permutations of the existing clusters are tested with respect to the fitness indicator, and the best one is selected as the new ordered partition.

In order to avoid cycling between the same ordered partitions, a Tabu List is added, recording a fixed number of ordered partitions which were considered at previous iterations. A parameter is used to determine the probability that the splitting of a cluster will be done through the mentioned approach or at random. This parameter is increased when a new ordered partition is already present in the Tabu List and decreased otherwise. In this way, cycling over the same solutions is detected and removed.

The approach is further improved in order to find an arbitrary number of clusters. This is done by initially choosing a small value for k and letting the algorithm run until no better solutions are found. At this point the following split operation is performed without the subsequent merging of one of the two sub-clusters, k is increased and the algorithm is run again for $k = k + 1$. This sequence of steps is repeated until a pre-fixed maximum value of k is reached, and the clustering result with the best $\mathcal{FI}_{\mathcal{P}}$ is retrieved.

3.2 From data analysis to MCDA

We present, in this section, a parallel between the fields of data analysis and MCDA, in the context of the problem of grouping the entities that exist in each of them, or clustering. Through this parallel, we wish to motivate the definition of clustering in MCDA based only on the relations that can be expressed in this context, therefore moving away from notions related to similarity.

3.2.1 General comparison

We present the definition of cluster analysis, as given by JAIN in [JAIN 2010]: “Cluster analysis is the formal study of methods and algorithms for grouping, or clustering, objects according to measured or perceived intrinsic characteristics or similarity”.

The fundamental entities that exist in this context, the objects, are generally defined through their evaluations on a series of attributes. It follows from the definition above that, in the context of clustering in data analysis, the objective is to place in the same group objects that are perceived to have the same characteristics, *i.e.* are similar.

In certain cases, a prior analysis of the available information on the objects may be performed, leading to the selection of a subset of attributes that are representative for the person that wishes to cluster them. In this case, a particular similarity relation which reflects with fidelity the perspective of this person is also selected.

In other cases, clustering is used as an exploratory analysis technique. As Jain has mentioned in [Jain 2010], "the aim of clustering is to find structure in data and is therefore exploratory in nature". In this context, the previously mentioned information may not be available, and so, clustering is used in order to highlight certain potential ways in which the set of objects are perceived.

When looking at the field of MCDA, one of the main differences from data analysis lies in the purpose of the approaches defined within it. The central focus of the MCDA field is to aid one or several persons, or decision-markers, in reaching certain decisions over the set of objects, which in this case become alternatives.

Furthermore, the attributes on which the alternatives are initially defined are transformed into criteria, so that the preferences of the DM are reflected through them. Just as in certain cases in data analysis, a prior step of analysing the available information is performed in this case too. However, usually this analysis is more extensive, the reason being the very aim of MCDA. We remind at this point the detailed overview of the decision aiding process in [Tsoukiàs 2007], which may be used to support this statement.

During the decision aiding process, a clear preference model is selected, and carefully tuned so that is reflects the perspective of the DM on the set of alternatives. The two main categories of such models are value functions and outranking relations. All of the preference models may be used to express either a relation of indifference, weak preference, strict preference or incomparability between any pair of alternatives [Roy 1996, Greco et al. 2010]. However, the problem of clustering in MCDA, when using value functions for modelling the preferences of a DM, quickly becomes trivial, due to the fact the problem is reduced to clustering on a single dimension.

We will continue by performing a comparison between the similarity relations, which may be expressed in an MCDA context too, and the previously mentioned preferential relations that can be constructed through the use of outranking relations. We begin by highlighting these differences through the use of several simple illustrative examples.

3.2.2 Illustrative examples

Let us first define the general characteristics of the illustrative examples that will be used further. In each example, we will consider several alternatives characterised by only three criteria. Each criterion is defined on an ordinal scale, containing the following five levels: *very bad*, *bad*, *medium*, *good*, *very good*. These levels implicitly reflect the preferences of a fictive DM. In the following discourse, we will refer to them either as attributes, when the preferential information is not taken into consideration, or as criteria otherwise.

In order to measure the similarity between two alternatives, we will use a simple matching coefficient [Everitt et al. 2001, Kaufman and Rousseeuw 1990], while considering that all attributes are equally significant. Therefore, if two alternatives have the same level on one attribute, this will account for an increase in the similarity measure equal to $\frac{1}{3}$. Therefore, if two alternatives have the same evaluations on all attributes, then the similarity measure is equal to 1, while if they have different evaluations on all attributes, then the similarity measure is equal to 0. We will denote this measure with S.

In order to determine how two alternatives compare, when considering the existing preferential information that is given implicitly by the five labels on each attribute, we will be using an outranking relation that is similar to ELECTRE I [Roy 1968], in order to simplify the discourse. An outranking relation first assesses if there is a weighted majority of criteria on which one alternative is at least as good as another, through a concordance degree. In our case, the *at least as good* operator simply translates into having an evaluation at least as high as that of the second alternative. We also consider that the criteria are equally significant, therefore, if an alternative has an evaluation at least as good as that of another, on one criterion, this accounts for an increase of the concordance degree with $\frac{1}{3}$. Following the construction of the concordance degree, if the first alternative has at least one criterion on which it contains an evaluation that is considered much worse than that of the second, the outranking relation is invalidated, and so brought to a value of 0. If this is not the case, then the outranking relation is equal to the concordance degree. We will denote the concordance degree with C.

After constructing the two outranking relations between a pair of alternatives, if both are above the median level of $\frac{1}{2}$, we consider that the two alternatives are indifferent. If one concordance degree is above this level and the other is not, then the first is considered strictly preferred to the second, while if both outranking relations are not above the median level, then they are considered incomparable [Roubens and Vincke 1985]. We will use the same median level for assessing if two alternatives are similar with respect to the previously described matching coefficient.

We begin with the first example in Table 3.1, and detail the construction of the similarity and outranking relations below.

Table 3.1: Example 1 dataset;

X/F	1	2	3
x	*good*	*bad*	*medium*
y	*bad*	*good*	*medium*

$$
\begin{array}{ll}
S(x,y) = 0+0+\frac{1}{3} = \frac{1}{3} < \frac{1}{2} & \rightarrow \text{x and y are dissimilar}\\
\left.\begin{array}{ll}
C(x,y) = \frac{1}{3}+0+\frac{1}{3} = \frac{2}{3} \geqslant \frac{1}{2} & \rightarrow \text{x outranks y}\\
C(y,x) = 0+\frac{1}{3}+\frac{1}{3} = \frac{2}{3} \geqslant \frac{1}{2} & \rightarrow \text{y outranks x}
\end{array}\right\} & \rightarrow \text{x and y are indifferent}
\end{array}
$$

The similarity measure between the two alternatives is equal only to $\frac{1}{3}$, due to the fact that they have matching evaluations on only the last attribute. The concordance degrees, however, are both equal to $\frac{2}{3}$, due to the fact that x has evaluations at least as good as y on the first and last criteria, while y, in turn, has evaluations at least as good as x on the last two criteria. Furthermore, as neither of the two alternatives has a much worse evaluation than the other on any criterion, we consider that they outrank each other. Therefore, the two alternatives are indifferent.

Through this simple example, we have highlighted a situation in which two alternatives that are perceived as dissimilar, when considering the preferences of a DM on their evaluations, they may be also perceived as indifferent.

We continue by considering another small example in Table 3.2, containing three alternatives.

Table 3.2: Example 2 dataset;

X/F	1	2	3
x	*good*	*medium*	*bad*
y	*medium*	*bad*	*good*
z	*bad*	*good*	*medium*

$$
\begin{array}{ll}
\left.\begin{array}{l}
S(x,y) = 0+0+0 = 0 < \frac{1}{2}\\
S(x,z) = 0+0+0 = 0 < \frac{1}{2}
\end{array}\right\} & \rightarrow \text{x, y and z are dissimilar}\\
S(y,z) = 0+0+0 = 0 < \frac{1}{2} & \\
\left.\begin{array}{ll}
C(x,y) = \frac{1}{3}+\frac{1}{3}+0 = \frac{2}{3} \geqslant \frac{1}{2} & \rightarrow \text{x outranks y}\\
C(y,x) = 0+0+\frac{1}{3} = \frac{1}{3} < \frac{1}{2} & \rightarrow \text{y does not outrank x}
\end{array}\right\} & \rightarrow \text{x is strictly preferred to y}\\
\left.\begin{array}{ll}
C(y,z) = \frac{1}{3}+0+\frac{1}{3} = \frac{2}{3} \geqslant \frac{1}{2} & \rightarrow \text{y outranks z}\\
C(z,y) = 0+\frac{1}{3}+0 = \frac{1}{3} < \frac{1}{2} & \rightarrow \text{z does not outrank y}
\end{array}\right\} & \rightarrow \text{y is strictly preferred to z}\\
\left.\begin{array}{ll}
C(z,x) = 0+\frac{1}{3}+\frac{1}{3} = \frac{2}{3} \geqslant \frac{1}{2} & \rightarrow \text{z outranks x}\\
C(x,z) = \frac{1}{3}+0+0 = \frac{1}{3} < \frac{1}{2} & \rightarrow \text{x does not outrank z}
\end{array}\right\} & \rightarrow \text{z is strictly preferred to x}
\end{array}
$$

We can observe that, in this case, all the alternatives are dissimilar from each other, as they contain different evaluations on all criteria. Furthermore, neither of them are indifferent, as in every case, one of them is strictly preferred to the another. However, we find that, as x is strictly preferred to y, y is strictly preferred to z and z is strictly preferred to x, a cycle on strict preference is created. Although we can clearly state that each alternative is strictly preferred to another, neither of them is the best one. This type of situation occurs only when considering the use of outranking relations for modelling the preferences of a DM.

Some common practices in MCDA, in order to deal with this kind of structure, consider all the alternatives in the cycle as indifferent to each other [Roy 1968]. Another approach is to constrain the outranking relation by requiring larger weighted majorities of criteria for supporting it, therefore breaking the weaker relations in the cycle [Roy and Skalka 1984]. In our case, we would require to raise the cut level of the outranking relation to above $\frac{2}{3}$, however this would lead to neither of the alternatives outranking each other, and therefore becoming incomparable. We will explore this type of relation further.

In the example from Table 3.3 we find two alternatives with rather contrasting evaluations.

Table 3.3: Example 3 dataset;

X/F	1	2	3
x	*v.good*	*v.bad*	*medium*
y	*v.bad*	*v.good*	*medium*

$$S(x,y) = 0 + 0 + \tfrac{1}{3} = \tfrac{1}{3} < \tfrac{1}{2} \qquad \rightarrow \text{ x and y are dissimilar}$$

$$\left.\begin{array}{l} \left.\begin{array}{r} C(x,y) = \frac{1}{3} + 0 + \frac{1}{3} = \frac{2}{3} \geqslant \frac{1}{2} \\ x_2 \text{ much worse than } y_2 \end{array}\right\} \rightarrow \text{ x does not outrank y} \\ \left.\begin{array}{r} C(y,x) = 0 + \frac{1}{3} + \frac{1}{3} = \frac{2}{3} \geqslant \frac{1}{2} \\ y_1 \text{ much worse than } x_1 \end{array}\right\} \rightarrow \text{ y does not outrank x} \end{array}\right\} \rightarrow \text{ x and y are incomparable}$$

Because both alternatives have very contrasting evaluations they are considered as incomparable. If we would not have accounted for this large performance difference, based on the concordance degrees alone, we would have considered them as indifferent.

In Table 3.4, we highlight a different case which leads to the construction of a relation of incomparability between two alternatives.

Table 3.4: Example 4 dataset;

X/F	1	2	3
x	*v.good*	*v.good*	*v.bad*
y	*medium*	*medium*	*v.good*

$$S(x,y) = 0 + 0 + 0 = 0 < \tfrac{1}{2} \qquad \rightarrow \text{ x and y are dissimilar}$$

$$\left.\begin{array}{l} \left.\begin{array}{r} C(x,y) = \frac{1}{3} + \frac{1}{3} + 0 = \frac{2}{3} \geqslant \frac{1}{2} \\ x_3 \text{ much worse than } y_3 \end{array}\right\} \rightarrow \text{ x does not outrank y} \\ \begin{array}{r} C(y,x) = 0 + 0 + \frac{1}{3} = \frac{1}{3} < \frac{1}{2} \end{array} \quad \rightarrow \text{ y does not outrank x} \end{array}\right\} \rightarrow \text{ x and y are incomparable}$$

In this case, although x may be seen as strictly preferred to y, since $C(x, y) \geqslant \frac{1}{2}$ and $C(y, x) < \frac{1}{2}$, due to the fact that x contains a much poorer performance on the last criterion, it is considered that it does not outrank y. Therefore, both alternatives are incomparable.

In conclusion, through the above examples, we have highlighted the different types of preferential relations that may be constructed, in the context of using outranking relations, while at the same time, without taking into account the existing preferential information, the alternatives may be considered as dissimilar. When the alternatives are similar, they will in most cases be considered indifferent, except when they have very contrasting evaluations of at least one criterion.

3.2.3 Detailed analysis of the relations

We now turn our attention towards a more detailed analysis of several different measures that are generally used in conjunction with the problem of clustering, in order to group together objects that are perceived as similar.

Because we consider the notion of similarity to be too restrictive in the context where a DM may express additional preferences over the attributes which define these objects, we will additionally analyse a measure that reflects the degree of indifference between these objects, considering the use of an outranking relation. We will refer from this point forward to the objects as alternatives, and to the attributes as criteria, although we will not use the additional preferential information given by the criteria when constructing the measures of similarity.

The aim of this analysis is to highlight if there are any major differences between the notions of similarity and those of indifference. Such differences would then motivate the use of the measures of indifference when clustering in an MCDA context.

In the following analysis, we will consider only those measures that are constructed using solely the evaluations of any pair of objects, or alternatives.

The analysis will be conducted over a dataset containing all the alternatives that may be constructed on a fixed number m of criteria. In order to make this dataset finite, each criterion will be defined on an ordinal scale containing l levels. Therefore, the number of alternatives in the dataset will be q^l. Furthermore, each criterion will be given a weight, denoting its relative significance with respect to the other criteria. These weights remain the same for the different measures that we will consider, although they will be used in a different manner depending on the particular measure that uses them.

The measures

Based on these considerations, we have selected three measures of similarity and one measure of indifference.

The measures of similarity are the converse of the weighted Manhattan distance, the converse of the weighted Euclidian distance, and the similarity measure that has been defined in [Bisdorff et al. 2011]. We will denote them with S_{L_1}, S_{L_2} and S. The first two of them are defined below:

$$S_{L_1}(x,y) := 1 - \sum_{i \in F} w_i \cdot |x_i - y_i|, \tag{3.9}$$

$$S_{L_2}(x,y) := 1 - \sqrt{\sum_{i \in F} w_i \cdot (x_i - y_i|)^2}. \tag{3.10}$$

The last measure has been created as a parallel to the outranking relation, however for measuring perceived similarity. Just like an outranking relation, this measure considers that the attributes are independent, and that a series of importance weights $w_i, \forall i \in F$ have been assigned to them. Furthermore, three discrimination thresholds are considered on each attribute. These thresholds are the similarity (σ), dissimilarity (δ) and strong dissimilarity (δ^+) thresholds, which may be seen as equivalents of the indifference (q), preference (p) and veto (v) thresholds, that are generally used in the construction of an outranking relation. We present below the definition of this measure, which has been brought to a $[0,1]$ interval. We have also stripped away the parts containing the strong dissimilarity threshold, as we will not analyse its impact at this point.

$$S_i(x,y) := \begin{cases} 1 & \text{, if } |x_i - y_i| \leqslant \sigma_i, \\ 0 & \text{, if } |x_i - y_i| \geqslant \delta_i, \text{ ,} \forall i \in F \\ \frac{1}{2} & \text{, otherwise.} \end{cases} \tag{3.11}$$

$$S(x,y) := \sum_{i \in F} w_i \cdot S_i(x,y). \tag{3.12}$$

The indifference measure is constructed, as in [Fernandez et al. 2010], as the minimum out of the two credibility degrees that are attached to the outranking relations that can be expressed between two alternatives. We have selected the outranking relation from [Bisdorff 2012], which has also been, in this case, stripped of the large performance differences polarization construction.

$$C_i(x,y) := \begin{cases} 1 & \text{, if } x_i - y_i \geqslant -q_i, \\ 0 & \text{, if } x_i - y_i \leqslant -p_i, \text{ ,} \forall i \in F \\ \frac{1}{2} & \text{, otherwise.} \end{cases} \tag{3.13}$$

$$O(x,y) := \sum_{i \in F} w_i \cdot C_i(x,y), \tag{3.14}$$

$$I(x,y) := \min\Big(O(x,y), O(y,x)\Big). \tag{3.15}$$

The analysis

The analysis consists in constructing the previously described measures over all the pairs of alternatives that can be generated, considering that the number of criteria, m, is fixed at 5, and that their scales contain 6 ordered levels, represented through numerical scales from 0 to 5. We have also considered three distinct scenarios, in order to increase the quality of the analysis:

- Scenario 1:
  - all the criteria are equally significant;
    ($w_i = \frac{1}{m}, \forall i \in 1..m$)
  - all scale levels can be discriminated;
    ($q_i = 0$, $p_i = 1$, $\sigma_i = 0$ and $\delta_i = 1$, $\forall i \in 1..m$)
- Scenario 2:
  - all the criteria are equally significant;
    ($w_i = \frac{1}{m}, \forall i \in 1..m$)
  - neighbouring levels on each scale cannot be discriminated;
    ($q_i = 1$, $p_i = 2$, $\sigma_i = 1$ and $\delta_i = 2$, $\forall i \in 1..m$)
- Scenario 3:
  - the criteria have different significances;
    ($w_1 = 0.4$, $w_2 = 0.3$ and $w_3 = w_4 = w_5 = 0.1$)
  - neighbouring levels on each scale cannot be discriminated.
    ($q_i = 1$, $p_i = 2$, $\sigma_i = 1$ and $\delta_i = 2$, $\forall i \in 1..m$)

The restriction to 5 criteria on ordinal scales with only 6 levels is motivated by the fact that the number of similarity and indifference measures that need to constructed for this analysis is quadratic in the number of alternatives generated. In this case, although we have a total of $7,776$ generated alternatives, the approximative number of pairs that need to be considered is $3 \cdot 10^7$. Increasing slightly the number of criteria or the number of ordinal levels on each scale would quickly raise the number of pairs that need to be considered to an unmanageable size.

In Figures 3.1, 3.2, 3.3 and 3.4, we highlight the distributions of the four considered measures, over the three scenarios. We have also interpolated the given results in order to approximate the results of the general case of using quantitative scales, which is impossible to analyse through this approach.

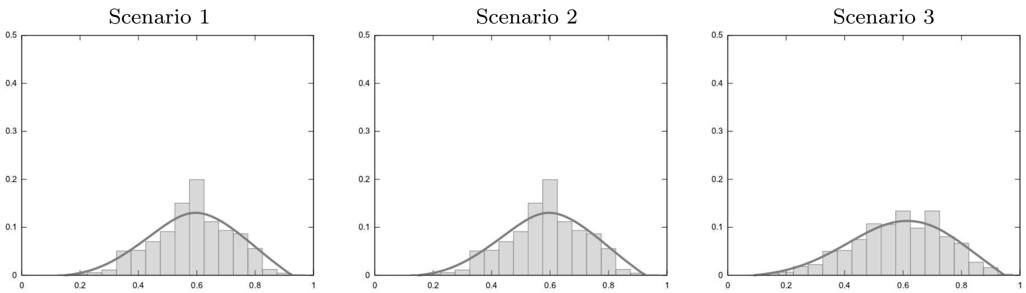


Figure 3.1: The distribution of the S_{L_1} similarity measure;

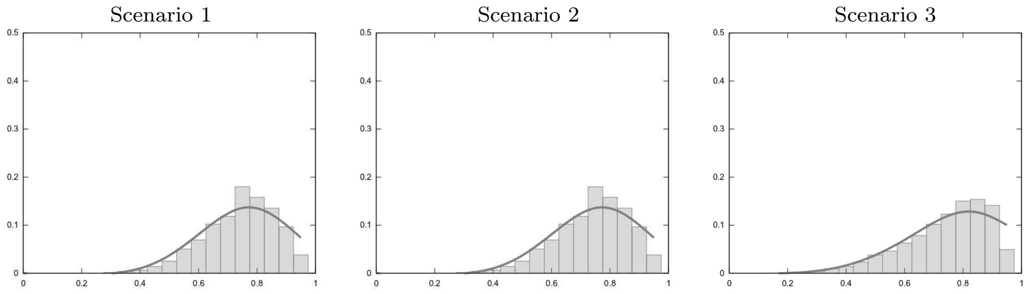


Figure 3.2: The distribution of the S_{L_2} similarity measure;

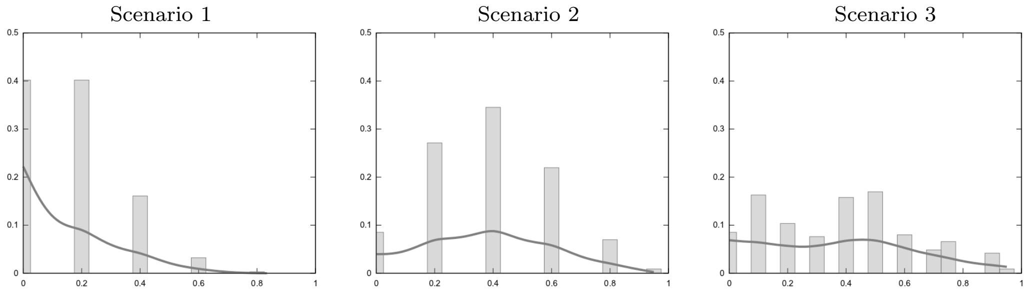


Figure 3.3: The distribution of the S similarity measure;

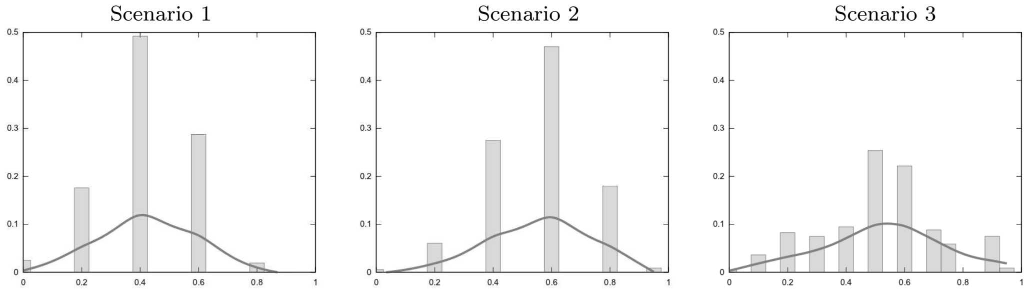


Figure 3.4: The distribution of the I indifference measure;

We find that the measures constructed from the weighted Manhattan and Euclidian distances have distributions close to the normal one, slightly shifted towards the right. No differences can be seen between the first and second scenarios, as the discrimination thresholds do not affect these measures. However, when placing more emphasis on the first two attributes, and less on the last three, we find that these distributions shift further towards the right. Nevertheless, we may conclude that most of the generated alternatives may be considered similar based on these measure, since the distributions are shifted to the right of the median level of $\frac{1}{2}$.

When looking at the S similarity measure, we find that, in the first scenario, the distribution is strongly centred towards the lower values range. This is expected, as in this scenario, only having the exact same values on one attribute leads to an increase of the similarity measure. The result of increasing the similarity threshold can be seen in the distribution of this measure for the second scenario, where higher values can be found. Furthermore, by discriminating between the different attributes, through the use of the significance weights, the measure takes a wider range of values. This is natural, as in the case of all the weights being equal to 0.2 there are only 6 distinct values, in the $[0, 1]$ interval, that can be constructed from them using only a, addition operator.

Most of the remarks related to the S similarity measure, can also be stated for the I indifference measure. We again find only 6 possible values that it can take, considering the first two scenarios, while in the last one this range increases. Furthermore, increasing the indifference and preference thresholds also leads to the shifting of the distribution to the higher values range. One clear difference, however, can be noticed in the shape of the distribution, which is centred more towards the right, in comparison to that of the S similarity measure.

We may conclude that, although more pairs alternatives are seen as dissimilar, when considering the S similarity measure, when using the I indifference measure we are able to find that most of them that are seen as indifferent. This remark is also consistent with the previous illustrative examples, where we were able to show that alternatives that are dissimilar may still be considered to be indifferent, when we take into account the additional preferential information on the set of attributes that define them.

In order to better compare these measures together, we present in Figures 3.5,3.6 and 3.7 the distribution of the absolute error between the indifference measure and each of the considered similarity measures.

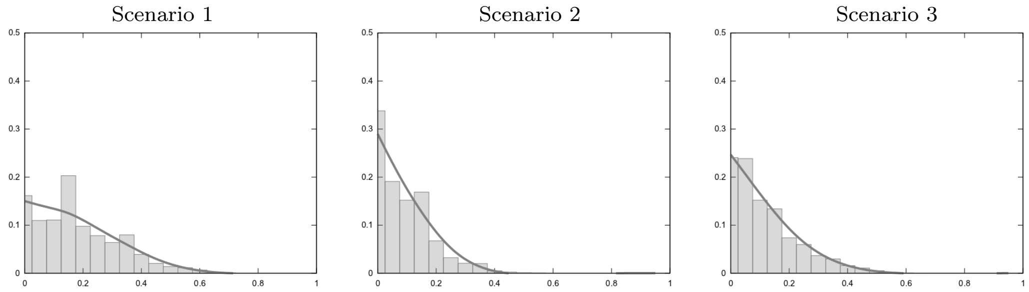


Figure 3.5: Distribution of the absolute error between I and S_{L_1};

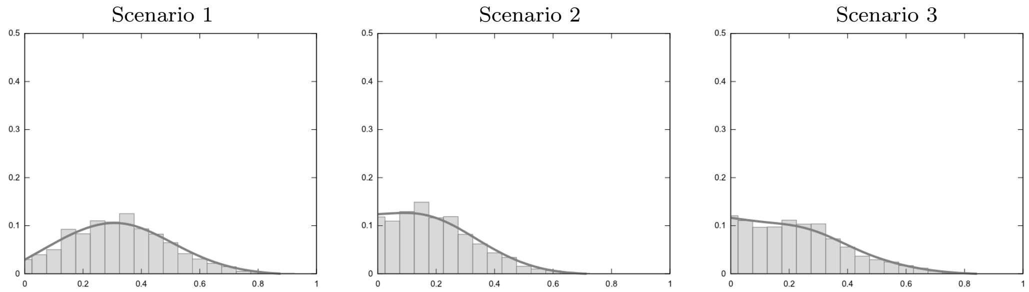


Figure 3.6: Distribution of the absolute error between I and S_{L_2};

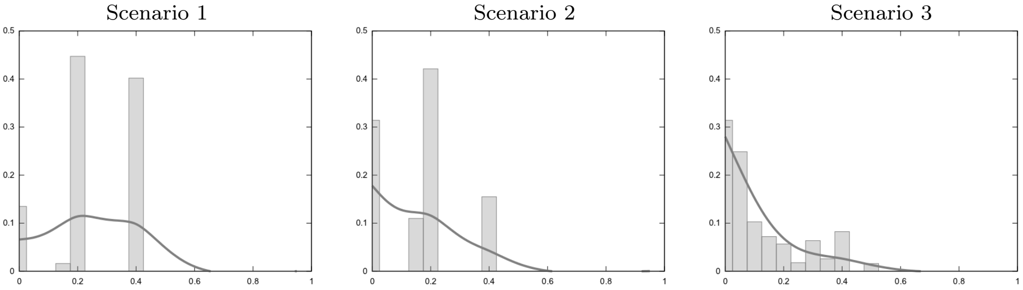


Figure 3.7: Distribution of the absolute error between I and S;

We notice that, although S_{L_1} and S come close to the values of the I measure, they also deviate from them with a rather high degree. We may conclude that the S_{L_2} measure is not very good in approximating I, which gives the degree of indifference between any pair of alternatives.

We finally provide a further comparison of the considered measures, by determining the percentage of times when alternatives that were considered indifferent, with respect to the I measure, were also considered to be similar, with respect to each of the other similarity measures, or when the alternatives that were considered not indifferent were also seen as dissimilar.

Table 3.5: Percentage of cases with similar and indifferent alternatives, or dissimilar and non-indifferent ones;

Measure	Scenario		
	1	2	3
S_{L_1}	46.5	73.2	74.5
S_{L_2}	32.8	67.4	72.0
S	72.8	63.9	70.8

Looking at the results in Table 3.5, we may conclude that the last similarity measure models the previously described property most consistently, across the three scenarios. The first two measures, although perform better on the last two scenarios, have very poor performances on the first. Nevertheless, all of the considered measures will in 30% of the cases consider that two indifferent alternatives are dissimilar, or that two non-indifferent alternatives are similar.

Taking into account the remarks related to the error distributions between the indifference measure and the three similarity measures, we may therefore conclude that they discriminate between the considered alternatives rather differently.

3.3 The problem of clustering in MCDA

Based on the overview of existing clustering approaches in MCDA, that we have seen in the beginning of this chapter, we were able to notice that many of them try to adapt approaches used in Data Analysis to the MCDA context. They either use similarity measures or construct distances from the preferential information, a practice which may be criticised.

In fact, Fernandez *et al.* [Fernandez et al. 2010] have already argued in favour of using a measure of indifference in order to replace the one of similarity. In the previous section, we have further analysed the differences between several commonly used similarity measures and the indifference measure proposed by these authors, and concluded that the latter has a rather different discrimination behaviour with respect to the alternatives that are being compared. Hence, we consider the use of an indifference measure instead of a similarity one well motivated.

Although the clustering approach proposed by Fernandez *et al.* in [Fernandez et al. 2010] uses a measure of indifference, while also the approach of Nemery *et al.* in [Nemery and de Smet 2005] may be seen to group the alternatives based on similar properties, both these approaches restrict themselves to the case of proposing an ordered set of clusters. We consider that complex structures, such as cycles on strict preference relations and those given through the relations of incomparability, may also be found in a MCDA clustering context.

To our knowledge, the topic of formally defining clustering in MCDA, considering all potential structures that the sets of alternatives may be in, has not yet been tackled, an issue which we wish to address here.

3.3.1 Defining clustering in MCDA

We will present in the following, several definitions of clustering in MCDA, and highlight the potential structures they may uncover and their potential use.

We start by reminding the original definition of clustering, which we call here **classical clustering**, as *the process of grouping objects that are similar and separating those that are not.*

Following the analysis that was previously detailed, we would like to provide several further definitions which will be based not on notions of similarity, but on notions of indifference, strict preference and incomparability.

We try to adhere to the existing terminology used in the MCDA literature around this topic and divide the different definitions based on the type of structure that we expect to find at the end of the clustering process. These structures are several in number, a fact that does not exist in the context of clustering in data analysis. In this latter case, the clusters hold only the property that the alternatives inside them are dissimilar from those inside other clusters.

We propose a similar definition to that of classical clustering, however using the notion of indifference, and call it **non-relational clustering**.

Definition 3.1 *We define **non-relational clustering** as the process of grouping alternatives that are indifferent and separating those that are not.*

The term non-relational has been chosen due to the fact that we are only looking, in this case, to group alternatives that are indifferent, while not proposing any of the other relations, of strict preference or incomparability between them. This may be considered also as nominal clustering, with the alternatives being grouped on reasons of them being indifferent from the perspective of the DM.

While in a non-relational clustering context, none or few relations of indifference may be found between different clusters of alternatives, any of the other relations of strict preference or incomparability may take their place. Finding a more structured clustering results, where only one such relation is predominant between any pair of clusters may be desirable.

Definition 3.2 *We call **relational clustering** the process of grouping alternatives that are indifferent and separating those that are not, such that, for any pair of clusters, the relations between the alternatives inside them are either of strict preference in one direction or another or of incomparability.*

Relational clustering only considers the relations between clusters in a pairwise manner. Considering that a relation of indifference may not be built between clusters, since we try to group the alternatives that are indifferent into clusters, while also modelling the relation of incomparability as the absence of a strict preference in either direction, we can characterise the structure of the clusters only through the relation of strict preference. Therefore, relational clustering may lead to one of the following structures on the set of clusters: partial or complete tournament and strict partial or strict complete order.

We may notice that the first two structures would exist only if the transitivity of the relations of strict preference between clusters would be broken by cycles on the relations of strict preference or in certain cases by relations of incomparability. Trying to find an ordered set of clusters by ignoring these situations may lead to a poorly supported clustering result. Furthermore, highlighting such situations may prove useful in the analysis of the considered set of alternatives. This may be especially practical during a process of preference elicitation, when analysing the relations between individual alternatives may be impossible for larger datasets.

Keeping in line with the evolution of clustering methods in the field of MCDA, we also identify a subcategory of relational clustering, that of ordered clustering.

Definition 3.3 *We define **ordered clustering** as the process of grouping alternatives that are indifferent and separating those that are not, such that the relations between the alternatives from different clusters support an ordered structure.*

The search for such structures may be motivated by the desire of a DM to find an ordered set of categories of alternatives, however in a context in which he is unsure on the number of these classes or their characteristics. At this point he may retrieve an ordered clustering result, which with the aid of certain descriptive measures may be used to make these classes become apparent to the DM.

All the potential structures that can be uncovered through clustering in the field of MCDA are also highlighted in Figure 3.8.

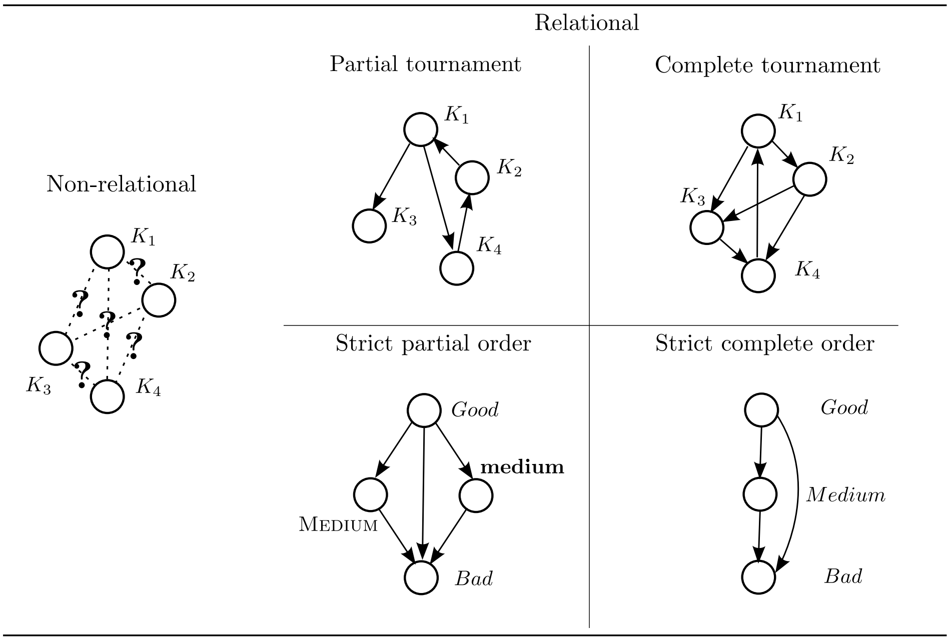


Figure 3.8: Potential structures of clusters in an MCDA context;

3.3.2 Motivations for clustering

Clustering is generally used in exploratory analysis in order to find a most likely way in which the data is structured [JAIN 2010]. This differentiates it from classification, which tries to fit a given model over the data. However, some weaker assumptions are also made when clustering, more precisely though the similarity or distance measure that is chosen.

When considering the MCDA domain, the aiding process is divided into four parts [TSOUKIÀS 2007], which we highlight in Figure 3.9. We start by situating the problem, identifying the alternatives and the type of decision the DM needs to take, then a model to represent the preferences of the DM is selected and tuned in order to reflect them, while finally a specific method is used to give a solution to the problem.

Three main types of decision problems have been formulated in [ROY 1985]: choice, sorting and ranking. We have additionally placed clustering along these

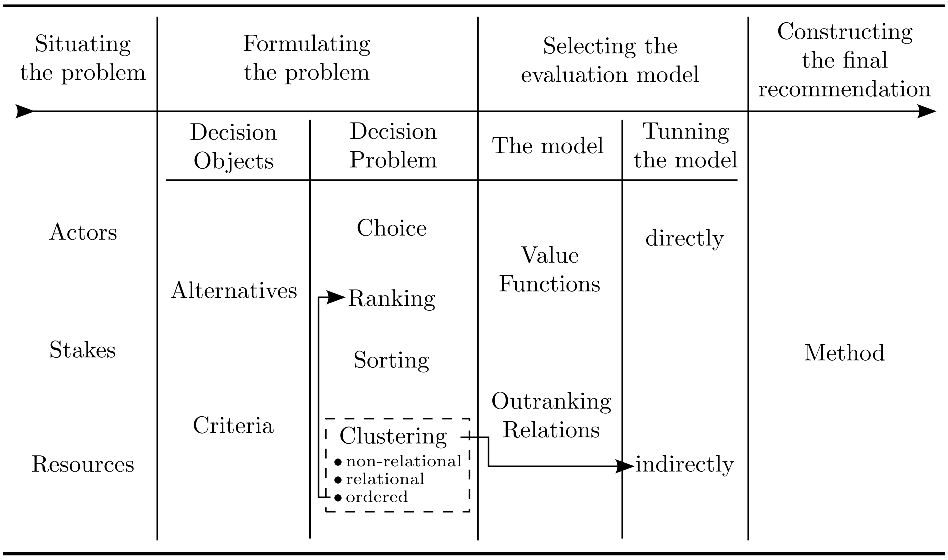


Figure 3.9: Clustering in a decision aiding process;

decision problems and wish to highlight its similarities and differences from the rest.

The choice problem proposes, as a recommendation, the best alternative or the set containing the best ones [Roy 1985]. From this perspective, it is quite different from clustering, which proposes a way in which all the alternatives can be grouped together. One similar aspect between the two lies in the fact that both consider the relations between alternatives in constructing the final result.

The problem of sorting is defined as the assignment of alternatives to predefined classes [Roy 1985], which may be ordered or not, therefore it can be seen as an adaptation of classification to the MCDA context. From this perspective it differs from clustering through the fact that the relations between the alternatives themselves are not used in constructing the final result, but the way in which each alternative, independently of the others, fits with respect to the given characteristics of each class. For example, in the ELECTRE TRI sorting approach [Yu 1992, Roy and Bouyssou 1993], the alternatives are compared to a set of profiles, or fictive alternatives, which are used to delimit each class.

The problem of ranking has been defined as that of ordering the alternatives from the best one to the worst, while also allowing for ties and incomparability to occur [Roy 1985]. This problem also uses only the relations between the alternatives in building the recommendation. The resulting equivalence classes will form either a strict partial or a strict complete order, therefore from a topological point of view the problem of ordered clustering is identical to it. We have illustrated this connection also in Figure 3.9. We may additionally argue that this definition of ranking is too broad, and consider ranking as defined in the context of ranking rules [Charon and Hudry 2010]. For example, a strict ranking with no ties and incomparability may be required when ordering the best candidates for a job position. In this context, the candidates can be asked for accepting the position one by one, from the best one to the worst until one accepts the position.

Looking now at clustering in MCDA, although the case of ordered clustering may be covered by the definition of ranking, both non-relational and relational clustering have no existing correspondents in the previously mentioned decision problems. Non-relational clustering looks for clusters of indifferent alternatives which are defined in a nominal manner, while relational clustering may uncover structures less strict than orders, which may exist if we are using as a preference model an outranking relation.

Furthermore, clustering may also be used in MCDA as an exploratory analysis technique, in order to aid a DM in understanding better the set of alternatives he is dealing with. The potential structures in which the alternatives are grouped could then be highlighted, additionally using measures to summarise these structures. In this way large sets of alternatives may also be more easily dealt with by the DM.

We furthermore wish to highlight the possibility of using clustering as a means of focusing a process of indirect elicitation of the parameters of a preference model, particularly one based on outranking relations. This may be especially beneficial when trying to extract the preferences of a DM over a large set of alternatives. In such cases, certain procedures need to be taken in order to focus the questioning protocol towards the pairs of alternatives that would most likely increase the convergence of the set of parameters towards the one that reflects fully the preferences of the DM. At this point, many preference elicitation procedures use approaches based on mathematical programs, which do not scale well.

Summary

In this chapter, several important differences have been pointed out between the relations of similarity and dissimilarity, and those present in an MCDA context, the relations of indifference, preference and incomparability.

A detailed overview of the existing clustering approaches in this fields has then been presented, highlighting several of the more representative algorithms. Following the differences between the relations generally used when clustering in data analysis, and those that exist in an MCDA context, additionally supported by the very few MCDA clustering approaches that take the latter into account, we have formally defined clustering in this context. The definitions that we propose use only the relations that can be naturally constructed when additional preferential information is available over the set of alternatives. These definitions will later be used in order to model the clustering problem and to propose several algorithmic approaches to solving it.

Finally, we have given several motivations for defining the problem of clustering in MCDA, through the comparison with the three classical decision problems, and we have further highlighted some of its potential applications. Several of them will be explored in the third part of the thesis.

Part II

Tackling the problem of clustering in MCDA

Chapter 4

Modelling the clustering problem

Contents

Abstract

We present, in this chapter, several ways in which the problem of clustering in the field of Multiple Criteria Decision Aid can be modelled. This is motivated by the formal definitions which were previously given to this problem.

We begin with several initial considerations, by defining a credibility calculus that is further used to model the three types of preferential relations that can be expressed between alternatives: indifference, strict preference and incomparability. Following this, we focus on several approaches of constructing these relations between sets of alternatives and propose different ways of expressing their credibility degrees. We also look at certain properties that these relations might hold, which may be used when building a particular clustering structure. Finally, we turn our attention towards modelling the clustering problems, considering the different structures that can be uncovered but also several measures that may be used to reflect their quality.

4.1 Working context

Let $X = \{x, y, z, ...\}$ be a set of n alternatives, that are defined on a family of criteria $F = \{1, 2, ..., m\}$. We consider ourselves inside a decision aiding framework, where the criteria have been previously constructed during the step of formulating the problem. Furthermore, a preference model based on an outranking relation has been chosen for modelling the way in which a DM perceives the alternatives. We also assume that this DM is interested in clustering the alternatives together, either in a non-relational, relational or ordered fashion.

The binary relations that can be constructed between the alternatives in this context, namely indifference, preference and incomparability are denoted with I, P and R respectively, while the universe containing them is denoted with O.

4.1.1 Credibility calculus

We begin by defining a credibility calculus, inspired by the one proposed in [Bisdorff 2000]. This calculus will then further be used in highlighting several ways in which the previously mentioned binary relations can be constructed along with several measures of credibility that may be attached to them. We will consider both the cases of a two-valued logic, *i.e.* boolean, or a three-valued logic, which will be defined in a bipolar setting.

We start with the boolean credibility calculus, which contains two logical states: true and false. Considering a set $\mathcal{P}$ of propositional statement, we associate to them the credibility function $r : \mathcal{P} \to [0, 1]$.

For a given proposition $p \in \mathcal{P}$, if $r(p) = 1$ then p is considered to be true, while if $r(p) = 0$ then p is considered to be false. All the possible values of r between 0 and 1 show a higher or lower degree of credibility with respect to the truth of proposition p. The limit case of $\frac{1}{2}$ may be considered as either true or false, but as we are working in a boolean logic, one statement needs to be selected, which in general is that the proposition is true.

In order to extract the boolean statements with respect to the truth of proposition p we attach a cut level $\lambda \in [\frac{1}{2}, 1)$. This level is motivated by the fact that a larger value of the credibility function is required in order to validate the statement. We model this through the crisp credibility function $r^* : \mathcal{P} \to \{0, 1\}$, which is defined as:

$$r^*(p) = \begin{cases} 1 & \text{, if } r(p) > \lambda, \\ 0 & \text{, otherwise.} \end{cases} \tag{4.1}$$

Let $\neg$, $\wedge$ and $\vee$ denote the logical operators negation, conjunction and disjunction respectively. From a finite set $\mathcal{P}$ of ground statements, grouping brackets and the basic logical operators, we may generate the set $\mathcal{E}$ of all well-formed finite statements as:

$$\begin{aligned} &\forall p \in \mathcal{P} : p \in \mathcal{E}, \\ &\forall x, y \in \mathcal{E} : \neg x | (x) | x \vee y | x \wedge y \ \in \mathcal{E}. \end{aligned} \tag{4.2}$$

We may extend the credibility denotation to all the well-formed finite statements $x, y \in \mathcal{E}$ through:

$$\begin{aligned} r(\neg x) &= 1 - r(x), \\ r(x \vee y) &= \max\big(r(x), r(y)\big), \\ r(x \wedge y) &= \min\big(r(x), r(y)\big). \end{aligned} \tag{4.3}$$

We may furthermore extend the calculus to the three-valued logic, which contains an additional logical state, that of indetermination. The credibility function r, in this case, is set in a bipolar value range, in the interval $[-1, 1]$.

For a given proposition $p \in \mathcal{P}$, if $r(p) = 1$ then p is considered to be true, while if $r(p) = -1$ then p is considered to be false. If $r(p) > 0$ then the statement is considered to be more true than false, whereas if $r(p) < 0$ then p is considered to be more false than true. The limit case of 0 is considered as a hesitation between stating that p is either true or false and as such p will be considered as indeterminate with respect to the two logical statements. We notice that in this case, we maintain the level of hesitation, unlike in the previous case where we were forced to decide between one truth level or the other.

We may again use a crisp credibility function $r^* : \mathcal{P} \rightarrow \{-1, 0, 1\}$ for determining the three truth statements, using another cut level $\lambda \in [0, 1)$:

$$r^*(p) = \begin{cases} 1 & \text{, if } r(p) > \lambda, \\ -1 & \text{, if } r(p) < -\lambda, \\ 0 & \text{, otherwise.} \end{cases} \tag{4.4}$$

In this case, we have used a double cut of the original credibility function. This is motivated by the fact that, when a certain fixed level, higher than the median, is required for validating a statement, in order to invalidate another statement, the credibility degree needs to be lower than the reflection of this fixed level with respect to the median.

The set $\mathcal{E}$ of all well-formed finite statements can be defined in the same way as before, as well as the credibility of basic logical operators on them, with the exception of negation which becomes simply $r(\neg x) = -r(x)$.

4.1.2 Modelling the preferential relations

As mentioned previously, we consider that the preferences of the DM are modelled through the use of an outranking relation.

From these relations, we can construct the three binary preferential relations of indifference, preference and incomparability in the following way [Roubens and Vincke 1985]:

$$\begin{aligned} x \,\mathrm{I}\, y &\iff x \,\mathrm{S}\, y \text{ and } y \,\mathrm{S}\, x, \\ x \,\mathrm{P}\, y &\iff x \,\mathrm{S}\, y \text{ and } y \,\not\mathrm{S}\, x, \\ x \,\mathrm{R}\, y &\iff x \,\not\mathrm{S}\, y \text{ and } y \,\not\mathrm{S}\, x. \end{aligned} \tag{4.5}$$

Usually the S relation has a degree of credibility attached to it, either in a boolean or bipolar setting. Using these degrees, we may apply the credibility calculus previously described and define the credibility of the I, P and R relations, in both boolean or bipolar settings, as:

$$\begin{aligned} r(x \,\mathrm{I}\, y) &= \min\big(r(x \,\mathrm{S}\, y), r(y \,\mathrm{S}\, x)\big), \\ r(x \,\mathrm{P}\, y) &= \min\big(r(x \,\mathrm{S}\, y), r(y \,\not\mathrm{S}\, x)\big), \\ r(x \,\mathrm{R}\, y) &= \min\big(r(x \,\not\mathrm{S}\, y), r(y \,\not\mathrm{S}\, x)\big) \end{aligned} \tag{4.6}$$

The definition of these credibility degrees directly follows the formal definitions from [Roubens and Vincke 1985], through the use of the credibility calculus herewith defined. The credibility degree of the indifference relation has also been similarly defined in [Fernandez et al. 2010].

We further wish to mention that, using the boolean credibility calculus, the relations of indifference, preference and incomparability are mutually exclusive. In the three-valued logic, however, this no longer applies. For instance, when both $x \,\mathrm{S}\, y$ and $y \,\mathrm{S}\, x$ are in a state of indetermination, then all of $x \,\mathrm{I}\, y$, $x \,\mathrm{P}\, y$, $y \,\mathrm{P}\, x$ and $x \,\mathrm{R}\, y$ are indeterminate too. When either $x \,\mathrm{S}\, y$ or $y \,\mathrm{S}\, x$ are in a state of indetermination, but not both at the same time, then two of $x \,\mathrm{I}\, y$, $x \,\mathrm{P}\, y$, $y \,\mathrm{P}\, x$ and $x \,\mathrm{R}\, y$ are also in a state of indetermination, while the rest are false. In these cases, as no relation is considered to be true, out of I, P and R, the ones that are indeterminate are considered equally probable. If an approach requires that only one type of relation exists between two alternatives, a random approach of selecting it among the ones that are indeterminate may be employed.

One additional mention involves the bipolar valued outranking relation with large performance differences polarization from [Bisdorff 2012], for which the degree of credibility of the relation of incomparability never rises above the median level.

4.2 Comparing sets of alternatives

Comparing pairs of alternatives can be consistently done using the I, P and R relations, through the credibility degrees previously defined. Extending them to sets of alternatives, however, is not as trivial.

4.2.1 Using representatives

Many clustering approaches use cluster representatives in order to characterise the objects inside them, either due to the fact that such an approach is required by the algorithm itself, or as a measure of characterising the final result [Jain 2010]. Furthermore, an approach of building cluster representatives in order to compare them with respect to the preferences of a DM has been used in [Valls 2002], more precisely in order to rank the clusters. Hence, we propose below the use of representatives in order to construct the preferential relations between sets of alternatives.

For any two sets of alternatives $C, D \subseteq X$, we denote their representatives through r_C and r_D. The relation between the two sets can be expressed by the relation between the two representative alternatives:

$$r_C \,\mathrm{O}\, r_D \rightarrow C \,\mathrm{O}\, D, \forall\, \mathrm{O} \in \{\mathrm{I}, \mathrm{P}, \mathrm{R}\}. \tag{4.7}$$

The degree of credibility will also be the same for the two sets as that of the two alternatives:

$$r(C \,\mathrm{O}\, D) = r(r_C \,\mathrm{O}\, r_D), \forall\, \mathrm{O} \in \{\mathrm{I}, \mathrm{P}, \mathrm{R}\}. \tag{4.8}$$

Finding a representative alternative for any set $C \subseteq X$ can be done in two ways:

- selecting an existing alternative from C;
- constructing a fictitious alternative.

Selecting an existing alternative from the set in order to play the role of a representative should be done based on a measure that reflects this property. If we consider the relation of indifference as the one used to characterise alternatives that cannot be distinguished from each other, based on the perspective of the DM on them, we can define the following fitness functions:

$$f^*_{avg}(r_C, C) = \frac{1}{|C|} \cdot \sum_{x \in C} r^*(r_C \,\mathrm{I}\, x); \tag{4.9}$$

$$f_{avg}(r_C, C) = \frac{1}{|C|} \cdot \sum_{x \in C} r(r_C \,\mathrm{I}\, x); \tag{4.10}$$

$$f_{min}(r_C, C) = \min_{x \in C} r(r_C \,\mathrm{I}\, x). \tag{4.11}$$

The first fitness function, f^*_{avg}, measures the ratio between the existing and the possible numbers of indifference relations between the representative and the alternatives in C. This function may be used in cases where it is important have the representative indifferent to as many alternatives in C, but no necessarily with a high degree of credibility.

The second fitness function, f_{avg}, takes into account also the degree of credibility of the relations of indifference between the representative and the alternatives in C, however it allows for compensation between these degrees to occur. This fitness function may be used where we are also interested in having a high degree of confidence attached to the relation of indifference between the representative of C and the alternatives inside.

The last function measures the minimum degree of credibility of the relations of indifference between r^C and the alternative in C. This is the most strict fitness measure and reflects the need to have all alternatives in C indifferent to the representative and in addition with the degrees of confidence as high as possible.

Following this, denoting with f either of the fitness functions before, the representative may be selected from the alternatives in C as:

$$r_C = \arg\max_{x \in C} f(x, C). \tag{4.12}$$

The representative may also be constructed as an artificial alternative, however this can be done only when considering a particular evaluation model. We will present several ways of constructing the representative of a set of alternatives, using the bipolar valued outranking relation with large performance differences polarization from [Bisdorff 2012] in Chapter 7.

4.2.2 Aggregating pairwise comparisons

Finding a representative may not always be a simple task. A different approach of constructing the relation between two sets of alternatives consists in aggregating all the relations between the alternatives of the two sets. In the approach of de Smet and Eppe [de Smet and Eppe 2009], the relation that appears most frequently, between the alternatives of two clusters, is selected as the relation between the two sets of alternatives.

We begin with a different approach, which uses outranking relations in order to compare the alternatives together. We proceed to building two global outranking relations between two sets of alternatives, from the outranking relations between the alternatives inside them, considering each ordered pair only once:

$$r(C \,\mathrm{S}\, D) = \frac{1}{|C| \cdot |D| - |C \cap D|} \sum_{x \in C, y \in D, x \neq y} r(x \,\mathrm{S}\, y). \tag{4.13}$$

From this, we may then construct the credibility of the I, P and R relations between the two sets, and then select the relation that has the highest degree of confidence as the relation between the two sets.

This approach, however, might not always be in accordance with the actual preferential relations that can be found between the alternatives of the two sets. We illustrate this through the following example.

Example 2 *We consider $C = \{x, y\}$ and $D = \{z, y\}$, two sets of alternatives, and the outranking relations between them shown in Table 4.1. The relations between the alternatives inside the same set are not represented, while the credibility degrees of the other relations are given on a bipolar scale from -1 to 1.*

Table 4.1: Bipolar credibility of the outranking relations in Example 2;

$r(\mathrm{S})$	x	y	z	t
x	–	–	0.2	1.0
y	–	–	−1.0	0.2
z	−0.2	1.0	–	–
t	−0.2	−0.2	–	–

The credibility degrees of the outranking relations between the two sets, as defined before, are: $r(C\,\mathrm{S}\,D) = 0.1$ and $r(D\,\mathrm{S}\,C) = 0.1$. As a result, the relation between the two sets can be considered as one of indifference. However, if we look at the individual preferential relations that can be constructed between the alternatives of the two sets, we find a different scenario. We notice that, alternative x outranks both the alternatives in D, since the credibility degrees of these relations are above the median level of 0, while the alternatives in D do not outrank x. Therefore x is preferred to z and t. When looking at alternative y, however, we notice that it is preferred only to alternative t, while alternative z from D is preferred to it.

We conclude that, due to the fact that no relations of indifference are exist between the alternatives of C and D, while three relations of preference going from C to D and only one from D to C are found, we may decide to state that C is also preferred to D.

Hence, another approach to constructing the preferential relations between two sets of alternatives, which is supported by the individual preferential relations between their alternatives, is to select the relation that appears with the highest frequency between them. This approach has also been used in [de Smet and Eppe 2009]. We define the credibility of any preferential relation $\mathrm{O} \in \{\mathrm{I}, \mathrm{P}, \mathrm{R}\}$ between two sets as:

$$r(C\,\mathrm{O}\,D) = \frac{1}{|C| \cdot |D| - \frac{|C \cap D|(|C \cap D| - 1)}{2}} \sum_{x \in C, y \in D, x < y} r^*(x\,\mathrm{O}\,y) \tag{4.14}$$

We have used the lexicographic operator $<$ in order to count each pair of alternatives only once, in the case of those from the intersection of the two sets.

Constructing the credibility degrees in this manner, however, is no longer in line with the credibility calculus we have previously defined. As there are four potential different types of relations that can appear between the alternatives of the two sets, it is reasonable to assume that in certain cases all of them may appear with a low frequency. Therefore, none of the four relations would hold a sufficiently high credibility, above the median level, in order to validate the relation. We need, in this case, to redefine the crisp credibility degree of any preferential relation $\mathrm{O} \in \{\mathrm{I}, \mathrm{P}, \mathrm{R}\}$ between two sets, r^*, as:

$$r^*(C\,\mathrm{O}\,D) = \begin{cases} 1 & \text{, if } r(C\,\mathrm{O}\,D) = \max\big(r(C\,\mathrm{I}\,D), r(C\,\mathrm{P}\,D), r(D\,\mathrm{P}\,C), r(C\,\mathrm{R}\,D)\big) \\ 0 & \text{, otherwise.} \end{cases} \tag{4.15}$$
$$\forall\, \mathrm{O} \in \{\mathrm{I}, \mathrm{P}, \mathrm{R}\}.$$

When working in a bipolar setting we can substitute the value 0 with -1.

We may further adapt this approach to also considering the degrees of credibility of the relations between the alternatives of the two sets as::

$$r(C\,\mathrm{O}\,D) = \frac{1}{|C| \cdot |D| - \frac{|C \cap D|(|C \cap D|-1)}{2}} \sum_{x \in C, y \in D, x<y} r(x\,\mathrm{O}\,y) \tag{4.16}$$

In this case too, we may not always have any of the existing relations holding a credibility degree higher than the median level, therefore the same approach as before, of constructing the relation between two sets, can be applied. This approach will also be used when we restrict the relation we can construct between two sets of alternatives to only a subset of $\mathrm{O} \in \{\mathrm{I}, \mathrm{P}, \mathrm{R}\}$. This will be required when searching for a particular type of structure in the clustering result.

4.2.3 Properties of the relations

In this subsection we discuss several properties of the relations between sets of alternatives which may be useful to have in the context of clustering.

Property 4.1 *The relation between any pair of clusters is restricted to exclude the indifference relation,* I.

Proof: *The definition of relational clustering in* MCDA *states that alternatives that are indifferent should be grouped together, while those that are not, should be separated, so that, for any pair of clusters, the relations between their alternatives are either of strict preference in one direction or the other, or of incomparability. Allowing for the creation of a relation of indifference between two clusters would go against both parts of this definition.* □

Property 4.2 *The binary relations between different clusters may be characterised solely based on the* P *relation.*

Proof: *Since the relation between two clusters can either be of a preference, in one direction or the other, or of incomparability, the absence of one of the first two implies the existence of the last.* □

Following this property, the potential structures that can be uncovered through clustering and characterised by the P relation are: partial or complete tournament, and strict partial or strict complete orders. These structures are depicted in Figure 4.1, where an arrow between two clusters represents the P relation.

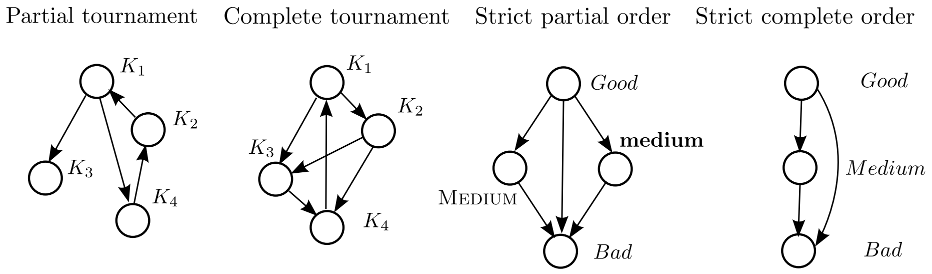

Partial tournament Complete tournament Strict partial order Strict complete order

Figure 4.1: Structures on the P relation;

Property 4.3 *The* P *relation is transitive on a set of clusters $K \in \mathcal{P}(X)$ iff:*

$$\max\big(r^*(C \not\mathrm{P}\, D), r^*(D \not\mathrm{P}\, E), r^*(C\, \mathrm{P}\, E)\big) = 1, \forall C, D, E \in K. \tag{4.17}$$

Proof: *The property of transitivity, for any binary relation* Q *over a set A, is:*

$$a\, \mathrm{Q}\, b \wedge b\, \mathrm{Q}\, c \rightarrow a\, \mathrm{Q}\, c\ , \forall a, b, c \in A$$

In our case, we have:

$$\begin{aligned}(C\, \mathrm{P}\, D \wedge D\, \mathrm{P}\, E) \rightarrow C\, \mathrm{P}\, E &\iff \neg(C\, \mathrm{P}\, D \wedge D\, \mathrm{P}\, E) \vee C\, \mathrm{P}\, E \\ &\iff (C \not\mathrm{P}\, D \vee D \not\mathrm{P}\, E) \vee C\, \mathrm{P}\, E\end{aligned}$$

Using our credibility calculus, this translates into:

$$\max\big(r^*(C \not\mathrm{P}\, D), r^*(D \not\mathrm{P}\, E), r^*(C\, \mathrm{P}\, E)\big) = 1.$$

□

This property is relevant in the context of ordered clustering.

Another property may be defined between pairs of sets of alternatives C and D where the first is strictly preferred to the second, *i.e.* $C \, \mathrm{P} \, D$. In such a case, finding an alternative from D that is strictly preferred to one from C may put serious doubt on the fact that $C \, \mathrm{P} \, D$. We define the property of preferential consistency in order to model these situations.

Any two sets of alternatives C and D are *preferentially consistent*, if a relation of strict preference can be expressed between then, while additionally no alternative from C is in a relation of strict preference with any alternative from D which contradicts the relation between the two sets. This property can easily be modelled through:

$$\min\left(r^*(C \, \mathrm{P} \, D), \max_{x \in C, y \in D} r^*(y \, \mathrm{P} \, x)\right) = 1. \tag{4.18}$$

This property is extended to a set of clusters, K, if any pair of distinct clusters in K, that are in a relation of preference, are preferentially consistent.

We acknowledge that searching for a clustering result which holds this property may be a very difficult task, in certain cases even impossible, especially when considering larger datasets. It may, however, prove useful to try to reduce the number of such contradictions between the proposed relation between two cluster and the individual relations between the alternatives inside them, and so, the property of preferential consistency may helpful in this case.

4.3 Clustering objectives

Following the definition we have given to clustering in MCDA, with its division into non-relational, relational and ordered subtypes, and the different ways of constructing relations between alternatives and sets of alternatives, we may now define several fitness functions that can be used to reflect the quality of a particular clustering result.

Many different ways of assessing the quality of a set of clusters have been proposed in the context of data analysis, a process which is also known as cluster validation [Jain and Dubes 1988]. We remind the three types of criteria that may be considered when validating a clustering result: internal, relative and external. The internal criteria use only the data itself in order to determine the quality of the clustering result, the relative criteria compare different structures together in order to select the best one, while external ones compare the set of clusters with a priori information.

In the following, we would like to define several fitness measures, that are based on two internal criteria.

The first criterion considers that a given number of clusters, k, is sought, which may be replaced through a single representative alternative for each cluster. In this case, we asses the degree of representation that each representative holds over the alternatives inside its cluster. For example, for a non-relational clustering result, we would like to have the representatives as indifferent as possible to the alternatives inside their clusters. Prefixing the number of clusters that are sought is essential in this case, otherwise the clustering result containing each alternative in the dataset as a separate cluster would maximize this type of fitness measure.

A second internal criterion models the fitness of clustering results containing any number of clusters. In order to be able to compare results with different numbers of clusters, we consider their fitness to be reflected by the degree with which each relation between two alternatives in the dataset fits in the structure that the clustering result proposes. For instance, if we are given a non-relational clustering result, then ideally, all the alternatives inside the same cluster would be indifferent with each other, while all those that are in different clusters would not be indifferent. Furthermore, the degrees of credibility of these relations would be maximal.

In the following, we will denote with K the set of clusters, which is of cardinality k, while for each cluster C in K we denote its representative with r_C. All cluster representatives are contained in the set R.

4.3.1 Non-relational clustering

We start with the problem of non-relational clustering, where we are looking to group alternatives that are indifferent and to separate those that are not.

In certain decision aiding contexts, such as that of eliciting the parameters of a preference model, only the use of crisp statements with respect to the credibility of an outranking relation between certain pairs of alternatives is possible [Dias and Mousseau 2006]. Therefore, we also propose to model the objective of non-relational clustering through the following fitness functions:

$$f^*_{\mathrm{NR\text{-}R}}(K,R) := \frac{1}{n} \cdot \sum_{C \in K} \sum_{x \in C} r^*(x \,\mathrm{I}\, r_C) \tag{4.19}$$

$$f^*_{\mathrm{NR}}(K) := \frac{2}{n(n-1)} \cdot \Big(\sum_{C \in K} \sum_{\substack{x,y \in C \\ x > y}} r^*(x \,\mathrm{I}\, y) + \sum_{\substack{C,D \in K \\ C > D}} \sum_{\substack{x \in C \\ y \in D}} r^*(x \,\not{\mathrm{I}}\, y) \Big). \tag{4.20}$$

In the formula above, the first fitness measure considers the first internal criterion, by measuring the average crisp credibility degree between the alternatives inside a cluster and the representative of that cluster.

The second fitness measure is divided in two parts. First we account for the relations of indifference between the alternatives inside the same cluster, while the second term considers the relations of non-indifference between the alternatives from different clusters. The formula is then brought to a $[0,1]$ value range, when the underlying outranking relation is defined using a boolean credibility, or to a $[-1, 1]$ interval in the case of bipolar-valued outranking relations.

We may extend these formulas so that the degrees of credibility attached to the indifference relations are also considered. The fitness measures then become:

$$f_{\text{NR-R}}(K, R) := \frac{1}{n} \cdot \sum_{C \in K} \sum_{x \in C} r(x \,\mathrm{I}\, r_C) \tag{4.21}$$

$$f_{\text{NR}}(K) := \frac{2}{n(n-1)} \cdot \Big(\sum_{C \in K} \sum_{\substack{x,y \in C \\ x > y}} r(x \,\mathrm{I}\, y) + \sum_{\substack{C,D \in K \\ C > D}} \sum_{\substack{x \in C \\ y \in D}} r(x \,\not\mathrm{I}\, y) \Big). \tag{4.22}$$

Clustering by trying to maximize these functions may correctly place alternatives that are strongly indifferent inside the same cluster, although certain alternatives that are not indifferent may also be grouped with them. The high credibility degrees of certain relations of indifference may compensate the weaker ones.

The following fitness measures address this issue, although they may prove harder to improve over difficult datasets:

$$f^{min}_{\text{NR-R}}(K, R) := \min_{C \in K} \min_{x \in C} r(x \,\mathrm{I}\, r_C) \tag{4.23}$$

$$f^{min}_{\text{NR}}(K) := \min \Big(\min_{C \in K} \big(\min_{\substack{x,y \in C \\ x > y}} r(C \,\mathrm{I}\, C) \big), \min_{\substack{C,D \in K \\ C > D}} \big(\min_{\substack{x \in C \\ y \in D}} r(C \,\not\mathrm{I}\, D) \big) \Big). \tag{4.24}$$

This characterisation of the non-relational clustering objective is the strictest, as the lowest credibility degree with respect to a desired relation between any alternative and a representative, or between any two alternatives in the dataset gives the overall fitness of the clustering result.

4.3.2 Relational and ordered clustering

In the case of relational clustering, in addition to grouping indifferent alternatives and separating those that are not indifferent, we also wish to have the alternatives in a group compare together in the same way with respect to any other alternative in the dataset.

The fitness measure with respect to the first internal criterion is:

$$\begin{aligned} f^*_{\text{R-R}}(K,R) :=& \frac{1}{n \cdot k} \cdot \Bigg(\sum_{C \in K} \sum_{x \in C} r^*(x \,\mathrm{I}\, r_C) + \\ & \sum_{\substack{C,D \in K \\ C \neq D}} \sum_{x \in C} \max \Big(\min \big(r^*(r_C \,\mathrm{P}\, r_D), r^*(x \,\mathrm{P}\, r_D) \big), \\ & \qquad \min \big(r^*(r_D \,\mathrm{P}\, r_C), r^*(y \,\mathrm{P}\, r_D) \big), \\ & \qquad \min \big(r^*(r_C \,\mathrm{R}\, r_D), r^*(x \,\mathrm{R}\, r_D) \big) \Big) \Bigg) \end{aligned} \tag{4.25}$$

In this case, we firstly consider the crisp credibility degrees of the relations of indifference between the alternatives in each cluster and the representative of that cluster, *i.e.* we desire that the alternatives in a cluster are indifferent to the cluster representative. Secondly, we measure how similarly, with respect to the representative of their cluster, they compare to the other cluster representatives. In order to measure this, we consider each of the three types of relations that may be placed between for any two clusters, excluding the relation of indifference. For each of these relations, we then apply a min operation over the crisp credibility degree between the cluster representatives, and the crisp credibility degree between the a considered alternative from the first cluster and the representative of the second. For example, considering the P relation, if the relations between r_C and r_D is P, *i.e.* $r^*(r_C \,\mathrm{P}\, r_D) = 1$, then the result of the min operation will be $r^*(x \,\mathrm{P}\, r_D)$, where $x \in C$. On the other hand, if $r^*(r_C \,\mathrm{P}\, r_D) = -1$, then no matter what value we have for $r^*(x \,\mathrm{P}\, r_D)$, we will get the minimum value of -1 from this operation. By applying a max operation over the results of the previously presented operation over the three types of relations, we find the crisp credibility degree of the type of relation that is present between r_C and r_D, but between x and r_D. In this example, we have considered that r^* is defined in a bipolar setting.

In a similar fashion, we present below the fitness measure with respect to the second internal criterion:

$$\begin{aligned} f^*_{\text{R}}(K) :=& \frac{2}{n(n-1)} \cdot \Bigg(\sum_{C \in K} \sum_{\substack{x,y \in C \\ x > y}} r^*(x \,\mathrm{I}\, y) + \\ & \sum_{\substack{C,D \in K \\ C > D}} \sum_{\substack{x \in C \\ y \in D}} \max \Big(\min \big(r^*(C \,\mathrm{P}\, D), r^*(x \,\mathrm{P}\, y) \big), \\ & \qquad \min \big(r^*(D \,\mathrm{P}\, C), r^*(y \,\mathrm{P}\, x) \big), \\ & \qquad \min \big(r^*(C \,\mathrm{R}\, D), r^*(x \,\mathrm{R}\, y) \big) \Big) \Bigg) \end{aligned} \tag{4.26}$$

We find that this measure is very similar to the previous one, only that we consider the relations between the clusters of alternatives, and not their representatives, and also asses the relations between all pairs of alternatives.

The following four formulas adapt the presented fitness measures in order to account for the valued credibility degrees:

$$\begin{aligned} f_{\text{R-R}}(K,R) :=& \frac{1}{n \cdot k} \cdot \Bigg(\sum_{C \in K} \sum_{x \in C} r(x \, \mathrm{I} \, r_C) + \\ & \sum_{\substack{C,D \in K \\ C \neq D}} \sum_{x \in C} \max \Big(\min \big(r^*(r_C \, \mathrm{P} \, r_D), r(x \, \mathrm{P} \, r_D) \big), \\ & \min \big(r^*(r_D \, \mathrm{P} \, r_C), (y \, \mathrm{P} \, r_D) \big), \\ & \min \big(r^*(r_C \, \mathrm{R} \, r_D), (x \, \mathrm{R} \, r_D) \big) \Big) \Bigg) \end{aligned} \tag{4.27}$$

$$\begin{aligned} f_{\mathrm{R}}(K) :=& \frac{2}{n(n-1)} \cdot \Bigg(\sum_{C \in K} \sum_{\substack{x,y \in C \\ x > y}} r^*(x \, \mathrm{I} \, y) + \\ & \sum_{\substack{C,D \in K \\ C > D}} \sum_{\substack{x \in C \\ y \in D}} \max \Big(\min \big(r^*(C \, \mathrm{P} \, D), r(x \, \mathrm{P} \, y) \big), \\ & \min \big(r^*(D \, \mathrm{P} \, C), r(y \, \mathrm{P} \, x) \big), \\ & \min \big(r^*(C \, \mathrm{R} \, D), r(x \, \mathrm{R} \, y) \big) \Big) \Bigg) \end{aligned} \tag{4.28}$$

$$\begin{aligned} f_{\text{R-R}}^{min}(K,R) :=& \min \Bigg(\min_{C \in K} \min_{x \in C} r^*(x \, \mathrm{I} \, r_C), \\ & \min_{\substack{C,D \in K \\ C \neq D}} \min_{x \in C} \max \Big(\min \big(r^*(r_C \, \mathrm{P} \, r_D), r^*(x \, \mathrm{P} \, r_D) \big), \\ & \min \big(r^*(r_D \, \mathrm{P} \, r_C), r^*(y \, \mathrm{P} \, r_D) \big), \\ & \min \big(r^*(r_C \, \mathrm{R} \, r_D), r^*(x \, \mathrm{R} \, r_D) \big) \Big) \Bigg) \end{aligned} \tag{4.29}$$

$$\begin{aligned} f_{\mathrm{R}}^{min}(K) :=& \min \Bigg(\min_{C \in K} \min_{\substack{x,y \in C \\ x > y}} r^*(x \, \mathrm{I} \, y), \\ & \min_{\substack{C,D \in K \\ C > D}} \min_{\substack{x \in C \\ y \in D}} \max \Big(\min \big(r^*(C \, \mathrm{P} \, D), r^*(x \, \mathrm{P} \, y) \big), \\ & \min \big(r^*(D \, \mathrm{P} \, C), r^*(y \, \mathrm{P} \, x) \big), \\ & \min \big(r^*(C \, \mathrm{R} \, D), r^*(x \, \mathrm{R} \, y) \big) \Big) \Bigg) \end{aligned} \tag{4.30}$$

The last subtype of clustering problems in MCDA is finding an ordered set of clusters. This objective may be modelled using the fitness functions from relational clustering, to which we add a mean of determining if the result is indeed an order or not. If this is not the case, then the fitness function should be lowered to the minimum value.

Following Property 4.3, we construct an indicator for having a transitive relation between the clusters in K as:

$$Tr(K) := \min_{\substack{C,D,E \in K \\ C \neq D \neq E}} \max\big(r^*(C \not\mathrm{P} D), r^*(D \not\mathrm{P} E), r^*(C \,\mathrm{P}\, E)\big). \tag{4.31}$$

In the case of the first internal criterion, which uses the cluster representatives, the relations between the clusters, from the equation above, will be given directly by the relations between the cluster representatives.

The fitness functions for ordered clustering may then be written as:

$$\begin{aligned}
f^*_{\text{O-R}}(K,R) &:= \min\big(Tr(K), f^*_{\text{R-R}}(K,R)\big), &(4.32)\\
f_{\text{O-R}}(K,R) &:= \min\big(Tr(K), f_{\text{R-R}}(K,R)\big), &(4.33)\\
f^{min}_{\text{O-R}}(K,R) &:= \min\big(Tr(K), f^{min}_{\text{R-R}}(K,R)\big), &(4.34)\\
f^*_{\text{O}}(K) &:= \min\big(Tr(K), f^*_{\text{R}}(K)\big), &(4.35)\\
f_{\text{O}}(K) &:= \min\big(Tr(K), f_{\text{R}}(K)\big), &(4.36)\\
f^{min}_{\text{O}}(K) &:= \min\big(Tr(K), f^{min}_{\text{R}}(K)\big). &(4.37)
\end{aligned}$$

4.3.3 Additional objectives

We may further formulate certain relational clustering objectives where the relations of strict preference between clusters need to be preferentially consistent. In order to model this we consider counting the relations that go against this property through:

$$\begin{aligned}
f^*_{\text{PC-R}}(K,R) := \sum_{\substack{C,D \in K \\ C \neq D}} \Big(&\sum_{x \in C} \min\big(r^*(r_C \,\mathrm{P}\, r_D), r^*(r_D \,\mathrm{P}\, x)\big) + \\
&\sum_{x \in D} \min\big(r^*(r_C \,\mathrm{P}\, r_D), r^*(x \,\mathrm{P}\, r_C)\big)\Big).
\end{aligned} \tag{4.38}$$

$$f^*_{\text{PC}}(K) := \sum_{\substack{C,D \in K \\ C \neq D}} \sum_{\substack{x \in C \\ y \in D}} \min\big(r^*(C \,\mathrm{P}\, D), r^*(y \,\mathrm{P}\, x)\big). \tag{4.39}$$

The first measure counts the relations between the alternatives in the first cluster which are strictly preferred by the representative of the second clusters, considering that the first cluster representative is strictly preferred to the second, but also the alternatives from the second cluster which are strictly preferred to the first cluster representative. Every ordered pair of distinct clusters is considered.

In the case of the second measure, the relations of strict preference between alternatives in the second cluster and those in the first cluster are taken into account, considering that the first cluster is considered to be strictly preferred to the second.

These formulas have been expressed in the context of using a binary credibility calculus. In case r^* is defined in a bipolar setting, the terms in each equation need to be brought to a $[0,1]$ value range.

In order to take into account also the credibility degrees of these relations that oppose the property of preferential consistency, we propose the following measures below:

$$\begin{aligned} f_{\text{PC-R}}(K,R) := \frac{1}{f^*_{\text{PC-R}}(K,R)} \cdot \sum_{\substack{C,D \in K \\ C \neq D}} \Bigg(& \sum_{x \in C} \max\Big(\min\left(r^*(r_C \mathrm{P} r_D), r^*(r_D \mathrm{P} x), r(r_D \mathrm{P} x) \right)\Big) + \\ & \sum_{x \in D} \max\Big(\min\left(r^*(r_C \mathrm{P} r_D), r^*(x \mathrm{P} r_C), r(x \mathrm{P} r_C) \right)\Big)\Bigg). \end{aligned} \tag{4.40}$$

$$f_{\text{PC}}(K) := \frac{1}{f^*_{\text{PC}}(K)} \cdot \sum_{\substack{C,D \in K \\ C \neq D}} \sum_{\substack{x \in C \\ y \in D}} \max\Big(\min\left(r^*(C \,\mathrm{P}\, D), r^*(y \,\mathrm{P}\, x), r(y \,\mathrm{P}\, x) \right)\Big). \tag{4.41}$$

We have included the credibility degrees of the relations that contradict the property of struct preference inside the min blocks so that when both the relation between the two clusters is one of strict preference, and in addition the relation between an alternative from the second cluster and one from the first cluster is also one of strict preference, the minimum between them is in fact the credibility degree of the relation that contradicts the property. Similar remarks may be made with respect to the first of the two equations from above.

The first factor in these equations has the role of bringing these measures to a $[0,1]$. The 1 value is achieved when all the relations that contradict the property of preferential consistency have the maximum credibility degree equal to 1. In this case too, we have only considered the use of boolean credibility degrees, adjustments being required when using bipolar credibility degrees.

Finally, we may consider even stricter measures, which retrieve the highest credibility degree among the relations that contradict the property of preferential consistency:

$$f^{max}_{\text{PC-R}}(K,R) := \max_{\substack{C,D\in K\\ C\neq D}} \Big(\max_{x\in C} \min\big(r^*(r_C \operatorname{P} r_D), r^*(r_D \operatorname{P} x), r(r_D \operatorname{P} x)\big),$$
$$\max_{x\in D} \min\big(r^*(r_C \operatorname{P} r_D), r^*(x \operatorname{P} r_C), r(x \operatorname{P} r_C)\big)\Big). \tag{4.42}$$

$$f^{max}_{\text{PC}}(K) := \max_{\substack{C,D\in K\\ C\neq D}} \max_{\substack{x\in C\\ y\in D}} \min\big(r^*(C \operatorname{P} D), r^*(y \operatorname{P} x), r(y \operatorname{P} x)\big). \tag{4.43}$$

All of the above presented measures need to be minimized in order to come closer to a preferentially consistent relational clustering result. This may be tackled as a multiple objective optimization problem, where on the one hand, one of the preferentially consistent fitness functions needs to be minimized, while on the other, a fitness function related to a particular clustering objective needs to be maximized.

Summary

In this chapter, we have focused on the task of modelling the problem of clustering in MCDA. This has been done in the context of having chosen as an evaluation model an outranking relation. Following the definition of a credibility calculus, we have consistently extended the credibility of any outranking relation into the credibilities of the three relations that can be built from then in this context: indifference, preference and incomparability.

A number of approaches of extending these relations and their credibility degrees to the case of sets of alternatives have then been proposed. Following this, several properties of these relations have been analysed.

The three classes of clustering problems in MCDA have then been modelled through the use of several fitness functions, which reflected the quality of a clustering result with different degrees of confidence. The least restrictive of them take into account only the crisp credibility degrees, the second account also for the actual credibility degrees, by using a mean operator, while the last considers that weakest credibility degree that supports the proposed structure as the overall fitness of the clustering result. Each of these fitness functions may be of interest in different practical applications.

Chapter 5

Solving the clustering problem

Contents

Abstract

Based on the different ways in which the problem of clustering in MCDA can be modelled, we proceed to proposing several approaches for solving it.

We begin with a few general remarks on the process of solving clustering in MCDA, and then consider the existing clustering approaches, and how well they fit with the way in which this problem has been defined in this paper.

We follow with adapting the classical K-MEANS algorithm to this case, and, as it is considered as a particular type of the self-organizing maps, as defined by KOHONEN, we explore several new map topologies, that would be well-suited to clustering in an MCDA context.

Finally, a second type of approach is presented, one which is inspired from Graph theory. Furthermore, this approach is a combination of a graph-theoretic clustering approach and a meta-heuristic, which may be used to find any of the presented types of clustering structures.

5.1 Initial considerations

5.1.1 On solving the problem

Most often, clustering problems are considered to be very difficult to solve. Although the problem of clustering in a single dimension has been shown to be solvable in a polynomial time with respect to the size of the problem [Fisher 1958], most of the studied problems that consider more than one dimension are considered NP-hard. For instance in [Gonzalez 1982], several definitions of the problem of clustering into k groups are studied. The results show that, considering several fitness functions, the problems are NP-hard even when the points that need to be grouped are restricted a two-dimensional euclidian space. More recently, the problem attached to the K-means clustering algorithm has been show to be NP-hard when only two dimensions are considered [Mahajan et al. 2009].

We consider the topic related to the complexity of the clustering problem in the MCDA field, as it has been defined in this work, to be out of our current scope. Nevertheless, due to the fact that this problem can be reduced to a classical clustering problem, by removing the preferential information that exists on each criterion, we may consider that the problem of clustering in MCDA is also difficult to solve. Furthermore, the added complexity of the evaluation models used, namely outranking relations, which allow for the construction of structures that lack transitivity over the relations of strict preference, or that contain relations of incomparability, increase the difficulty of solving this problem.

As there are no constraints with respect to how the alternatives have to be grouped together, which would in some way reduce the complexity of this problem, the only clear way of finding the optimal solution, with respect to a particular fitness measure, is to consider the entire polytope of potential solutions. This means that all possible partitions need to be explored, leading to the selection of the best one with respect to a considered fitness measure. As the number of partitions of a set is equal to the Bell number, such an approach of solving this problem is clearly exponential in complexity.

An efficient way of enumerating these partitions, both with respect to time and space complexity, may be found in [Orlov 2002].

5.1.2 Existing and new approaches

As previously mentioned in Chapter 3, most of the clustering approaches developed around the field of MCDA have been using concepts and measures that are not native to this field, but to that of Data analysis. In the context of MCDA, the information that accompanies the attributes that define the objects is broader, containing in addition the preference of a DM on them.

Several existing approaches do in fact tackle this problem, not by using measures of similarity, but measures of indifference, strict preference and incomparability.

Among these approaches we mention the algorithms proposed in [de Smet and Guzman 2004, de Smet and Eppe 2009, Nemery and de Smet 2005, Fernandez et al. 2010].

de Smet and Guzman, to our knowledge, were the first to extend the classical K-means algorithm to the mcda context in [de Smet and Guzman 2004], which was later further extended in [de Smet and Eppe 2009] so that relations between clusters can also be expressed. In this work, they proposed to use, instead of a Euclidian distance, a distance that is constructed from the similarity between the relations that two alternatives have towards all the alternatives in the dataset. This approach holds a few shortcomings, as by adding or removing alternatives from the dataset, the distances between the alternatives that remain change, although the alternatives themselves remain the same.

The approach from Nemery and de Smet in [Nemery and de Smet 2005], considering the definition of only the preference degrees between the alternatives, may be seen as an ordered clustering approach, based on our definitions. Furthermore, through the use of the coherence indicator, it may also be seen to improve the property of preferential consistency that we have herein defined. However, this approach is not suited for evaluation models that allow for incomparability between alternatives to occur. The output of the method is also restricted to a strict complete order between the clusters.

Fernandez *et al.* have advocated the use of an indifference measure in order to group the alternatives together, and have furthermore defined this measure in [Fernandez et al. 2010] as we have. However, they constructed a clustering approach using only this measure, and did not define measures for the relations of preference and incomparability. The approach they propose can be therefore seen as a non-relational clustering approach. A parallel can easily be drawn between a K-medoids algorithm, and the approach they propose, although certain interactions occur between the cluster centroids. Furthermore, the assignment of the alternatives to a cluster is also conditioned by several neighbourhood parameters. An order between the constructed clusters is proposed, either based on the relations between the cluster centroids or the relations between the alternatives inside the clusters. Hence, the method may also be seen as an ordered clustering approach, where a strict complete order is proposed, although this structure is not actively constructed during the clustering process.

It may be observed that several existing mcda clustering approaches adhere, with higher or lower degrees, to certain definitions that we have given to clustering in this field. However, not all the potential structures can be uncovered through them, an issue which we wish to address further.

In the remainder of this chapter, we will proceed to defining several algorithms which may be used for solving the mcda clustering problems, as defined in this paper.

We will begin with an extension of the most popular classical clustering algorithm, the K-means algorithm. This will be done in the context of K-means being a particular case of the Kohonen self-organizing maps [Kohonen et al. 2001]. Therefore, we will also consider extending the self-organizing maps to the MCDA context, by proposing several map topologies which would be well-suited in this case.

Finally, we will also consider an approach inspired from Graph theory.

5.2 Self-organizing maps for MCDA clustering

5.2.1 General description

The general idea behind the self-organizing maps (SOM), which were proposed by Kohonen in [Kohonen et al. 2001], is to map a dataset, containing objects in an m-dimensional space, onto a lower dimensional network of neurons. In addition, certain topological properties of this network are enforced. The notion of neighbourhood is essential in this case, as neurons that are connected will have a tendency to remain closer to each other. Therefore a SOM may be seen, when mapping on a certain dataset, as a compromise between trying to determine the structure of the data and trying to validate the structure given through the topology of the map. The trade-off between these two objectives is usually given by a learning rate which affects the level of impact of neighbouring neurons and which decreases over time.

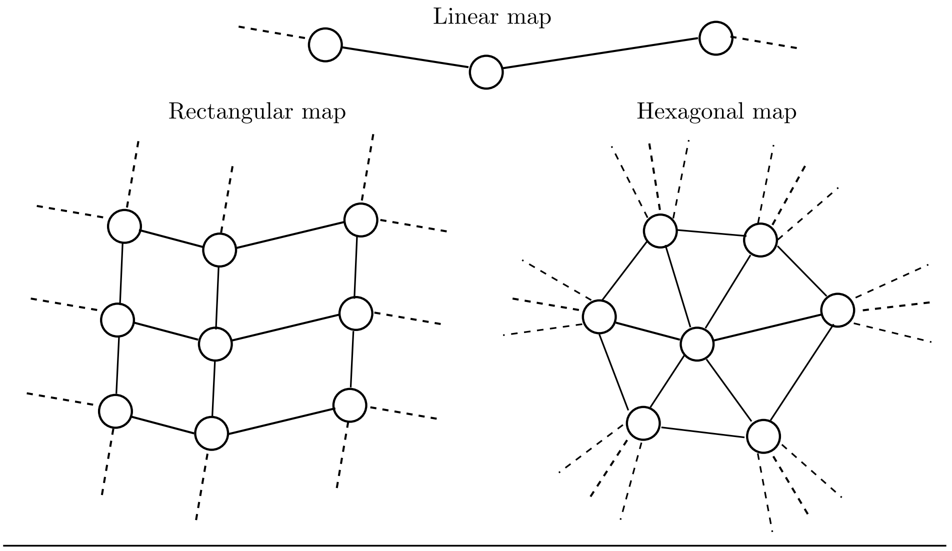


Figure 5.1: Common topologies of SOMs;

In order to allow for an easy visualisation, the SOMs are usually represented in one or two dimensions. The one-dimensional SOMs are in the form of a chain, while the two-dimensional SOMs form either a rectangular or a hexagonal grid.

We will denote with $C = \{c_1, ..., c_k\}$ the set of k neurons of the map, and with $N(c_l, r)$, $\forall l \in 1..k$, the neighbourhood of the node c_l considering also a radius parameter r.

The main structure of the algorithm is presented in Algorithm 2.

Algorithm 2 SOM algorithm;

```
Input: $k$
Output: $K$
$\alpha \leftarrow \textsc{InitLearningRate}()$
$r \leftarrow \textsc{InitNRadius}()$
$C \leftarrow \textsc{InitSOM}(k)$
while $alpha > 0$ do
    $M \leftarrow \{\}$
    for all $x \in X$ do
        $d \leftarrow \{\}$
        for $l = 1$ to $k$ do
            $d_l \leftarrow \textsc{dist}(x, c_l)$
        $i \leftarrow \textsc{SelectMinDistIndex}(d)$
        $M_i \leftarrow M_i \cup x$
    for $l = 1$ to $k$ do
        $\textsc{Update}(c_l, M_l, \mathrm{N}(c_l, r), \alpha)$
    $\alpha \leftarrow \textsc{UpdateLearningRate}()$
    $r \leftarrow \textsc{UpdateNRadius}()$
return $K$
```

The algorithm starts with an initialization of the neighbourhood and learning rate parameters, but also placing the nodes of the map in the feature space. This may be done either randomly, or by spreading it uniformly across the feature space.

A number of iterations then follow. During each iteration, the objects are mapped to the node in the network that they are closest to. Based on this assignment, the map nodes are then updated. Usually this is done using the average of the evaluations of the objects that are assigned to each node, but also factoring in the evaluations of the objects assigned to neighbouring nodes, in a proportion equal to the learning rate parameter. After the nodes have been updated, the learning rate and neighbourhood size parameters are also adjusted, and a new iteration begins. The learning rate and neighbourhood size parameters are usually decreased across these iterations, so that the effects of the objects that are assigned to neighbouring nodes in the map get weaker.

Following the lapse of a certain amount of time, the completion of a certain number of iterations, or in this case, the decrease of the learning rate parameter to the minimum level, the algorithm stops, and the last mapping of the objects is retrieved.

Based on the description above, using a SOM can be easily seen as a clustering approach, if we consider that each node is a cluster centroid. The K-MEANS algorithm is, in fact, a particular case of SOMs, where all the nodes in the map are completely disconnected from each other. In this case, only the objects that are assigned to a specific node have an effect on it.

5.2.2 K-MEANS and K-MEDOIDS

We begin with the simplest case of SOMs, the K-MEANS algorithm, and try to adapt it to the context of clustering in MCDA.

K-MEANS generally uses a measure of similarity, or a distance measure, in order to propose a pre-fixed number of clusters which minimize the distance between the objects and the centroid of the cluster to which they are assigned. As the name implies, the centroids are constructed from the mean evaluations of the objects inside a cluster, although a variant that selects the most centrally located object, the K-MEDOIDS, has also been developed for contexts where the mean cannot be defined. We will extend both these algorithms to the MCDA case.

We begin by defining a distance based on the credibility of the relation of indifference, which is the measure that is used in this case to group alternatives together. Given the credibility r of the relation of indifference between two alternatives $x, y \in X$, from Equation 4.1.2, we define the distance between two alternatives as follows:

$$d_I(x,y) = 1 - r(x\,\mathrm{I}\,y) \text{ , for } r(x\,\mathrm{I}\,y) \in [0,1], \tag{5.1}$$

$$d_I(x,y) = 1 - \frac{r(x\,\mathrm{I}\,y)+1}{2} \text{ , for } r(x\,\mathrm{I}\,y) \in [-1,1]. \tag{5.2}$$

We have highlighted above how this distance is constructed given either a boolean credibility or a bipolar credibility attached to the relation of indifference.

As the number of clusters, k, is usually given, the fitness of a set of clusters given by this approach is given by the average distance from the alternatives in a cluster to the centroid of that cluster:

$$f_I(K) = \frac{1}{n} \sum_{l \in 1..k} \sum_{x \in K_l} d_I(x, c_l). \tag{5.3}$$

We also highlight the way in which the centroids are constructed. In the case of the K-MEANS adaptation, the centroid is given by the mean of the evaluations of the alternatives inside the cluster on the criteria defined on quantitative scales and by the median on those defined on qualitative scales:

$$c_{l\,i} = \frac{1}{|K_l|} \cdot \sum_{x \in K_l} x_i, \forall i \in F. \tag{5.4}$$

In the case of the K-MEDOIDS adaptation, the centroid is selected as the alternative inside the cluster that minimizes the average distance to the other alternatives in the cluster:

$$c_l = \arg\min_{x \in K_l} \sum_{y \in K_l} d_I(x,y). \tag{5.5}$$

We highlight the main steps of the K-MEANS algorithm in Algorithm 3.

Algorithm 3 K-MEANS algorithm;

```
Input: $k$
Output: $K$
$C \leftarrow$ InitializeCentroids($k$)
while not BreakCondition() do
    $K \leftarrow$ ConstructPartition($C$)
    $C \leftarrow$ UpdateCentroids($K$)
return $K$
```

In the case of the adapted algorithms, the initialization can be done either by selecting randomly alternatives from the dataset as the initial centroids, or selecting the first one at random and then selecting each consecutive one as the farthest from the centroids selected thus far.

The breaking condition for the algorithm can be the completion of prefixed number of iteration, the lapse of a certain amount of time, or having minimal changes to the centroids for a certain number of iterations.

The construction of the partition around the centroids is done in the same way for both the K-MEANS adaptation and the K-MEDOIDS one, by placing each alternative in the cluster corresponding to the centroid that is closest to it.

The centroids are then updated following Equation (5.4) in the case of the K-MEANS adaptation, or following Equation (5.5) in the case of the K-MEDOIDS adaptation.

We denote the first algorithm with D-SOM$_{\text{AVG}}$ and the second with D-SOM$_{\text{MED}}$, as both algorithms correspond to **disconnected self-organizing maps**.

The time complexity of both algorithm, as with the classical K-MEANS algorithm, is linear with respect to the number of alternatives that are clustered.

5.2.3 New topologies and adaptivity

We continue by considering self-organizing maps that hold certain topologies over the map nodes. In order for them to reflect a particular desired clustering structure, we propose that they are linked through relations of strict preference, so that an order between them may be constructed. In Figure 5.2 we highlight two potential topologies.

The first only links the neighbouring nodes so that a chain is constructed, while the second contains links between all the nodes of the map. We call the first a **chained self-organizing map** and the second an **ordered self-organizing map**.

We first take the case of the chained self-organizing map and detail it further. Considering a map of k nodes, we denote it through $C = \{c_1, c_2, ..., c_k\}$. The last

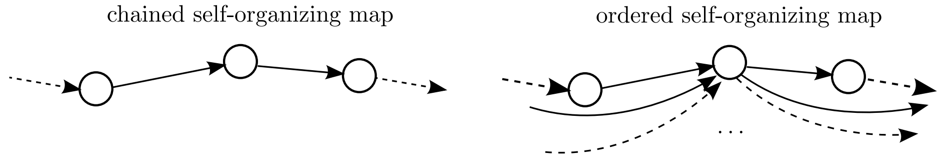


Figure 5.2: New topologies of SOMs;

node, c_k, is considered to be the bottom one, while the first one is the top one. Each node $c_l \in C$ is connected to the nodes c_{l-1} and c_{l+1}, except for the first and last nodes which only connect to c_2, respectively to c_{k-1}. The relation between any node c_l and its neighbours is:

$$c_l \,\mathrm{P}\, c_{l+1}, \forall l \in 1..k-1, \tag{5.6}$$

$$c_l \,\not\mathrm{P}\, c_{l-1}, \forall l \in 2..k. \tag{5.7}$$

We define the mapping of the alternatives from X to the nodes from C through $K = \{K_1, ..., K_k\}$, where K is constructed by assigning each alternative in X to the closest node in the map based on one of the distance measures defined in the previous subsection.

The difference from traditional SOMs lies in the effect that objects assigned to a node in the map have on the neighbouring nodes. Classically, an object assigned to a particular node has an effect of pulling towards it both the node it has been assigned to and the neighbours of this node, however, in our case, it will pull towards it the node it has been assigned to, but push away its neighbours based on the type of relation between the current node and its neighbours. In this way, the objective is to map each node in a dense region with respect to the relations of indifference, but also in a way in which these nodes are ordered with respect to the relations of strict preference that give the topology of the map.

We adopt two procedures for constructing the nodes from a given mapping, similar to the ones used in the case of the K-MEANS and K-MEDOIDS adaptations discussed before. The first constructs the node from the evaluations of the alternatives that were mapped to it and those of the neighbouring nodes, while the second selects one of the alternatives that were mapped to it based on a fitness measure.

The first approach is highlighted below:

$$\begin{aligned} &1) \quad c_{li} = \frac{1}{|K_l|} \cdot \sum_{x \in K_l} x_i, \forall i \in F; \\ &2) \quad c_{li} = \frac{\Big(|K_l| \cdot c_{li} + \sum\limits_{\substack{x \in K_{l+1}: \\ x_i > c_{li}}} x_i + \sum\limits_{\substack{x \in K_{l-1}: \\ x_i < c_{li}}} x_i\Big)}{|K_l| + \big|\{x \in K_{l+1}\colon\ x_i > c_{li}\}\big| + \big|\{x \in K_{l-1}\colon\ x_i < c_{li}\}\big|}, \forall i \in F. \end{aligned} \tag{5.8}$$

Like in the K-MEANS adaptation, initially the node is constructed as the mean of the evaluations of the alternatives mapped to it. However, as we additionally wish to take into account the alternatives from the neighbouring node, a second step is further defined. In this step, the evaluations of the node on each criterion $i \in F$ are updated to take into account also the evaluations of the alternatives that were assigned to the node below the current, and of those assigned to the node above. However, only the evaluations that are above the evaluations of the current node are taken into account in the case of the alternatives from the lower node, while only the evaluations that are below evaluations of the current node are taken into account in the case of the alternatives from the upper node.

We have omitted, in the above equations, the learning rate parameter, in order to make these equations easier to read, which should be multiplied with all the elements related to the neighbouring nodes.

The second approach to building the node is to select an existing alternative that was assigned to the current one following the formula below:

$$c_l = \arg\max_{x \in K_l} \Big(\sum_{y \in K_l} r(x \,\mathrm{I}\, y) + \sum_{y \in K_{l+1}} r(x \,\mathrm{P}\, y) + \sum_{y \in K_{l-1}} r(y \,\mathrm{P}\, x) \Big) \tag{5.9}$$

This formula is rather easy to read. An alternative from K_l is selected as c_l based on how much it is indifferent to the other alternatives in K_l, strictly preferred by those from upper cluster K_{l-1} and strictly preferred to those from the lower cluster K_{l+1}. The learning rate parameter should be added to the last two terms in order to complete the formula.

The algorithms of clustering through the use of chained self-organizing maps follow the general outline of the self-organizing maps, and are denoted with C-SOM$_{\text{AVG}}$ and C-SOM$_{\text{MED}}$, depending on the two approaches to constructing the nodes, which have been previously described.

In the case of the ordered self-organizing maps, each node in the map c_l is connected to all the other ones so that:

$$c_l \,\mathrm{P}\, c_m, \forall l \in 1..k-1, m > l, \tag{5.10}$$

$$c_l \,\not\mathrm{P}\, c_m, \forall l \in 2..k, m < l. \tag{5.11}$$

All the considerations made for chained self-organizing maps apply also to the ordered self-organizing maps, with the formulas used for updating the nodes in the map being updated below.

The formulas for constructing a node from the evaluations of the alternatives mapped to it and those of the alternatives mapped to its neighbours become:

$$\begin{aligned}
&1)\quad c_{li} = \frac{1}{|K_l|} \cdot \sum_{x\in K_l} x_i, \forall i \in F;\\
&2)\quad c_{li} = \frac{\Big(|K_l| \cdot c_{li} + \sum\limits_{m>l} \sum\limits_{\substack{x\in K_m:\\ x_i>c_{li}}} x_i + \sum\limits_{m<l} \sum\limits_{\substack{x\in K_m:\\ x_i<c_{li}}} x_i\Big)}{|K_l| + \big|\{x\in K_m\colon\ m>l, x_i>c_{li}\}\big| + \big|\{x\in K_m\colon\ m<l, x_i<c_{li}\}\big|}, \forall i\in F.
\end{aligned} \tag{5.12}$$

The formula for the second approach to updating the map nodes becomes:

$$c_l = \operatorname*{arg\,max}_{x\in K_l} \Big(\sum_{y\in K_l} r(x\,\mathrm{I}\,y) + \sum_{m>l} \sum_{y\in K_m} r(x\,\mathrm{P}\,y) + \sum_{m<l} \sum_{y\in K_m} r(y\,\mathrm{P}\,x) \Big) \tag{5.13}$$

The clustering algorithms that use ordered self-organizing maps are denoted with O-SOM$_{\text{AVG}}$ and O-SOM$_{\text{MED}}$, depending on the previously described approaches for updating the nodes.

We remind the fact that these topologies of self-organizing maps should be used in order to extract ordered sets of clusters containing indifferent alternatives.

It may be noticed that a parameter related to the size of the map is required in when employing the use of a clustering approach based on self-organizing maps. In the cases we have described, this is given directly by the number of clusters that we wish to extract. However, in certain cases this information may not be available, or it may not reflect the actual number of clusters that exist in the dataset. Therefore, we propose to update the described algorithms so that the size of the map changes if needed.

The algorithms will perform a normal sequence of steps, starting with a given value for the map size, or if none is provided, a random one. After the algorithm converges to a final mapping, *i.e.* a clustering result, a further step is performed. Firstly, all the nodes in the map are compared with each other, and if any pair is seen as indifferent, then we conclude that the size of the map is too large, due to the fact that two nodes have converged towards the same cluster of indifferent alternatives. Hence, one of the two nodes is removed and another sequence of the two previously described steps is performed. Secondly, given that no two nodes in the map are indifferent, we look for any alternative that has been assigned to a node to which it is not indifferent. In this case we may conclude that we have potentially missed a cluster of alternatives, and therefore increase the size of the map by adding the selected alternative to it. Furthermore, in the case of the chained and ordered self-organizing maps. based on the relations between the selected alternative and the other nodes in the map, it is also inserted at the appropriate position. This is done by successively considering each node in the map, based on the order of the map, and insert the new node before the first node that outranks the alternative with a lower credibility degree than the credibility degree of the alternative outranking the considered node. Following this, the described sequence of steps is reiterated. When no increase or decrease of the map size is required, the clustering results is output as the final one.

In Figure 5.3 we highlight the way in which the size of the map is updated in the case of a chained self-organizing map.

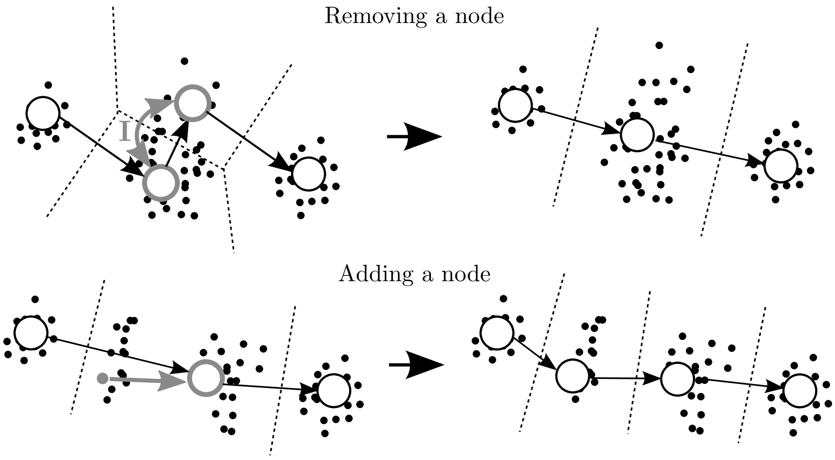


Figure 5.3: Adapting the number of nodes in the chained SOM;

All of the six algorithms based on SOMs described thus far have their adaptive variants denoted with the addition of a star superscript, *e.g.* D-SOM$_{\text{AVG}}$ becomes D-SOM$^{*}_{\text{AVG}}$.

5.3 CLIP: CLustering using Indifference and Preference relations

We present in this section an algorithm for solving the problem of clustering in MCDA, which may be used to find any of the structures presented so far. We will also explore the possibility of determining automatically the most likely number of clusters that are present in a given dataset.

Due to complexity issues related to the problem of clustering in general, but also particularly in the field of MCDA, this approach is divided in two steps:

1. partition using the indifference relation;
2. refine using all preferential relations.

The first step aims at building a "good enough " initial partition, with respect to any of the clustering objectives, either by limiting the amount of information considered, or using approximative algorithms. For example, we may proceed to apply either one of the self-organizing maps adaptations in place of this step. The second step is then used to refine this initial result through a meta-heuristic approach with the aim of getting closer to the optimal result based on a selected clustering objective. An external resource on this approach can be found in [Meyer and Olteanu 2013].

We continue by detailing more the first step of the approach.

5.3.1 First step: Partitioning using indifferences

As mentioned before, any of the previous methods may be used at this step. Some of these methods require the number of clusters to be given beforehand, although we have presented several algorithms that overcome this limitation. Even in those cases, using a random value for the initial number of clusters may require considerable changes to it until finding a good result.

We define, in this case, an approach that tries first to identify the densest areas in the graph induced from the relations of indifference, which have a high probability of forming a cluster. Not only is the property of being in a dense region important, but also the placement with respect to the other dense regions.

Defining a core

Considering the graph constructed by the relations of indifference, $G(X, \mathrm{I})$, we call a **core** a set of alternatives $Y \subseteq X$ which is placed in a dense region in this graph and has strong connections to alternatives that are placed in this regions but weak ones to alternatives from different regions. Following this definition, we will also consider that all the alternatives inside the core are indifferent to each other, therefore it forms a clique in $G(X, \mathrm{I})$. In addition, a core will not overlap with other sets of alternatives which are better defined in this respect.

We define for any core a profile Q which measures how the core compares to the alternatives in X overall.

$$Q(Y, x) = \frac{1}{|Y|}\big(|\{y \in Y : x \,\mathrm{I}\, y\}| - |\{y \in Y : x \not\mathrm{I} y\}|\big), \forall x \in X, \forall Y \subseteq X. \tag{5.14}$$

This profile therefore balances, for each alternative in X, the number of alternatives inside the core that are indifferent to it, and the number of those that are not. We use a bipolar scale so that a positive value in this profile would highlight the fact that a majority of elements in the core are indifferent to it, while a negative value would highlight the fact that such a majority is not achieved. Furthermore, in order for the core to reflect with a high degree the definition we have given it, its profile should contain large values, irrespective of their sign.

This definition of the core profile falls in line with those clustering problems which account only for the crisp credibilities of the relations between the alternatives. We may, therefore, extend the definition of the profile in order account also for the credibility degrees attached to the relations of indifference:

$$Q(Y,x) = \frac{1}{|Y|}\Big|\sum_{y \in Y} r(x \,\mathrm{I}\, y)\Big|, \forall x \in X, \forall Y \subseteq X. \tag{5.15}$$

The properties of a core can now be summarised as:

$$x \,\mathrm{I}\, y,\ \forall x, y \in Y, \tag{5.16}$$

$$f_C(Y) > f_C(Z),\ \forall Z \subseteq X : Z \cap Y \neq \emptyset, \tag{5.17}$$

where the fitness function of a core is:

$$f_C(Y) := \frac{1}{|X|}\sum_{x \in X} \big|Q(Y,x)\big| \tag{5.18}$$

The fitness measure quantifies the consistency with which the alternatives in the core are indifferent to the alternatives that are inside the same dense region, and not indifferent to the ones outside of it. In the case where two sets of alternatives that are candidates to be a core have the same value for f_C, one of them is selected at random.

Exact approach of finding the cores

It comes rather clearly that finding the best set of cluster cores is a combinatorial problem, as from the start we do not know the number of dense regions in the graph, nor how the cliques are distributed over the graph induced by the relations of indifference.

Finding the cores in an exact manner would therefore require us to enumerate all the cliques in the graph. A simple backtracking algorithm may be used in this case.

In this algorithm, we pass three parameters, the first representing the clique that we are building, the second containing the candidate alternatives for extending it, and lastly the alternative which was last added. The first parameter is straightforward, while the second will contain the alternatives that are indifferent to all the alternatives currently inside this partially constructed core. The last parameter is used in conjunction with the lexicographic order operator (<), in order for the algorithm to pass through each clique only once.

Algorithm 4 Clique enumeration;

```
Input: $Y = \{\}, Z = X, x^{'}$
for all $x \in Z, x^{'} < x$ do
    $Y^{'} \leftarrow Y \cup \{x\}$
    $Z^{'} \leftarrow \{\}$
    for all $y \in Z$ do
        if $x \, \mathrm{I} \, y$ then
            $Z^{'} \leftarrow Z^{'} \cup \{y\}$
    CliqueEnum($Y^{'}, Z^{'}, x$)
    $Y^{'}$ is a clique
```

Following the enumeration of all cliques in G, the ones that have the property highlighted in Equation (5.17) are selected as cores.

It has been show in [Moon and Moser 1965] that there exists a potentially exponential number of maximal cliques in a graph. A maximal clique is a clique which is not contained in another, therefore the number of cliques in a graph is also potentially exponential, making the process of enumerating them exponential in time complexity. For this reason, we look to overcome this difficulty through an approximative approach, which we will present in the next subsection. Before doing so, we first look at how the selected cores are transformed into a partition of X.

Expanding the cores

Each cluster is built by first adding the elements of the corresponding core to it, and second by adding the alternatives that were left out to the cluster containing the core to which they belong best:

$$\begin{aligned} &1)\ K_l = C_l, \forall l \in 1..k, \\ &2)\ K_l = K_l \cup \{x\} : l = \underset{m \in 1..k}{\arg\max} \big(Q(C_m, x)\big),\ \forall x \in X - \bigcup_{n \in 1..k} C_n, \end{aligned} \tag{5.19}$$

This greedy approach is sufficient at this point as the finer changes to the partition will be tackled at the next step. This will also be shown through empirical results in the next chapter.

5.3.2 Approximative approach for the first step

The approximative approach for finding the best set of cores can be described as a population-based meta-heuristic, and more precisely an evolutionary algorithm [Talbi 2009]. The general framework of this type of approach is detailed in Algorithm 5.

Algorithm 5 Population-based meta-heuristic;

```
P ← GenerateInitialPopulation()
while not BreakCondition() do
    P' ← GenerateNewPopulation(P)
    P ← SelectNewPopulation(P, P')
    EvaluatePopulation(P)
return best individual
```

In our case, each individual Y in the population P is a clique in G, encoded as a set and is evaluated using the fitness function in Equation (5.18). As opposed to the general outline presented above, we are interested not only in the individual that is the best one globally, with respect to the fitness function, but also in all the others that are locally best. The notion of locality translates to two sets having a non empty intersection. The approach that we will describe resembles closely the Glow-worm meta-heuristic from [Krishnanand and Ghose 2005].

We present the main steps of the proposed approach below.

Initializing the population

We mention two strategies for building the initial population:

- random generation;
- diversification procedures.

The first strategy is straightforward. It requires the size of the population to be specified beforehand, but may generate individuals that overlap and leave parts of the graph uncovered.

The second strategy addresses this issue, by iteratively generating the individuals so that they are placed uniformly in the solution space. In our case they are generated so that they will not overlap with the previously generated ones. In this way we may also find out an approximate number of cliques that are required to cover the graph G, and so determine the size of the population directly. Of course, an upper limit may be specified, as having a sparse graph will require many cliques in order to cover it.

Generating the new population

As each individual is a clique in G, and may vary in size, the only meaningful operator for generating new individuals is the mutation operator, which performs a change to an individual irrespective of the others. In our case, this operator will either remove an alternative from a set, or add one that is linked to all the ones in the clique, following the heuristic function below:

$$h_{add}(Y, x) := \frac{1}{|Y|} \sum_{y \in X} \operatorname{sgn} Q(Y, y) \cdot Q(\{x\}, y), \tag{5.20}$$

$$h_{remove}(Y, x) := -h_{add}(Y, x). \tag{5.21}$$

For any individual Y, a pool of potential changes is generated from the set $Z = Y \cup \{x \in X - Y : x \,\mathrm{I}\, y, \forall y \in Y\}$. The alternatives from Y will be characterised by h_{remove}, while those in the set of alternatives that are indifferent to the entire set Y will be characterised by h_{add}. Generating all possible individuals may not always be a desired strategy as on denser graphs they may be very large in number, therefore a prefixed number of individuals may be generated from each one in the old population. When generating a number of new individuals from Y, several operations characterised by the above heuristic functions may be selected through mechanisms such as Roulette Wheel Selection, Stochastic Universal Sampling, Tournament Selection or Rank-based Selection [Talbi 2009].

Selecting the new population

After generating a new set of individuals, several strategies may be used to construct the new population:

- keep only the new population and discard the old one;
- keep the best individual from the old population and the ones in the new;
- select the best individuals from both the old and the new population.

In the case of this problem, we will be keeping throughout the steps of the method a record of the individuals that maximize the fitness function locally, as this is the result that we wish to retrieve. However, the property of these individuals being locally optimal with respect to the fitness function can only be based on the information available from the prior exploration of the solution. We denote these individuals with P^*.

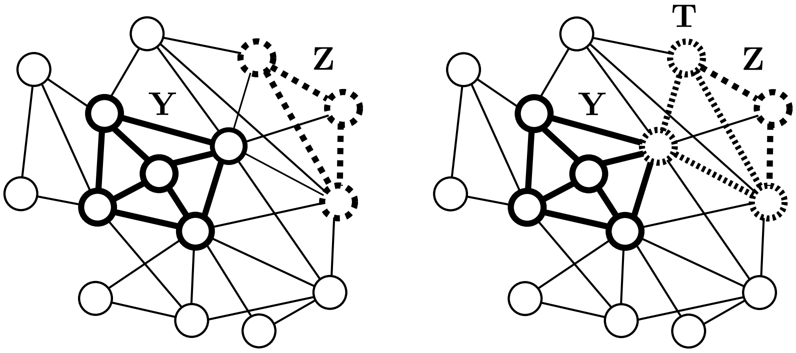
In order to continue exploring the neighbourhood of the supposedly locally optimal individuals, we generate the new population by keeping a record of P^* and then selecting the remaining individuals from the previously constructed new population $P^{'}$.

$$P \leftarrow P^* \cup \textsc{Select}(P^{'}).$$

Evaluating the new population

After the new population has been generated, the individuals from P^* need to be updated. One method of doing this is to compare all the individuals from $P^{'}$ with the ones in P^*. If we find at least one that overlaps with an individual from P^* and has a higher fitness function it will replace it. Also, if an individual from $P^{'}$ does not overlap with any from P^*, and has a higher fitness function than all the ones that overlap with it from $P^{'}$, it will be placed in P^*.

This approach of updating P^* however only considers the individuals from the current population and does not benefit from past knowledge. We may, for instance, have a region in the graph G that is large and moderately dense with respect to the edges inside it. A centrally located clique may not overlap at all with the ones from the edge and as a result those two may be considered by the previous approach to be locally optimal when compared to the centrally located clique. This is illustrated in Figure 5.4, where inside apparently one dense region in the graph two non-overlapping cliques may be considered as cores.



$f_C(Y) > f_C(Z)$
$Y \cap Z = \emptyset$
$\to Y, Z$ cores

$f_C(Y) > f_C(T) > f_C(Z)$
$Y \cap T \neq \emptyset$ and $T \cap Z \neq \emptyset$
$\to Y$ core

Figure 5.4: Finding two non-overlapping cliques in one dense region;

As a result, we define a map $B : X \to \mathbb{R}$, which records for all alternatives the highest fitness value for any previously explored clique which contained that alternative. P^* is updated by taking all individuals in the new population, $Y \in P$ and verifying if the following property is met:

$$f_C(x) \geq B(x), \forall x \in Y. \tag{5.22}$$

After selecting the individuals that meet this property, a final check needs to be made in the case where there are several individuals that overlap and have the same value of the fitness function. In this case one of them is removed at random until no overlapping individuals remain. This is done in order to ensure that only one clique is mapped over a dense region in the graph.

Following this, B is updated based on the fitness values of the new population P:

$$B(x) = \max\big(B(x), f_C(Y)\big), \forall x \in Y, \forall Y \in P. \tag{5.23}$$

In this way, even if an individual from the previous generation no longer exists in the new one, the information from it is preserved in B.

5.3.3 Second step: Refining using preferences

Until now, we have built a partition K of X following the first step of this approach. This partition was built either through the presented greedy heuristic of first finding a set of cores and then expanding them, or through other approaches that are also heuristic in nature.

This second step may be used to improve further the initial partition by making small changes at a time to it, in order to guide it towards the optimal result for any of the presented clustering objectives.

These small changes are translated into moving one alternative at a time from one cluster to another. We will present in what follows several heuristic functions for characterising these moves, which aim at improving the fitness of the partition K following a selected clustering objective.

Heuristic functions

We start with the first clustering objective, that of non-relational clustering, and consider each of the three types of fitness function. The heuristics for moving an alternative x from cluster K_l to cluster K_m are defined as:

$$\begin{aligned} h^*_{\mathrm{NR}}(x, K_l, K_m) \; := \; & \frac{1}{2 \cdot |X|} \cdot \Big(\sum_{y \in K_m} r^*(x \,\mathrm{I}\, y) + \sum_{y \in X - K_m} r^*(x \,\not\mathrm{I} y) \\ & - \sum_{y \in K_l} r^*(x \,\mathrm{I}\, y) - \sum_{y \in X - K_l} r^*(x \,\not\mathrm{I} y) \Big) \end{aligned} \tag{5.24}$$

$$\begin{aligned} h_{\mathrm{NR}}(x, K_l, K_m) \; := \; & \frac{1}{2 \cdot |X|} \cdot \Big(\sum_{y \in K_m} r(x \,\mathrm{I}\, y) + \sum_{y \in X - K_m} r(x \,\not\mathrm{I} y) \\ & - \sum_{y \in K_l} r(x \,\mathrm{I}\, y) - \sum_{y \in X - K_l} r(x \,\not\mathrm{I} y) \Big) \end{aligned} \tag{5.25}$$

$$\begin{aligned} h^{min}_{\mathrm{NR}}(x, K_l, K_m) \; := & \min \Big(\min_{y \in K_m} \big(r(x \,\mathrm{I}\, y) \big), \min_{y \in X - K_m} \big(r(x \,\not\mathrm{I} y) \big) \Big) \\ & - \min \Big(\min_{y \in K_l} \big(r(x \,\mathrm{I}\, y) \big), \min_{y \in X - K_l} \big(r(x \,\not\mathrm{I} y) \big) \Big) \end{aligned} \tag{5.26}$$

Each heuristic is constructed from two terms. This first measures how well x would belong to K_m, while the second measures how well x belongs to the cluster it is in, K_l. The second term is subtracted from the first in order to determine if x belongs better to K_m or K_l.

Each heuristic measures these affirmations differently. The first heuristic takes into account the crisp credibility degrees, and is therefore suited for the fitness function that also uses these crisp credibility degrees. The second heuristic uses the valued credibility degrees, while the last one looks at the weakest credibility degree that supports each of the two affirmations.

As both terms, in all the considered heuristics, are in the same value range, if the heuristic function gives a positive value, this means that x is better suited to be in K_m, while if the result is negative, then x is better to be kept in K_l.

Going to the relational clustering objectives we define the following heuristic functions:

$$
\begin{aligned}
h^*_{\mathrm{R}}(x, K_l, K_m) :=& \frac{1}{2 \cdot |X|} \cdot \Bigg(\sum_{y \in K_m} r^*(x \,\mathrm{I}\, y) \\
&\quad + \sum_{\substack{K_n \in K \\ K_n \neq K_m}} \sum_{y \in K_n} \max \Big(\min \big(r^*(K_m \,\mathrm{P}\, K_n), r^*(x \,\mathrm{P}\, y) \big), \\
&\qquad\qquad \min \big(r^*(K_n \,\mathrm{P}\, K_m), r^*(y \,\mathrm{P}\, x) \big), \\
&\qquad\qquad \min \big(r^*(K_m \,\mathrm{R}\, K_n), r^*(x \,\mathrm{R}\, y) \big) \Big) \\
&\quad - \sum_{y \in K_l} r^*(x \,\mathrm{I}\, y) \\
&\quad - \sum_{\substack{K_n \in K \\ K_n \neq K_l}} \sum_{y \in K_n} \max \Big(\min \big(r^*(K_l \,\mathrm{P}\, K_n), r^*(x \,\mathrm{P}\, y) \big), \\
&\qquad\qquad \min \big(r^*(K_n \,\mathrm{P}\, K_l), r^*(y \,\mathrm{P}\, x) \big), \\
&\qquad\qquad \min \big(r^*(K_l \,\mathrm{R}\, K_n), r^*(x \,\mathrm{R}\, y) \big) \Big) \Bigg)
\end{aligned}
\tag{5.27}
$$

$$
\begin{aligned}
h_{\mathrm{R}}(x, K_l, K_m) :=& \frac{1}{2 \cdot |X|} \cdot \Bigg(\sum_{y \in K_m} r(x \,\mathrm{I}\, y) \\
&\quad + \sum_{\substack{K_n \in K \\ K_n \neq K_m}} \sum_{y \in K_n} \max \Big(\min \big(r^*(K_m \,\mathrm{P}\, K_n), r(x \,\mathrm{P}\, y) \big), \\
&\qquad\qquad \min \big(r^*(K_n \,\mathrm{P}\, K_m), r(y \,\mathrm{P}\, x) \big), \\
&\qquad\qquad \min \big(r^*(K_m \,\mathrm{R}\, K_n), r(x \,\mathrm{R}\, y) \big) \Big) \\
&\quad - \sum_{y \in K_l} r(x \,\mathrm{I}\, y) \\
&\quad - \sum_{\substack{K_n \in K \\ K_n \neq K_l}} \sum_{y \in K_n} \max \Big(\min \big(r^*(K_l \,\mathrm{P}\, K_n), r(x \,\mathrm{P}\, y) \big), \\
&\qquad\qquad \min \big(r^*(K_n \,\mathrm{P}\, K_l), r(y \,\mathrm{P}\, x) \big), \\
&\qquad\qquad \min \big(r^*(K_l \,\mathrm{R}\, K_n), r(x \,\mathrm{R}\, y) \big) \Big) \Bigg)
\end{aligned}
\tag{5.28}
$$

$$\begin{aligned}
h_{\mathrm{R}}^{min}(x, K_l, K_m) :=\min\Bigg(& \min_{y\in K_m} \big(r(x\,\mathrm{I}\,y)\big), \\
& \min_{\substack{K_n\in K\\ K_n\neq K_m\\ y\in K_n}} \max\Big(\min\big(r^*(K_m\,\mathrm{P}\,K_n), r(x\,\mathrm{P}\,y)\big), \\
& \qquad \min\big(r^*(K_n\,\mathrm{P}\,K_m), r(y\,\mathrm{P}\,x)\big), \\
& \qquad \min\big(r^*(K_m\,\mathrm{R}\,K_n), r(x\,\mathrm{R}\,y)\big)\Big)\Bigg) \\
- \min\Bigg(& \min_{y\in K_l} \big(r(x\,\mathrm{I}\,y)\big), \\
& \min_{\substack{K_n\in K\\ K_n\neq K_l\\ y\in K_n}} \max\Big(\min\big(r^*(K_l\,\mathrm{P}\,K_n), r(x\,\mathrm{P}\,y)\big), \\
& \qquad \min\big(r^*(K_n\,\mathrm{P}\,K_l), r(y\,\mathrm{P}\,x)\big), \\
& \qquad \min\big(r^*(K_l\,\mathrm{R}\,K_n), r(x\,\mathrm{R}\,y)\big)\Big)\Bigg) \qquad (5.29)
\end{aligned}$$

In comparison to the heuristic functions used for non-relational clustering, the heuristic functions for relational clustering further develop the terms that are used to asses the quality of x being in K_l or K_m when compared to the alternatives from the other clusters than these two. As with the relational clustering fitness functions, a combination of a max and a min operator is used for taking into account only the credibility degrees of the same type of relation between x and the alternatives inside other clusters, as the relation between the cluster in which x is supposed to be placed (either K_m or K_m) and the other clusters.

Determining the effect that moving one alternative from a cluster to another has on the property of transitivity of the relations of strict preference between all the clusters is rather computationally expensive. In order to do so, we would need to recalculate, for each move, the relations between the two clusters that are trading the alternative, and all the other clusters, so that the property of transitivity could then be assessed. Therefore, when considering the problem of ordered clustering, no additional heuristic functions are proposed.

Meta-heuristic approach

Using the operations of moving one alternative from one cluster to another in K, which may be characterized by the heuristic functions that were previously defined, we can employ any single solution-based meta-heuristic [Talbi 2009], such as local search, simulated annealing [Kirkpatrick et al. 1983] or tabu search [Glover 1989, Glover 1990] in order to find the optimal clustering result. We present in what follows the general outlines of these three meta-heuristics.

Local Search Local search is one of the oldest and simplest single-solution based meta-heuristics. It can be briefly described as follows:

Algorithm 6 Local search meta-heuristic;

```
s ← InitialSolution()
N ← GenerateNeighbours(s)
while not NoImprovingNeighbours(N) do
    s ← SelectImprovingNeighbour(N)
    N ← GenerateNeighbours(s)
return s
```

In a few words, this approach starts from an initial solution, then it selects a neighbouring solution that would increase the fitness function that needs to be optimized, until no such neighbours exist.

In our case, the set of neighbours will be all the partitions that can be constructed from the current one by moving one alternative from one cluster to another, or taking it out of its current cluster and constructing a singleton cluster with it.

The selection mechanism can be any of the following:

- Best improvement: takes the neighbouring solution with the highest fitness value;
- First improvement: takes the first neighbouring solution which improves the fitness function;
- Random selection: an improving neighbour is selected randomly.

The first two strategies are deterministic, where the first converges very quickly with respect to the number of iterations of the algorithm, although it needs to generate at each step all the neighbours of the current solution. The second strategy generates the neighbours incrementally and select the first one that improves the fitness function. The improvement with each iteration will be smaller than in the first strategy, though as the set of neighbours will not be completely generated, except when the solution gets closer to a locally optimal one, each iteration will be faster. The last strategy is stochastic, and may be applied when there is a high risk of having locally optimal solutions. In this case, the algorithm may be run several times and the best solution may be then selected. Escaping a locally optimal solution may also be done by accepting non-improving solutions. This is the case of the two meta-heuristics that will follow.

The local search meta-heuristic can be easily adapted to our problem. The initial solution will be given by the partition found by the first step of our approach. Then, instead of generating all partitions from this initial one, by moving one alternative from a cluster to another, or outside of them all, we list all these moves along with the heuristics defined in the beginning of this subsection. As these heuristics give a direct indication of how the fitness function of the resulting partition will change, it is not necessary to build these partitions and evaluate them. All three selection procedures may be applied, where a Roulette Wheel mechanism may be used for the random selection, for instance.

Simulated Annealing The simulated annealing meta-heuristic [Kirkpatrick et al. 1983] is based on the process of heating and then slowly cooling a substance in order to obtain a strong crystalline structure. It can be described by the following series of steps:

Algorithm 7 Simulated Annealing meta-heuristic;

```
$s \leftarrow$ InitialSolution()
$T \leftarrow T_{max}$
while $T > T_{min}$ do
    while not EquilibriumCondition() do
        $s^{'} \leftarrow$ GenerateRandomNeighbour($s$)
        if $f(s^{'}) - f(s) \geq 0$ then
            $s \leftarrow s^{'}$
        else
            if $random(0,1) > e^{\frac{f(s)-f(s^{'})}{T}}$ then
                $s \leftarrow s^{'}$
    $T \leftarrow$ UpdateTemperature($T$)
return $s$
```

The algorithm starts with an initial solution and an initial temperature variable. As this temperature variable decreases towards its minimum value, a series of operations are made on the current solution. A neighbour is randomly generated and if it improves the fitness function from that of the old solution then it replaces it. The algorithm also allows for non-improving neighbours to be selected, with a probability proportional to the decrease in fitness and to the temperature variable. Therefore, in the beginning of the algorithm, non-improving solutions have a higher probability of being selected, and so we tend to explore the solution space, while towards the end, the algorithm converges to a final solution. For each temperature level a fixed number of steps are performed.

We adapt this approach to our problem by taking the initial partition from the first step of our algorithm as the starting solution. The initial temperature is fixed at a high value and decreases linearly based on a running time constraint. The number of iterations for each temperature can vary while each randomly generated neighbour represents a move characterised by the heuristic functions defined before.

Tabu Search The Tabu search meta-heuristic [Glover 1989, Glover 1990] is very similar to the local search meta-heuristic having the best improvement selection scheme. It moves away from it by being able to select a non-improving solution when no neighbours that improve the fitness function exist, therefore escaping from locally optimal solutions. The neighbour with the smallest decrease in the fitness function is selected, therefore this approach is a deterministic one. As a result, cycles can be formed where the algorithm iterates over the same sequence of solutions.

The general sequence of steps that define this meta-heuristic are found below:

Algorithm 8 Tabu Search meta-heuristic;

```
$s \leftarrow$ InitialSolution()
InitializeLists($T, MTM, LTM$)
InitializeCriteria($AC, IC, DC$)
while not StopCriterion() do
    $s^{'} \leftarrow$ GenerateBestNeighbour($s, T, AC$)
    $s \leftarrow s^{'}$
    UpdateLists($T, MTM, LTM$)
    UpdateCriteria($AC, IC, DC$)
    if $IC$ holds then
        Intensify()
    if $DC$ holds then
        Diversify()
return best $s$
```

In order to avoid cycles, Tabu search discards the neighbours that were previously visited, by storing a fixed number of the last visited neighbours in a tabu list (T). This is also regarded as a short-term memory, as storing in memory all the previously generated solutions is impractical.

In order to improve on this aspect, a tabu list may contain not previously generated solutions, but tabu moves, which describe a particular operation of generating a new neighbour. In this way we improve on the memory requirements for storing the tabu list, but we lose some information and may deny moves that will generate neighbours that have not yet been constructed. To overcome this, for some conditions called aspiration criteria (AC), a tabu move may still be used to generate a new solution. The most common of these criteria is to select a tabu move if the generated solution is better than any of the previously generated ones.

Some other mechanisms consist in storing a medium-term memory (MTM), which contains all the locally best solutions found previously, and a long-term memory (LTM) which stores information on the previously generated solutions from the search. The medium-term memory is used to give priority to solutions that have common characteristics to the best solutions found until that point. This operation is done if certain criteria hold (IC), and has an effect of intensifying the search around similar solutions. The long-term memory, on the other hand, keeps track of the characteristics of all the previously generated solutions in order to give priority to solutions that move away from them. If certain criteria hold (DC) this operation is done, having a diversification effect and from which the search space is explored better.

Due to the added complexity of defining the aspiration criteria, the medium-term and the long-term memories, we have decided not to adapt at this stage the Tabu search for the problem of clustering in MCDA. Following the results in the next chapter, we may decide on the need to explore this option in the future.

Summary

Looking back at this chapter, we have briefly considered the complexity of this problem, which we deem as difficult. An analysis of the existing MCDA clustering approaches has also been made, in order to determine how well they fit with the model that we have previously proposed. Several of these approaches, do indeed fit well with our definitions, however, they explore only a few of the structures that may be uncovered through this process.

Hence, two types of clustering approaches have been further presented. The first takes a classical clustering algorithm, K-MEANS, and adapts it to the MCDA case. We also look at the generalization of this approach, the self-organizing maps, and propose several new topologies, which would be well-suited for finding ordered sets of clusters. We will consider, in the next chapter, if these topologies offer any increase in performance, when looking for this type of clustering result. The second type of approach that we have proposed, is inspired from Graph Theory, and is combined with a meta-heuristic so that it may be tailored for finding any of the clustering structures that we have defined. We will determine the performance of this approach too, in the next chapter.

Chapter 6

Validating the clustering approaches

Contents

Abstract

The clustering approaches proposed previously are validated in this chapter, through a series of experiments over large sets of artificially constructed MCDA problem instances. These problem instances are constructed so that a wide range of potential MCDA problems are considered.

The proposed approaches are initially analysed independently, considering all their variants and using the appropriate fitness measures. Following this, the variants that perform the best from each of these methods are compared with each other, but also with other existing clustering approaches.

6.1 Generating the benchmarks

In order to test the proposed MCDA clustering approaches, we have generated a large set of problem instances, on which to run and evaluate them.

In order for these tests to properly reflect the performance of the algorithms, we construct these benchmarks so that the alternatives inside them hold a wide array of structures.

6.1.1 Benchmarks characteristics

Each benchmark corresponds to an MCDA clustering problem and contains:

- a set of alternatives, X;
- a set of criteria F;
- the parameters of a selected preference model, reflecting the perspective of a fictive DM;
- the evaluations of the alternatives on the set of criteria.

The benchmarks, that have been created for the purpose of validating the proposed algorithms, contain 100 alternatives, which are defined on a set of 11 criteria. In order to simplify the implementation of the given algorithms, each criterion is defined on a ratio scale in the interval $[0, 1]$, however this is done without loss of generality.

We have chosen these parameters related to the size of each problem instance so that the problems would be considered large and difficult with respect to normal MCDA applications, but also be small enough so that the algorithms would not require a very long time to run. In this way we may simulate the performance of the presented algorithms over problems instances that resemble those that are generally found in real-life MCDA applications.

In order to compare the alternatives together we have chosen to use the bipolar-valued outranking relation from [Bisdorff et al. 2008], to which we have removed the weak veto threshold, in order to simplify the model. Any type of outranking relations may be used instead, for instance the relations used in the ELECTRE [Keeney and Raiffa 1976, Roy 1993] or PROMETHEE [Brans and Mareschal 2002] methods. Due to the variety of structures which we will embed in the set of benchmarks, structures that are given by the resulting relations of indifference, strict preference and incomparability, the results of the algorithms will be representative for clustering using any outranking relation which allows for these relations and structures to be constructed.

One mention needs to be made in regard to several of the presented adaptations of the SOM approaches to MCDA, which require that the outranking model contains preference thresholds on each criterion. In these cases, the algorithms would need to be adapted for the use of different outranking relations.

The parameters of the considered outranking relation are:

- preference direction on each criterion (increasing or decreasing);
- significance weights: $w_i \in (0,1), \forall i \in F\colon \sum\limits_{i \in F} w_i = 1$;
- indifference thresholds: $q_i \in [0,1), \forall i \in F\colon 0 \leqslant q_i < p_i$;
- preference thresholds: $p_i \in (0,1), \forall i \in F\colon q_i < p_i < v_i$;
- veto thresholds: $v_i \in (0,1], \forall i \in F\colon p_i < v_i \leqslant 1$;
- cut level: $\lambda \in (0,1]$.

We consider that the direction of preference of the fictive DM on each criterion is increasing. Furthermore, the criteria are considered equally important, therefore they are given equal significance weights. We motivate this choice through the fact that, by increasing the relative significance of a criterion in relation to the rest, we decrease the power of these criteria of affecting the resulting outranking relation, therefore reducing the degree with which we can discriminate between the alternatives. This remark does, of course, not pertain to the second part of the definition of an outranking relation, that which accounts for large performance differences.

We will address the issue of selecting contrasting criteria significance weights, but also that of using different types of outranking relations, in the future.

Coming back to our chosen preference model, the indifference thresholds are set as $q_i = \frac{1}{6(k+1)}, \forall i \in F$, while the preference thresholds are $p_i = 2 \cdot q_i$. Furthermore, the veto thresholds are fixed at a value of 0.75, while the cut threshold is set at the smallest possible value it can take, $\lambda = 0$, due to the fact that the criteria weights reflect exactly the point of view of the fictive DM on the set of alternatives, and so there is no need for larger weighted majorities of criteria for supporting the outranking relation between them.

The selection of the discrimination thresholds is further motivated by the way in which the evaluations of the alternatives are generated. We present here several remarks related to this issue, while a more thorough description of this process will be presented in the next subsection.

Considering that we wish to construct k clusters of alternatives, we first define k normal distributions on each criterion $i \in F$. We denote with $\mathcal{N}_l$, $\forall l \in 1..k$, all the normal distributions in the form of $\mathcal{N}(\mu_l, \sigma_l^2)$. Each alternative will have its evaluation on a given criterion generated using one of these distributions.

The centres of the k distributions are equally spaced apart, $\mu_l = \frac{l+1}{k+1}$, $\forall l \in 1..k$, while their standard deviations are fixed to $\frac{1}{2(k+1)}$. This is highlighted in Figure 6.1.

Each distribution is made to overlap with a neighbouring one by as much as two thirds, however, this means that only approximatively 15% of the values generated according to them may be confused with the values generated by a neighbouring distribution. The previously mentioned indifference and preference thresholds have been chosen so that a significant percentage of the evaluations generated using the same normal distribution may be seen as being strictly preferred to others. This is motivated by the desire to be able to construct alternatives that are not

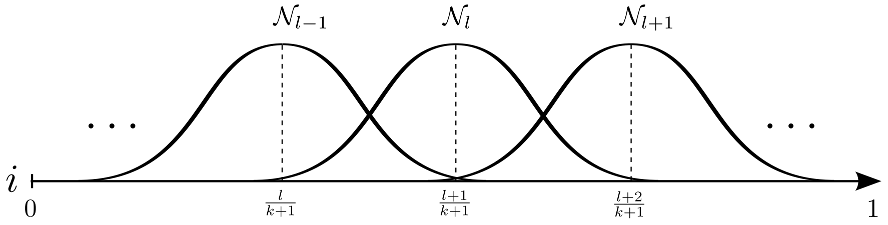


Figure 6.1: A series of normal distribution on criterion i;

strongly indifferent with each other, even when they are built using the same normal distributions across all criteria. In this way, the generated problem instances are representative for difficult decision problems.

6.1.2 Generating the evaluations

When generating the evaluations of the alternatives inside a benchmark, we first need to consider the number of clusters that we wish to generate and their sizes. In our case, we will be fixing the number of generated clusters of alternatives to 10. This number is neither too small, so that the resulting structure is not trivial, while it is also not too large considering the number of alternatives in the benchmarks. The size of each cluster depends on the selection, for each benchmark, of one of the following scenarios:

- equal sizes;
- linearly increasing sizes;
- quadratically increasing sizes;
- cluster sizes characterised by a sine function with the period equal to the number of clusters;
- cluster sizes characterised by a sine function with the period equal to half the number of clusters.

After assigning the alternatives to each cluster, we then consider different generators for constructing the evaluations of the alternatives from each cluster. We propose 12 generators, which will construct sets of clusters of various strength. We will denote these generators with letter from $\mathcal{A}$ to $\mathcal{L}$.

The first generator, $\mathcal{A}$, considering an alternative from cluster $l \in 1..k$, constructs its evaluations on each criterion $i \in F$ using the $\mathcal{N}_l$ normal distributions. This generator should produce the strongest clusters out of all the ones that will be considered further, which furthermore are ordered from the best one ($l = k$) to the worst ($l = 1$). This is due to the fact that, although the alternatives generated in this way may not be very strongly indifferent to each other, they will be strictly preferred by most of the alternatives generated using distribution with higher means, while in turn they will be strictly preferred to most of the alternatives generated using distributions with lower means.

The second generator, $\mathcal{B}$, considers that the evaluations of an alternative from cluster $l \in 1..k$, on each criterion, may be drawn from either of the $\mathcal{N}_{l-1}, \mathcal{N}_l, \mathcal{N}_{l+1}$ normal distributions, with equal probability. In this way, we further weaken the previously mentioned relations of strict preference that can occur between alternatives from neighbouring clusters, considering their indexes $1..k$.

In Figure 6.2 we illustrate the generation of two alternatives on three criteria according to these first two generators.

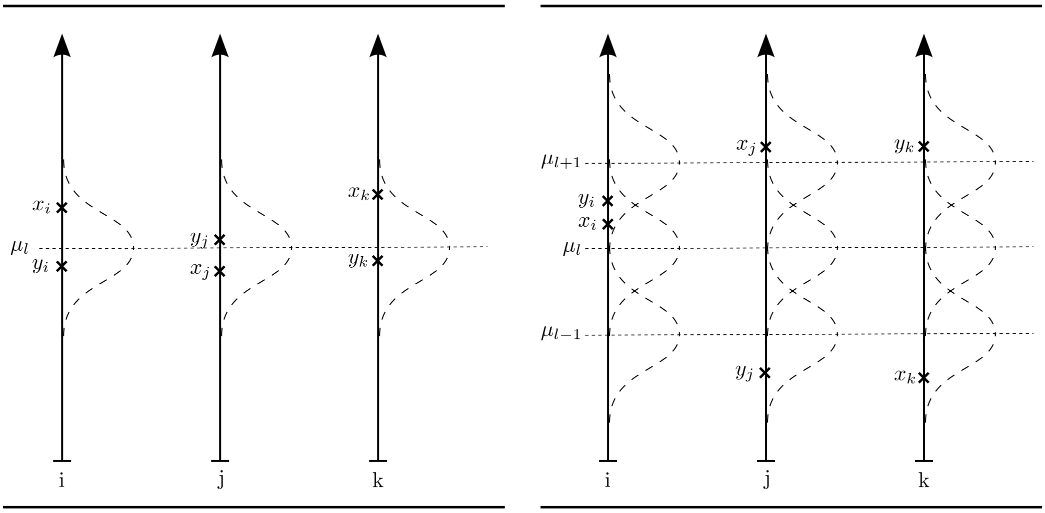


Figure 6.2: Illustrating generators $\mathcal{A}$ (left) and $\mathcal{B}$ (right) for $x, y \in K_l$;

Both the alternatives that are being generated belong to cluster K_l. Hence, their evaluations are constructed on each criterion using the normal distribution with the centre at μ_l. We notice that these two alternatives have a high chance of having close evaluations on all criteria. When considering the second generator, for each alternative and on each criterion we select with equal probability a normal distribution with the centres at μ_{l+1}, μ_l or μ_{l-1}. In this way, the two alternatives have a lower chance of being given evaluations that are close to each other on every criterion. However, on certain criteria they will still have evaluations that are close to each other, while on others, the probability that one of their evaluations is higher than the other is 50%, therefore, they still have a high chance of becoming indifferent. In the illustrated example, the two alternatives are in both cases indifferent.

The following three generators consider distributions farther away from the l^{th} normal distribution, that was initially used in the case of the A generator. Generator $\mathcal{C}$ considers $\mathcal{N}_{l-2}, \mathcal{N}_{l-1}, \mathcal{N}_{l+1}, \mathcal{N}_{l+2}$ with probabilities 33.3%, 16.6%, 16.6% and 33.3% respectively, generator $\mathcal{D}$ considers $\mathcal{N}_{l-4}, \mathcal{N}_{l-1}, \mathcal{N}_l, \mathcal{N}_{l+1}, \mathcal{N}_{l+4}$ with probabilities 16.6%, 25%, 16.6%, 25% and 16.6% respectively, while generator $\mathcal{E}$ considers $\mathcal{N}_{l-4}, \mathcal{N}_{l-1}, \mathcal{N}_l, \mathcal{N}_{l+1}, \mathcal{N}_{l+4}$ with probabilities 33.3%, 16.6%, 16.6% and 33.3% respectively.

It follows from these considerations, that the generated clusters are progressively weaker in structure, considering the order in which the generators have been presented.

A series of another five generators is derived from the already presented ones, denoted with $\mathcal{F}$, $\mathcal{G}$, $\mathcal{H}$, $\mathcal{I}$ and $\mathcal{J}$. These generators function in the same way as the previous ones, with the exception that a 33.3% probability exists for an alternative from a cluster at index $l < 5$ to be given the maximum evaluation on one randomly chosen criterion. This same probability is given to the alternatives from the clusters with indexes $l > 5$ to be given the minimum evaluation on a randomly chosen criterion. In this way we further move away from the initial ordered structure given by the $\mathcal{A}$ generator, by giving very poor evaluations to certain alternatives from the clusters ranked higher, and giving very good evaluations to certain alternatives from the clusters that are ranked lower. Through this approach we additionally construct alternatives that are incomparable with others.

Finally, two more generators are considered. We denote them with $\mathcal{K}$ and $\mathcal{L}$. The first generator is derived from the $\mathcal{J}$ generator, while further adding a 33.3% probability for any alternative in the mid range of clusters to be given both a very good evaluation on one criterion, and a very bad evaluation on another. In this way, the alternatives that are placed in the mid ranked clusters will also become incomparable to others. The $\mathcal{L}$ generator increases the probabilities used by $\mathcal{K}$ from 33.3% to 66.6%.

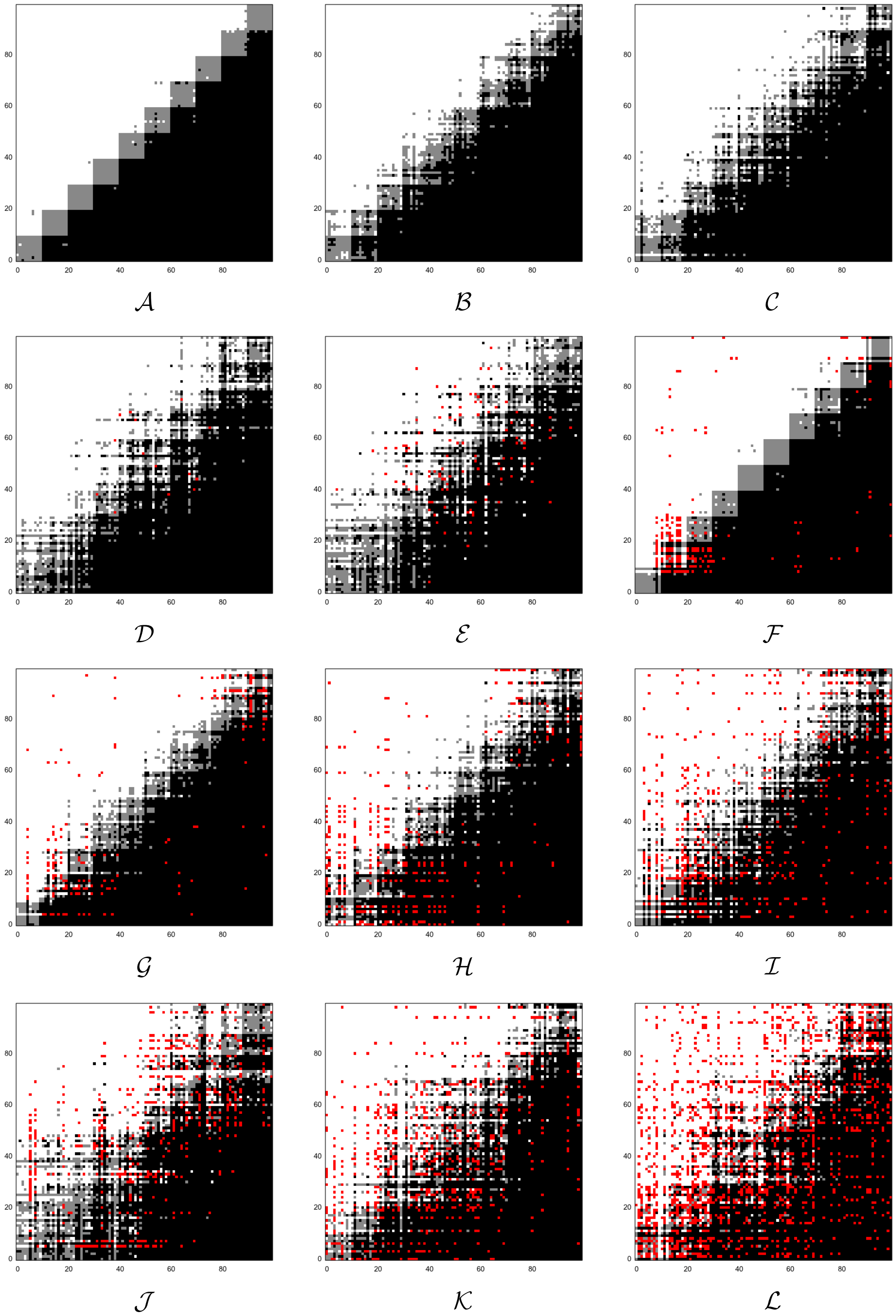
By constructing five problem instances using each of the previously described generators, for each of the five scenarios related to the cluster sizes, we end up with a total of 300 problem instances containing 100 alternatives each.

6.1.3 Analysing the benchmarks

Before presenting the experimental results of running several algorithms on each of the constructed benchmarks, we first take a closer look at the generated set of benchmarks.

In Figure 6.3, we present several illustrations of problem instances constructed using the 12 evaluations generators. They are illustrated in an order we consider increasing with respect to the difficulty of finding back the original sets of clusters.

Each illustration is in the form of a matrix, containing the dominating preferential relations between every pair of alternatives in the problem instance, *i.e.* the ones with the highest degrees of credibility. The relations of indifference are illustrated with gray, those of strict preference over the considered alternative with black, those of strict preference in favour of the considered alternative with white, while with red we depict the relations of incomparability. For instance, the first vertical line, in the first image, corresponds to the relations between the first alternative and all the others. As in this case, this alternative is placed in the bottom cluster, it is indifferent to the ones in the same cluster, but preferred by all the rest. Only the scenario where all the clusters are equal in size have been illustrated below.



$\mathcal{A}$ $\mathcal{B}$ $\mathcal{C}$

$\mathcal{D}$ $\mathcal{E}$ $\mathcal{F}$

$\mathcal{G}$ $\mathcal{H}$ $\mathcal{I}$

$\mathcal{J}$ $\mathcal{K}$ $\mathcal{L}$

Figure 6.3: Preferential relations for problems built with the 12 generators;

We may notice that we start with a rather clear order between clusters of indifferent alternatives. Almost all the alternatives in a cluster are indifferent to all the rest, therefore showing that the $\mathcal{A}$ generator is well suited for building this type of structure.

As we progress further, we find that this structure is more and more distorted, with alternatives from neighbouring clusters becoming indifferent, while those from the same cluster moving away from the initial setting of being indifferent to all the others inside the same cluster. Furthermore, only the last two generators from the first five produce a few alternatives that are incomparable to others, this being reflected by the few red dots. This is an artefact of having the alternatives in the mid clusters begin evaluated in certain cases using normal distribution 4 levels above or 4 levels below the one at the current level. In such a case, alternatives that would otherwise be indifferent, contain both very good and very bad evaluations, on certain criteria, which makes them seem incomparable to others.

Continuing with the following five generators, we find that they closely resemble the initial five, with the exception that more relations of incomparability appear inside the lower and the higher clusters. This is a natural effect of the considered generators.

Finally, the last two generators contain even more relations of incomparability.

6.2 SOM adaptations

Using the previously described set of benchmarks, we proceed to executing the proposed algorithms over them and analysing the results.

We begin with the adaptations of the self-organizing maps (SOM) to clustering in MCDA. We remind the three classes of these approaches as: the disconnected SOM on indifferences (denoted with D-SOM), the chained SOM one indifferences (denoted with C-SOM) and the ordered SOM on indifferences (denoted with O-SOM).

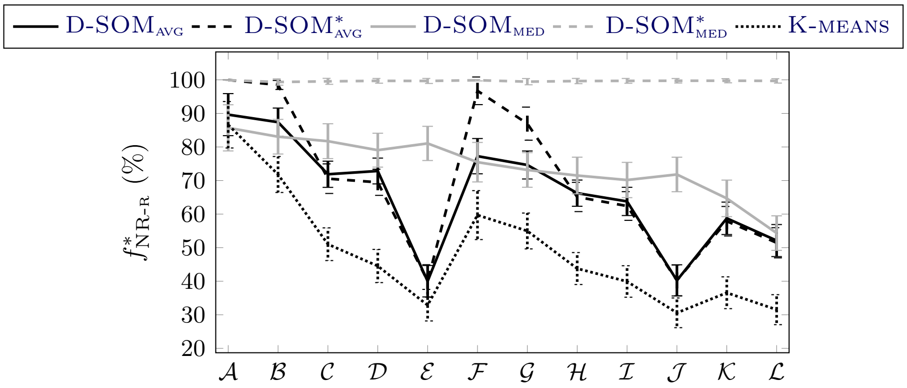
Each of the three types of SOMs on indifferences are split into approaches that construct the nodes in the map from the evaluations of the alternatives that have been mapped to them (denoted with a AVG subscript), and those that select an existing alternative out of those that have been mapped to them (denoted with a MED subscript).

For each of the 6 mentioned approaches, variants that adapt the size of the map (denoted with $*$ superscripts) have also been created.

6.2.1 The disconnected SOM on indifferences

We begin with the disconnected SOMs, and illustrate the average results, with respect to the measures that are directly linked to them, $f^*_{\text{NR-R}}$ and $f_{\text{NR-R}}$. All of the 4 variants of this approach, as well as the K-MEANS algorithm have been executed 50 times over each of the 300 benchmarks, so that the results are representative. Furthermore, the algorithms were left to run for at most one minute. During this time, the algorithms have been able to converge to a final solution. For each of these algorithms, the desired number of clusters was set to 10, which is the number of clusters that were initially generated inside each benchmark. Nevertheless, the actual structure in the benchmarks may contain a different number of clusters, due to the increasing interactions between the alternatives in neighbouring clusters, and the inclusion of large performance differences in the more difficult benchmarks.

We illustrate the results of D-SOM$_{\text{AVG}}$, D-SOM$^*_{\text{AVG}}$, D-SOM$_{\text{MED}}$, D-SOM$^*_{\text{MED}}$ and K-MEANS, with respect to the $f^*_{\text{NR-R}}$ fitness measure, in Figure 6.4. This fitness measure reflects the percentage of alternatives that are indifferent, with respect to the crisp credibility degree of this relation, towards the map node to which they are assigned.

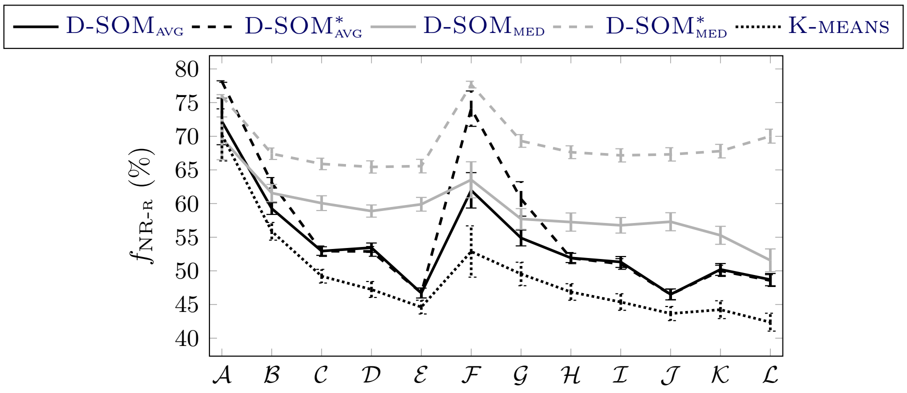


D-SOM$_{\text{AVG}}$	89.6 (6.3)	87.4 (4.2)	71.9 (3.9)	72.8 (3.8)	40.0 (4.7)	77.3 (5.3)	74.6 (4.2)	66.2 (3.9)	63.8 (4.2)	40.2 (4.6)	58.7 (4.8)	52.1 (4.8)
D-SOM$^*_{\text{AVG}}$	100.0 (0.0)	98.5 (1.3)	70.6 (4.4)	69.5 (3.9)	40.1 (4.8)	96.7 (4.1)	87.0 (4.9)	65.1 (4.4)	62.4 (4.3)	40.2 (4.6)	57.9 (4.4)	51.5 (4.5)
D-SOM$_{\text{MED}}$	85.7 (6.9)	83.1 (5.2)	81.7 (5.2)	79.0 (5.0)	81.1 (5.1)	75.5 (5.9)	73.2 (5.2)	71.5 (5.5)	70.2 (5.3)	71.8 (5.1)	64.7 (5.4)	54.3 (5.1)
D-SOM$^*_{\text{MED}}$	99.9 (0.1)	99.3 (0.9)	99.6 (0.9)	99.7 (0.7)	99.7 (0.7)	99.9 (0.2)	99.5 (0.9)	99.6 (0.8)	99.7 (0.7)	99.7 (0.7)	99.7 (0.6)	99.7 (0.7)
K-MEANS	86.6 (6.9)	71.8 (5.4)	51.0 (4.9)	44.5 (4.9)	32.9 (4.7)	59.7 (7.3)	55.0 (5.3)	43.8 (4.8)	39.9 (4.7)	30.6 (4.4)	36.6 (4.7)	31.5 (4.5)

Figure 6.4: Comparing D-SOM variants using the $f^*_{\text{NR-R}}$ measure;

We can clearly observe that all of the proposed approaches perform better than K-MEANS. This means that we are able to achieve a result which contains more alternatives in each cluster that are indifferent to the representative of that cluster. We additionally observe that the non-adaptive variants perform rather similarly, with a small increase in performance of D-SOM$_{\text{MED}}$ over D-SOM$_{\text{AVG}}$. However, the adaptive variant D-SOM$^*_{\text{MED}}$ provides significant improvements over all the other approaches, managing to propose results which contain almost all the alternatives in the dataset being indifferent to the representative of the cluster they are in.

We turn our attention towards characterising these results using the $f_{\text{NR-R}}$ measure, in Figure 6.5. This measure additionally accounts for the degree of credibility between the alternatives inside a cluster and the representative of that cluster.



D-SOM$_{\text{AVG}}$	72.2 (3.5)	59.3 (0.9)	52.9 (0.6)	53.4 (0.7)	46.7 (0.7)	62.0 (2.6)	54.9 (1.2)	51.9 (0.7)	51.3 (0.8)	46.5 (0.8)	50.2 (0.9)	48.7 (0.9)
D-SOM$^*_{\text{AVG}}$	78.1 (0.1)	63.1 (0.8)	52.9 (0.8)	52.9 (0.7)	46.7 (0.7)	74.1 (2.6)	60.7 (2.6)	51.9 (0.8)	51.0 (0.8)	46.5 (0.8)	50.0 (0.8)	48.6 (0.9)
D-SOM$_{\text{MED}}$	69.7 (3.2)	61.6 (1.3)	60.1 (1.1)	58.9 (0.9)	59.9 (1.0)	63.5 (2.7)	57.7 (1.5)	57.2 (1.4)	56.8 (1.2)	57.3 (1.3)	55.3 (1.3)	51.6 (1.7)
D-SOM$^*_{\text{MED}}$	76.0 (0.2)	67.4 (0.9)	65.9 (0.9)	65.4 (0.9)	65.6 (1.0)	77.7 (0.5)	69.3 (0.9)	67.6 (1.0)	67.2 (1.0)	67.3 (1.0)	67.8 (1.0)	70.0 (1.0)
K-MEANS	70.2 (3.9)	55.9 (1.3)	49.2 (1.0)	47.2 (1.2)	44.6 (1.0)	52.9 (3.8)	49.5 (1.7)	46.9 (1.2)	45.4 (1.2)	43.7 (1.0)	44.2 (1.3)	42.4 (1.3)

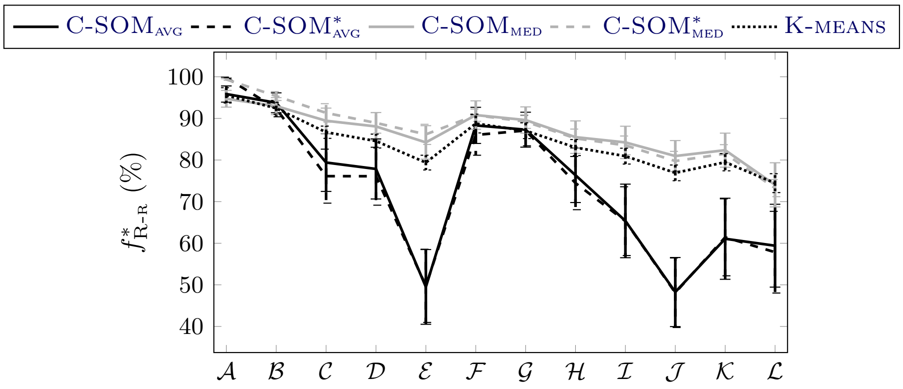
Figure 6.5: Comparing D-SOM variants using the $f_{\text{NR-R}}$ measure;

We find similar remarks as those made when considering the previous fitness measure. K-MEANS is again performing the worst, while the D-SOM$_{\text{MED}}$ approaches perform better than the D-SOM$_{\text{AVG}}$ ones. Furthermore, D-SOM$^*_{\text{MED}}$ clearly dominates the rest.

6.2.2 The chained SOM on indifferences

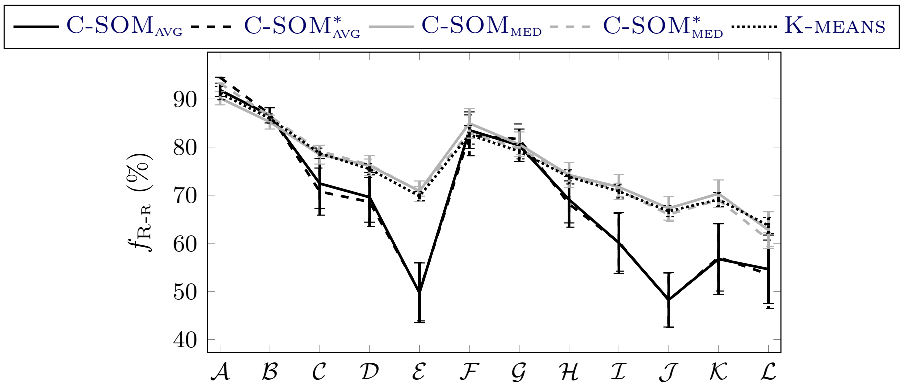
We continue with the chained SOMs, and illustrate the average results, with respect to the $f^*_{\text{R-R}}$ and $f_{\text{R-R}}$ fitness measures, in the same experimental framework as before. We fix the order between the clusters given by the mapping of the C-SOM, so that the two fitness measures will describe how well this approach is able to find such a structure.

We illustrate the results of the four C-SOM variants, but also those of the K-MEANS algorithm, in Figures 6.6 and 6.7, following 50 executions of the algorithms over each of the 300 problem instances.



C-SOM$_{\text{AVG}}$	95.9 (2.0)	93.8 (2.4)	79.4 (6.9)	77.9 (7.2)	49.5 (9.0)	88.3 (4.3)	87.3 (4.2)	76.3 (6.6)	65.4 (8.8)	48.3 (8.3)	61.1 (9.7)	59.4 (10.0)
C-SOM$^*_{\text{AVG}}$	99.8 (0.1)	92.4 (2.0)	76.1 (6.5)	76.1 (7.0)	49.8 (8.8)	86.0 (4.9)	87.1 (3.7)	74.5 (6.4)	65.3 (8.3)	48.1 (8.4)	61.4 (9.3)	57.8 (9.8)
C-SOM$_{\text{MED}}$	94.7 (2.0)	93.0 (2.0)	89.4 (3.0)	88.1 (3.3)	84.2 (4.0)	90.8 (3.4)	89.6 (3.2)	85.5 (3.9)	84.2 (3.9)	81.0 (3.7)	82.4 (4.1)	74.0 (5.3)
C-SOM$^*_{\text{MED}}$	99.4 (0.2)	95.4 (1.1)	91.3 (2.2)	88.9 (2.5)	86.2 (2.4)	90.8 (1.5)	89.2 (1.6)	85.2 (2.2)	83.5 (2.3)	79.8 (2.3)	81.5 (2.2)	73.2 (2.1)
K-MEANS	95.5 (1.8)	92.4 (1.5)	86.7 (1.5)	84.6 (1.7)	79.4 (1.8)	88.8 (2.3)	87.0 (1.9)	83.0 (1.9)	80.9 (1.9)	76.9 (1.9)	79.5 (2.1)	74.4 (2.3)

Figure 6.6: Comparing C-SOM variants using the $f^*_{\text{R-R}}$ measure;



C-SOM$_{\text{AVG}}$	91.8 (1.4)	86.6 (1.6)	72.4 (5.2)	69.6 (5.2)	49.7 (6.2)	83.5 (3.8)	80.4 (3.4)	69.1 (4.8)	60.1 (6.3)	48.3 (5.7)	56.7 (7.3)	54.6 (7.1)
C-SOM$^*_{\text{AVG}}$	94.5 (0.1)	86.8 (1.4)	70.7 (4.9)	68.6 (5.1)	49.9 (6.1)	82.4 (4.2)	81.7 (3.1)	68.2 (4.8)	60.2 (6.0)	48.2 (5.7)	57.1 (7.0)	53.6 (7.1)
C-SOM$_{\text{MED}}$	90.2 (1.4)	85.2 (1.4)	78.4 (2.0)	76.2 (2.0)	70.8 (2.1)	84.9 (3.1)	80.6 (2.6)	74.2 (2.6)	71.7 (2.6)	67.3 (2.5)	70.3 (2.8)	62.7 (3.8)
C-SOM$^*_{\text{MED}}$	93.2 (0.1)	86.4 (0.6)	79.2 (1.1)	76.3 (1.2)	70.8 (1.0)	83.0 (1.2)	79.7 (1.2)	73.9 (1.4)	71.1 (1.4)	66.0 (1.5)	69.1 (1.5)	60.7 (1.5)
K-MEANS	91.2 (1.4)	85.9 (0.9)	78.7 (1.1)	75.4 (1.1)	69.8 (1.0)	82.5 (1.9)	79.1 (1.6)	73.8 (1.5)	70.8 (1.4)	66.7 (1.1)	69.1 (1.5)	63.7 (1.7)

Figure 6.7: Comparing C-SOM variants using the $f_{\text{R-R}}$ measure;

We may observe in the first figure, that the K-MEANS algorithm is able to consistently provide results which contain alternatives that are indifferent to the representative of their cluster, but that are also strictly preferred to the representatives from the clusters placed below in the order, and strictly preferred by the representatives of the clusters that are placed above. Placing the chain topology over the

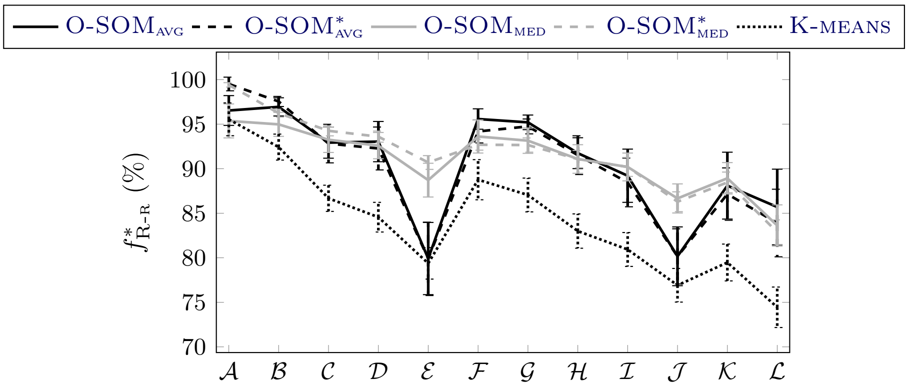
SOM is able to bring improvements only when the nodes in the map are chosen from the existing alternatives, while the approach of constructing the nodes from the averages of the evaluations of the alternatives in each cluster is offering poorer results.

Considering the results with respect to the $f_{\text{R-R}}$ measure, we notice almost no improvements of the chained self-organizing maps over K-MEANS with respect to it. Furthermore, the approaches that build the map nodes from the averages of the evaluations of the alternatives assigned to them give poorer results that the rest. We may, at this point, deduce either that the C-SOM is not able to find good clusters of indifferent alternatives that are also ordered, or that the approach that clusters using measures of similarity is sufficient to find such a structure. We will investigate this further considering the ordered self-organizing maps.

6.2.3 The ordered SOM on indifferences

We conclude our experiments of using the SOM adaptations to MCDA, with the ordered SOMs.

We illustrate the results of the four variants of this approach with respect to the $f^*_{\text{R-R}}$ measure, and those of K-MEANS, in Figure 6.8.

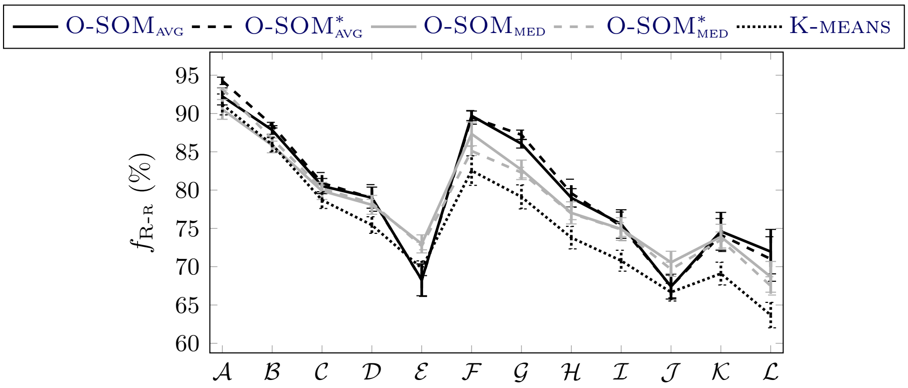


O-SOM$_{\text{AVG}}$	96.5 (1.7)	96.9 (1.0)	92.9 (1.8)	93.0 (2.3)	79.9 (4.1)	95.6 (1.1)	95.2 (0.8)	91.7 (1.7)	89.2 (3.0)	80.2 (3.3)	88.1 (3.8)	85.7 (4.2)
O-SOM$^*_{\text{AVG}}$	99.5 (0.8)	97.5 (0.6)	92.8 (2.2)	92.3 (2.4)	79.9 (4.1)	94.2 (1.3)	94.7 (0.8)	91.5 (2.2)	88.5 (2.7)	80.2 (3.1)	87.1 (2.9)	83.9 (3.8)
O-SOM$_{\text{MED}}$	95.4 (1.9)	95.0 (1.3)	93.2 (1.4)	92.6 (1.5)	88.7 (1.9)	93.6 (1.9)	93.1 (1.4)	91.1 (1.6)	90.2 (1.6)	86.7 (1.6)	88.9 (1.8)	83.6 (2.3)
O-SOM$^*_{\text{MED}}$	99.4 (0.3)	96.2 (0.4)	94.3 (0.6)	93.6 (0.6)	90.7 (0.8)	92.7 (0.6)	92.7 (0.8)	91.1 (1.0)	90.2 (1.1)	86.3 (1.1)	88.5 (1.2)	82.9 (1.4)
K-MEANS	95.5 (1.8)	92.4 (1.5)	86.7 (1.5)	84.6 (1.7)	79.4 (1.8)	88.8 (2.3)	87.0 (1.9)	83.0 (1.9)	80.9 (1.9)	76.9 (1.9)	79.5 (2.1)	74.4 (2.3)

Figure 6.8: Comparing O-SOM variants using the $f^*_{\text{R-R}}$ measure;

In this case, we observe that the ordered self-organizing maps are able to propose better results than K-MEANS. While the adaptive variants provide only in certain cases marginal improvements over the non-adaptive ones, we also find that the approaches that select the map nodes from the existing alternatives are not always performing better than the other variants, as was the case for the other topologies. Nevertheless, it appears that the approaches that select the nodes from the existing alternatives offer more stable results across all of the tested benchmarks.

In Figure 6.9 we present the results with respect to the $f_{\text{R-R}}$ measure, which additionally measures the degree of credibility of the considered relations.



O-SOM$_{\text{AVG}}$	92.2 (1.1)	87.8 (0.6)	80.5 (1.0)	79.0 (1.4)	68.3 (2.1)	89.7 (0.6)	86.1 (0.6)	79.0 (1.2)	75.6 (1.9)	67.4 (1.6)	74.6 (2.5)	72.0 (2.9)
O-SOM$^*_{\text{AVG}}$	94.3 (0.5)	88.5 (0.4)	80.9 (1.4)	79.0 (1.7)	68.2 (2.1)	89.5 (0.9)	87.2 (0.6)	79.6 (1.8)	75.3 (1.9)	67.4 (1.5)	74.1 (2.2)	71.0 (2.9)
O-SOM$_{\text{MED}}$	90.5 (1.3)	85.8 (1.0)	79.9 (1.1)	78.1 (1.2)	73.0 (1.2)	87.3 (1.6)	82.6 (1.3)	77.0 (1.4)	74.9 (1.5)	70.6 (1.5)	73.9 (1.6)	68.7 (2.0)
O-SOM$^*_{\text{MED}}$	93.2 (0.1)	86.7 (0.3)	80.2 (0.5)	78.3 (0.5)	72.7 (0.5)	85.1 (0.6)	82.3 (0.7)	76.9 (0.8)	74.9 (0.9)	69.7 (0.9)	73.5 (1.0)	67.5 (1.2)
K-MEANS	91.2 (1.4)	85.9 (0.9)	78.7 (1.1)	75.4 (1.1)	69.8 (1.0)	82.5 (1.9)	79.1 (1.6)	73.8 (1.5)	70.8 (1.4)	66.7 (1.1)	69.1 (1.5)	63.7 (1.7)

Figure 6.9: Comparing O-SOM variants using the $f_{\text{R-R}}$ measure;

Most of the statements from before apply here too, only that the differences between the different fitness values become smaller. This is a natural effect, since we are also measuring the credibility degrees of the relations of indifference between the alternatives inside a cluster and the map node to which they are assigned, but also the credibility degrees of the relations of strict preference between these alternatives and the other nodes in the map.

We may conclude that both the D-SOM and the O-SOM approaches provide significant improvements over K-MEANS, considering the fitness measures that we have investigated.

6.3 CLIP and other approaches

We continue by evaluating the performance of the second type of approach for solving the problem of clustering in MCDA that has been proposed in this work, the CLIP approach.

The performance of each of the two steps of the method will be independently evaluated, followed by a comparative analysis between all of the proposed approaches and several existing ones.

6.3.1 First step of CLIP

The first step of the CLIP approach aims at constructing an initial partition of the alternatives, with the purpose of maximizing the number of relations of indifference inside a cluster and minimizing the number of such relations between different clusters. However, this initial partition does not need to be the optimal one with respect to this definition, but come only close to it. The second step of the CLIP approach then further processes this initial partition, in order to filter the relations between clusters, so that different structures can be extracted.

Besides using, for the first step, any other clustering approach that groups alternatives that are indifferent and separates alternatives that are not, we have also proposed an approach that is inspired from Graph Theory, and uses the concept of cluster core to find the dense regions in the graph induced from the relations of indifference. The approach can either be exact, requiring an enumeration of all maximal cliques in the graph, or approximative, in the form of a meta-heuristic similar to the Glow-worm meta-heuristic [Krishnanand and Ghose 2005], and specifically developed for this case.

Table 6.1: Results of the exact approach of the first step of CLIP;

	Measure									
	f^*_{NR} (%)	f_{NR} (%)	f^{min}_{NR} (%)	f^*_{R} (%)	f_{R} (%)	f^{min}_{R} (%)	Tr (%)	f^*_{PC} (%)	f_{PC} (%)	f^{max}_{PC} (%)
$\mathcal{A}$	99.1 (0.0)	91.1 (0.0)	31.3 (0.0)	99.0 (0.0)	91.0 (0.0)	31.3 (0.0)	100.0	0.1 (0.0)	3.4 (0.0)	25.8 (0.0)
$\mathcal{B}$	93.6 (0.0)	84.0 (0.0)	1.6 (0.0)	90.3 (0.0)	81.7 (0.0)	1.1 (0.0)	100.0	6.6 (0.0)	75.2 (0.0)	97.3 (0.0)
$\mathcal{C}$	90.4 (0.0)	77.3 (0.0)	0.9 (0.0)	85.6 (0.0)	75.0 (0.0)	0.9 (0.0)	100.0	9.6 (0.0)	66.1 (0.0)	95.3 (0.0)
$\mathcal{D}$	88.9 (0.0)	76.6 (0.0)	0.0 (0.0)	83.3 (0.0)	72.0 (0.0)	0.0 (0.0)	96.0	11.0 (0.0)	66.1 (0.0)	99.6 (0.0)
$\mathcal{E}$	87.1 (0.0)	78.2 (0.0)	0.0 (0.0)	79.6 (0.0)	67.2 (0.0)	0.0 (0.0)	88.0	7.6 (0.0)	60.7 (0.0)	90.5 (0.0)
$\mathcal{F}$	98.4 (0.0)	89.4 (0.0)	0.0 (0.0)	95.2 (0.0)	85.8 (0.0)	0.0 (0.0)	16.0	2.9 (0.0)	78.0 (0.0)	88.5 (0.0)
$\mathcal{G}$	93.2 (0.0)	83.1 (0.0)	0.0 (0.0)	86.2 (0.0)	76.4 (0.0)	0.0 (0.0)	40.0	9.0 (0.0)	75.8 (0.0)	97.3 (0.0)
$\mathcal{H}$	89.6 (0.0)	77.5 (0.0)	0.0 (0.0)	79.8 (0.0)	69.0 (0.0)	0.0 (0.0)	80.0	11.2 (0.0)	67.1 (0.0)	97.6 (0.0)
$\mathcal{I}$	89.1 (0.0)	77.9 (0.0)	0.0 (0.0)	77.4 (0.0)	66.3 (0.0)	0.0 (0.0)	60.0	11.4 (0.0)	64.7 (0.0)	96.4 (0.0)
$\mathcal{J}$	89.0 (0.0)	81.4 (0.0)	0.0 (0.0)	72.3 (0.0)	62.3 (0.0)	0.0 (0.0)	40.0	6.3 (0.0)	60.4 (0.0)	90.0 (0.0)
$\mathcal{K}$	90.0 (0.0)	80.4 (0.0)	0.0 (0.0)	74.8 (0.0)	64.6 (0.0)	0.0 (0.0)	40.0	8.5 (0.0)	62.9 (0.0)	93.3 (0.0)
$\mathcal{L}$	93.0 (0.0)	85.7 (0.0)	0.0 (0.0)	69.3 (0.0)	61.5 (0.0)	0.0 (0.0)	0.0	3.8 (0.0)	61.0 (0.0)	86.9 (0.0)

In Table 6.1 we present the results of using the exact approach of extracting cluster cores, which are then transformed into a clustering result by adding each alternative to the core that it is most indifferent to, considering only the crisp credibility degrees. The approach has been executed only once on each problem instance, as it is deterministic.

In order to evaluate the performance of this approach we need only to look at the first three columns, which measure how well the proposed clusters are reflected by the relations of indifference between the alternatives.

The first measure considers only the crisp credibility degrees of these relations, which is the only information that the clustering approach directly uses, therefore this is the most relevant evaluation measure, in this case. We notice very high values for this measure, across all types of benchmarks, therefore highlighting the good performance of the approach. Furthermore, the average credibility degrees of the relations that support the proposed sets of clusters, depicted in the second column, are also quite high. However, the last fitness measure of this type, in the third column, shows that in most cases there exists at least one relation between two alternatives that is strongly opposing the proposed clustering result. Nevertheless, the first step of the CLIP approach is not designed to improve this last measure.

We additionally observe that the fitness of the results decreases, as we consider more and more difficult problem instances. The first five types of problems do not have relations of incomparability actively constructed inside them, therefore we notice a steady decrease of the first two fitness measures. The following five types of problem instances are the same as the previous five, except that in this case more relations of incomparability are embedded in them. Therefore, we notice that the fitness on the $\mathcal{F}$ type of problem instances, with respect to the first two measures increase following the results for $\mathcal{E}$, only to decrease again over the following four types of benchmarks. The last two benchmark types are the same as the $\mathcal{I}$ type of benchmarks, only with much more relations of incomparability inside them.

Considering the following three fitness measures, which account also for the type of relation between two clusters, and measure the overall support that only one type of relation is present between two given clusters, we find, as expected, slightly lower measures than the previous ones. In this case, the relations between clusters are constructed using the approach of aggregating the outranking relations between the alternatives of any pair of clusters, and deriving the preferential relation from them.

Furthermore, although over the first types of benchmarks the approach is able to consistently find ordered sets of clusters, as seen in the seventh column, when considering the more difficult problem instances this is no longer the case. For the last type of benchmarks, for instance, the algorithm never finds an order between the proposed clusters, which is not surprising considering the abundance of relations of incomparability that exist between the alternatives inside them.

Finally, the measures related to the property of preferential consistency show a relatively small number of relations between pairs of alternatives that contradict a relation of strict preference between the clusters in which they are placed. However, the average degrees of credibility of these relations, in some cases are rather high, while with the exception of the first type of benchmarks, we always find at least one such relation with a very high credibility degree, as seen in the last column of the table.

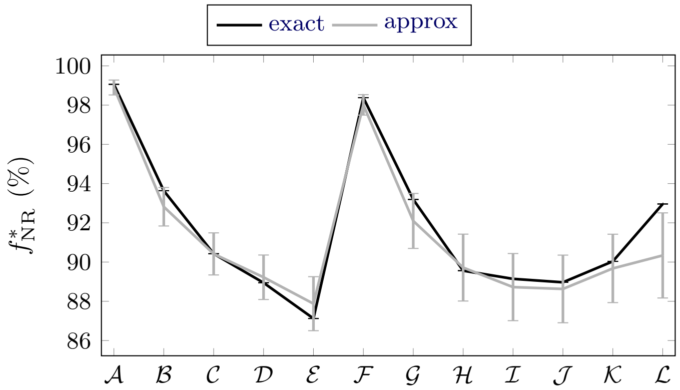
In the following set of results, in Table 6.2, we highlight the performance of the approximative approach of finding cluster cores. As in the previous case, these cores are then used for constructing the initial clustering result.

The difference between the performance of this approach and that of the previous one are not immediately visible, as the two perform similarly. One aspect that may be quickly noticed is that fact that this approach is stochastic, as seen through the non-zero standard deviations next to the average of the results on each measure.

Table 6.2: Results of the approximative approach of the first step of CLIP;

	Measure									
	f^*_{NR} (%)	f_{NR} (%)	f^{min}_{NR} (%)	f^*_{R} (%)	f_{R} (%)	f^{min}_{R} (%)	Tr (%)	f^*_{PC} (%)	f_{PC} (%)	f^{max}_{PC} (%)
$\mathcal{A}$	98.9 (0.4)	91.0 (0.2)	27.6 (10.2)	98.8 (0.7)	90.9 (0.5)	27.5 (10.3)	100.0	0.2 (0.7)	8.5 (20.6)	30.2 (17.9)
$\mathcal{B}$	92.8 (1.0)	84.0 (0.6)	3.9 (6.1)	90.6 (2.3)	82.4 (1.7)	3.5 (5.7)	100.0	4.4 (3.2)	69.2 (14.9)	88.4 (15.3)
$\mathcal{C}$	90.4 (1.1)	77.8 (0.6)	3.9 (6.3)	87.7 (2.4)	76.5 (1.4)	3.0 (5.1)	100.0	5.4 (3.3)	64.2 (6.0)	90.5 (11.3)
$\mathcal{D}$	89.3 (1.2)	77.3 (0.7)	0.0 (0.1)	85.4 (2.4)	73.5 (1.4)	0.0 (0.0)	99.8	7.5 (3.4)	62.7 (4.4)	91.9 (9.9)
$\mathcal{E}$	87.9 (1.5)	79.1 (1.0)	0.0 (0.0)	81.8 (2.9)	68.6 (1.5)	0.0 (0.0)	95.2	7.3 (2.5)	59.5 (1.9)	88.1 (8.2)
$\mathcal{F}$	98.0 (0.5)	89.2 (0.3)	0.0 (0.0)	94.3 (1.4)	85.1 (1.0)	0.0 (0.0)	20.6	3.8 (2.0)	81.7 (11.5)	91.1 (8.3)
$\mathcal{G}$	92.1 (1.5)	82.4 (1.1)	0.0 (0.0)	84.4 (2.7)	75.2 (2.0)	0.0 (0.0)	66.9	10.1 (3.1)	76.7 (3.2)	98.6 (2.7)
$\mathcal{H}$	89.7 (1.7)	77.9 (1.2)	0.0 (0.0)	80.3 (2.7)	69.3 (1.7)	0.0 (0.0)	87.8	10.2 (3.2)	66.7 (2.9)	95.2 (5.2)
$\mathcal{I}$	88.8 (1.8)	77.8 (1.3)	0.0 (0.0)	77.4 (3.0)	66.3 (1.8)	0.0 (0.0)	77.4	10.8 (3.0)	63.4 (2.4)	94.2 (5.8)
$\mathcal{J}$	88.6 (1.7)	81.4 (1.4)	0.0 (0.0)	73.7 (3.0)	62.9 (1.8)	0.0 (0.0)	66.3	6.3 (2.1)	59.5 (2.0)	87.5 (7.8)
$\mathcal{K}$	89.6 (1.8)	80.2 (1.4)	0.0 (0.0)	74.0 (3.1)	64.2 (1.9)	0.0 (0.0)	54.0	8.7 (2.5)	63.1 (2.4)	94.0 (5.3)
$\mathcal{L}$	90.4 (2.2)	83.6 (1.8)	0.0 (0.0)	65.4 (3.2)	58.9 (2.1)	0.0 (0.0)	5.9	4.3 (1.4)	62.4 (2.3)	89.8 (6.1)

In Figure 6.10 we present the results of the two approaches, considering the f^*_{NR} fitness measure.

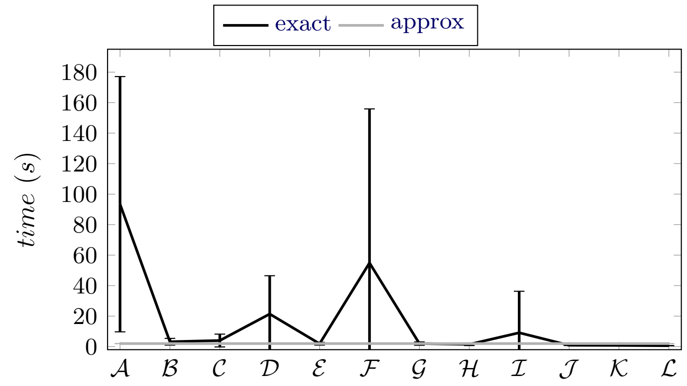


exact	99.1 (0.0)	93.6 (0.0)	90.4 (0.0)	88.9 (0.0)	87.1 (0.0)	98.4 (0.0)	93.2 (0.0)	89.6 (0.0)	89.1 (0.0)	89.0 (0.0)	90.0 (0.0)	93.0 (0.0)
approx	98.9 (0.4)	92.8 (1.0)	90.4 (1.1)	89.2 (1.1)	87.9 (1.4)	98.0 (0.5)	92.1 (1.4)	89.7 (1.7)	88.7 (1.7)	88.6 (1.7)	89.7 (1.7)	90.3 (2.2)

Figure 6.10: Comparing the two approaches for the first step of CLIP, using the f^*_{NR} measure;

Looking at the average performance of the two approaches, we find that the loss in performance of the approximative approach with respect to the exact one is marginal. In certain cases, it is even performing better than the first. This result derives from the definition of the cluster cores, and the procedure of constructing a set of clusters from them, which only guarantees that the alternatives placed in the same cluster will be mostly indifferent to the core, but not to the other alternatives that are grouped together with them.

The execution times required by each of the two approaches are highlighted in Figure 6.11.



exact	93.4 (83.7)	3.2 (2.2)	4.0 (4.2)	21.4 (25.1)	1.9 (0.9)	54.9 (101.0)	2.0 (1.1)	1.5 (0.8)	9.1 (27.2)	0.9 (0.2)	0.8 (0.1)	0.6 (0.1)
approx	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)	2.0 (0.0)

Figure 6.11: Comparing the two approaches for the first step of CLIP, using the execution times;

Considering the above, but also the very high execution times of the first method, for certain types of benchmarks, we conclude that the approximative approach of constructing the cluster cores is better suited than the exact one. The high execution times of the exact approach are directly linked to the number of indifference relations that exist between the alternatives inside a benchmark, but also to these relations being between the same alternatives. Therefore, the exact method performs slowly over the benchmarks of type $\mathcal{A}$, due to the fact that the alternatives inside each cluster are mostly indifferent to each other. This is also the case for the benchmarks of type $\mathcal{F}$, due to the fact that they only differ from those of type $\mathcal{A}$ through the addition of large performance differences for certain alternatives on only one criterion.

6.3.2 Second step of CLIP

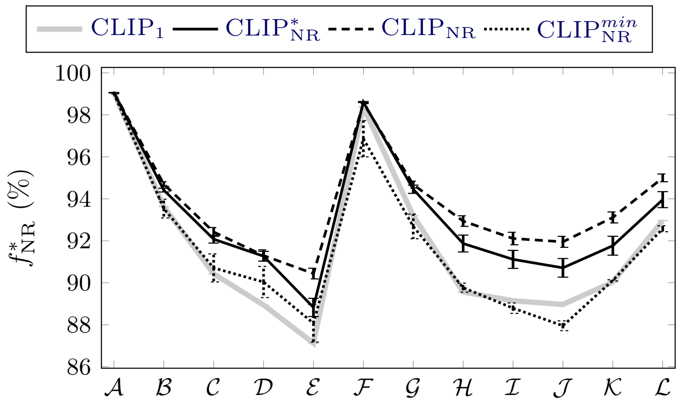
We continue with the second step of the CLIP approach, which takes the initial result from the first step and performs small changes to it, in an incremental fashion using a meta-heuristic approach, in order to bring it closer to a particular type of desired structure.

In the following, we will consider that the exact approach of the first step is used, so that all of the approaches of refining the initial partition, that we will consider, start from the same result. In practice, we may resort to using the approximative variant of the first step.

We denote the first step of the CLIP approach with CLIP$_1$, and any of the variants of the second step with CLIP$^{\text{fitness type}}_{\text{structure type}}$. The fitness type will be denoted with either a $*$, if the fitness function uses only the crisp credibility degrees of the relations, with no symbol if the average credibility degrees of the supporting relations are reflected through the fitness function, or with a *min*, if the minimum credibility degree supporting the proposed results is used as a fitness for the set of clusters. The structure type can be any of: NR (non-relational set of clusters), PT (partial tournament), CT (complete tournament), SPO (strict partial order) or SCO (strict complete order).

For each of the approaches related to the second step of the CLIP method, we have considered both a local search and a simulated annealing implementations [Talbi 2009]. However, as the simulated annealing approach performed better than the local search approach on all the considered benchmarks, we have chosen to highlight further only the results with respect to it. Furthermore, the simulated annealing parameters have been tuned over a selected few benchmarks.

We begin by analysing the approaches of constructing a non-relational clustering result. We highlight the improvements of the $\text{CLIP}^*_{\text{NR}}$, CLIP_{NR} and $\text{CLIP}^{min}_{\text{NR}}$ approaches in relation to the quality of the results given by CLIP_1, using each of the three fitness measures f^*_{NR}, f_{NR} and f^{min}_{NR}. Each algorithm was left to run for up to one minute, although most have been observed to converge to a final solution within a $1-2$ second execution time.



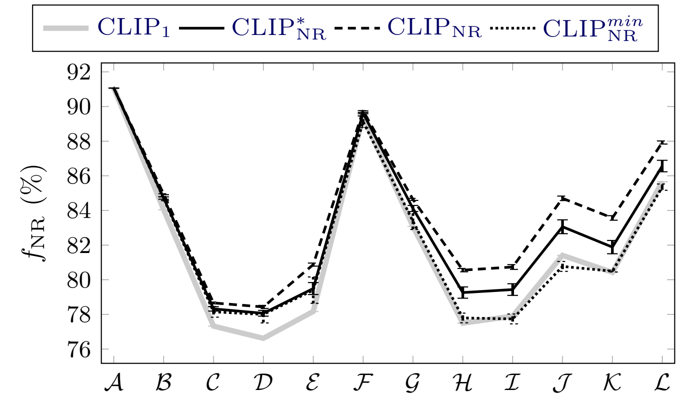
CLIP_1	99.1 (0.0)	93.6 (0.0)	90.4 (0.0)	88.9 (0.0)	87.1 (0.0)	98.4 (0.0)	93.2 (0.0)	89.6 (0.0)	89.1 (0.0)	89.0 (0.0)	90.0 (0.0)	93.0 (0.0)
$\text{CLIP}^*_{\text{NR}}$	99.1 (0.0)	94.4 (0.1)	92.1 (0.2)	91.3 (0.2)	88.8 (0.4)	98.6 (0.0)	94.5 (0.2)	91.9 (0.4)	91.1 (0.4)	90.7 (0.4)	91.8 (0.5)	94.0 (0.4)
CLIP_{NR}	99.1 (0.0)	94.7 (0.1)	92.4 (0.2)	91.3 (0.3)	90.4 (0.3)	98.6 (0.0)	94.7 (0.2)	92.9 (0.3)	92.1 (0.3)	92.0 (0.3)	93.1 (0.3)	95.0 (0.2)
$\text{CLIP}^{min}_{\text{NR}}$	99.0 (0.0)	93.5 (0.4)	90.7 (0.7)	90.0 (0.7)	88.1 (0.9)	96.9 (0.9)	92.7 (0.6)	89.8 (0.2)	88.8 (0.2)	88.0 (0.2)	90.1 (0.0)	92.6 (0.1)

Figure 6.12: Comparing CLIP_1, $\text{CLIP}^*_{\text{NR}}$, CLIP_{NR} and $\text{CLIP}^{min}_{\text{NR}}$ using f^*_{NR};

Looking at the results from Figure 6.12, we observe that the $\text{CLIP}^*_{\text{NR}}$ and CLIP_{NR} approaches improve on the results of the CLIP_1 approach, with respect to the f^*_{NR} fitness measure. However, we find that the CLIP_{NR} approach performs slightly better than the $\text{CLIP}^*_{\text{NR}}$ approach. The latter, as it directly uses the crisp credibility degrees inside the heuristic function of the meta-heuristic, was expected to perform better than the former approach, which considers the valued credibility degrees. Nevertheless, the improvements do not go above a 2% margin.

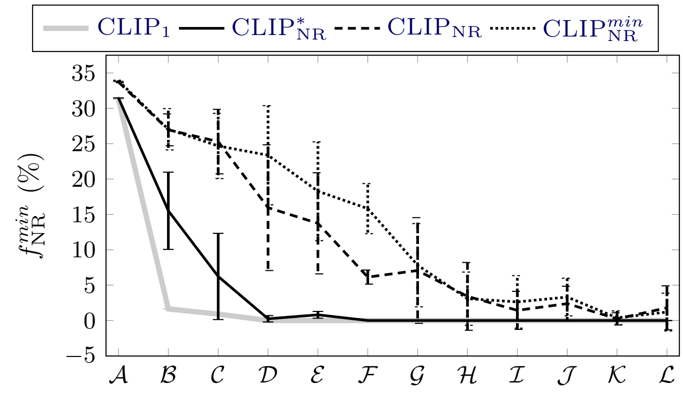
The $\text{CLIP}^{min}_{\text{NR}}$ approach may be seen to lead to a decrease in performance with respect to the considered fitness measure, which may be expected, as it tries to raise the minimum degree of credibility supporting the clustering results, and all of them.

The results of the same approaches are further highlighted in Figure 6.13, from the perspective of the f_{NR} fitness measure, while in Figure 6.14 we present them from perspective of the f_{NR}^{min} fitness measure.



CLIP$_1$	91.1 (0.0)	84.0 (0.0)	77.3 (0.0)	76.6 (0.0)	78.2 (0.0)	89.4 (0.0)	83.1 (0.0)	77.5 (0.0)	77.9 (0.0)	81.4 (0.0)	80.4 (0.0)	85.7 (0.0)
CLIP$^*_{\mathrm{NR}}$	91.1 (0.0)	84.7 (0.1)	78.3 (0.1)	78.1 (0.2)	79.5 (0.3)	89.6 (0.0)	84.1 (0.2)	79.3 (0.3)	79.4 (0.3)	83.1 (0.4)	81.9 (0.4)	86.6 (0.3)
CLIP$_{\mathrm{NR}}$	91.1 (0.0)	84.9 (0.0)	78.7 (0.1)	78.4 (0.1)	80.9 (0.1)	89.8 (0.0)	84.6 (0.1)	80.6 (0.1)	80.7 (0.1)	84.7 (0.1)	83.6 (0.1)	87.9 (0.1)
CLIP$^{min}_{\mathrm{NR}}$	91.1 (0.0)	84.7 (0.1)	78.1 (0.3)	78.0 (0.5)	79.4 (0.8)	89.1 (0.3)	83.4 (0.5)	77.8 (0.3)	77.7 (0.3)	80.8 (0.3)	80.5 (0.0)	85.3 (0.2)

Figure 6.13: Comparing CLIP$_1$, CLIP$^*_{\mathrm{NR}}$, CLIP$_{\mathrm{NR}}$ and CLIP$^{min}_{\mathrm{NR}}$ using f_{NR};



CLIP$_1$	31.3 (0.0)	1.6 (0.0)	0.9 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
CLIP$^*_{\mathrm{NR}}$	31.5 (0.0)	15.5 (5.5)	6.2 (6.1)	0.2 (0.5)	0.8 (0.5)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
CLIP$_{\mathrm{NR}}$	33.7 (0.1)	27.0 (2.3)	25.3 (4.6)	16.0 (8.9)	13.7 (7.2)	6.1 (1.0)	7.1 (7.5)	3.4 (4.8)	1.4 (2.7)	2.4 (2.5)	0.2 (0.9)	1.8 (3.1)
CLIP$^{min}_{\mathrm{NR}}$	33.9 (0.1)	27.0 (2.9)	24.7 (4.6)	23.4 (7.0)	18.3 (7.0)	15.8 (3.5)	7.8 (5.9)	3.1 (3.8)	2.6 (3.7)	3.3 (2.7)	0.4 (1.0)	1.2 (2.7)

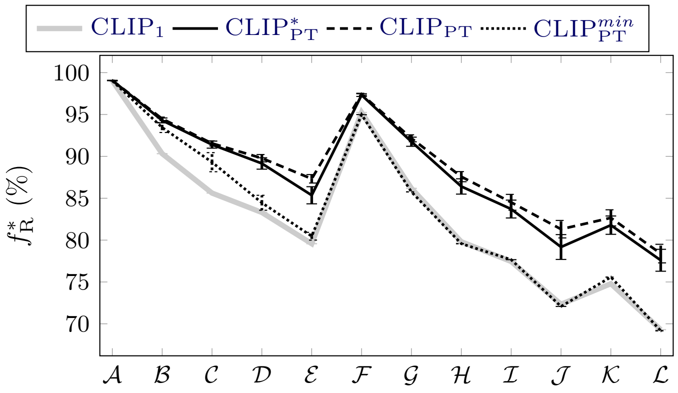
Figure 6.14: Comparing CLIP$_1$, CLIP$^*_{\mathrm{NR}}$, CLIP$_{\mathrm{NR}}$ and CLIP$^{min}_{\mathrm{NR}}$ using f^{min}_{NR};

In the first case, the CLIP_{NR} approach outperforms the others across all the considered benchmarks. Furthermore, it finds these results more consistently than the other approaches, as seen through the standard deviation values. This is an expected result, as this approach is meant to improve the initial clustering results given by CLIP_1 in this way.

The following set of results highlights the last and most restrictive fitness measure for a non-relational clustering result, f_{NR}^{min}. We can observe that CLIP_1 only in a few cases finds results with the minimum support of a relation between two alternatives above 0. Each of the following methods improves slightly this results, however it is clear that the $\text{CLIP}_{\text{NR}}^{min}$ approach is able to perform better. This is especially evident over the increasingly more difficult benchmarks. Nevertheless, across all the benchmarks, the relation that holds the minimum support for the proposed clustering result rarely goes above a 35% credibility degree, hence the structure inside each benchmark is not trivial.

Continuing with the approaches that look for a relational clustering result, we highlight the results of the methods that search for the least restrictive structure of this type: $\text{CLIP}_{\text{PT}}^*$, CLIP_{PT} and $\text{CLIP}_{\text{PT}}^{min}$. These methods allow for any type of structure between clusters, ranging from a partial tournament to a strict complete order, to be found.

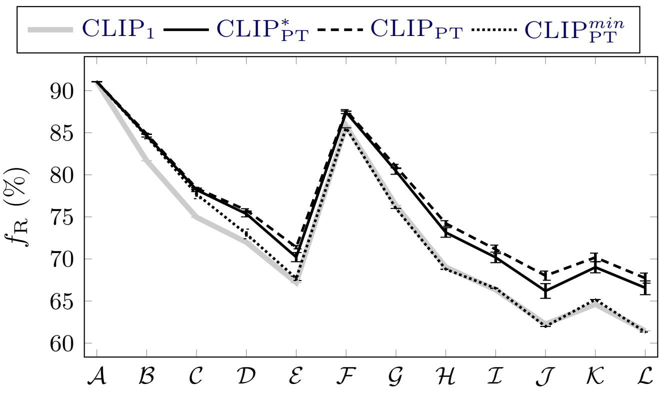
In Figure 6.15, we first highlight the performance of these algorithm considering the first fitness measure, f_{R}^*.



CLIP_1	99.0 (0.0)	90.3 (0.0)	85.6 (0.0)	83.3 (0.0)	79.6 (0.0)	95.2 (0.0)	86.2 (0.0)	79.8 (0.0)	77.4 (0.0)	72.3 (0.0)	74.8 (0.0)	69.3 (0.0)
$\text{CLIP}_{\text{PT}}^*$	99.1 (0.0)	94.3 (0.3)	91.4 (0.4)	89.2 (0.7)	85.4 (1.0)	97.3 (0.2)	91.7 (0.5)	86.4 (0.9)	83.7 (1.1)	79.2 (1.5)	81.8 (1.1)	77.6 (1.3)
CLIP_{PT}	99.1 (0.0)	94.5 (0.2)	91.5 (0.3)	89.8 (0.4)	87.3 (0.5)	97.4 (0.1)	92.2 (0.4)	87.6 (0.6)	84.5 (0.9)	81.3 (1.0)	82.7 (1.0)	78.4 (1.1)
$\text{CLIP}_{\text{PT}}^{min}$	99.0 (0.0)	93.4 (0.6)	89.3 (1.1)	84.5 (0.9)	80.5 (0.5)	95.0 (0.0)	85.8 (0.0)	79.6 (0.0)	77.7 (0.0)	72.1 (0.0)	75.6 (0.0)	69.2 (0.0)

Figure 6.15: Comparing CLIP_1, $\text{CLIP}_{\text{PT}}^*$, CLIP_{PT} and $\text{CLIP}_{\text{PT}}^{min}$ using f_{R}^*;

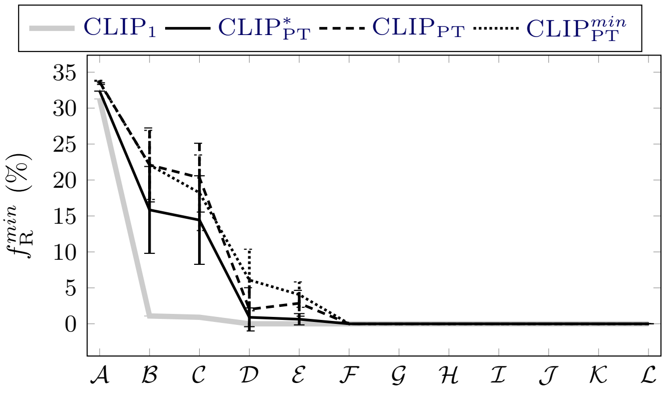
As with the case of the approaches oriented towards finding a non-relational clustering result, we observe that the second of the approaches oriented towards finding a relational clustering result, CLIP_{PT}, performs the best with respect to this fitness measure, and not the first one. Nevertheless, the $\text{CLIP}_{\text{PT}}^*$ approach comes very close to its performance, while the CLIP$^{min}_{\text{PT}}$ does not seem to improve the results in this regard.



CLIP$_1$	91.0 (0.0)	81.7 (0.0)	75.0 (0.0)	72.0 (0.0)	67.2 (0.0)	85.8 (0.0)	76.4 (0.0)	69.0 (0.0)	66.3 (0.0)	62.3 (0.0)	64.6 (0.0)	61.5 (0.0)
CLIP$^*_{\text{PT}}$	91.1 (0.0)	84.7 (0.2)	78.2 (0.2)	75.4 (0.3)	70.2 (0.5)	87.4 (0.1)	80.4 (0.4)	73.1 (0.6)	70.2 (0.6)	66.2 (0.9)	69.0 (0.7)	66.6 (0.8)
CLIP$_{\text{PT}}$	91.1 (0.0)	84.9 (0.1)	78.4 (0.1)	75.8 (0.2)	71.4 (0.2)	87.6 (0.1)	81.0 (0.3)	74.2 (0.3)	71.1 (0.5)	68.0 (0.5)	70.2 (0.5)	67.7 (0.6)
CLIP$^{min}_{\text{PT}}$	91.1 (0.0)	84.6 (0.3)	77.7 (0.5)	73.0 (0.6)	67.7 (0.3)	85.6 (0.0)	76.0 (0.0)	68.8 (0.0)	66.5 (0.0)	62.0 (0.0)	65.2 (0.0)	61.3 (0.0)

Figure 6.16: Comparing CLIP$_1$, CLIP$^*_{\text{PT}}$, CLIP$_{\text{PT}}$ and CLIP$^{min}_{\text{PT}}$ using f_{R};

When considering the second fitness function, f_{R}, the results from Figure 6.16 illustrate that the CLIP$_{\text{PT}}$ approach performs better than the rest, with as much as a 5% increase in the fitness of the initial results given by CLIP$_1$. CLIP$^*_{\text{PT}}$ closely follows it, while CLIP$^{min}_{\text{PT}}$ is again last.



CLIP$_1$	31.3 (0.0)	1.1 (0.0)	0.9 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
CLIP$^*_{\text{PT}}$	32.4 (0.0)	15.8 (6.0)	14.4 (6.2)	0.9 (1.3)	0.6 (0.8)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
CLIP$_{\text{PT}}$	33.6 (0.1)	22.1 (5.1)	20.3 (4.8)	2.0 (3.0)	2.9 (1.8)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
CLIP$^{min}_{\text{PT}}$	33.6 (0.3)	22.1 (4.8)	18.2 (5.3)	6.1 (4.3)	4.1 (1.8)	0.0 (0.1)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)

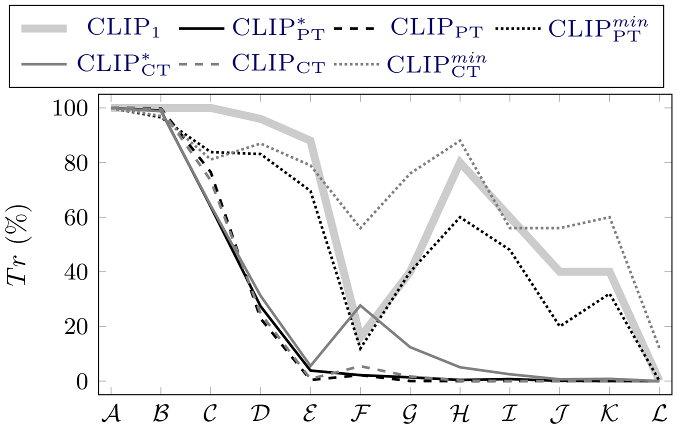
Figure 6.17: Comparing CLIP$_1$, CLIP$^*_{\text{PT}}$, CLIP$_{\text{PT}}$ and CLIP$^{min}_{\text{PT}}$ using f^{min}_{R};

$\text{CLIP}^{min}_{\text{PT}}$ is however performing the best when considering the f^{min}_{R} fitness measure. This is seen in Figure 6.17.

It is easily noticeable that finding a good result from the perspective of this fitness measure is very difficult, even more so than in the case of a non-relation clustering result. In this case, we try to find a good relational clustering result with the lowest support of at least one relation between alternatives as high as possible. However, as soon as we consider the benchmarks containing relations of incomparability embedded inside them, neither of the considered approaches is able to increase the measure above the minimum level.

When considering the other approaches that look for increasingly stricter relational clustering results (complete tournament, strict partial and strict complete orders), similar remarks as those presented so far can be made. We will therefore not present in such detail the results of these approaches.

We will continue, however, to analyse how the different relational clustering approaches perform, when considering the transitivity of the relations that are found between the clusters that they propose. This is highlighted in Figure 6.18.



CLIP_1	100.0	100.0	100.0	96.0	88.0	16.0	40.0	80.0	60.0	40.0	40.0	0.0
$\text{CLIP}^*_{\text{PT}}$	100.0	99.0	63.8	27.4	3.8	2.2	1.4	0.4	0.7	0.1	0.0	0.0
CLIP_{PT}	100.0	99.8	76.5	22.9	0.4	2.2	0.0	0.0	0.0	0.0	0.0	0.0
$\text{CLIP}^{min}_{\text{PT}}$	100.0	96.6	83.8	83.1	69.6	11.9	40.0	60.0	48.0	20.0	32.0	0.0
$\text{CLIP}^*_{\text{CT}}$	100.0	98.8	64.2	31.2	5.5	27.7	12.4	5.0	2.5	0.6	0.8	0.0
CLIP_{CT}	100.0	99.7	73.4	24.7	0.8	5.4	1.7	0.2	0.0	0.0	0.0	0.0
$\text{CLIP}^{min}_{\text{CT}}$	100.0	97.0	81.1	87.0	79.0	56.0	76.0	88.0	56.0	56.0	60.0	12.0

Figure 6.18: Comparing CLIP_1 and all the approaches forming the second step of CLIP, using the Tr transitivity measure;

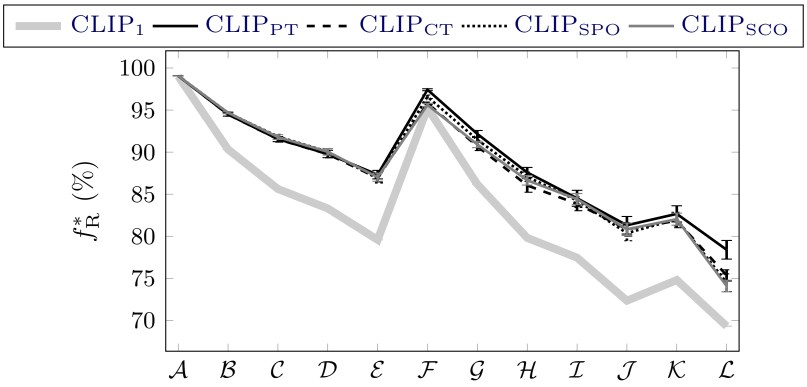
Due to the fact that the approaches that look for either a strict partial order or a strict complete order between the resulting set of clusters always propose sets of clusters that hold the maximum value for this measure, we have omitted to display them in Figure 6.18.

We may easily observe that most of the considered approaches lead to a decrease in the number of sets of clusters containing an order between them being proposed.

The approaches looking to optimize the f_{R}^{min} fitness function deviate slightly from this remark. The $\mathrm{CLIP}_{\mathrm{PT}}^{min}$ approach, however, always leads to a decrease in this measure as opposed to the CLIP_1 approach, while the $\mathrm{CLIP}_{\mathrm{CT}}^{min}$ approach finds, in certain cases, more results containing relations that are transitive between the proposed clusters. We consider this to be a result of restricting the relations between clusters to only those of strict preference.

The last remark is also noticeable considering the approaches that try to optimize the other two types of fitness functions, f_{R}^{*} and f_{R}, as the $\mathrm{CLIP}_{\mathrm{CT}}^{*}$ and $\mathrm{CLIP}_{\mathrm{CT}}$ approaches more often find ordered sets of clusters as opposed to $\mathrm{CLIP}_{\mathrm{PT}}^{*}$ and $\mathrm{CLIP}_{\mathrm{PT}}$.

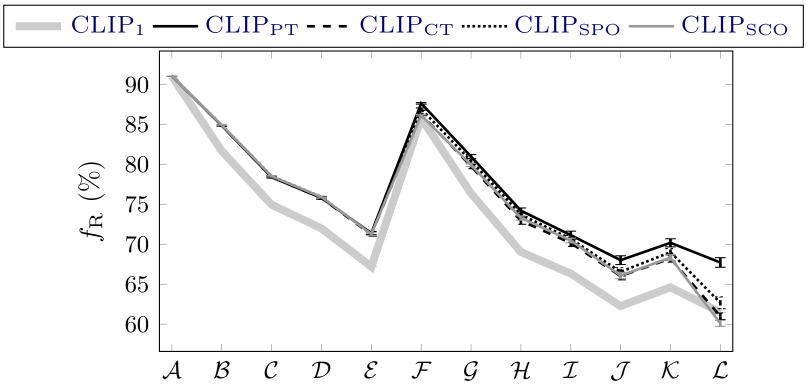
The following analysis considers each of the three fitness functions and the methods that have been shown to perform best according to them. We begin with the f_{R}^{*} fitness measure. We highlight the results of the $\mathrm{CLIP}_{\mathrm{PT}}$, $\mathrm{CLIP}_{\mathrm{CT}}$, $\mathrm{CLIP}_{\mathrm{SPO}}$ and $\mathrm{CLIP}_{\mathrm{SCO}}$ approaches, as they have been previously shown to perform slightly better than the $\mathrm{CLIP}_{\mathrm{PT}}^{*}$, $\mathrm{CLIP}_{\mathrm{CT}}^{*}$, $\mathrm{CLIP}_{\mathrm{SPO}}^{*}$ and $\mathrm{CLIP}_{\mathrm{SCO}}^{*}$ approaches.



CLIP_1	99.0 (0.0)	90.3 (0.0)	85.6 (0.0)	83.3 (0.0)	79.6 (0.0)	95.2 (0.0)	86.2 (0.0)	79.8 (0.0)	77.4 (0.0)	72.3 (0.0)	74.8 (0.0)	69.3 (0.0)
$\mathrm{CLIP}_{\mathrm{PT}}$	99.1 (0.0)	94.3 (0.3)	91.4 (0.4)	89.2 (0.7)	85.4 (1.0)	97.3 (0.2)	91.7 (0.5)	86.4 (0.9)	83.7 (1.1)	79.2 (1.5)	81.8 (1.1)	77.6 (1.3)
$\mathrm{CLIP}_{\mathrm{CT}}$	99.1 (0.0)	94.3 (0.3)	91.1 (0.5)	89.2 (0.6)	86.7 (0.6)	95.9 (0.1)	90.9 (0.5)	86.2 (0.9)	84.4 (0.8)	81.2 (0.8)	82.5 (0.8)	75.4 (0.7)
$\mathrm{CLIP}_{\mathrm{SPO}}$	99.1 (0.0)	94.5 (0.2)	91.4 (0.4)	89.5 (0.5)	86.8 (0.6)	96.7 (0.2)	91.7 (0.4)	87.6 (0.5)	85.0 (0.7)	81.0 (1.0)	82.8 (0.7)	75.7 (0.9)
$\mathrm{CLIP}_{\mathrm{SCO}}$	99.1 (0.0)	94.6 (0.2)	91.6 (0.3)	89.8 (0.5)	87.0 (0.5)	95.7 (0.2)	91.3 (0.3)	87.1 (0.6)	85.1 (0.6)	81.9 (0.6)	82.6 (0.7)	74.9 (0.7)

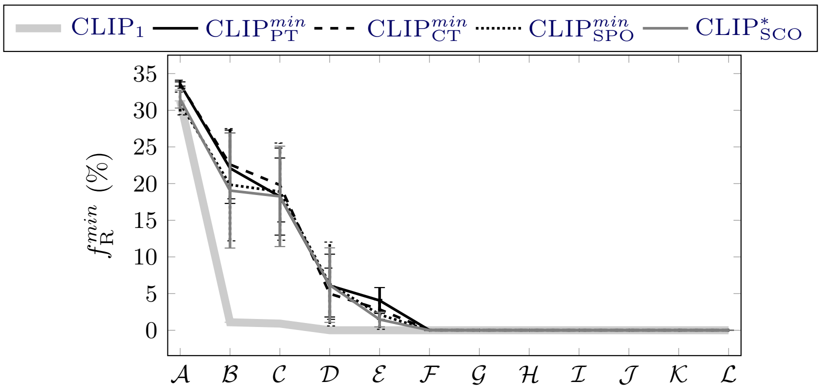
Figure 6.19: Comparing the second step approaches of CLIP, using the f_{R}^{*} fitness measure;

In Figure 6.19 we observe that the $\mathrm{CLIP}_{\mathrm{PT}}$ approach provides the best improvement over the initial solution given by CLIP_1, considering the f_{R}^{*} fitness measure. Nevertheless, all of the considered approaches perform very close to $\mathrm{CLIP}_{\mathrm{PT}}$, while the $\mathrm{CLIP}_{\mathrm{SPO}}$ and $\mathrm{CLIP}_{\mathrm{SCO}}$ approaches furthermore consider that the clusters are ordered.



CLIP$_1$	91.0 (0.0)	81.7 (0.0)	75.0 (0.0)	72.0 (0.0)	67.2 (0.0)	85.8 (0.0)	76.4 (0.0)	69.0 (0.0)	66.3 (0.0)	62.3 (0.0)	64.6 (0.0)	61.5 (0.0)
CLIP$_{\mathrm{PT}}$	91.1 (0.0)	84.9 (0.1)	78.4 (0.1)	75.8 (0.2)	71.4 (0.2)	87.6 (0.1)	81.0 (0.3)	74.2 (0.3)	71.1 (0.5)	68.0 (0.5)	70.2 (0.5)	67.7 (0.6)
CLIP$_{\mathrm{CT}}$	91.1 (0.0)	84.9 (0.1)	78.4 (0.1)	75.8 (0.2)	71.2 (0.2)	86.3 (0.1)	79.8 (0.3)	73.0 (0.4)	70.2 (0.4)	66.0 (0.5)	68.2 (0.5)	61.0 (0.4)
CLIP$_{\mathrm{SPO}}$	91.1 (0.0)	84.9 (0.0)	78.5 (0.1)	75.9 (0.1)	71.3 (0.2)	86.9 (0.1)	80.5 (0.2)	73.7 (0.3)	70.7 (0.4)	66.6 (0.5)	69.0 (0.4)	62.7 (0.7)
CLIP$_{\mathrm{SCO}}$	91.1 (0.0)	84.9 (0.0)	78.5 (0.1)	75.9 (0.1)	71.3 (0.2)	86.1 (0.1)	79.9 (0.2)	73.3 (0.3)	70.5 (0.3)	66.1 (0.4)	68.3 (0.4)	60.1 (0.4)

Figure 6.20: Comparing the second step approaches of CLIP, using the f_{R} fitness measure;



CLIP$_1$	31.3 (0.0)	1.1 (0.0)	0.9 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
CLIP$_{\mathrm{PT}}^{min}$	33.6 (0.3)	22.1 (4.8)	18.2 (5.3)	6.1 (4.3)	4.1 (1.8)	0.0 (0.1)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
CLIP$_{\mathrm{CT}}^{min}$	33.6 (0.5)	22.6 (4.7)	19.8 (5.0)	5.0 (3.5)	2.8 (1.4)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
CLIP$_{\mathrm{SPO}}^{min}$	30.9 (1.6)	19.8 (7.6)	18.9 (6.6)	6.3 (5.7)	2.1 (2.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
CLIP$_{\mathrm{SCO}}^{min}$	31.5 (1.2)	19.0 (7.8)	18.3 (6.8)	6.2 (5.1)	1.5 (1.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)

Figure 6.21: Comparing the second step approaches of CLIP, using the f_{R}^{min} fitness measure;

In Figure 6.20, we find similar remarks as before, however in this case considering the f_{R} fitness measure. It is further clearer, when looking at the last types of benchmarks, that the approaches that do not consider placing relations of incomparability between clusters perform slightly poorer. This is due to the abundance of relations of incomparability inside these types of benchmarks. Furthermore, even when these relations are placed between clusters, trying to build a strict partial order between these clusters is taking away from the fitness of the results, as seen by the difference between the CLIP_{PT} and CLIP_{SPO} approaches.

Finally, we consider the last fitness measure, and the approaches that are best suited for it in Figure 6.21.

In this case too, we find that all of the considered approaches are able to raise the fitness of the results, however only over the first types of benchmarks. Furthermore, the approaches that do not restrict the result to a strictly ordered set of clusters, either complete or partial, seem to perform better both with respect to the average of the considered fitness measure, but also with respect to the consistency with which these results are found.

We conclude that the CLIP_{PT} approach is performing best when considering the f^*_{R} and f_{R} fitness measures for characterising the clustering results. Additionally, the $\text{CLIP}^{min}_{\text{PT}}$ is performing best with respect to the f^{min}_{PT} fitness measure. However, the equivalents of these approaches, which search for ordered structures between the proposed clusters, also perform well considering all of these fitness measures.

Similar conclusions can be drawn when searching for a non-relational clustering result, with CLIP_{NR} holding the best performance when considering the f^*_{NR} and f_{NR} fitness measures, and $\text{CLIP}^{min}_{\text{NR}}$ performing the best when considering f^{min}_{NR}.

6.3.3 Comparative analysis

In the following, we will consider two additional clustering approaches:

- the classical K-means approach (denoted with K-means);
- the Multi-criteria Relational Clustering approach from [de Smet and Eppe 2009] (denoted with MRC).

The classical K-means approach tries to minimize the average Euclidian distance between the alternatives in the dataset and the centroids of the clusters in which they are assigned.

The second approach is an adaptation of K-means, however it uses a distance constructed from the preferential relations that can be found between the alternatives. Furthermore, certain preferential relations are also extracted between every pair of clusters, therefore the approach is suited to finding a relational clustering result. In our experimental framework, we will be using the crisp credibility degrees of the preferential relations that can be constructed from the outranking relation selected in the beginning of this chapter.

We begin by illustrating the results of the K-means algorithm with respect to all the fitness measures defined in this work.

Considering that all the criteria are defined on the same scales, and furthermore that they are equally important, we are able to notice in Table 6.3 that the K-means

Table 6.3: Results of the K-MEANS algorithm;

	Measure									
	f^*_{NR} (%)	f_{NR} (%)	f^{min}_{NR} (%)	f^*_{R} (%)	f_{R} (%)	f^{min}_{R} (%)	Tr (%)	f^*_{PC} (%)	f_{PC} (%)	f^{max}_{PC} (%)
$\mathcal{A}$	94.9 (2.2)	89.1 (1.1)	2.6 (4.3)	94.7 (2.2)	88.9 (1.2)	1.2 (3.3)	100.0	0.0 (0.0)	0.1 (0.9)	25.0 (5.1)
$\mathcal{B}$	90.6 (1.1)	83.8 (0.5)	9.0 (5.1)	89.7 (1.4)	83.3 (0.6)	3.9 (4.4)	99.8	0.2 (0.2)	50.1 (14.4)	59.3 (7.6)
$\mathcal{C}$	87.1 (0.8)	77.2 (0.3)	6.8 (4.5)	84.5 (1.4)	76.3 (0.5)	4.8 (4.3)	99.0	2.6 (1.0)	56.9 (1.3)	74.5 (6.6)
$\mathcal{E}$	85.6 (1.0)	76.8 (0.4)	0.1 (0.4)	81.9 (1.4)	73.4 (0.5)	0.0 (0.1)	95.7	5.3 (1.3)	57.9 (1.0)	80.4 (5.4)
$\mathcal{D}$	84.0 (0.9)	78.5 (0.4)	0.0 (0.0)	76.4 (1.7)	67.2 (0.7)	0.0 (0.0)	92.8	8.8 (1.7)	58.7 (0.9)	84.2 (5.4)
$\mathcal{F}$	88.7 (2.3)	83.8 (1.7)	0.0 (0.0)	86.2 (2.3)	80.9 (1.6)	0.0 (0.0)	82.6	1.2 (0.7)	72.1 (7.1)	90.1 (7.7)
$\mathcal{G}$	87.3 (1.6)	81.0 (1.0)	0.0 (0.0)	84.0 (1.6)	77.0 (0.9)	0.0 (0.0)	82.2	1.7 (0.8)	62.4 (3.0)	85.1 (7.2)
$\mathcal{H}$	86.3 (1.2)	77.6 (0.7)	0.0 (0.0)	80.2 (1.4)	70.9 (0.6)	0.0 (0.0)	84.8	4.6 (1.3)	59.3 (1.3)	83.2 (5.6)
$\mathcal{J}$	85.8 (1.1)	78.1 (0.6)	0.0 (0.0)	77.5 (1.3)	67.9 (0.6)	0.0 (0.0)	85.9	7.6 (1.5)	59.3 (1.0)	85.3 (5.3)
$\mathcal{I}$	85.6 (0.9)	81.3 (0.6)	0.0 (0.0)	72.9 (1.4)	62.4 (0.7)	0.0 (0.0)	89.4	10.6 (1.9)	59.5 (0.9)	86.7 (5.1)
$\mathcal{K}$	86.1 (1.2)	80.0 (0.8)	0.0 (0.0)	75.4 (1.3)	65.6 (0.7)	0.0 (0.0)	81.0	7.8 (1.5)	59.5 (1.0)	85.1 (5.1)
$\mathcal{L}$	87.1 (1.1)	82.7 (0.9)	0.0 (0.0)	67.9 (1.3)	58.2 (1.0)	0.0 (0.0)	64.6	8.2 (1.8)	60.8 (1.2)	87.4 (4.8)

algorithm performs rather well with respect to most of the presented measures. This is a direct consequence of the previous mentions, but also following the consideration from Chapter 3. When non-zero indifference thresholds are considered in conjunction with the use of an outranking relation for comparing alternatives together, the Euclidian distance is able, in more than 60% of the cases, to characterise indifferent alternatives through a distance lower than the median level, and non-indifferent alternatives with a distance above this level. Hence, while considering the first non-relational fitness measure, the K-MEANS approach performs rather well.

The second non-relational fitness measure is also showing a rather good performance of the K-MEANS algorithm, however the third illustrates that the approach very often proposes results that are strongly opposed by at least one pair of alternatives.

In the case of the relational clustering fitness measures, the performance of the algorithm decreases. This is an expected result, as these functions discriminate more between the different types of relations that exist between alternatives.

Looking at the Tr column, we find that in many cases the clusters of alternatives that are found are in an ordered structure. We consider this to be an effect of grouping alternatives that have similar evaluations. As there is a certain guarantee that the evaluations are close to each other, the way in which these alternatives compare to the rest is also closely linked.

Finally, the last three columns illustrate a moderate number of relations between clusters that oppose the property of preferential consistency. The last two columns also show results that are similar to other approaches we have tested so far.

We continue with the MRC approach, and illustrate its results in Table 6.4.

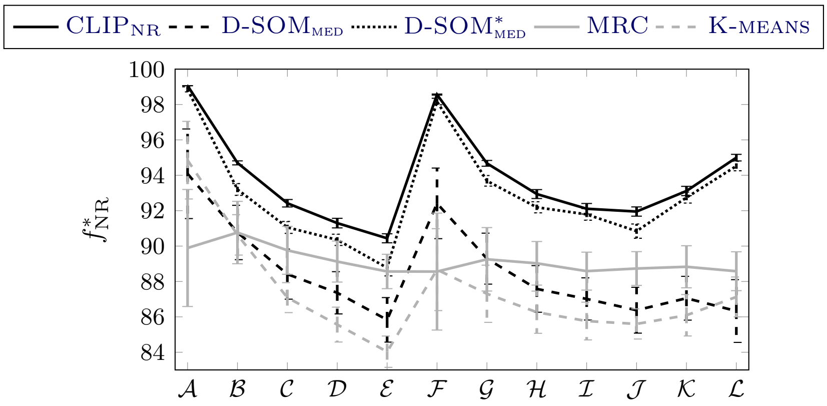
The first three columns show slightly better results as those of the K-MEANS approach, when considering the non-relational fitness measures. However, the improvements over the relational fitness measures are considerably better, which is not

Table 6.4: Results of the MRC algorithm;

	Measure									
	f^*_{NR} (%)	f_{NR} (%)	f^{min}_{NR} (%)	f^*_{R} (%)	f_{R} (%)	f^{min}_{R} (%)	Tr (%)	f^*_{PC} (%)	f_{PC} (%)	f^{max}_{PC} (%)
$\mathcal{A}$	89.9 (3.3)	85.6 (2.2)	0.1 (0.2)	89.8 (3.3)	85.5 (2.3)	0.1 (0.2)	100.0	0.0 (0.2)	0.7 (4.5)	22.5 (6.3)
$\mathcal{B}$	90.8 (1.8)	83.5 (0.8)	5.6 (3.8)	90.6 (1.8)	83.3 (0.8)	4.9 (3.9)	100.0	0.0 (0.0)	22.2 (19.8)	49.5 (3.1)
$\mathcal{C}$	89.8 (1.4)	77.9 (0.4)	12.6 (3.7)	89.2 (1.4)	77.7 (0.5)	11.9 (3.8)	99.9	0.4 (0.1)	53.3 (1.8)	59.3 (3.4)
$\mathcal{E}$	89.1 (1.2)	77.7 (0.4)	0.0 (0.2)	88.4 (1.1)	75.3 (0.4)	0.0 (0.2)	99.8	0.9 (0.2)	54.4 (0.5)	62.1 (2.8)
$\mathcal{D}$	88.6 (1.0)	79.8 (0.4)	0.0 (0.0)	86.6 (1.1)	70.8 (0.4)	0.0 (0.0)	97.0	1.7 (0.3)	54.9 (0.4)	64.3 (2.3)
$\mathcal{F}$	88.6 (3.3)	82.2 (2.5)	0.0 (0.0)	87.5 (3.3)	80.6 (2.4)	0.0 (0.0)	100.0	0.1 (0.2)	23.2 (21.0)	46.8 (11.8)
$\mathcal{G}$	89.3 (1.8)	80.7 (1.1)	0.0 (0.0)	87.5 (1.7)	77.9 (0.9)	0.0 (0.0)	100.0	0.1 (0.1)	49.6 (11.6)	60.0 (5.2)
$\mathcal{H}$	89.0 (1.2)	77.6 (0.7)	0.0 (0.0)	85.9 (1.1)	72.4 (0.5)	0.0 (0.0)	99.2	0.8 (0.2)	54.9 (0.8)	64.4 (3.6)
$\mathcal{J}$	88.6 (1.1)	78.2 (0.6)	0.0 (0.0)	84.4 (0.9)	70.0 (0.4)	0.0 (0.0)	97.7	1.4 (0.3)	54.7 (0.5)	64.1 (2.6)
$\mathcal{I}$	88.7 (1.0)	81.8 (0.7)	0.0 (0.0)	82.2 (0.7)	66.2 (0.4)	0.0 (0.0)	93.6	2.1 (0.3)	55.1 (0.4)	65.7 (3.1)
$\mathcal{K}$	88.8 (1.2)	80.1 (0.8)	0.0 (0.0)	82.7 (0.9)	68.0 (0.5)	0.0 (0.0)	96.5	1.6 (0.3)	55.0 (0.5)	66.0 (3.5)
$\mathcal{L}$	88.6 (1.1)	82.1 (0.9)	0.0 (0.0)	75.7 (0.9)	61.0 (1.0)	0.0 (0.0)	89.0	1.7 (0.4)	55.4 (0.7)	67.7 (4.7)

surprising considering the nature of the approach. We furthermore notice that the results that are proposed fall into ordered structures more often then those of the previous approach.

We may also notice several good values with respect to the preferential consistency fitness measures, in the last three columns.



CLIP$_{\mathrm{NR}}$	99.1 (0.0)	94.7 (0.1)	92.4 (0.2)	91.3 (0.3)	90.4 (0.3)	98.6 (0.0)	94.7 (0.2)	92.9 (0.3)	92.1 (0.3)	92.0 (0.3)	93.1 (0.3)	95.0 (0.2)
D-SOM$_{\mathrm{MED}}$	94.1 (2.5)	90.8 (1.5)	88.4 (1.4)	87.4 (1.2)	85.8 (1.3)	92.4 (2.0)	89.3 (1.4)	87.6 (1.3)	87.0 (1.2)	86.4 (1.3)	87.1 (1.2)	86.3 (1.8)
D-SOM$^*_{\mathrm{MED}}$	98.9 (0.1)	93.2 (0.3)	91.1 (0.3)	90.4 (0.3)	88.8 (0.5)	98.2 (0.2)	93.7 (0.3)	92.2 (0.3)	91.8 (0.3)	90.8 (0.4)	92.7 (0.3)	94.5 (0.3)
MRC	89.9 (3.3)	90.8 (1.8)	89.8 (1.4)	89.1 (1.2)	88.6 (1.0)	88.6 (3.3)	89.3 (1.8)	89.0 (1.2)	88.6 (1.1)	88.7 (1.0)	88.8 (1.2)	88.6 (1.1)
K-MEANS	94.9 (2.2)	90.6 (1.1)	87.1 (0.8)	85.6 (1.0)	84.0 (0.9)	88.7 (2.3)	87.3 (1.6)	86.3 (1.2)	85.8 (1.1)	85.6 (0.9)	86.1 (1.2)	87.1 (1.1)

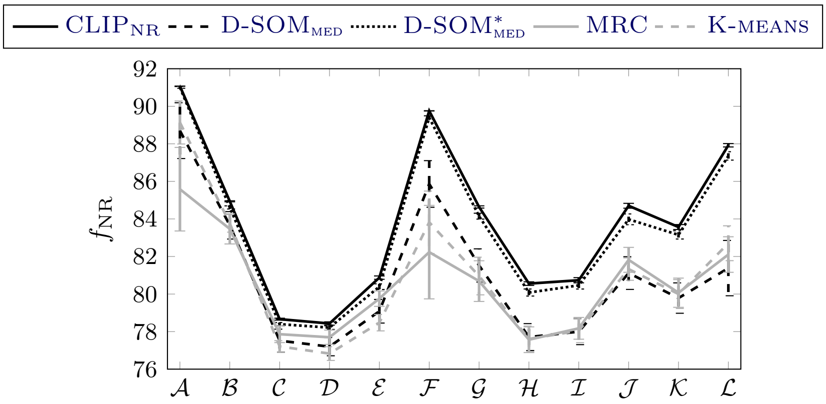
Figure 6.22: Comparison using the f^*_{NR} fitness measure;

We continue by comparing these two approaches as well as the ones presented before, from the perspective of each of the non-relational and relational fitness measures defined in this work. We will illustrate only one CLIP variant, in each case, and both a non-adaptive and an adaptive SOM variant. The inclusion of the non-adaptive variant is done in order to compare fairly the performance of these algorithms and the K-MEANS and MRC approaches. These last two are not able to adapt to the problem, and require that the number of clusters is given beforehand.

In Figure 6.22, we illustrate the results of the CLIP$_{\mathrm{NR}}$, D-SOM$_{\mathrm{MED}}$, D-SOM$^{*}_{\mathrm{MED}}$, MRC and K-MEANS algorithms with respect to the f^{*}_{NR} fitness measure.

It it quite clear that the CLIP approach outperforms the others from the perspective of this fitness measure. The K-MEANS algorithm performs overall the worst, however this is not a surprising outcome, as it uses notions of similarity for grouping alternatives together. The MRC approach performs more or less the same across all the considered benchmarks, while the D-SOM$_{\mathrm{MED}}$ approach in certain cases performs better, while in others worse than MRC. The D-SOM$^{*}_{\mathrm{MED}}$ approach is seen to perform clearly better than the non-adaptive variant.

In Figure 6.23 we consider the fitness measure that accounts for the average degree of credibility of the relations of indifference between the alternatives placed inside the same cluster, and those of non-indifference between the alternatives placed in different clusters. The same approaches are used in this case too.



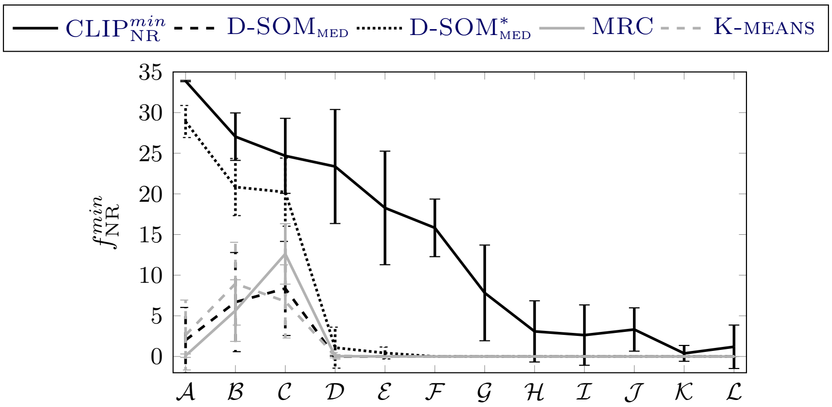
CLIP$_{\mathrm{NR}}$	91.1 (0.0)	84.9 (0.0)	78.7 (0.1)	78.4 (0.1)	80.9 (0.1)	89.8 (0.0)	84.6 (0.1)	80.6 (0.1)	80.7 (0.1)	84.7 (0.1)	83.6 (0.1)	87.9 (0.1)
D-SOM$_{\mathrm{MED}}$	88.7 (1.5)	83.7 (0.7)	77.5 (0.6)	77.2 (0.5)	79.1 (0.6)	85.9 (1.2)	81.5 (0.9)	77.7 (0.7)	78.0 (0.7)	81.1 (0.9)	79.8 (0.8)	81.4 (1.5)
D-SOM$^{*}_{\mathrm{MED}}$	91.0 (0.0)	84.6 (0.1)	78.4 (0.1)	78.2 (0.1)	80.4 (0.2)	89.4 (0.1)	84.1 (0.2)	80.1 (0.2)	80.5 (0.2)	84.0 (0.3)	83.2 (0.2)	87.4 (0.2)
MRC	85.6 (2.2)	83.5 (0.8)	77.9 (0.4)	77.7 (0.4)	79.8 (0.4)	82.2 (2.5)	80.7 (1.1)	77.6 (0.7)	78.2 (0.6)	81.8 (0.7)	80.1 (0.8)	82.1 (0.9)
K-MEANS	89.1 (1.1)	83.8 (0.5)	77.2 (0.3)	76.8 (0.4)	78.5 (0.4)	83.8 (1.7)	81.0 (1.0)	77.6 (0.7)	78.1 (0.6)	81.3 (0.6)	80.0 (0.8)	82.7 (0.9)

Figure 6.23: Comparison using the f_{NR} fitness measure;

The CLIP variant and the adaptive D-SOM perform better than the rest, however the difference in performance between all the considered approaches is smaller than in the previous case. The K-MEANS algorithm can be also seen to perform better than before, while the MRC approach has a similar behaviour to it. The good performance of K-MEANS is motivated, in this case, by the similarities between the indifference measure and the similarity measure constructed from the

Euclidian distance, which, considering the use of the same criteria scales, and their equal significance weights, are rather high. We remind at this point the comparison performed between these measures in Chapter 3.

Finally, we turn towards the f_{NR}^{min} fitness measure, and highlight the results of the considered approaches through it in Figure 6.24. In this case the selected CLIP approach is the $\mathrm{CLIP}_{\mathrm{NR}}^{min}$.



$\mathrm{CLIP}_{\mathrm{NR}}^{min}$	33.9 (0.1)	27.0 (2.9)	24.7 (4.6)	23.4 (7.0)	18.3 (7.0)	15.8 (3.5)	7.8 (5.9)	3.1 (3.8)	2.6 (3.7)	3.3 (2.7)	0.4 (1.0)	1.2 (2.7)
D-SOM$_{\text{MED}}$	2.0 (4.0)	6.7 (6.1)	8.4 (5.8)	0.0 (0.1)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
D-SOM$^*_{\text{MED}}$	28.9 (2.0)	20.8 (3.5)	20.2 (4.2)	1.1 (2.5)	0.4 (0.7)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
MRC	0.1 (0.2)	5.6 (3.8)	12.6 (3.7)	0.0 (0.2)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
K-MEANS	2.6 (4.3)	9.0 (5.1)	6.8 (4.5)	0.1 (0.4)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)

Figure 6.24: Comparison using the f_{NR}^{min} fitness measure;

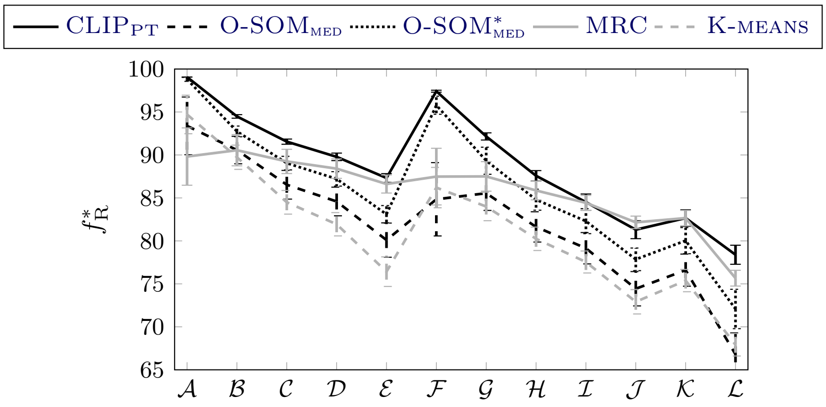
The $\mathrm{CLIP}_{\mathrm{NR}}^{min}$ approach clearly outperforms all the other approaches, through very large performance gains in certain cases. This is motivated by the fact that this approach is specifically tailored for improving the f_{NR}^{min} fitness measure.

The adaptive D-SOM approach may also be seen to improve this measure slightly, in comparison to the other approaches.

All of the remaining approaches perform poorly with respect to this measure, highlighting the fact that, in most cases, the proposed clustering results have at least one relation between a pair of alternatives that is completely opposed to the result found.

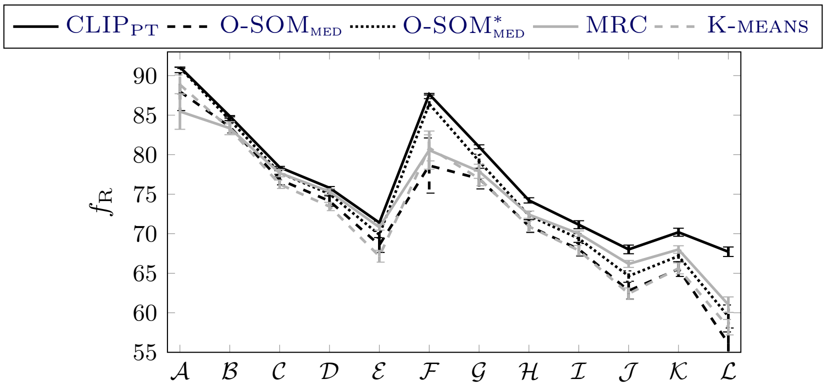
We continue with the relational clustering fitness measures, and start with the results highlighting f_{R}^*, in Figure 6.25, and those highlighting f_{R}, in Figure 6.26.

In both cases we have selected the O-SOM$_{\text{MED}}$ approach from the SOM adaptations, and $\mathrm{CLIP}_{\mathrm{PT}}$ from the CLIP variations.



CLIP$_{\mathrm{PT}}$	99.1 (0.0)	94.5 (0.2)	91.5 (0.3)	89.8 (0.4)	87.3 (0.5)	97.4 (0.1)	92.2 (0.4)	87.6 (0.6)	84.5 (0.9)	81.3 (1.0)	82.7 (1.0)	78.4 (1.1)
O-SOM$_{\mathrm{MED}}$	93.4 (3.4)	90.6 (1.6)	86.5 (1.7)	84.6 (1.7)	80.1 (2.0)	84.8 (4.3)	85.5 (2.0)	81.6 (1.8)	79.1 (1.8)	74.4 (2.0)	76.6 (1.9)	66.8 (2.5)
O-SOM$^*_{\mathrm{MED}}$	98.8 (0.2)	92.7 (0.6)	89.1 (0.8)	87.2 (0.8)	83.1 (1.0)	95.8 (1.0)	89.3 (1.6)	84.8 (1.2)	82.3 (1.5)	77.9 (1.3)	80.0 (1.5)	72.1 (2.3)
MRC	89.8 (3.3)	90.6 (1.8)	89.2 (1.4)	88.4 (1.1)	86.6 (1.1)	87.5 (3.3)	87.5 (1.7)	85.9 (1.1)	84.4 (0.9)	82.2 (0.7)	82.7 (0.9)	75.7 (0.9)
K-MEANS	94.7 (2.2)	89.7 (1.4)	84.5 (1.4)	81.9 (1.4)	76.4 (1.7)	86.2 (2.3)	84.0 (1.6)	80.2 (1.4)	77.5 (1.3)	72.9 (1.4)	75.4 (1.3)	67.9 (1.3)

Figure 6.25: Comparison using the f^*_{R} fitness measure;



CLIP$_{\mathrm{PT}}$	91.1 (0.0)	84.9 (0.1)	78.4 (0.1)	75.8 (0.2)	71.4 (0.2)	87.6 (0.1)	81.0 (0.3)	74.2 (0.3)	71.1 (0.5)	68.0 (0.5)	70.2 (0.5)	67.7 (0.6)
O-SOM$_{\mathrm{MED}}$	88.0 (2.4)	83.5 (0.8)	76.8 (0.7)	74.1 (0.7)	68.6 (0.9)	78.6 (3.5)	77.0 (1.3)	71.0 (0.8)	68.0 (0.9)	62.8 (1.1)	65.5 (0.9)	56.1 (1.4)
O-SOM$^*_{\mathrm{MED}}$	91.0 (0.1)	84.4 (0.2)	77.7 (0.3)	75.0 (0.3)	69.9 (0.4)	86.4 (0.7)	79.2 (0.9)	72.3 (0.5)	69.4 (0.6)	64.6 (0.7)	67.2 (0.7)	59.5 (1.5)
MRC	85.5 (2.3)	83.3 (0.8)	77.7 (0.5)	75.3 (0.4)	70.8 (0.4)	80.6 (2.4)	77.9 (0.9)	72.4 (0.5)	70.0 (0.4)	66.2 (0.4)	68.0 (0.5)	61.0 (1.0)
K-MEANS	88.9 (1.2)	83.3 (0.6)	76.3 (0.5)	73.4 (0.5)	67.2 (0.7)	80.9 (1.6)	77.0 (0.9)	70.9 (0.6)	67.9 (0.6)	62.4 (0.7)	65.6 (0.7)	58.2 (1.0)

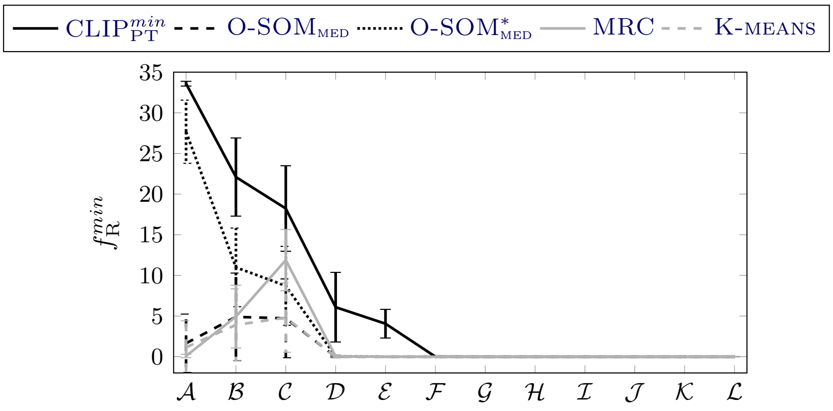
Figure 6.26: Comparison using the f_{R} fitness measure;

Considering the first fitness measure, the CLIP$_{\mathrm{PT}}$ approach can be seen to perform very well, while the MRC algorithm comes very close to it, on certain benchmarks. This is in part due to the fact that this approach is considered a relational clustering approach, while furthermore it uses directly the crisp degrees of credibility of the preferential relations.

The K-means and O-SOM$_{\text{MED}}$ approaches do not perform as well as the others, while the adaptive O-SOM$_{\text{MED}}$ variant comes closer to the first two.

In Figure 6.26, we observe that all of the considered approaches have close performances. The CLIP variant performs overall the best, being closely followed by the O-SOM$^*_{\text{MED}}$ approach. The MRC approach comes much closer to the CLIP variant on certain types of benchmarks, and especially on those containing many relations of incomparability. The O-SOM$_{\text{MED}}$ and K-means approaches can also be seen to perform rather well on certain types of benchmarks.

When considering the most restrictive relational fitness measure, in Figure 6.27, the chosen CLIP variant is CLIP$^{min}_{\text{PT}}$.



CLIP$^{min}_{\text{PT}}$	33.6 (0.3)	22.1 (4.8)	18.2 (5.3)	6.1 (4.3)	4.1 (1.8)	0.0 (0.1)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
O-SOM$_{\text{MED}}$	1.7 (3.6)	4.9 (5.4)	4.7 (4.8)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
O-SOM$^*_{\text{MED}}$	27.7 (3.9)	11.0 (4.8)	8.7 (4.9)	0.0 (0.1)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
MRC	0.1 (0.2)	4.9 (3.9)	11.9 (3.8)	0.0 (0.2)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
K-means	1.2 (3.3)	3.9 (4.4)	4.8 (4.3)	0.0 (0.1)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)

Figure 6.27: Comparison using the f^{min}_{R} fitness measure;

In this case, the CLIP approach clearly performs better than the rest. The MRC, O-SOM$_{\text{MED}}$ and K-means approaches very often propose results that are strongly opposed by at least one pair of alternatives. The adaptive variant of the O-SOM$_{\text{MED}}$ approach is able to raise this measure over the first types of benchmarks. In the case of the more difficult benchmarks, none of the considered approaches to raise the minimum degree of confidence of the supporting relations, which in mostly due to the abundance of incomparability relations between the alternatives.

Summary

In the above, we have presented a detailed analysis of the proposed MCDA clustering approaches. A wide range of benchmarks has been created for this purpose, and the approaches have been executed many times over each of them so that the results that have been found are representative for their performance using each of the illustrated fitness measures.

We may conclude that, in the case of the SOM adaptation, the disconnected SOMs provide clear improvements when considering the relations of indifference between the alternatives inside a cluster and the cluster representative, as opposed to approaches that use measures of similarity. Furthermore, the ordered SOMs additionally construct ordered clusters which are better fitted than the K-means approach. The chained SOMs do not provide such benefits as the previous two SOM adaptations. In addition, the approaches that select the map nodes from the existing alternatives are seen to perform overall better than those that construct the map nodes from the averages of the evaluations of the alternatives in each cluster, while the adaptive variants of all of the proposed approaches provide improvements over the results of the non-adaptive ones. In the case of the D-SOMs, these improvements are more significant.

In the case of the CLIP approach, the approximative approach from the first step may be clearly preferred to the exact one, due to the small execution time and the very close fitness results. The different second step variants have been shown to improve the initial results, considering each fitness measure, although the approaches that take into account the full credibility degrees of the preferential relations perform better than the rest even when the fitness measures account only for the crisp credibilities.

When comparing these approaches together, but also to other clustering approaches, we may conclude that the CLIP approach performs better in the large majority of cases than all the considered algorithms. This is mainly a result of the second step of the approach, which makes small changes to the initial result in order to improve any of the considered fitness measures.

Part III

Exploring large preferential datasets

Chapter 7

Measures for characterising sets of alternatives

Contents

Abstract

We focus on several measures that may be used in order to characterise sets of alternatives, with the purpose of using them in order to extract information from clustering results. Before defining these measures, we first need to select a particular preference model, which is based on a particular outranking relation. For each measure, we present several approaches to constructing them, which are then analysed and validated following several experiments over a set of artificially generated benchmarks. The approaches that perform the best and the context in which this happens will be highlighted in the end. The proposed measures will later in the thesis be used in order to analyse an illustrative case study.

7.1 The preference model

We will be focusing on the use of the bipolar valued outranking relation with large performance differences polarization, as defined in [Bisdorff 2012], in order to model the preferences of a DM. We motivate this choice only by our familiarity with this particular type of outranking relation, although the definitions of the measures that follow are independent of this model. Only a few of the approaches for constructing these measures will be strongly linked to this outranking relation, others may be applied to other similarly defined outranking relations, while the rest are independent from it.

In this section, we present the chosen outranking relation and then model it through linear constraints, which may be then used in different mathematical programs that construct the proposed measures in an exact way.

7.1.1 The bipolar valued outranking relation

The bipolar valued outranking relation follows the general definition of an outranking relation, *i.e.* combining a weighted majority of independent criteria which validate or not the fact that an alternative is at least as good as another, and the absence of a large counter performance of the first with respect to the second [Roy 1993].

This outranking relation uses the bipolar credibility degree r, where a negative value signifies that fact that we have a stronger support for the absence of the relation, while a strictly positive value signifies the contrary. A median value of 0 denotes a state of indetermination where the relation can neither be validated nor invalidated.

Constructing the global concordance degree

We consider a set of n alternatives, $X = \{x, y, z, ...\}$, which are defined on a family of criteria $F = \{1, 2, ..., m\}$. We state, without loss of generality, that all criteria are defined on ratio scales on the interval $[0, 1]$, with the evaluation of an alternative $x \in X$ being denoted through x_i. For each criterion $i \in F$ we consider an importance weight $w_i \in (0, 1)$, with $\sum\limits_{i \in F} w_i = 1$, and three discrimination thresholds $0 \leqslant q_i < p_i < v_i \leq 1$. These are, in the order in which they appear, the indifference, the preference and the large counter performance discrimination thresholds.

When considering the first part of the definition of the outranking relation, we need to determine beforehand if an alternative is at least as good as another on each criterion. We therefore define the *local concordance degree* on criterion $i \in F$,

$\forall x, y, \in X$, as:

$$C_i(x,y) = \begin{cases} 1 & \text{, if } x_i - y_i \geq -q_i; \\ -1 & \text{, if } x_i - y_i \leq -p_i; \\ 0 & \text{, otherwise.} \end{cases} \tag{7.1}$$

The *global concordance degree* then aggregates the local concordance degrees as:

$$C(x,y) = \sum_{i \in F} w_i \cdot C_i(x,y), \forall x, y \in X. \tag{7.2}$$

As the sum of the weights is 1, it is clear that $C(x,y) \in [-1,1], \forall x, y \in X$.

Taking into account large performance differences

In order to deal with large performance differences between the evaluations of the alternatives on any criterion, following from [Bisdorff 2012], we define the *local large counter performance degree* on criterion $i \in F$, $\forall x, y, \in X$, as:

$$LCP_i(x,y) = \begin{cases} 1 & \text{if } x_i - y_i \leq -v_i; \\ -1 & \text{otherwise.} \end{cases} \tag{7.3}$$

As in the definition of an outranking relation, we need only to determine if there exists a large counter performance on at least one criterion, therefore we additionally define the *global large counter performance degree* as:

$$LCP(x,y) = \max_{i \in F} LCP_i(x,y), \forall x, y \in X. \tag{7.4}$$

Using the global concordance and large performance difference degrees we may then construct the credibility of the outranking relation in the following way:

$$r(x \,\mathrm{S}\, y) = \begin{cases} C(x,y) & \text{, if } LCP(x,y) = LCP(y,x) = -1; \\ 1 & \text{, if } C(x,y) \geq \lambda \text{ and } LCP(x,y) = -1 \text{ and } LCP(y,x) = 1; \\ -1 & \text{, if } C(x,y) \leq -\lambda \text{ and } LCP(x,y) = 1 \text{ and } LCP(y,x) = -1; \\ 0 & \text{otherwise.} \end{cases} \tag{7.5}$$

The bipolar valued outranking relation places equal emphasis on the two concepts that are used in its construction. In the case where no large performance differences are encountered between the two alternatives, the credibility of the outranking relation is equal to the global concordance degree. If, however, the global concordance degree is greater than the bipolar cut level $\lambda \in [0,1)$, meaning that there is sufficient support with respect to the statement that the first alternative is at least as good as the second, and if additionally only the second alternative has a large counter performance with respect to the first, then the initial statement is reinforced. In this case we can clearly state that the first alternative outranks the second, therefore a polarization of the credibility degree is performed to the value of 1. A similar judgement is made in the case of polarizing the credibility degree to -1, when the global concordance degree is lower than the symmetric of the bipolar cut level, and furthermore, only the second alternative has a much lower performance than the first on at least one criterion. All the other cases, which contain conflicts between the global concordance degree and the large counter performance degrees, place the credibility degree of the outranking relation at a neutral level. This is also the case when we find large counter performances for both the considered alternatives.

7.1.2 Modelling the relation through linear constraints

In order to construct mathematical programs that model the considered bipolar valued outranking relation, we first decompose it into linear constraints. We do this so that we may be able to infer either the evaluations of the alternatives, the credibility degrees of the outranking relations between them, or the parameters of the outranking relation, provided that two of these three pieces of information are given.

We extend here the work from [Meyer et al. 2008] to modelling this particular type of outranking relation.

The three levels of the local concordance degree for any criterion $i \in F$ can be decomposed, for any pair of alternatives $x \neq y \in X$, using two binary variables $\alpha_i, \beta_i \in \{0, 1\}$ *s.t.* :

$$C_i(x, y) = \alpha_i(x, y) - \beta_i(x, y). \tag{7.6}$$

$C_i(x, y) = 1$ if $\alpha_i(x, y) = 1$ and $\beta_i(x, y) = 0$, $C_i(x, y) = -1$ if $\alpha_i(x, y) = 0$ and $\beta_i(x, y) = 1$ and $C_i(x, y) = 0$ if $\alpha_i(x, y)$ and $\beta_i(x, y)$ are both 0 or both 1.

The local concordance degree can then be rewritten through the following constraints:

$$\left\{ \begin{array}{rcccl} (-1 + q_i)(1 - \alpha_i(x, y)) & \leqslant & x_i - y_i + q_i & < & (1 + q_i)\alpha_i(x, y); \\ (-1 + p_i)\beta_i(x, y) & < & x_i - y_i + p_i & \leqslant & (1 + p_i)(1 - \beta_i(x, y)). \end{array} \right. \tag{7.7}$$

It can be observed, since $p_i > q_i \geqslant 0$, that $x_i - y_i + q_i \geq 0 \Rightarrow x_i - y_i + p_i > 0$ and $x_i - y_i + q_i \leq 0 \Rightarrow x_i - y_i + q_i < 0$. From this, $x_i - y_i + q_i \geq 0 \iff \alpha_i(x, y) = 1$ and $\beta_i(x, y) = 0$ and $x_i - y_i + p_i \leq 0 \iff \alpha_i(x, y) = 0,$ and $\beta_i(x, y) = 1.$ When

$-p_i < x_i - y_i < -q_i$ then $\alpha_i(x,y) = 0$ and $\beta_i(x,y) = 0$. $\alpha_i(x,y)$ and $\beta_i(x,y)$ cannot be simultaneously 1.

The global concordance degree then becomes:

$$C(x,y) = \sum_{i \in F} w_i(\alpha_i(x,y) - \beta_i(x,y)). \tag{7.8}$$

This can be rewritten as $C(x,y) = \sum_{i \in F} w_i^{'}(x,y)$, where:

$$w_i^{'}(x,y) = \begin{cases} w_i & \text{if } C_i(x,y) = 1; \\ -w_i & \text{if } C_i(x,y) = -1; \\ 0 & \text{otherwise.} \end{cases} \tag{7.9}$$

We therefore model the global concordance degree through the following linear constraints:

$$\begin{cases} -w_i \ \leqslant \ w_i^{'}(x,y) \ \leqslant \ w_i; \\ w_i + \alpha_i(x,y) - \beta_i(x,y) \ \leqslant \ w_i^{'}(x,y); \\ w_i^{'}(x,y) \ \leqslant \ -w_i + \alpha_i(x,y) - \beta_i(x,y) + 1; \\ -[\alpha_i(x,y) + \beta_i(x,y)] \ \leqslant \ w_i^{'}(x,y) \ \leqslant \ \alpha_i(x,y) + \beta_i(x,y). \end{cases} \tag{7.10}$$

Since $\alpha_i(x,y)$ and $\beta_i(x,y)$ cannot be simultaneously 1, it is clear that $C_i(x,y) = 1 \Rightarrow w_i^{'}(x,y) = w_i$, $C_i(x,y) = -1 \Rightarrow w_i^{'}(x,y) = -w_i$, and $C_i(x,y) = 0 \Rightarrow w_i^{'}(x,y) = 0$.

The local large counter performance degree takes only two values, 1 and -1, therefore it can be easily modelled through a single binary variable $\alpha_i^{'} \in \{0,1\}, \forall i \in F$:

$$LCP_i(x,y) = 2 \cdot \alpha_i^{'}(x,y) - 1. \tag{7.11}$$

The local large counter performance can then be rewritten through the following constraints:

$$(-1 + v_i)\alpha_i^{'}(x,y) \ < \ x_i - y_i + v_i \ \leqslant \ (1 + v_i)(1 - \alpha_i^{'}(x,y)). \tag{7.12}$$

In order to model the credibility degree of the outranking relation through linear constraints we first define three additional binary variables for each pair of alternatives $x \neq y \in X$: one for the global large counter performance degree, $\alpha^{''}$, and two for modelling if the global concordance degree is greater than the cut level λ, or lower than its symmetric with respect to the median level, through $\alpha^+, \alpha^- \in \{0,1\}$. We construct the following linear constraints:

$$\begin{cases} \alpha_i^{'}(x,y) \ \leqslant \ \alpha^{''}(x,y) \ \leqslant \ \sum\limits_{i \in F} \alpha_i^{'}(x,y); \\ -2(1-\alpha^+(x,y)) \ \leqslant \ \sum\limits_{i \in F} w_i^{'}(x,y) - \lambda \ < \ 2\alpha^+(x,y); \\ -2\alpha^-(x,y) \ < \ \sum\limits_{i \in F} w_i^{'}(x,y) + \lambda \ \leqslant \ 2(1-\alpha^-(x,y)). \end{cases} \tag{7.13}$$

Since $\alpha^{''}(x,y)$ can only take values in $\{0,1\}$, if there are no large counter performances against x then $\alpha^{''}(x,y)$ is forced to 0 by the right part of the first constraint. If there is at least one criterion on which x has a much lower performance than y, then the left part of the first constraint forces $\alpha^{''}(x,y)$ to 1.

The second constraint will force $\alpha^+(x,y)$ to 0 if the global concordance degree given by $\sum\limits_{i \in F} w_i^{'}(x,y)$ is lower than λ and place it at 1 otherwise. Similarly, the last constraint forces $\alpha^-(x,y)$ to 0 if the concordance degree is greater than $-\lambda$ and forces it to 1 otherwise.

Using these additional variables, the credibility degree of the outranking relation is modelled through:

$$\begin{cases} \sum\limits_{i \in F} w_i^{'}(x,y) - \alpha^{''}(x,y)(1+\lambda) & \leqslant & r(x \,\mathrm{S}\, y) & \leqslant & \sum\limits_{i \in F} w_i^{'}(x,y) + \alpha^{''}(y,x)(1+\lambda); \\ -2 + \alpha^+(x,y) + \alpha^{''}(x,y) & \leqslant & r(x \,\mathrm{S}\, y) & \leqslant & 2 - \alpha^+(x,y) + \alpha^{''}(x,y); \\ -2 + \alpha^-(x,y) + \alpha^{''}(y,x) & \leqslant & r(x \,\mathrm{S}\, y) & \leqslant & 2 - \alpha^-(x,y) + \alpha^{''}(y,x); \\ -1 + 2\alpha^{''}(y,x) - \alpha^-(x,y) & \leqslant & r(x \,\mathrm{S}\, y) & \leqslant & 1 - 2\alpha^{''}(x,y) + \alpha^+(x,y). \end{cases} \tag{7.14}$$

The first set of constraints set $r(x \,\mathrm{S}\, y)$ to the value of the weighted sum, *i.e.* the global concordance degree, in case that no large counter performance exists between the alternatives. In case such a counter performance exists, the constraints are relaxed. Therefore, if a large counter performance exists against x, *i.e.* $\alpha^{''}(x,y) = 1$, then $r(x \,\mathrm{S}\, y)$ will either be set to 0, or to -1, hence the term on the left is lowered to allow for these cases.

The second set of constraints sets $r(x \,\mathrm{S}\, y)$ to 0 when the concordance degree is greater than the cut level, *i.e.* $\alpha^+(x,y) = 1$, and a large counter performance exists against the first alternative, *i.e.* $\alpha^{''}(x,y) = 1$. In the same way, the third set of constraints sets $r(x \,\mathrm{S}\, y)$ to 0 when $\alpha^-(x,y) = 1$ and $\alpha^{''}(y,x) = 1$.

The last set of constraints deals with the polarisation of $r(x \,\mathrm{S}\, y)$ to 1 if $\alpha^{''}(y,x) = 1$ and $\alpha^-(x,y) = 0$, and to -1 if $\alpha^{''}(x,y) = 1$ and $\alpha^+(x,y) = 0$, which is exactly the behaviour of $r(x \,\mathrm{S}\, y)$ that we are looking for.

7.1.3 Formulating a mixed-integer linear program

Using the linear decomposition of the outranking relation presented before, we may formulate a mixed-integer linear program for it.

We consider the following potential inputs and/or outputs of this mathematical program:

1. the evaluations of the alternatives on the criteria: $x_i, \forall x \in X, i \in F$;
2. the credibility degrees of the outranking relation: $r(x \,\mathrm{S}\, y), \forall x, y \in X$;
3. the parameters of the outranking relation: $\lambda, q_i, p_i, v_i \forall i \in F$.

Using the mixed integer program, that is listed below, we may set as output and retrieve any of the above, given that the other two are known beforehand, and set as inputs. For example, given the evaluations of the alternatives and the credibility degrees of the outranking relation, we can extract the parameters that were used in its construction. In the mathematical program below we consider this case. We denote with g the evaluations of the alternatives on the criteria, and with s the credibility degrees of the outranking relation between any pair of alternatives.

MIP-$\tilde{\mathrm{S}}_{\mathrm{LDP}}$:

Input:		
	$g(x,i) \in [0,1]$	$\forall i \in F, \forall x \in X$
	$s(x,y) \in [-1,1]$	$\forall x \neq y \in X$
Output:		
	$\lambda \in [0,1]$	
	$w_i, q_i, p_i, v_i \in [0,1]$	$\forall i \in F$
Variables:		
	$\alpha_i(x,y), \beta_i(x,y), \alpha^{'}_i(x,y) \in \{0,1\}$	$\forall i \in F, \forall x \neq y \in X$
	$\alpha^{''}(x,y), \alpha^{+}(x,y), \alpha^{-}(x,y) \in \{0,1\}$	$\forall x \neq y \in X$
	$w^{'}_i(x,y) \in [-1,1]$	$\forall i \in F, \forall x \neq y \in X$
Parameters:		
	$\gamma_1 \in]0,1[$	
Constraints:		
s.t.	$\sum_{i=1}^{n} w_i = 1$	
	$w_i \geqslant \gamma_1$	$\forall i \in F$
	$(-1+q_i)(1-\alpha_i(x,y)) \leqslant g(x,i) - g(y,i) + q_i$	$\forall i \in F, \forall x \neq y \in X$
	$g(x,i) - g(y,i) + q_i + \gamma_1 \leqslant (1+q_i)\alpha_i(x,y)$	$\forall i \in F, \forall x \neq y \in X$
	$(-1+p_i)\beta_i(x,y) + \gamma_1 \leqslant g(x,i) - g(y,i) + p_i$	$\forall i \in F, \forall x \neq y \in X$
	$g(x,i) - g(y,i) + p_i \leqslant (1+p_i)(1-\beta_i(x,y))$	$\forall i \in F, \forall x \neq y \in X$
	$(-1+v_i)\alpha^{'}_i(x,y) + \gamma_1 \leqslant g(x,i) - g(y,i) + v_i$	$\forall i \in F, \forall x \neq y \in X$
	$g(x,i) - g(y,i) + v_i \leqslant (1+v_i)(1-\alpha^{'}_i(x,y))$	$\forall i \in F, \forall x \neq y \in X$
	$-w_i \leqslant w^{'}_i(x,y)$	$\forall i \in F, \forall x \neq y \in X$
	$w^{'}_i(x,y) \leqslant w_i$	$\forall i \in F, \forall x \neq y \in X$
	$w_i + \alpha_i(x,y) - \beta_i(x,y) - 1 \leqslant w^{'}_i(x,y)$	$\forall i \in F, \forall x \neq y \in X$
	$w^{'}_i(x,y) \leqslant -w_i + \alpha_i(x,y) - \beta_i(x,y) + 1$	$\forall i \in F, \forall x \neq y \in X$
	$-(\alpha_i(x,y) + \beta_i(x,y)) \leqslant w^{'}_i(x,y)$	$\forall i \in F, \forall x \neq y \in X$
	$w^{'}_i(x,y) \leqslant \alpha_i(x,y) + \beta_i(x,y)$	$\forall i \in F, \forall x \neq y \in X$
	$\alpha^{'}_i(x,y) \leqslant \alpha^{''}(x,y)$	$\forall i \in F, \forall x \neq y \in X$
	$\alpha^{''}(x,y) \leqslant \sum_{i \in F} \alpha^{'}_i(x,y)$	$\forall x \neq y \in X$

$$\begin{array}{ll}
-2(1-\alpha^{+}(x,y))+\lambda \leqslant \sum_{i\in F} w_i^{'}(x,y) & \forall x \neq y \in X \\
\sum_{i\in F} w_i^{'}(x,y)+\gamma_1 \leqslant 2\alpha^{+}(x,y)+\lambda & \forall x \neq y \in X \\
-2\alpha^{-}(x,y)-\lambda+\gamma_1 \leqslant \sum_{i\in F} w_i^{'}(x,y) & \forall x \neq y \in X \\
\sum_{i\in F} w_i^{'}(x,y) \leqslant 2(1-\alpha^{+}(x,y))-\lambda & \forall x \neq y \in X \\
s(x,y) \leqslant \sum_{i\in F} w_i^{'}(x,y)+\alpha^{''}(y,x)(1+\lambda) & \forall x \neq y \in X \\
\sum_{i\in F} w_i^{'}(x,y)-\alpha^{''}(x,y)(1+\lambda) \leqslant s(x,y) & \forall x \neq y \in X \\
s(x,y) \leqslant 2-\alpha^{+}(x,y)+\alpha^{''}(x,y) & \forall x \neq y \in X \\
-2+\alpha^{+}(x,y)+\alpha^{''}(x,y) \leqslant s(x,y) & \forall x \neq y \in X \\
s(x,y) \leqslant 2-\alpha^{-}(x,y)+\alpha^{''}(y,x) & \forall x \neq y \in X \\
-2+\alpha^{-}(x,y)+\alpha^{''}(y,x) \leqslant s(x,y) & \forall x \neq y \in X \\
s(x,y) \leqslant 1-2\alpha^{''}(x,y)+\alpha^{+}(x,y) & \forall x \neq y \in X \\
-1+2\alpha^{''}(y,x)-\alpha^{-}(x,y) \leqslant s(x,y) & \forall x \neq y \in X \\
q_i \geqslant \gamma_1 & \forall i \in F \\
p_i \geqslant q_i+\gamma_1 & \forall i \in F \\
v_i \geqslant p_i+\gamma_1 & \forall i \in F
\end{array}$$

The inputs will be considered as parameters while the outputs as variables. Furthermore, the inputs may not be complete, while the output may not be unique. It may also be the case that the set of inputs define a system which cannot be modelled through the bipolar valued outranking relation, that we have defined before, and so no solution may be found.

The γ_1 parameter has been added in order to bring the weak inequations to strict ones. It should be chosen very small.

We will denote the body of the mixed integer linear program as MIP-$\tilde{\text{S}}_{\text{LDP}}$, in order to simplify the definition of other mathematical programs that follow.

7.2 Descriptive measures for sets of alternatives

Having clearly defined the evaluation model used for expressing the preferences of a DM, we proceed to presenting several measures that may be used to characterise sets of alternatives and propose several approaches for constructing them.

Clustering is used to group large numbers of objects, or in this case alternatives, when little information on this structure is available, or when this information is costly to extract. Hence, the measures that we will propose may be used to describe clusters of alternatives.

We will define in what follows several types of measures, called profiles, which can be used as a summary of the information found in any considered dataset. Following the way in which they will be defined, these are: central profiles, bounding profiles and separating profiles.

The central profiles hold strong similarities to those proposed in [Perny 1998] and [Belacel 2000], where the problem of nominal sorting is addressed. The separating profiles can also be associated with the profiles used in the problem of sorting into ordered categories. For instance, the ELECTRE TRI [Yu 1992, Roy and Bouyssou 1993] method is a popular choice for solving this problem.

7.2.1 Central profiles

The topic of finding a representative for each cluster has already been discussed in Chapter 4, where one alternative from a cluster is selected based on a fitness function. Considering a cluster C, we quickly remind the three fitness functions that can be used to characterise the potential representative alternative r_C:

$$f^*(r_C, C) = \frac{1}{|C|} \cdot \sum_{x \in C} r^*(r_C \,\mathrm{I}\, x); \tag{7.15}$$

$$f(r_C, C) = \frac{1}{|C|} \cdot \sum_{x \in C} r(r_C \,\mathrm{I}\, x); \tag{7.16}$$

$$f^{min}(r_C, C) = \min_{x \in C} r(r_C \,\mathrm{I}\, x). \tag{7.17}$$

These functions measure in different ways the degree with which the representative is indifferent to the alternatives inside the set C.

We also call such a representative a **central profile** for a set of alternatives. Ideally, a central profile should be selected, or constructed, so that it maximizes one of the fitness functions presented above.

We also provide an intuitive illustration of this profile in Figure 7.1. The outranking relations are illustrated through lines with arrows, and are present between both the representative and the alternatives inside the cluster C.

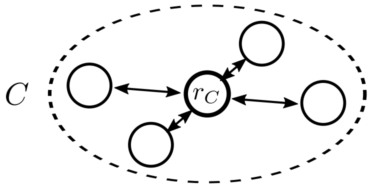


Figure 7.1: Central profile;

Below we present several ways in which this profile can be found.

Selecting the fittest alternative

Choosing an existing alternative from C, to play the role of a central profile, is done as follows:

$$r_C = \arg\max_{x \in C} f(x, C). \tag{7.18}$$

We have depicted the formula for selecting a central profile using only the second fitness function, however, we may replace it with any of the three we have presented before.

Building from average evaluations

Constructing a fictive alternative as a central profile may potentially lead to better results based on the selected fitness function. For instance in the K-MEANS algorithm, the clusters are built around centroids, which are constructed from the averages of the evaluations, on each attribute, of the objects inside a particular cluster. We could apply this same approach to building a central profile for a set of alternatives:

$$r_{Ci} = \frac{1}{|C|} \sum_{x \in C} x_i, \forall i \in F. \tag{7.19}$$

This approach is only suited when the attributes or criteria are defined on numerical scales, however we may use a median operator when confronted with ordinal scales and simply selecting the value which appears with the highest frequency when dealing with nominal scales.

Using a mathematical program

Due to the differences between the classical measures of similarity and the one of indifference, which we have illustrated in the first part of the thesis, building a central profile from the means of the evaluations of the alternatives inside the set may not be well suited. For this reason we propose to use the mathematical program that has been defined for the bipolar-valued outranking relation with large differences polarization, in order to build such a profile.

We will consider as input the evaluations of the alternatives inside C, and the parameters of the outranking relation. The credibility degrees of the outranking relation between any pair of alternatives will be constructed by the mathematical program implicitly, therefore they do not need to be given. The output of this program will be the evaluations of the representative of C, on each criterion.

In order to model the first fitness function from Equation (7.15) we need to model the r^* crisp credibility degree of the indifference relation between the representative alternative and each alternative in the set C. As r^* can take values in $\{-1, 0, 1\}$, we define two binary variables $\alpha_{\mathrm{I}}^{+}, \alpha_{\mathrm{I}}^{-}$ for each alternative in $x \in C$ in order to model this:

$$r^*(r_C \,\mathrm{I}\, x) = \alpha_{\mathrm{I}}^{+}(x) - \alpha_{\mathrm{I}}^{-}(x). \tag{7.20}$$

The first variable needs to be 1 when $r(r_C \,\mathrm{I}\, x) \geqslant \lambda$, and 0 otherwise, while the second variable needs to be 1 when $r(r_C \,\mathrm{I}\, x) \leqslant -\lambda$, and 0 otherwise. To do this, we will first need to determine the crisp credibility degrees of the outranking relation between the profile and the alternatives, and vice versa. This is done in the same way as determining when the global concordance degree is above the cut level, or below its symmetric with respect to the median level. We define the binary variables $\alpha^+_{\mathrm{S}\,1}$ and $\alpha^-_{\mathrm{S}\,1}$ for each alternative, which tell us if the outranking relation between the profile and an alternatives in C is above or equal to λ ($\alpha^+_{\mathrm{S}\,1}$) or below or equal to $-\lambda$ ($\alpha^-_{\mathrm{S}\,1}$). Similarly we define $\alpha^+_{\mathrm{S}\,2}$ and $\alpha^-_{\mathrm{S}\,2}$ when considering the outranking relation between an alternative in C and r_C. We model this through the following linear constraints:

$$\begin{cases} -2(1-\alpha^+_{\mathrm{S}\,1}(x)) \;\leqslant\; r(r_C \,\mathrm{S}\, x) - \lambda \;<\; 2\alpha^+_{\mathrm{S}\,1}(x); \\ -2\alpha^-_{\mathrm{S}\,1}(x) \;<\; r(r_C \,\mathrm{S}\, x) + \lambda \;\leqslant\; 2(1-\alpha^-_{\mathrm{S}\,1}(x)); \\ -2(1-\alpha^+_{\mathrm{S}\,2}(x)) \;\leqslant\; r(x \,\mathrm{S}\, r_C) - \lambda \;<\; 2\alpha^+_{\mathrm{S}\,2}(x); \\ -2\alpha^-_{\mathrm{S}\,2}(x) \;<\; r(x \,\mathrm{S}\, r_C) + \lambda \;\leqslant\; 2(1-\alpha^-_{\mathrm{S}\,2}(x)). \end{cases} \tag{7.21}$$

At this point, we model the linear constraints used for defining α^+_{I} and α^-_{I} below:

$$\begin{cases} 1-\alpha^+_{\mathrm{I}}(x) \;\leqslant\; \alpha^+_{\mathrm{S}\,1}(x) + \alpha^+_{\mathrm{S}\,2}(x) - 2 \;<\; \alpha^+_{\mathrm{I}}(x), \\ \alpha^-_{\mathrm{S}\,1}(x) \;\leqslant\; \alpha^-_{\mathrm{I}}(x), \\ \alpha^-_{\mathrm{S}\,2}(x) \;\leqslant\; \alpha^-_{\mathrm{I}}(x), \\ \alpha^-_{\mathrm{I}}(x) \;\leqslant\; \alpha^-_{\mathrm{S}\,1}(x) + \alpha^-_{\mathrm{S}\,2}(x). \end{cases} \tag{7.22}$$

It is clear that, when $r(r_C \,\mathrm{S}\, x) \geqslant \lambda$ and $r(x \,\mathrm{S}\, r_C) \geqslant \lambda$, hence $\alpha^+_{\mathrm{S}\,1}(x) = \alpha^+_{\mathrm{S}\,2}(x) = 1$, then $\alpha^+_{\mathrm{I}}(x) = 1$, while it is 0 otherwise. This result is given by the first set of inequations from above.

When either $\alpha^-_{\mathrm{S}\,1}(x) = 1$, *i.e.* $r(r_C \,\mathrm{S}\, x) \leqslant -\lambda$, or $\alpha^-_{\mathrm{S}\,2}(x) = 1$, *i.e.* $r(x \,\mathrm{S}\, r_C) \leqslant -\lambda$, then $\alpha^-_{\mathrm{I}}(x) = 1$ and 0 otherwise. This is given by the last three sets of inequations, which model a max operator between $\alpha^-_{\mathrm{S}\,1}(x)$ and $\alpha^-_{\mathrm{S}\,2}(x)$.

The mathematical program for constructing a central profile based on the first fitness function from Equation (7.15) is as follows:

MIP-CPR*:

$\vdots$

MIP-$\tilde{\mathrm{S}}_{\mathrm{LDP}}$

$\vdots$

Input:

	$g(x,i) \in [0,1]$	$\forall i \in F, \forall x \in C$
	$\lambda \in [0,1]$	
	$w_i, q_i, p_i, v_i \in [0,1]$	$\forall i \in F$

Output:

	$g(r_C,i) \in [0,1]$	$\forall i \in F$

Objective function:

max $\quad \sum\limits_{x \in C} \big(\alpha^{*+}(x) - \alpha^{*-}(x)\big)$

Variables:

$$\begin{array}{lll}
 & \alpha_{\mathrm{S}\,1}^{+}(x) \in \{0,1\} & \forall x \in C \\
 & \alpha_{\mathrm{S}\,1}^{-}(x) \in \{0,1\} & \forall x \in C \\
 & \alpha_{\mathrm{S}\,2}^{+}(x) \in \{0,1\} & \forall x \in C \\
 & \alpha_{\mathrm{S}\,2}^{-}(x) \in \{0,1\} & \forall x \in C \\
 & \alpha_{\mathrm{I}}^{+}(x) \in \{0,1\} & \forall x \in C \\
 & \alpha_{\mathrm{I}}^{-}(x) \in \{0,1\} & \forall x \in C
\end{array}$$

Constraints:

$$\begin{array}{lll}
\text{s.t.} & -2(1-\alpha_{\mathrm{S}\,1}^{+}(x)) + \lambda \leqslant s(r_C, x) & \forall x \in C \\
 & s(r_C, x) + \gamma_1 \leqslant 2\alpha_{\mathrm{S}\,1}^{+}(x) + \lambda & \forall x \in C \\
 & -2\alpha_{\mathrm{S}\,1}^{-}(x) - \lambda + \gamma_1 \leqslant s(r_C, x) & \forall x \in C \\
 & s(r_C, x) \leqslant 2(1-\alpha_{\mathrm{S}\,1}^{-}(x)) - \lambda & \forall x \in C \\
 & -2(1-\alpha_{\mathrm{S}\,2}^{+}(x)) + \lambda \leqslant s(x, r_C) & \forall x \in C \\
 & s(x, r_C) + \gamma_1 \leqslant 2\alpha_{\mathrm{S}\,2}^{+}(x) + \lambda & \forall x \in C \\
 & -2\alpha_{\mathrm{S}\,2}^{-}(x) - \lambda + \gamma_1 \leqslant s(x, r_C) & \forall x \in C \\
 & s(x, r_C) \leqslant 2(1-\alpha_{\mathrm{S}\,2}^{-}(x)) - \lambda & \forall x \in C \\
 & 1 - \alpha_{\mathrm{I}}^{+}(x) \leqslant \alpha_{\mathrm{S}\,1}^{+}(x) + \alpha_{\mathrm{S}\,2}^{+}(x) - 2 & \forall x \in C \\
 & \alpha_{\mathrm{S}\,1}^{+}(x) + \alpha_{\mathrm{S}\,2}^{+}(x) - 2 + \gamma_1 \leqslant \alpha_{\mathrm{I}}^{+}(x) & \forall x \in C \\
 & \alpha_{\mathrm{S}\,1}^{-}(x) \leqslant \alpha_{\mathrm{I}}^{-}(x) & \forall x \in C \\
 & \alpha_{\mathrm{S}\,2}^{-}(x) \leqslant \alpha_{\mathrm{I}}^{-}(x) & \forall x \in C \\
 & \alpha_{\mathrm{I}}^{-}(x) \leqslant \alpha_{\mathrm{S}\,1}^{-}(r_C) + \alpha_{\mathrm{S}\,2}^{-}(x) & \forall x \in C
\end{array}$$

In order to model the fitness function from Equation (7.16) we need to determine the minimum credibility degree out of the two outranking relations, $r_C \,\mathrm{S}\, x$ and $x \,\mathrm{S}\, r_C$, $\forall x \in C$. In order to do this, we define a binary variable for each alternative in C to act as an indicator that tells us which of the two outranking relations has a lower confidence degree. We denote this variable with δ and model the credibility degree of the I relation between the representative and any alternative in C through:

$$\begin{cases}
\delta(x) - 1 \;\leqslant\; r(r_C \,\mathrm{S}\, x) - r(x \,\mathrm{S}\, r_C) \;<\; \delta(x), \\
r(r_C \,\mathrm{S}\, x) - 2\delta(x) \;\leqslant\; r(r_C \,\mathrm{I}\, x) \;\leqslant\; r(r_C \,\mathrm{S}\, x), \\
r(x \,\mathrm{S}\, r_C) - 2(1-\delta(x)) \;\leqslant\; r(r_C \,\mathrm{I}\, x) \;\leqslant\; r(x \,\mathrm{S}\, r_C).
\end{cases} \tag{7.23}$$

The first set of inequations places $\delta(x)$ at 1 if $r(x \,\mathrm{S}\, r_C) \leqslant r(r_C \,\mathrm{S}\, x)$ and at 0 otherwise. The following sets of inequations set the credibility degree of $r_C \,\mathrm{I}\, x$ to the minimum between $r(r_C \,\mathrm{S}\, x)$ and $r(x \,\mathrm{S}\, r_C)$.

The mathematical program for constructing a central profile based on the second fitness function from Equation (7.16) is defined below:

MIP-CPR:

⋮

MIP-$\tilde{\mathrm{S}}_{\mathrm{LDP}}$

⋮

Input:

$$\begin{array}{ll}
g(x,i) \in [0,1] & \forall i \in F, \forall x \in C \\
\lambda \in [0,1] & \\
w_i, q_i, p_i, v_i \in [0,1] & \forall i \in F
\end{array}$$

Output:

$$\begin{array}{ll}
g(r_C, i) \in [0,1] & \forall i \in F
\end{array}$$

Objective function:

$$\max \quad \sum_{x \in C} ri(x)$$

Variables:

$$\begin{array}{lll}
 & \delta(x) \in \{0,1\} & \forall x \in C \\
 & ri(x) \in [0,1] & \forall x \in C
\end{array}$$

Constraints:

$$\begin{array}{lll}
\text{s.t.} & \delta(x) - 1 \leqslant s(r_C, x) - s(x, r_C) & \forall x \in C \\
 & s(r_C, x) - s(x, r_C) + \gamma_1 \leqslant \delta(x) & \forall x \in C \\
 & ri(x) \leqslant s(r_C, x) & \forall x \in C \\
 & ri(x) \leqslant s(x, r_C) & \forall x \in C \\
 & s(r_C, x) - 2\delta(x) \leqslant ri(x) & \forall x \in C \\
 & s(x, r_C) - 2(1 - \delta(x)) \leqslant ri(x) & \forall x \in C
\end{array}$$

Finally, in order to model the fitness function from Equation (7.17) we need to determine the minimum between the credibilities of the I relation between the representative r_C and every alternative in C. We again use binary indicators to determine if an alternative $x \in C$ has the minimum degree of the indifference relation to r_C with respect to any other $y \in C$. We denote this indicator with $\delta^{'}$.

$$\begin{cases} \delta^{'}(x,y) - 1 \ < \ r(r_C \,\mathrm{I}\, x) - r(r_C \,\mathrm{I}\, y) \ \leqslant \ \delta^{'}(x,y), \\ r(r_C \,\mathrm{I}\, x) - 2 \sum\limits_{y \in C} \delta^{'}(x,y) \ \leqslant \ \min\limits_{y \in C} r(r_C \,\mathrm{I}\, y) \ \leqslant r(r_C \,\mathrm{I}\, x). \end{cases} \tag{7.24}$$

The first set of inequations sets the indicator $\delta^{'}$ between x and y to 1 if the credibility degree of the indifference relation between the representative and x is greater than the one between the representative and the other alternative y. Therefore, if x is the alternative with the minimum degree of the relation of indifference to r_C then all its indicators will be 0. This is then used in the second set of inequations in order to model $\min\limits_{y \in C} r(r_C \,\mathrm{S}\, y)$.

The mathematical program for this last fitness function is:

MIP-CPR^{min}:

⋮

MIP-CPR

⋮

Input:

$$\begin{array}{lll}
 & g(x,i) \in [0,1] & \forall i \in F, \forall x \in C \\
 & \lambda \in [0,1] & \\
 & w_i, q_i, p_i, v_i \in [0,1] & \forall i \in F
\end{array}$$

Output:

$$\begin{array}{lll}
 & g(r_C, i) \in [0,1] & \forall i \in F
\end{array}$$

Objective function:

$$\begin{array}{ll}
\max & rimin
\end{array}$$

Variables:

$$\begin{array}{lll}
 & \delta^{'}(x,y) \in \{0,1\} & \forall x \neq y \in C \\
 & rimin \in [0,1] &
\end{array}$$

Constraints:

$$\begin{array}{lll}
\text{s.t.} & \delta^{'}(x,y) - 1 + \gamma_1 \leqslant ri(x) - ri(y) & \forall x, y \in C \\
 & ri(x) - ri(y) \leqslant \delta^{'}(x,y) & \forall x, y \in C \\
 & rimin \leqslant ri(x) & \forall x \in C \\
 & ri(x) - 2 \sum\limits_{y \in C} \delta^{'}(x,y) \leqslant rimin & \forall x \in C
\end{array}$$

Using a meta-heuristic approach

Due to the number of variables that need to be defined in the mathematical programs that we have presented, using such an approach might quickly become impractical when the number of alternatives and criteria increase. When constraints on the time it takes for a central profile to be constructed exist, we may resort to finding a solution that is not the optimal one with respect to the fitness functions defined for a central profile, but which comes close to it.

For this reason we present a meta-heuristic approach for constructing a central profile. Considering a set of alternatives C and its representative r_C, which will contain evaluations on all the m criteria in F, we define $2 \cdot m$ potential operations we can make on r_C, by either increasing or decreasing each evaluation on the m criteria.

We characterise the operation of increasing the evaluation of r_C on criterion i, considering each of the three fitness functions that we try to maximize, through the following heuristic functions:

$$\begin{aligned} h^*(r_C, i) := \frac{1}{|C|} \cdot \Big(& \big|\{x \in C \colon\ x_i - r_{Ci} > q_i \text{ and } r^*(r_C \,\mathrm{I}\, x) = 0\}\big| \\ & - \big|\{x \in C \colon\ x_i - r_{Ci} < -q_i \text{ and } r^*(r_C \,\mathrm{I}\, x) = 0\}\big|\Big) \quad (7.25) \\ h(r_C, i) := \frac{1}{|C|} \cdot \Big(& \big|\{x \in C \colon\ x_i - r_{Ci} > q_i\}\big| \\ & - \big|\{x \in C \colon\ x_i - r_{Ci} < -q_i\}\big|\Big) \quad (7.26) \\ h^{min}(r_C, i) := \frac{1}{|C|} \cdot \Big(& \big|\{x \in C \colon\ x_i - r_{Ci} > q_i \text{ and } r(r_C \,\mathrm{I}\, x) \leqslant r(r_C \,\mathrm{I}\, y), \forall y \in C\}\big| \\ & - \big|\{x \in C \colon\ x_i - r_{Ci} < -q_i \text{ and } r(r_C \,\mathrm{I}\, x) \leqslant r(r_C \,\mathrm{I}\, y), \forall y \in C\}\big|\Big). \quad (7.27) \end{aligned}$$

These heuristics can be thought of as voting procedures, where each alternative in C votes in favour of increasing the evaluation of r_C on criterion i, in disfavour, or sustains from voting. This is reflected by the two terms in each of the above equations. The first term counts the number of alternatives which have an evaluation higher than that of r_C by more than the q_i threshold. This means that those alternatives are not considered indifferent to r_C on criterion i. Moreover, those alternatives are preferred to it, therefore, from their perspective, the evaluation of r_C should be increased. The second term counts in a similar way the alternatives from whose perspective the evaluation of r_C on criterion i should be decreased. The alternatives which are already indifferent with r_C on criterion i do not require an increase or decrease in the evaluation of r_C.

In the first heuristic function only the alternatives which are not overall indifferent to the profile are considered. This is due to the fact that we wish to maximize the number of alternatives which are indifferent to the central profile with respect to the crisp credibility degrees. In the second heuristic function all the alternatives in

C are considered, which is due to the fact that we look to increase the average degree of indifference between the central profile and all the alternatives in C. Finally, the last heuristic function considers only the alternatives that hold the minimum degree of credibility with respect to the relations of indifference between the central profile and them, as these credibility degrees give the value of the last fitness function of a central profile.

The heuristic functions of decreasing the evaluation of r_C on any criterion are retrieved by reversing the sign of the previously defined heuristic functions.

Using these fitness functions we may then apply any single-solution meta-heuristic [Talbi 2009], such as those described in Chapter 5.

For any of these algorithms, we may select one out of several initialization strategies, such as:

- random generation: r_C is given random evaluations on all criteria;
- random selection: r_C is selected as an alternative from C randomly;
- fittest selection: r_C is selected as the alternative from C having the best fitness with respect to a selected fitness function from Equations (7.15),(7.16),(7.17);
- generation from mean evaluations: r_C is constructed as in Equation (7.19).

The amount with which the evaluation of r_C is changed, either positively or negatively on a selected criterion i, will also be set to the smallest amount that will have an impact on the way in which r_C compares to the alternatives in C. This is better highlighted in Figure 7.2.

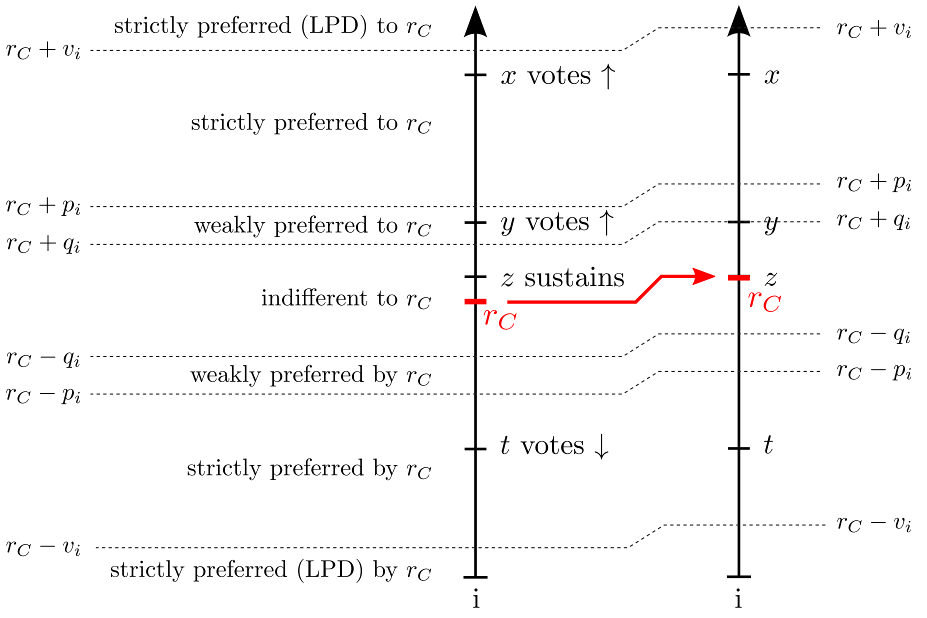


Figure 7.2: Detailing the heuristic for changing r_C for a set C of 4 alternatives;

In this example, we consider a set $C = \{x, y, z, t\}$ of 4 alternatives and their central profile r_C. We highlight how the evaluation of the central profile on a particular criterion i should be increased. We notice that, as x is strictly preferred to r_C on criterion i, we need to increase the evaluation of r_C in order to make x appear indifferent to r_C. This is also the case for y, which is weakly preferred to r_C. However, the evaluation of z is already indifferent to that of r_C, therefore from this perspective we do not need to increase or decrease the evaluation of r_C. Finally, t has an evaluation that is strictly preferred by that of r_C, therefore we would need to decrease the evaluation of r_C in order to make t indifferent on this criterion to r_C.

As we have two alternatives in favour of increasing the evaluation of r_C, and only one against, we have a positive fitness measure for performing this operation. However, we would only add to the evaluation of r_C the smallest amount which changes at least one of the comparisons between it and the alternatives in C. The first alternative, x, would require r_C to be increased by an amount that brings the first dotted line below the evaluation of x on i just above it. This amount is $x_i - r_{Ci} - p_i + \varepsilon$, where $\varepsilon > 0$ and $\varepsilon \ll 1$, an in this case x would become only weakly preferred to r_{Ci}. However, this amount can be seen to be larger than the amounts that would be required in order for the other alternatives to compare differently to r_C, therefore we will not increase r_{Ci} by this amount. The use of ε is necessary following the definition of the outranking relation. The smallest amount that would impact the way in which at least one alternative compares to r_C on i is equal to $y_i - r_{Ci} - q_i$, which would make y become indifferent to r_C on criterion i, while all the other alternatives will remain in the same state as before. Therefore the increase of r_{Ci} would be this amount.

Having a positive fitness value for this operation does not imply that we would perform it straight away. All the operations of both increasing and decreasing the evaluations of r_C on all criteria, characterised through the described heuristic measure, may be used in a meta-heuristic approach. By making the smallest change to the evaluation of the profile, which will affect at least one of the alternatives in C in the way they compare to the profile, ensures that the meta-heuristic will be able to explore the search space from one solution to the closest of its neighbouring solutions, and not perform large changes to the central profile, in a way similar to a random search.

7.2.2 Bounding profiles

Depending on the level of indifference between the alternatives inside a set, a central profile may be able to represent them with a higher or a lower degree. If a central profile is not sufficient in order to characterise the alternatives in this set, it may be better suited to find or construct two profiles which bound between them the alternatives inside the set. We call these the upper and lower **bounding profiles** and denote them with r_C^+ and r_C^- for any set C.

We define the following fitness functions, in a way similar to those of a central profile, in order to characterise an upper bounding profile:

$$f^{*+}(r_C^+, C) = \frac{1}{|C| \cdot (2 \cdot |C| + 1)} \cdot \Big(2 \cdot |C| \cdot \sum_{x \in C} r^*(r_C^+ \,\mathrm{S}\, x) + \sum_{x \in C} r^*(x \,\mathrm{S}\, r_C^+)\Big); \tag{7.28}$$

$$f^{+}(r_C^+, C) = \frac{1}{|C| \cdot (2 \cdot |C| + 1)} \cdot \Big(2 \cdot |C| \cdot \sum_{x \in C} r(r_C^+ \,\mathrm{S}\, x) + \sum_{x \in C} r(x \,\mathrm{S}\, r_C^+)\Big); \tag{7.29}$$

$$f^{min+}(r_C^+, C) = \frac{1}{(2 \cdot |C| + 1)} \Big(2 \cdot |C| \cdot \min_{x \in C} r(r_C^+ \,\mathrm{S}\, x) + \min_{x \in C} r(x \,\mathrm{S}\, r_C^+)\Big); \tag{7.30}$$

The first term from each of the above equations models the fact that the upper profile needs to be at least as good as all the alternatives inside C. Each type of function considers increasingly stronger statements related to this fact. However, considering only this aspect, we may easily construct the upper profile by setting the maximum evaluations on all criteria. This is not desired, as we want the upper and lower profiles to bound the alternatives in C as tightly as possible. For this reason we have added the second term, through which we try to have the alternatives in C be additionally at least as good as the upper profile, therefore making them indifferent to it.

The first term has been weighted so that improvements with respect to the second term are used only to discriminate profiles that have the same performance with respect to the first. In other words, the fitness functions are multi-objective, where the first objective completely dominates the second.

We provide an intuitive illustration of these profiles in Figure 7.3. The outranking relations are illustrated through lines with arrows. The thicker ones are the most important relations in the definition of each profile, while the thinner ones have a less important role.

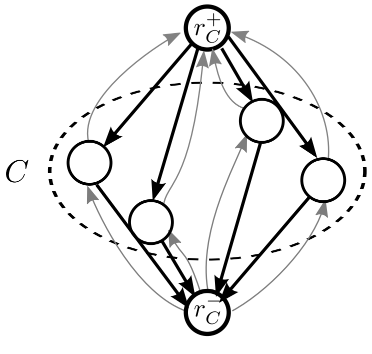


Figure 7.3: Bounding profiles;

It easily follows that for compact sets of alternatives, which contain alternatives that are mostly indifferent with each other, the upper profile may become identical to the central profile, which is consistent with the definition of the bounding profiles as extensions of the central one.

We define the following fitness functions for characterising a lower profile:

$$f^{*-}(r_C^-, C) = \frac{1}{|C| \cdot (2 \cdot |C| + 1)} \cdot \Big(\sum_{x \in C} r^*(r_C^- \, \mathrm{S} \, x) + 2 \cdot |C| \cdot \sum_{x \in C} r^*(x \, \mathrm{S} \, r_C^-) \Big); \tag{7.31}$$

$$f^{-}(r_C^-, C) = \frac{1}{|C| \cdot (2 \cdot |C| + 1)} \cdot \Big(\sum_{x \in C} r(r_C^- \, \mathrm{S} \, x) + 2 \cdot |C| \cdot \sum_{x \in C} r(x \, \mathrm{S} \, r_C^-) \Big); \tag{7.32}$$

$$f^{min-}(r_C^-, C) = \frac{1}{(2 \cdot |C| + 1)} \Big(\min_{x \in C} r(r_C^- \, \mathrm{S} \, x) + 2 \cdot |C| \cdot \min_{x \in C} r(x \, \mathrm{S} \, r_C^-) \Big); \tag{7.33}$$

It is clearly visible that these functions are the same as those characterising an upper profile, only that the two objectives have been reversed in their importance. Therefore, it is mainly desired to have the alternatives inside the set are at least as good as the lower profile, while secondly it is desired that the lower profile is also at least as good as them.

We continue by presenting several ways in which these profiles can be found.

Selecting the fittest alternative

Firstly, each profile can be selected from the alternatives in C so that the fitness functions that characterise them are maximized:

$$r_C^+ = \arg\max_{x \in C} f^+(x, C), \tag{7.34}$$

$$r_C^- = \arg\max_{x \in C} f^-(x, C). \tag{7.35}$$

We have again only represented the formulas using the f^+ and f^- fitness functions, although they could be replaced at any time with any of the others.

Building from top and bottom evaluations

A similar approach as that of building a central profile, from the averages of the evaluations of the alternatives inside a cluster, can also be envisioned in the case of bounding profiles. In this case however, the upper bounding profile is constructed from the maximum evaluations of the alternatives in C on each criterion, while the lower bounding profile is constructed from the minimum ones:

$$r_{Ci}^+ = \max_{x \in C} x_i, \forall i \in F, \tag{7.36}$$

$$r_{Ci}^- = \min_{x \in C} x_i, \forall i \in F. \tag{7.37}$$

This approach guarantees that the upper bounding profile will be at least as good as all the alternatives in C, while they in turn will be at least as good as the lower bounding profile.

Using a mathematical program

We may furthermore consider using a mathematical program in order to build the bounding profiles so that they provide the highest possible values for any of the proposed fitness functions.

Modelling the first fitness function for both bounding profiles is done similarly to the first fitness function of the central profiles. Like in that case, we define four variables for indicating if the value of the outranking relation between a profile and an alternative, or the other way around, is above the cut threshold λ or below its symmetric $-\lambda$. These variables are, for each bounding profile: $\alpha^+_{\mathrm{S}+1}, \alpha^+_{\mathrm{S}+2}, \alpha^-_{\mathrm{S}+1}, \alpha^-_{\mathrm{S}+2}, \alpha^+_{\mathrm{S}-1}, \alpha^+_{\mathrm{S}-2}, \alpha^-_{\mathrm{S}-1}, \alpha^-_{\mathrm{S}-2}$. We will not detail the way in which they are constructed, but reference the way in which their counterparts, used for building central profiles, have been defined in Equation (7.21).

For instance, the crisp credibility degree of the outranking relations between the upper bounding profile r^+_C of a set C and an alternative $x \in C$ are given by:

$$\begin{aligned} r^*(r^+_C \,\mathrm{S}\, x) &= \alpha^+_{\mathrm{S}\,1}(x) - \alpha^-_{\mathrm{S}\,1}(x), \\ r^*(x \,\mathrm{S}\, r^+_C) &= \alpha^+_{\mathrm{S}\,2}(x) - \alpha^-_{\mathrm{S}\,2}(x). \end{aligned} \tag{7.38}$$

The mixed integer program can then be quickly written as:

MIP-BPR*:

$\vdots$

MIP-$\tilde{\mathrm{S}}_{\mathrm{LDP}}$

$\vdots$

Input:

$g(x,i) \in [0,1] \qquad \forall i \in F, \forall x \in C$

$\lambda \in [0,1]$

$w_i, q_i, p_i, v_i \in [0,1] \qquad \forall i \in F$

Output:

$g(r^+_C, i) \in [0,1] \qquad \forall i \in F$

$g(r^-_C, i) \in [0,1] \qquad \forall i \in F$

Objective function:

$\max \quad 2 \cdot |C| \cdot \sum_{x \in C} \big(\alpha^+(r^+_C, x) - \alpha^-(r^+_C, x)\big) + \sum_{x \in C} \big(\alpha^+(x, r^+_C) - \alpha^-(x, r^+_C)\big)$

$\max \quad \sum_{x \in C} \big(\alpha^+(r^-_C, x) - \alpha^-(r^-_C, x)\big) + 2 \cdot |C| \cdot \sum_{x \in C} \big(\alpha^+(x, r^-_C) - \alpha^-(x, r^-_C)\big)$

Variables:

$\alpha^+_{\mathrm{S}+1}(x) \in \{0,1\} \qquad \forall x \in C$

$\alpha^-_{\mathrm{S}+1}(x) \in \{0,1\} \qquad \forall x \in C$

$\alpha^+_{\mathrm{S}+2}(x) \in \{0,1\} \qquad \forall x \in C$

$\alpha^-_{\mathrm{S}+2}(x) \in \{0,1\} \qquad \forall x \in C$

$$
\begin{array}{lll}
 & \alpha^{+}_{\mathrm{S}\,-1}(x) \in \{0,1\} & \forall x \in C \\
 & \alpha^{-}_{\mathrm{S}\,-1}(x) \in \{0,1\} & \forall x \in C \\
 & \alpha^{+}_{\mathrm{S}\,-2}(x) \in \{0,1\} & \forall x \in C \\
 & \alpha^{-}_{\mathrm{S}\,-2}(x) \in \{0,1\} & \forall x \in C \\
\multicolumn{3}{l}{\textit{Constraints}:} \\
\text{s.t.} & -2(1-\alpha^{+}_{\mathrm{S}\,+1}(x)) + \lambda \leqslant s(r^{+}_{C}, x) & \forall x \in C \\
 & s(r^{+}_{C}, x) + \gamma_1 \leqslant 2\alpha^{+}_{\mathrm{S}\,+1}(x) + \lambda & \forall x \in C \\
 & -2\alpha^{-}_{\mathrm{S}\,+1}(x) - \lambda + \gamma_1 \leqslant s(r^{+}_{C}, x) & \forall x \in C \\
 & s(r^{+}_{C}, x) \leqslant 2(1-\alpha^{-}_{\mathrm{S}\,+1}(x)) - \lambda & \forall x \in C \\
 & -2(1-\alpha^{+}_{\mathrm{S}\,+2}(x)) + \lambda \leqslant s(x, r^{+}_{C}) & \forall x \in C \\
 & s(x, r^{+}_{C}) + \gamma_1 \leqslant 2\alpha^{+}_{\mathrm{S}\,+2}(x) + \lambda & \forall x \in C \\
 & -2\alpha^{-}_{\mathrm{S}\,+2}(x) - \lambda + \gamma_1 \leqslant s(x, r^{+}_{C}) & \forall x \in C \\
 & s(x, r^{+}_{C}) \leqslant 2(1-\alpha^{-}_{\mathrm{S}\,+2}(x)) - \lambda & \forall x \in C \\
 & -2(1-\alpha^{+}_{\mathrm{S}\,-1}(x)) + \lambda \leqslant s(r^{-}_{C}, x) & \forall x \in C \\
 & s(r^{-}_{C}, x) + \gamma_1 \leqslant 2\alpha^{+}_{\mathrm{S}\,-1}(x) + \lambda & \forall x \in C \\
 & -2\alpha^{-}_{\mathrm{S}\,-1}(x) - \lambda + \gamma_1 \leqslant s(r^{-}_{C}, x) & \forall x \in C \\
 & s(r^{-}_{C}, x) \leqslant 2(1-\alpha^{-}_{\mathrm{S}\,-1}(x)) - \lambda & \forall x \in C \\
 & -2(1-\alpha^{+}_{\mathrm{S}\,-2}(x)) + \lambda \leqslant s(x, r^{-}_{C}) & \forall x \in C \\
 & s(x, r^{-}_{C}) + \gamma_1 \leqslant 2\alpha^{+}_{\mathrm{S}\,-2}(x) + \lambda & \forall x \in C \\
 & -2\alpha^{-}_{\mathrm{S}\,-2}(x) - \lambda + \gamma_1 \leqslant s(x, r^{-}_{C}) & \forall x \in C \\
 & s(x, r^{-}_{C}) \leqslant 2(1-\alpha^{-}_{\mathrm{S}\,-2}(x)) - \lambda & \forall x \in C
\end{array}
$$

The second fitness functions for each profile are easy to model, as the credibility degree of the outranking relation is given by the s variable defined in the MIP-$\tilde{\mathrm{S}}_{\mathrm{LDP}}$ mathematical program.

$$
\begin{array}{lll}
\multicolumn{3}{l}{\text{MIP-BPR}:} \\
\vdots & & \\
\multicolumn{3}{l}{\text{MIP-}\tilde{\mathrm{S}}_{\mathrm{LDP}}} \\
\vdots & & \\
\multicolumn{3}{l}{\textit{Input}:} \\
 & g(x,i) \in [0,1] & \forall i \in F, \forall x \in C \\
 & \lambda \in [0,1] & \\
 & w_i, q_i, p_i, v_i \in [0,1] & \forall i \in F \\
\multicolumn{3}{l}{\textit{Output}:} \\
 & g(r^{+}_{C}, i) \in [0,1] & \forall i \in F \\
 & g(r^{-}_{C}, i) \in [0,1] & \forall i \in F \\
\multicolumn{3}{l}{\textit{Objective function}:} \\
\max & \multicolumn{2}{l}{2 \cdot |C| \cdot \sum\limits_{x \in C} s(r^{+}_{C}, x) + \sum\limits_{x \in C} s(x, r^{+}_{C})} \\
\max & \multicolumn{2}{l}{\sum\limits_{x \in C} s(r^{-}_{C}, x) + 2 \cdot |C| \cdot \sum\limits_{x \in C} s(x, r^{-}_{C})}
\end{array}
$$

When considering the last of the fitness functions, the same approach as that used for central profiles is applied. We define a binary variable δ^{+}_{1} for each pair of alternatives $x \neq y \in C$, which acts as an indicator for the statement $r(r^{+}_{C}\,\mathrm{S}\,x) > r(r^{+}_{C}\,\mathrm{S}\,y)$ being true. This is modelled through:

$$
\left\{
\begin{array}{lll}
\delta^{+}_{1}(x,y) - 1 & < \; r(r^{+}_{C}\,\mathrm{S}\,x) - r(r^{+}_{C}\,\mathrm{S}\,y) & \leqslant \; \delta^{+}_{1}(x,y), \\
r(r^{+}_{C}\,\mathrm{S}\,x) - 2 \sum\limits_{y \in C} \delta^{+}_{1}(x,y) & \leqslant \; \min\limits_{y \in C} r(r^{+}_{C}\,\mathrm{S}\,y) & \leqslant r(r^{+}_{C}\,\mathrm{S}\,x).
\end{array}
\right.
\tag{7.39}
$$

The same approach is used to determine $\min\limits_{y \in C} r(y \,\mathrm{S}\, r_C^+)$, $\min\limits_{y \in C} r(r_C^- \,\mathrm{S}\, y)$, $\min\limits_{y \in C} r(y \,\mathrm{S}\, r_C^-)$ by using the indicator binary variables δ_2^+, δ_1^- and δ_2^- respectively. The minimum outranking degrees will be denoted with $smin_1^+$ and $smin_2^+$, for the upper bounding profile, and with $smin_1^-$ and $smin_2^-$, for the lower one.

Below we present the mixed-integer linear program for constructing the upper and lower bounding profiles for a set C.

MIP-BPR^{min}:		
$\vdots$		
MIP-$\tilde{\text{S}}_{\text{LDP}}$		
$\vdots$		
Input:		
	$g(x,i) \in [0,1]$	$\forall i \in F, \forall x \in C$
	$\lambda \in [0,1]$	
	$w_i, q_i, p_i, v_i \in [0,1]$	$\forall i \in F$
Output:		
	$g(r_C^+, i) \in [0,1]$	$\forall i \in F$
	$g(r_C^-, i) \in [0,1]$	$\forall i \in F$
Objective function:		
max	$2 \cdot \lvert C\rvert \cdot smin_1^+ + smin_2^+$	
max	$smin_1^- + 2 \cdot \lvert C\rvert \cdot smin_2^-$	
Variables:		
	$\delta_1^+(x,y) \in \{0,1\}$	$\forall x \neq y \in C$
	$\delta_2^+(x,y) \in \{0,1\}$	$\forall x \neq y \in C$
	$\delta_1^-(x,y) \in \{0,1\}$	$\forall x \neq y \in C$
	$\delta_2^-(x,y) \in \{0,1\}$	$\forall x \neq y \in C$
	$smin_1^+ \in [0,1]$	
	$smin_2^+ \in [0,1]$	
	$smin_1^- \in [0,1]$	
	$smin_2^- \in [0,1]$	
Constraints:		
s.t.	$\delta_1^+(x,y) - 1 + \gamma_1 \leqslant s(r_C^+, x) - s(r_C^+, y)$	$\forall x \in C$
	$s(r_C^+, x) - s(r_C^+, y) \leqslant \delta_1^+(x,y)$	$\forall x, y \in C$
	$\delta_2^+(x,y) - 1 + \gamma_1 \leqslant s(x, r_C^+) - s(y, r_C^+)$	$\forall x \in C$
	$s(x, r_C^+) - s(y, r_C^+) \leqslant \delta_2^+(x,y)$	$\forall x, y \in C$
	$smin_1^+ \leqslant s(r_C^+, x)$	$\forall x \in C$
	$s(r_C^+, x) - 2 \sum\limits_{y \in C} \delta_1^+(x,y) \leqslant smin_1^+$	$\forall x \in C$
	$smin_2^+ \leqslant s(x, r_C^+)$	$\forall x \in C$
	$s(x, r_C^+) - 2 \sum\limits_{y \in C} \delta_2^+(x,y) \leqslant smin_2^+$	$\forall x \in C$
	$\delta_1^-(x,y) - 1 + \gamma_1 \leqslant s(r_C^-, x) - s(r_C^-, y)$	$\forall x \in C$
	$s(r_C^-, x) - s(r_C^-, y) \leqslant \delta_1^-(x,y)$	$\forall x, y \in C$
	$\delta_2^-(x,y) - 1 + \gamma_1 \leqslant s(x, r_C^-) - s(y, r_C^-)$	$\forall x \in C$
	$s(x, r_C^-) - s(y, r_C^-) \leqslant \delta_2^-(x,y)$	$\forall x, y \in C$
	$smin_1^- \leqslant s(r_C^-, x)$	$\forall x \in C$
	$s(r_C^-, x) - 2 \sum\limits_{y \in C} \delta_1^-(x,y) \leqslant smin_1^-$	$\forall x \in C$
	$smin_2^- \leqslant s(x, r_C^-)$	$\forall x \in C$
	$s(x, r_C^-) - 2 \sum\limits_{y \in C} \delta_2^-(x,y) \leqslant smin_2^-$	$\forall x \in C$

Using a meta-heuristic approach

In the case of bounding profiles we may again resort to using an approximative approach, when the datasets are large and there are time constraints related to the task of finding them.

We characterise the operation of increasing the evaluation of r_C^+ on criterion i, considering each of the three fitness functions that we try to maximize, through the following heuristic functions:

$$\begin{aligned}
h^{*+}(r_C^+, i) := \frac{1}{|C|(2|C|+1)} \cdot \Big(&2 \cdot |C| \cdot \big|\{x \in C\colon\ x_i - r_{Ci}^+ > q_i \text{ and } r^*(r_C^+ \,\mathrm{S}\, x) = 0\}\big| \\
&-\big|\{x \in C\colon\ x_i - r_{Ci}^+ < -q_i \text{ and } r^*(x \,\mathrm{S}\, r_C^+) = 0\}\big|\Big) \quad (7.40) \\
h^{+}(r_C^+, i) := \frac{1}{|C|(2|C|+1)} \cdot \Big(&2 \cdot |C| \cdot \big|\{x \in C\colon\ x_i - r_{Ci}^+ > q_i\}\big| \\
&-\big|\{x \in C\colon\ x_i - r_{Ci}^+ < -q_i\}\big|\Big) \quad (7.41) \\
h^{min+}(r_C^+, i) := \frac{1}{|C|(2|C|+1)} \cdot \Big(&2 \cdot |C| \cdot \big|\{x \in C\colon\ x_i - r_{Ci}^+ > q_i \text{ and} \\
&\qquad r(r_C^+ \,\mathrm{S}\, x) \leqslant r(r_C^+ \,\mathrm{S}\, y), \forall y \in C\}\big| \\
&-\big|\{x \in C\colon\ x_i - r_{Ci}^+ < -q_i \text{ and} \\
&\qquad r(x \,\mathrm{S}\, r_C^+) \leqslant r(y \,\mathrm{S}\, r_C^+), \forall y \in C\}\big|\Big). \quad (7.42)
\end{aligned}$$

In the equations above we count in the first term the number of alternatives which require that the evaluation of the upper bounding profile to be increased in order for this profile to outrank them on criterion i. The second term counts the number of alternatives which require this evaluation to be decreased so that they will outrank the upper bounding profile. The first term is weighted so that it dominates the second term, and then the second term is subtracted from it. In each equation only a subset of alternatives from C are considered, following from the fitness functions that are used to characterise the quality of the upper bounding profile.

The same heuristic functions, however negated, are used for characterising a decrease in the evaluation of an upper bounding profile.

The heuristic functions that are used in the case of the lower bounding profile are the same as those of the upper bounding profile, except that the weights of the two terms are reversed.

These functions can then be used in a single-solution meta-heuristic, in order to construct both types of profiles.

7.2.3 Separating profiles

We furthermore consider a particular case of profile, which is best suited for characterising the separation between two sets of alternatives that are considered to be ordered. The upper set is delimited from below, while the lower one from above. We call these **separating profiles**. Due to their practical use for characterising ordered clustering results, we consider an ordered set of clusters $K = \{K_1, \cdots, K_k\}$, where K_1 is the top cluster, and denote the separating profiles with $r_l^{\pm}, \forall l \in 1..k-1$. Such profiles may be considered to be similar to the ones defined in the ELECTRE TRI method [Yu 1992, Roy and Bouyssou 1993].

We provide an intuitive illustration of such a profile in Figure 7.4.

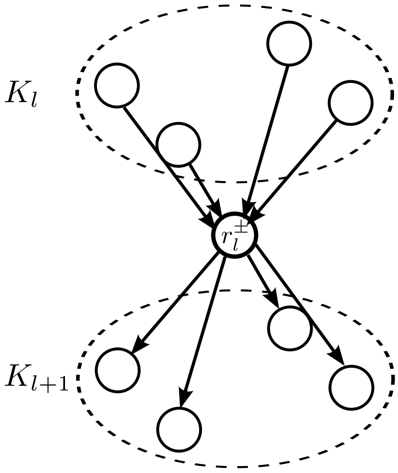


Figure 7.4: Separating profile;

We define the following fitness functions for characterising a separating profile:

$$f^{*\pm}(r_l^{\pm}, K) = \frac{1}{|K_l| + |K_{l+1}|} \cdot \Big(\sum_{x \in K_l} r^*(r_l^{\pm} \,\mathrm{S}\, x) + \sum_{x \in K_{l+1}} r^*(x \,\mathrm{S}\, r_l^{\pm}) \Big); \tag{7.43}$$

$$f^{\pm}(r_l^{\pm}, K) = \frac{1}{|K_l| + |K_{l+1}|} \cdot \Big(\sum_{x \in K_l} r(r_l^{\pm} \,\mathrm{S}\, x) + \sum_{x \in K_{l+1}} r(x \,\mathrm{S}\, r_l^{\pm}) \Big); \tag{7.44}$$

$$f^{min\pm}(r_l^{\pm}, K) = \min \Big(\min_{x \in K_l} r(r_l^{\pm} \,\mathrm{S}\, x), \min_{x \in K_{l+1}} r(x \,\mathrm{S}\, r_l^{\pm}) \Big); \tag{7.45}$$

We notice that we look for profiles that outrank the alternatives in the lower cluster but which are in turn outranked by the alternatives from the cluster above.

Selecting the fittest alternative

We select the $f^{\pm}$ fitness function, in order to highlight the procedure of selecting the fittest alternative from two neighbouring clusters, in order to act as a separating profile. Any of the three functions defined before may take its place. The separating profile between clusters K_l and K_{l+1} from K can be selected from one of the two clusters in the following way:

$$r^{\pm}_{l} = \underset{x \in K_l \cup K_{l+1}}{\arg\max}\ f^{\pm}(x, K). \tag{7.46}$$

Building from the alternatives evaluations

The separating profile may be additionally constructed from the evaluations of the alternatives inside the two clusters it separates.

Since it may be considered as the mid point between the central profiles of the two clusters, we construct it as the average between the average evaluations of both clusters:

$$r^{\pm}_{l\ i} = \frac{1}{2}\Big(\frac{1}{|K_{l+1}|}\sum_{x \in K_{l+1}} x_i + \frac{1}{|K_l|}\sum_{x \in K_l} x_i\Big), \forall i \in F. \tag{7.47}$$

Using a mathematical program

A mathematical program can also be used to construct the best fitting separating profile given two clusters, K_l and K_{l+1}, where the first is considered strictly preferred to the second.

Like in the case of central profiles, the first fitness function, $f^{*\pm}$, can be modelled by adding four binary variables $\alpha^{+}_{\mathrm{S}\,1}, \alpha^{+}_{\mathrm{S}\,2}, \alpha^{-}_{\mathrm{S}\,1}, \alpha^{-}_{\mathrm{S}\,2}$, and using linear constraints such as those in Equation (7.21).

The mathematical program becomes in this case:

MIP-SPR*:

⋮

MIP-$\tilde{\mathrm{S}}_{\mathrm{LDP}}$

⋮

Input:

	$g(x, i) \in [0, 1]$	$\forall i \in F, \forall x \in K_l \cup K_{l+1}$
	$\lambda \in [0, 1]$	
	$w_i, q_i, p_i, v_i \in [0, 1]$	$\forall i \in F$

Output:

	$g(r^{\pm}_{l}, i) \in [0, 1]$	$\forall i \in F$

Objective function:

$$\max \quad \sum_{x\in K_l}\left(\alpha^{+}_{\mathrm{S}\,2}(x)-\alpha^{-}_{\mathrm{S}\,2}(x)\right)+\sum_{x\in K_{l+1}}\left(\alpha^{+}_{\mathrm{S}\,1}(x)-\alpha^{-}_{\mathrm{S}\,1}(x)\right)$$

Variables:

$$\begin{array}{ll}
\alpha^{+}_{\mathrm{S}\,1}(x)\in\{0,1\} & \forall x\in K_{l+1}\\
\alpha^{-}_{\mathrm{S}\,1}(x)\in\{0,1\} & \forall x\in K_{l+1}\\
\alpha^{+}_{\mathrm{S}\,2}(x)\in\{0,1\} & \forall x\in K_{l}\\
\alpha^{-}_{\mathrm{S}\,2}(x)\in\{0,1\} & \forall x\in K_{l}
\end{array}$$

Constraints:

$$\begin{array}{lll}
\text{s.t.} & -2(1-\alpha^{+}_{\mathrm{S}\,1}(x))+\lambda\leqslant s(r_l^{\pm},x) & \forall x\in K_{l+1}\\
 & s(r_l^{\pm},x)+\gamma_1\leqslant 2\alpha^{+}_{\mathrm{S}\,1}(x)+\lambda & \forall x\in K_{l+1}\\
 & -2\alpha^{-}_{\mathrm{S}\,1}(x)-\lambda+\gamma_1\leqslant s(r_l^{\pm},x) & \forall x\in K_{l+1}\\
 & s(r_l^{\pm},x)\leqslant 2(1-\alpha^{-}_{\mathrm{S}\,1}(x))-\lambda & \forall x\in K_{l+1}\\
 & -2(1-\alpha^{+}_{\mathrm{S}\,2}(x))+\lambda\leqslant s(x,r_l^{\pm}) & \forall x\in K_{l}\\
 & s(x,r_l^{\pm})+\gamma_1\leqslant 2\alpha^{+}_{\mathrm{S}\,2}(x)+\lambda & \forall x\in K_{l}\\
 & -2\alpha^{-}_{\mathrm{S}\,2}(x)-\lambda+\gamma_1\leqslant s(x,r_l^{\pm}) & \forall x\in K_{l}\\
 & s(x,r_l^{\pm})\leqslant 2(1-\alpha^{-}_{\mathrm{S}\,2}(x))-\lambda & \forall x\in K_{l}
\end{array}$$

The second fitness function, $f^{\pm}$, only uses the s variable already defined in the mathematical program MIP-$\tilde{\mathrm{S}}_{\mathrm{LDP}}$. The mathematical program for finding a profile according to this function is:

MIP-SPR:

$\vdots$

MIP-$\tilde{\mathrm{S}}_{\mathrm{LDP}}$

$\vdots$

Input:

$$\begin{array}{ll}
g(x,i)\in[0,1] & \forall i\in F,\forall x\in K_l\cup K_{l+1}\\
\lambda\in[0,1] & \\
w_i,q_i,p_i,v_i\in[0,1] & \forall i\in F
\end{array}$$

Output:

$$\begin{array}{ll}
g(r_l^{\pm},i)\in[0,1] & \forall i\in F
\end{array}$$

Objective function:

$$\max \quad \sum_{x\in K_{l+1}} s(r_l^{\pm},x)+\sum_{x\in K_l} s(x,r_l^{\pm})$$

In the case of the last fitness function, $f^{\pm}$, two binary variables δ^{+} and δ^{-} are defined, the first for each pair of alternatives $x\neq y\in K_l$, while the second for each pair $x\neq y\in K_{l+1}$. These variables are used as indicators for $r(r_l^{\pm}\,\mathrm{S}\,x)>r(r_l^{\pm}\,\mathrm{S}\,y),\forall x\in K_{l+1}$ and $r(x\,\mathrm{S}\,r_l^{\pm})>r(y\,\mathrm{S}\,r_l^{\pm}),\forall x\in K_l$ respectively being true. They are constructed in exactly the same way as the other indicator variables used in conjunction with the last fitness functions in the case of both central and bounding profiles. The minimum credibility degree of the outranking relations between the profile and the alternatives in the lower cluster, denoted with $smin^{+}$, and that of the outranking relations between the alternatives in the upper cluster and the separating profile, denoted with $smin^{-}$, are constructed from the previously described binary variables.

Below we present the mixed-integer linear program for constructing the separating profile between the clusters K_l and K_{l+1}:

MIP-SPR^{min}:		
$\vdots$		
MIP-$\tilde{\text{S}}_{\text{LDP}}$		
$\vdots$		
Input:		
	$g(x,i) \in [0,1]$	$\forall i \in F, \forall x \in K_l \cup K_{l+1}$
	$\lambda \in [0,1]$	
	$w_i, q_i, p_i, v_i \in [0,1]$	$\forall i \in F$
Output:		
	$g(r_l^{\pm}, i) \in [0,1]$	$\forall i \in F$
Objective function:		
max	$smin^+ + smin^-$	
Variables:		
	$\delta^+(x,y) \in \{0,1\}$	$\forall x \neq y \in K_{l+1}$
	$\delta^-(x,y) \in \{0,1\}$	$\forall x \neq y \in K_l$
	$smin^+ \in [0,1]$	
	$smin^- \in [0,1]$	
Constraints:		
s.t.	$\delta^+(x,y) - 1 + \gamma_1 \leqslant s(r_l^{\pm}, x) - s(r_l^{\pm}, y)$	$\forall x, y \in K_{l+1}$
	$s(r_l^{\pm}, x) - s(r_l^{\pm}, y) \leqslant \delta^+(x,y)$	$\forall x, y \in K_{l+1}$
	$\delta^-(x,y) - 1 + \gamma_1 \leqslant s(x, r_l^{\pm}) - s(y, r_l^{\pm})$	$\forall x, y \in K_l$
	$s(x, r_l^{\pm}) - s(y, r_l^{\pm}) \leqslant \delta^-(x,y)$	$\forall x, y \in K_l$
	$smin^+ \leqslant s(r_l^{\pm}, x)$	$\forall x \in K_{l+1}$
	$s(r_l^{\pm}, x) - 2 \sum\limits_{y \in C} \delta^+(x,y) \leqslant smin^+$	$\forall x \in K_{l+1}$
	$smin^- \leqslant s(x, r_l^{\pm})$	$\forall x \in K_l$
	$s(x, r_l^{\pm}) - 2 \sum\limits_{y \in C} \delta_2^+(x,y) \leqslant smin^-$	$\forall x \in K_l$

Using a meta-heuristic approach

The meta-heuristic approach to building a separating profile between two clusters, K_l and K_{l+1}, is aided by the following heuristic functions which reflect the quality the operation of increasing the evaluation of the separating profile on criterion i is.

$$
\begin{aligned}
h^*(r_l^{\pm}, i) := \frac{1}{|K_l| + |K_{l+1}|} \cdot \Big(& \big|\{x \in K_{l+1}\colon\ x_i - {r_l^{\pm}}_i > q_i \text{ and } r^*(r_l^{\pm} \,\mathrm{S}\, x) = 0\}\big| \\
& - \big|\{x \in K_l\colon\ x_i - {r_l^{\pm}}_i < -q_i \text{ and } r^*(x \,\mathrm{S}\, r_l^{\pm}) = 0\}\big| \Big) \qquad (7.48) \\
h(r_l^{\pm}, i) := \frac{1}{|K_l| + |K_{l+1}|} \cdot \Big(& \big|\{x \in K_{l+1}\colon\ x_i - {r_l^{\pm}}_i > q_i\}\big| \\
& - \big|\{x \in K_l\colon\ x_i - {r_l^{\pm}}_i < -q_i\}\big| \Big) \qquad (7.49) \\
h^{min}(r_l^{\pm}, i) := \frac{1}{|K_l| + |K_{l+1}|} \cdot \Big(& \big|\{x \in K_{l+1}\colon x_i - {r_l^{\pm}}_i > q_i \wedge r(r_l^{\pm} \,\mathrm{S}\, x) \leqslant r(r_l^{\pm} \,\mathrm{S}\, y), \forall y \in K_{l+1}\}\big| \\
& - \big|\{x \in K_l\colon x_i - {r_l^{\pm}}_i < -q_i \wedge r(x \,\mathrm{S}\, r_l^{\pm}) \leqslant r(y \,\mathrm{S}\, r_l^{\pm}), \forall y \in K_l\}\big| \Big). (7.50)
\end{aligned}
$$

These functions balance those alternatives from the lower cluster, for which the evaluation of the separating profile should be increase so that the profile will outrank these alternatives on criterion i, and those alternatives from the upper cluster, for which the evaluation of the profile should be decreased so that they will outrank it on the same criterion. These functions are valued in the $[-1, 1]$ interval and it follows from their definition that the heuristic functions that characterise the operation of decreasing the evaluation of the separating profile are simply the ones above but with the reversed sign.

Finally, like for the other two types of profiles, these heuristic functions may be used with a single-solution meta-heuristic in order to construct the separating profile.

7.3 Validating the approaches

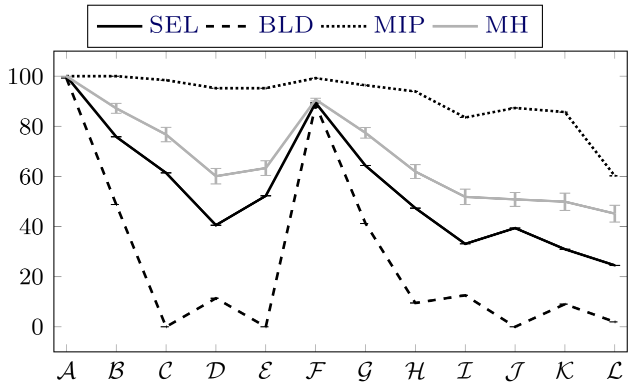
In order to test and compare the different approaches of building central, bounding and separating profiles, we will again be using the set of benchmarks from Chapter 6. These benchmarks contain 100 alternatives, which are defined on 11 criteria, while different weaker of stronger structures are considered, forming a total of 300 problem instances.

We will consider each type of profiles independently of the others, however, as for all of them we have proposed four types of approaches for constructing them, we will be using the same notations for these approaches. Therefore the approaches of selecting an existing alternative as a profile, be it central, bounding or separating, will be denoted with SEL. The approaches that build these profiles from the evaluations of the alternatives will be denoted with BLD, while those that use a mixed-integer program with MIP. In the case of the last type of approach, a simulated annealing implementation has been selected, which we denote with MH. This approach will start from the results given by the SEL approach, which may be considered as an initialization heuristic. Furthermore, over the considered benchmarks the SEL approach is very quick, making it a good candidate for giving the initial solution to the MH approach, although BLD could equally be used in its place. All the approaches, except MH, are deterministic, therefore they have been executed only once over each problem instance, while the latter has been repeated 50 times over each problem instance.

For each of the considered benchmarks, the 10 clusters of alternatives, that were initially constructed in each benchmark, are used when constructing each type of profile. Over the considered benchmarks, the first two approaches for constructing any type of profile are very fast and so no execution time limit was needed. However, the approaches that use mixed-integer programs have the potential of requiring a very long time in order to retrieve an optimal solution. The main reason for the use of the MIP is to illustrate how close the other approaches come to an optimal solution, although, due to certain constraints, we have limited the execution times of this type of approaches to ten minutes for any individual profile. In order to simulate an on-line use of the approaches of constructing profiles that describe sets of alternatives, we have additionally limited the execution of the meta-heuristic approaches to a total of 20 seconds. Nevertheless, this time was sufficient for the approaches to converge to a final solution over the considered benchmarks. A more extensive analysis, considering different benchmark sizes will follow in the future.

7.3.1 Extracting central profiles

We start with the results for building central profiles. In Figure 7.5 we present the average fitness values of the central profiles extracted using each of the four methods, considering the f^* fitness measure from Equation (7.15). The standard deviation is also depicted within brackets.



SEL	100.0 (0.0)	75.8 (0.0)	61.4 (0.0)	40.5 (0.0)	52.2 (0.0)	89.3 (0.0)	64.3 (0.0)	47.4 (0.0)	33.1 (0.0)	39.4 (0.0)	31.0 (0.0)	24.5 (0.0)
BLD	99.2 (0.0)	48.8 (0.0)	0.0 (0.0)	11.4 (0.0)	0.0 (0.0)	88.6 (0.0)	41.3 (0.0)	9.5 (0.0)	12.6 (0.0)	0.0 (0.0)	9.0 (0.0)	1.9 (0.0)
MIP	100.0 (0.0)	100.0 (0.0)	98.4 (0.0)	95.2 (0.0)	95.2 (0.0)	99.2 (0.0)	96.4 (0.0)	93.9 (0.0)	83.5 (0.0)	87.3 (0.0)	85.7 (0.0)	60.2 (0.0)
MH	100.0 (0.0)	87.2 (1.9)	76.7 (2.9)	60.1 (3.1)	63.3 (2.9)	90.7 (0.5)	77.4 (2.1)	61.9 (2.7)	51.8 (3.1)	50.8 (2.7)	49.9 (3.4)	45.2 (3.4)

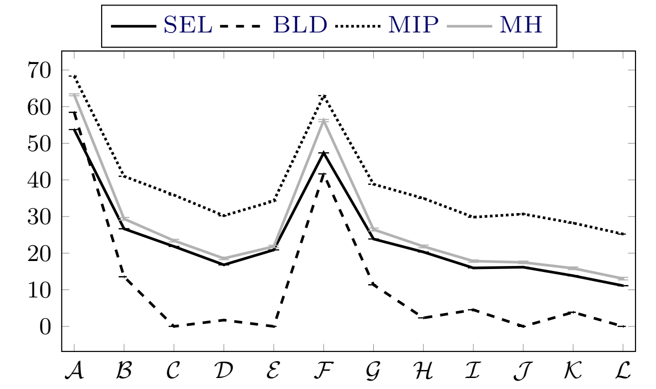
Figure 7.5: Building central profiles with respect to the f^* fitness measure;

We quickly notice that the approach of using a mathematical program performs the best, which is expected as it retrieves the optimal result with respect to the considered fitness measure. In certain cases, however, this approach may not be able to find this result, due to the fact that we have restricted the approach to a maximum running time of ten minutes for any profile.

The approaches of selecting and building a central profile perform the worst, with the latter showing rather poor overall performances compared to the other approaches.

The meta-heuristic approach offers significant improvements over the approach of selecting a central profile, which is used as a starting point for this method. Over the benchmark types that contain more heterogeneous clusters of alternatives, the improvements are more significant than on the benchmark types containing clusters of alternatives which have closer evaluations.

In the following figure we illustrate the results of the methods for building central profiles, considering the f fitness measure from Equation (7.16).

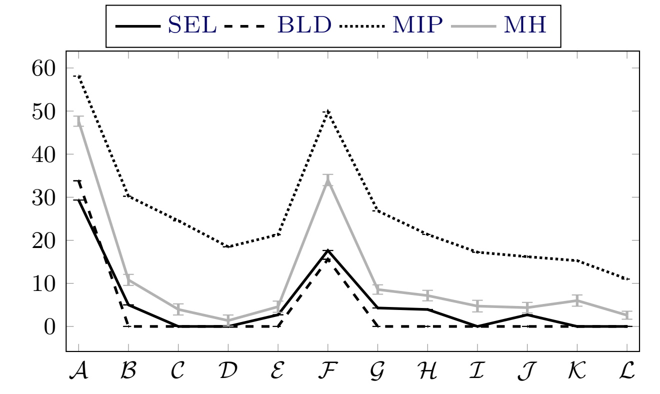


	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	$\mathcal{D}$	$\mathcal{E}$	$\mathcal{F}$	$\mathcal{G}$	$\mathcal{H}$	$\mathcal{I}$	$\mathcal{J}$	$\mathcal{K}$	$\mathcal{L}$
SEL	53.8 (0.0)	26.7 (0.0)	21.8 (0.0)	16.8 (0.0)	20.9 (0.0)	47.4 (0.0)	23.9 (0.0)	20.3 (0.0)	16.0 (0.0)	16.2 (0.0)	13.8 (0.0)	11.1 (0.0)
BLD	58.4 (0.0)	13.6 (0.0)	0.0 (0.0)	1.7 (0.0)	0.0 (0.0)	41.6 (0.0)	11.4 (0.0)	2.3 (0.0)	4.5 (0.0)	0.0 (0.0)	3.9 (0.0)	0.0 (0.0)
MIP	68.4 (0.0)	41.0 (0.0)	35.9 (0.0)	30.2 (0.0)	34.3 (0.0)	63.0 (0.0)	38.8 (0.0)	35.0 (0.0)	29.8 (0.0)	30.7 (0.0)	28.2 (0.0)	25.2 (0.0)
MH	63.2 (0.3)	29.4 (0.4)	23.4 (0.4)	18.6 (0.3)	21.8 (0.3)	56.2 (0.3)	26.5 (0.4)	21.8 (0.4)	17.8 (0.3)	17.5 (0.3)	15.9 (0.4)	13.0 (0.3)

Figure 7.6: Building central profiles with respect to the f fitness measure;

Looking at Figure 7.6, we find that the overall performance of the approaches decreases, which is natural as this fitness measure is more restrictive.

Similar remarks as before can be made, with the addition of the fact that the improvements of the meta-heuristic approach over the approach of selecting an existing alternative as a central profile is much smaller. The approach of constructing a central profile from the average evaluations of the alternatives inside each cluster is again seen to perform rather poorly.



	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	$\mathcal{D}$	$\mathcal{E}$	$\mathcal{F}$	$\mathcal{G}$	$\mathcal{H}$	$\mathcal{I}$	$\mathcal{J}$	$\mathcal{K}$	$\mathcal{L}$
SEL	29.3 (0.0)	5.0 (0.0)	0.0 (0.0)	0.0 (0.0)	2.7 (0.0)	17.6 (0.0)	4.3 (0.0)	3.9 (0.0)	0.0 (0.0)	2.7 (0.0)	0.0 (0.0)	0.0 (0.0)
BLD	33.8 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	15.6 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
MIP	58.1 (0.0)	30.2 (0.0)	24.5 (0.0)	18.5 (0.0)	21.3 (0.0)	49.8 (0.0)	26.8 (0.0)	21.4 (0.0)	17.2 (0.0)	16.2 (0.0)	15.3 (0.0)	11.0 (0.0)
MH	47.7 (1.2)	10.8 (1.3)	4.0 (1.3)	1.4 (1.3)	4.6 (1.3)	34.0 (1.3)	8.6 (1.1)	7.2 (1.2)	4.7 (1.3)	4.4 (1.2)	6.0 (1.3)	2.6 (0.9)

Figure 7.7: Building central profiles with respect to the f^{min} fitness measure;

Finally, in Figure 7.7, we present the results of building central profiles with respect to the f^{min} fitness measure from Equation (7.17).

In addition to the overall lower values of this fitness measure, the meta-heuristic approach provides better improvements over the SEL approach, as opposed to the previous case. However, the MIP approach may be seen to perform much better than all the rest.

In Table 7.11 we present the average running times of the presented algorithms, across all benchmarks, while considering each fitness function separately.

Table 7.11: Average running times of methods for building central profiles;

Approach	Time (s)					
	f^*		f		f^{min}	
SEL	< 1		< 1		< 1	
BLD	< 1		< 1		< 1	
MIP	140	(283)	486	(355)	615	(433)
MH	20		20		20	

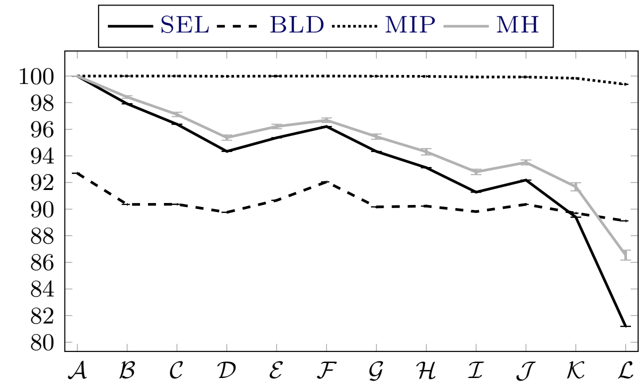
We find that the first two approaches have very short execution times, the approach of using a mathematical program takes a very long time to find the optimal result, while the meta-heuristic approach is stopped after taking 2 seconds to execute for each central profile of the 10 clusters of alternatives.

Considering the overall high execution time required by the MIP approach, as seen in Table 7.11, we may conclude that this approach is only suited when the clusters are small in size, and the number of criteria on which the alternatives are defined is also small. The approach of selecting an existing alternative seems to perform rather well, considering the very small execution time required, while the meta-heuristic approaches offer good improvements when considering the f^* and f^{min} fitness measures. Finally, the approach of constructing a central profile from the average evaluations of the alternatives inside a cluster is not at all suited for constructing central profiles.

7.3.2 Extracting bounding profiles

We continue by using the same experimental framework in order to test the different methods of building, in this case, bounding profiles. Both the upper and lower profiles are evaluated with respect to fitness functions that are similarly constructed and valued. Therefore, we will be using the average of these fitness functions in order to illustrate the overall performance of the four types of approaches for building bounding profiles.

In Figure 7.8 we presented the results of the four methods, when looking for bounding profiles defined through the f^{*+} and f^{*-} fitness measures, from Equations (7.28) and (7.31).



SEL	100.0 (0.0)	97.9 (0.0)	96.4 (0.0)	94.3 (0.0)	95.4 (0.0)	96.2 (0.0)	94.3 (0.0)	93.1 (0.0)	91.3 (0.0)	92.2 (0.0)	89.4 (0.0)	81.2 (0.0)
BLD	92.7 (0.0)	90.4 (0.0)	90.4 (0.0)	89.8 (0.0)	90.7 (0.0)	92.0 (0.0)	90.2 (0.0)	90.2 (0.0)	89.8 (0.0)	90.4 (0.0)	89.7 (0.0)	89.1 (0.0)
MIP	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	100.0 (0.0)	99.9 (0.0)	99.9 (0.0)	99.8 (0.0)	99.4 (0.0)
MH	100.0 (0.0)	98.4 (0.1)	97.1 (0.2)	95.4 (0.2)	96.2 (0.2)	96.7 (0.2)	95.4 (0.2)	94.3 (0.2)	92.8 (0.2)	93.5 (0.2)	91.7 (0.3)	86.5 (0.4)

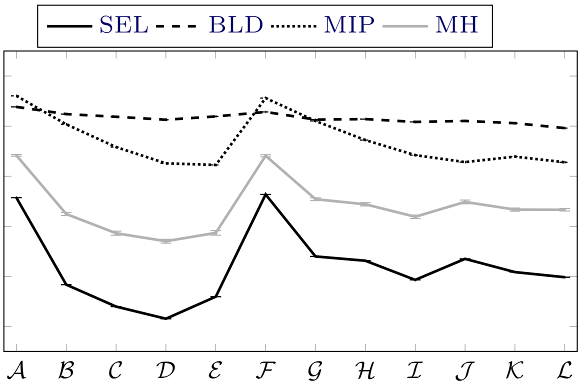
Figure 7.8: Building bounding profiles with respect to the f^{*+}, f^{*-} fitness measures;

We may notice that, in almost all cases, the approach of using a mathematical program finds results that maximize the fitness measures. This means that the bounding profiles found by this method are indifferent to all the alternatives inside their cluster. Comparing these results to those of the mathematical program approach for finding central profiles, we may deduce that the latter was not always able to find an optimal solution. Since the mathematical programs for finding bounding profiles are simpler, they have been able to find the optimal bounding profiles which in this particular case are identical to the optimal central profiles. Nevertheless, on other problem instances where a bounding profile may not become indifferent to all the alternatives inside a cluster, using this approach for constructing a central profile is not adequate.

The SEL approach perform rather well across most of the benchmarks, while the meta-heuristic approach provides larger improvements over the more difficult problem instances. The BLD approach performs worse than all the rest across the easier problems instances, although this changes when considering the more difficult ones.

In Figure 7.9 we continue with the results of the methods that look for bounding profiles that are defined by the f^{+} and f^{-} fitness measures from Equations (7.29) and (7.32).

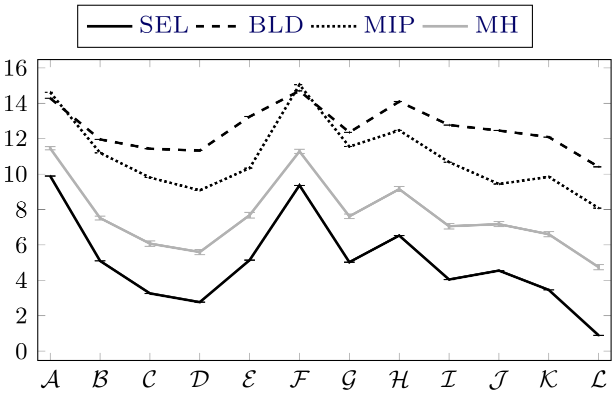
In this case, the credibility degrees of the outranking relations are also taken into account, and not just the crisp ones. We find that in this case the SEL approach performs the worst, while the MH approach is able to provide significant improvements on its solutions. Furthermore, the MIP approach is not always performing the best, which indicates that, in certain cases, it was not able to find an optimal result for certain bounding profiles, within the given 10 minute time frame.



SEL	75.7 (0.0)	58.3 (0.0)	54.0 (0.0)	51.5 (0.0)	55.9 (0.0)	76.3 (0.0)	64.0 (0.0)	63.1 (0.0)	59.3 (0.0)	63.5 (0.0)	60.8 (0.0)	59.8 (0.0)
BLD	93.8 (0.0)	92.4 (0.0)	91.9 (0.0)	91.3 (0.0)	91.9 (0.0)	92.8 (0.0)	91.3 (0.0)	91.4 (0.0)	90.8 (0.0)	91.0 (0.0)	90.6 (0.0)	89.6 (0.0)
MIP	96.0 (0.0)	90.4 (0.0)	85.8 (0.0)	82.6 (0.0)	82.2 (0.0)	95.6 (0.0)	91.0 (0.0)	87.2 (0.0)	84.2 (0.0)	82.8 (0.0)	83.9 (0.0)	82.8 (0.0)
MH	84.2 (0.2)	72.4 (0.3)	68.6 (0.4)	67.0 (0.4)	68.7 (0.5)	84.1 (0.2)	75.4 (0.3)	74.4 (0.3)	71.9 (0.3)	74.9 (0.4)	73.3 (0.3)	73.3 (0.2)

Figure 7.9: Building bounding profiles with respect to the f^+, f^- fitness measures;

The approach of building the bounding profiles from the best and worst evaluations of the alternatives inside a cluster performs the best, in this case. We can deduce from this, that as the upper bounding profiles constructed in this way are guaranteed to outrank all the alternatives inside their corresponding cluster, while the lower bounding profiles are guaranteed to be outranked by them, the most important components of the f^+ and f^- fitness measures are being maximized. Most of the other approaches do not seem to be able to do this.



SEL	9.9 (0.0)	5.1 (0.0)	3.3 (0.0)	2.8 (0.0)	5.1 (0.0)	9.4 (0.0)	5.0 (0.0)	6.5 (0.0)	4.0 (0.0)	4.5 (0.0)	3.5 (0.0)	0.9 (0.0)
BLD	14.3 (0.0)	12.0 (0.0)	11.4 (0.0)	11.3 (0.0)	13.2 (0.0)	14.7 (0.0)	12.4 (0.0)	14.1 (0.0)	12.8 (0.0)	12.5 (0.0)	12.1 (0.0)	10.4 (0.0)
MIP	14.6 (0.0)	11.2 (0.0)	9.8 (0.0)	9.1 (0.0)	10.3 (0.0)	15.1 (0.0)	11.6 (0.0)	12.5 (0.0)	10.7 (0.0)	9.4 (0.0)	9.8 (0.0)	8.1 (0.0)
MH	11.5 (0.1)	7.5 (0.1)	6.1 (0.1)	5.6 (0.2)	7.7 (0.2)	11.3 (0.1)	7.6 (0.1)	9.2 (0.1)	7.1 (0.2)	7.2 (0.1)	6.6 (0.1)	4.7 (0.2)

Figure 7.10: Building bounding profiles with respect to the f^{min+}, f^{min-} measures;

In Figure 7.10 we present the results of the methods that look for bounding profiles that are defined by the fitness measures from Equations (7.30) and (7.33).

We may draw similar conclusions as before. The BLD approach performs the best, while the MIP approach is not able to find the optimal results during its fixed execution time. The SEL approach performs the worst, highlighting the fact that existing alternatives from each cluster are not good candidates for bounding the other alternatives in the cluster, based on the f^{min+} and f^{min-} fitness measures. Finally, the MH approach is able to provide significant improvements over the result of the SEL approach, however, as these initial results are quite poor, the results of this approach are also not as good as those of the other approaches.

In Table 7.12 we present the average execution times of the four approaches.

Table 7.12: Average running times of methods for building bounding profiles;

Approach	Time (s)		
	f^{*+}, f^{*-}	f^{+}, f^{-}	f^{min+}, f^{min-}
SEL	< 1	< 1	< 1
BLD	< 1	< 1	< 1
MIP	93 (157)	194 (139)	87 (29)
MH	20	20	20

The first two approaches are in this case too very quick, while the mixed integer programs is the slowest. The meta-heuristic approach is given a fixed time of 20 second for each problem instance.

In relation to the execution times of the MIP approach for extracting central profiles, the same approach for constructing bounding profiles is much quicker. Nevertheless, this approach should be used only on small problem instances, or when an exact solution is required.

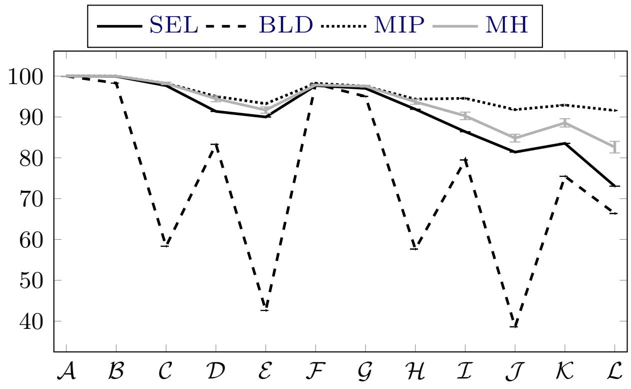
Furthermore, the approach of building the bounding profiles from the evaluations of the alternatives performs very well when considering the last two sets of fitness measures, and since it has a very quick execution time, it is the best choice for these cases. When considering the first set of fitness measures, the approach of selecting an existing alternative as a bounding profile seems well-motivated, except when the problem instance contains alternatives with very contrasting evaluations.

7.3.3 Extracting separating profiles

We conclude this validation section with the experimental results of the approaches for building separating profiles. These profiles are meant to be placed between two clusters of alternatives which are considered to be in a relation of strict preference. Therefore, the alternatives from the cluster that is strictly preferred to the second need to outrank the separating profile, while the alternatives from the second cluster need to be outranked by it.

Considering the problem instances that we have used until now in our experiments, we will define the separating profiles between the 10 clusters of alternatives that have been originally constructed. We will also use the order in which these clusters have been generated, considering that the last cluster is at the top of the order, while the first is at the bottom. Hence, 9 separating profiles will be generated for each problem instance. Although in the case of the more difficult problem instances the order between the initial clusters may be weaker, or not at all present, this simulates the attempts of constructing separating profiles between clusters that are not very well delimited.

In Figure 7.11 we present the results of the four methods, when looking for separating profiles defined through the f^* fitness measure, from Equation (7.43).



SEL	100.0 (0.0)	99.9 (0.0)	97.7 (0.0)	91.4 (0.0)	90.0 (0.0)	97.7 (0.0)	97.0 (0.0)	91.9 (0.0)	86.4 (0.0)	81.4 (0.0)	83.5 (0.0)	73.1 (0.0)
BLD	100.0 (0.0)	98.3 (0.0)	58.3 (0.0)	83.3 (0.0)	42.6 (0.0)	98.1 (0.0)	95.0 (0.0)	57.6 (0.0)	79.5 (0.0)	38.6 (0.0)	75.5 (0.0)	66.4 (0.0)
MIP	100.0 (0.0)	100.0 (0.0)	98.2 (0.0)	95.1 (0.0)	93.3 (0.0)	98.2 (0.0)	97.5 (0.0)	94.4 (0.0)	94.6 (0.0)	91.8 (0.0)	92.9 (0.0)	91.6 (0.0)
MH	100.0 (0.0)	100.0 (0.0)	98.2 (0.3)	94.5 (0.7)	91.7 (0.8)	97.8 (0.2)	97.5 (0.2)	93.7 (0.6)	90.3 (0.9)	84.8 (0.9)	88.6 (1.0)	82.6 (1.4)

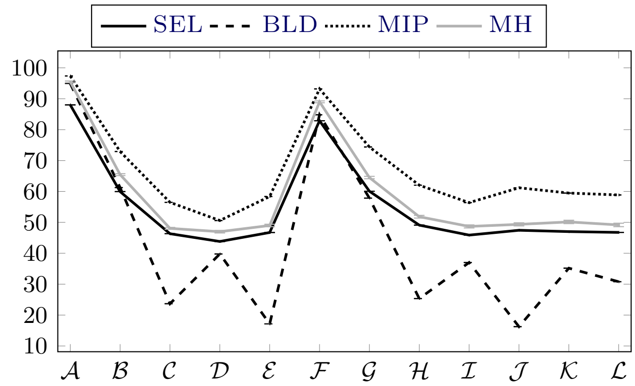
Figure 7.11: Building separating profiles with respect to the f^* fitness measure;

We observe that over the problem instances which contain clusters of alternatives that are compact ($\mathcal{A}$, $\mathcal{B}$, $\mathcal{F}$ and $\mathcal{G}$), we are able to delimit these clusters through profiles that are outranked by all the alternatives in the upper cluster, and that outrank in turn all the alternatives from the lower cluster.

The MIP approach performs the best in this case, while the SEL approach is also able to perform very well. The meta-heuristic which improves on the result of the latter provides only small improvements, which are motivated by the very good quality of the starting solution.

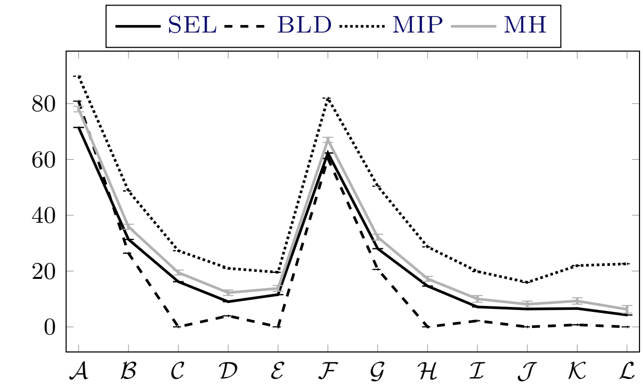
The BLD approach, however, performs rather poorly, especially across the benchmarks which contain alternatives with increasingly contrasting evaluations. Since the separating profiles generated by this approach use the average between the average evaluations of the alternatives in each cluster, this confirms that the average operator is not properly suited when measures based on outranking relations are involved.

In Figures 7.12 and 7.13 we presented the results when looking for separating profiles defined through the fitness measures from Equations (7.44) and (7.45).



SEL	88.0 (0.0)	60.0 (0.0)	46.3 (0.0)	43.8 (0.0)	46.7 (0.0)	82.9 (0.0)	60.1 (0.0)	49.1 (0.0)	45.9 (0.0)	47.4 (0.0)	47.0 (0.0)	46.8 (0.0)
BLD	94.9 (0.0)	61.2 (0.0)	23.7 (0.0)	39.8 (0.0)	17.1 (0.0)	84.8 (0.0)	57.8 (0.0)	25.3 (0.0)	37.0 (0.0)	16.3 (0.0)	35.2 (0.0)	30.8 (0.0)
MIP	97.4 (0.0)	72.9 (0.0)	56.5 (0.0)	50.6 (0.0)	58.4 (0.0)	93.2 (0.0)	74.4 (0.0)	62.0 (0.0)	56.4 (0.0)	61.2 (0.0)	59.5 (0.0)	58.9 (0.0)
MH	95.5 (0.3)	65.4 (0.3)	48.1 (0.3)	47.0 (0.4)	49.0 (0.4)	89.0 (0.3)	64.5 (0.4)	51.8 (0.4)	48.7 (0.5)	49.4 (0.5)	50.1 (0.5)	49.2 (0.6)

Figure 7.12: Building separating profiles with respect to the f fitness measure;



SEL	71.4 (0.0)	31.4 (0.0)	16.2 (0.0)	9.0 (0.0)	11.6 (0.0)	62.3 (0.0)	28.0 (0.0)	14.6 (0.0)	7.1 (0.0)	6.4 (0.0)	6.6 (0.0)	4.3 (0.0)
BLD	80.9 (0.0)	26.4 (0.0)	0.0 (0.0)	4.0 (0.0)	0.0 (0.0)	60.4 (0.0)	20.6 (0.0)	0.0 (0.0)	2.2 (0.0)	0.0 (0.0)	0.8 (0.0)	0.0 (0.0)
MIP	89.8 (0.0)	48.7 (0.0)	27.3 (0.0)	21.0 (0.0)	19.6 (0.0)	82.0 (0.0)	50.4 (0.0)	28.7 (0.0)	19.8 (0.0)	16.0 (0.0)	22.0 (0.0)	22.6 (0.0)
MH	78.0 (1.0)	35.9 (0.9)	19.5 (0.9)	12.2 (1.0)	13.8 (1.0)	67.0 (0.9)	32.1 (1.1)	17.2 (0.9)	10.0 (1.2)	8.1 (1.1)	9.3 (1.1)	6.3 (1.4)

Figure 7.13: Building separating profiles with respect to the f^{min} fitness measure;

We provide a single set of comments on these results, since they are very similar.

In these two cases, the fitness of the separating profiles naturally decreases, although we find the same conclusions as in the case of the first fitness measure. The MIP approach performs the best, followed by the MH approach which brings small improvements to the solution given by SEL. Finally, the BLD approach performs worse than the others, with very large differences over certain types of problem instances.

Table 7.13: Average running times of methods for building separating profiles;

Approach	Time (s)					
	f^*		f		f^{min}	
SEL	0		0		0	
BLD	0		0		0	
MIP	18	(19)	139	(91)	203	(174)
MH	20		20		20	

Considering the average execution times of these approaches, as depicted in Table 7.13, conclude that selecting an existing alternative to be a separating profile is the best approach, regardless of the fitness measure used. The meta-heuristic approach provides very small improvements, while taking around 2 seconds to execute for each separating profile. The MIP approach, although is able to find the best results, is considerably disadvantaged by the long execution time it requires. Finally, the BLD approach does not seem well-suited for finding good separating profiles with respect to the three fitness measures that have been defined for them.

Summary

In this chapter, we have proposed several measures that may be used to characterise different sets of alternatives. Each of these measures has been defined based on the assumption that outranking relations are used to compare the alternatives together. Several methods have also been proposed for constructing each type of measure, furthermore considering a particular definition of an outranking relation. Each of these measures may be used to describe clustering results.

Central profiles may be used as representative alternatives, and replace those inside a cluster, while bounding profiles may be used when less compact clusters are considered. In this case, a central profile may not represent, with a high enough degree, the alternatives inside the cluster. Furthermore, separating profiles may be used in conjunction with a set of ordered clusters of alternatives.

We have performed a number of tests over a set of artificially constructed benchmarks in order to evaluate the proposed methods. The conclusions show that although the mathematical program approach outperforms the others in most cases, with respect to the quality of the results, the high execution time required by it makes it impractical for problem instances that contain more than 100 alternatives.

We have also shown that constructing central or separating profiles using an average operator over the evaluations of the alternatives inside each cluster results in these profiles being of very poor quality. In these cases, selecting an existing alternative as a central or separating profile is better suited. When considering the bounding profiles, building the upper and lower bounding profiles using the best and worst evaluations of the alternatives inside the cluster gives very good results in terms of quality, while the approach of selecting an existing alternative performs in this case worse. An exception may be made when considering the fitness measure which accounts only for the crisp credibility degrees of the outranking relations.

Finally, the meta-heuristic approach may be seen to increase the quality of the initial solution given by approach that selects the best fitting alternative, although in the case of the separating profiles the improvements are not significant.

We conclude by stating that the operational quality of the three types of profiles, and the fitness measures used to define them, will be in part explored in the last chapter of the thesis, while a more comprehensive analysis will be performed in a future work.

Chapter 8

Clustering for large MCDA applications

Contents

Abstract

In this chapter, we will consider the case of clustering large sets of alternatives in a MCDA context. We begin with a general introduction to this problem, describe the general limitations and challenges that come with the need to process large datasets, and then go over some of the more popular approaches to handling them. We continue with proposing several methods for clustering MCDA datasets in this context and conclude with some experimental results that highlight and compare the performance of these methods.

Due to the fact that the topic of clustering large datasets is very broad, through the proposed approaches and experimental results that follow, we only address a small part of this topic. Our incursion serves mostly to motivate further research around this topic, but also to set up a certain algorithmic framework, which will be used in the case study that concludes the thesis.

8.1 Clustering large datasets

8.1.1 General introduction

As seen up to this point, clustering in MCDA may be used to simplify the exploration of certain datasets, which may still be taken into consideration through a direct approach of analysing each individual element inside it and its interactions with the rest. In fact, many MCDA approaches aim at having an interaction between the one or several DMs and the analyst while discussing over the entire set of alternatives on which a decision needs to be taken. Such a process is used either during the initial stages of decision aiding when the preferences of the DM need to be extracted, or at the later ones when a recommendation is given and needs to be motivated.

Nevertheless, typical MCDA applications contain a number of alternatives around the order of 10, while if the number of criteria on which they are defined passes the value of 10, then the problem is considered to be difficult.

The size of MCDA datasets, compared to those from Data Mining, is therefore very small. Even 50 years ago, datasets that were considered large in Data Mining contained several thousands of objects [Jain et al. 1999], while nowadays large datasets contain even several millions.

Tackling, or even finding problems in MCDA where the number of alternatives is of the same order as that of the large datasets found in Data Mining may not be feasible. Still, the possibility of being confronted with decision problems where the number of alternatives is of the order of hundreds or even thousands may be very real. We will explore such a large dataset on which preferential information may easily be expressed in the following chapter.

8.1.2 Limitations and challenges

When dealing with large datasets several issues arise which need to be properly handled in order to be able to perform the desired operation on them, in this case clustering.

The first one represents the amount of time that is required for the algorithm to be able to give a result. Generally, algorithms which have a time complexity exponential in the number of objects in the dataset are considered inefficient even when the datasets are rather small. Usually such algorithms provide a good quality of the clustering result, however their use on large or even medium sized datasets becomes impractical. For instance, this was one the issues which required the use of a meta-heuristic approach in place of the exact one inside the second clustering approach presented in this work, the CLIP algorithm. In this case, the overall quality of the results was decreased in favour of a faster execution.

In Data Mining, when considering clustering of large datasets, anything having a complexity higher than linear is considered impractical. Furthermore, for very large datasets even this level is too high, and so the clustering approach is used only on a subset of the original dataset. The information extracted from this is then used over the entire dataset through a much simpler algorithm. The K-MEANS algorithm, having a time complexity linear in the size of the dataset, is often used for clustering large datasets, despite its limitations with respect to the quality of the end result.

Another issue related to handling large sets of data lies in the memory requirements of the clustering algorithm. For instance, the K-MEANS algorithm may only need to keep in memory the k centroids for each cluster, while agglomerative hierarchical approaches require that the similarity matrix be kept in memory, which is quadratic in size with respect to the number of objects in the dataset.

Following from the previous issue, the dimensionality of the dataset, or the number of attributes (or criteria) which describe each object, is also a factor to be considered. For the previously mentioned K-MEANS algorithm, which has very few memory requirements, having at each iteration to compute the distance between each object and the k centroids may lead to a considerable decrease in the performance of the algorithm is this operation requires a large amount of computation to be done. Nevertheless, in the MCDA context, where each criterion needs to reflect a certain point of view of the DM on the alternatives, which furthermore has a certain significance, issues related to the dimensionality of the data may very rarely occur.

The last issue to consider when clustering large datasets lies in the balance between the constraints placed on the clustering algorithm in order to make it time and memory efficient, and the detriment to the quality of the clustering result which follows from them.

8.1.3 Main approaches

The most popular approaches to clustering large sets of data can be divided into four categories [JAIN 2010]:

- Data summarization;
- Distributed computing;
- Incremental clustering;
- Sampling-based methods.

Several methods belonging to the first category are the BIRCH method [ZHANG et al. 1996] and also the divide-and-conquer approach from [STEINBACH et al. 2000]. In general, for this type of approaches, the initial dataset is split into a number of subsets, which are incrementally loaded into the main memory. Each subset is processed so that a reduced number of objects are generated from it. These objects, together with those generated from the other subsets, forming a smaller dataset, are then clustered using a standard clustering procedure. The first step can be several times, over the summaries of the previous dataset provided that the new dataset is still too large for the standard clustering approach which is used to produce the final result.

Approaches of this type are efficient with respect to the running time of the algorithm, as the problem is reduced to several smaller ones. Furthermore, the memory requirements are also reduced. However, the quality of the clustering results may be affected due to the way in which the dataset is split, and also following the summarization procedure, which may lose some of the original information.

The approaches from the distributed computing category contain a series of steps that can be performed in parallel, therefore speeding their overall running time by multiplying the computational resources that are used at each step. These approaches may provide more accurate clustering results, as they do not perform summaries on the original data, though a much larger number of resources need to be available.

Incremental clustering approaches only store the cluster representation information into the main memory, which is updated by considering each alternative at a time. A single pass over the dataset is usually performed, making such approaches efficient both with respect to the time complexity, but also with respect to the memory resources used. The overall quality of the clustering results may however be affected due to the order in which the objects are taken into consideration. The COBWEB algorithm [Fisher 1987] belongs to this category.

The last category of clustering approaches for large datasets take one or several samples out of the original dataset, which are then clustered. The information gathered from this is then used to cluster the original dataset, usually by a much simpler procedure which may involve a single pass through the dataset. The CURE algorithm [Guha et al. 1998] belongs to this category of clustering approaches.

8.2 MCDA clustering algorithms for large datasets

We continue by extending the MCDA clustering algorithms presented in this work so that large datasets of decision alternatives may be handled. For the four categories of large data clustering approaches three of them will be tackled further. These are the divide and conquer, incremental and clustering through sampling approaches.

8.2.1 A Divide and Conquer approach

The divide and conquer approach splits the larger problem into several smaller and much simpler ones. These problems can then be together solved in a much shorter time and with fewer computational resources. However, this gain in time and memory space is made with a trade-off in solution quality. The main steps of this approach are illustrated in Figure 8.1, and will be described further.

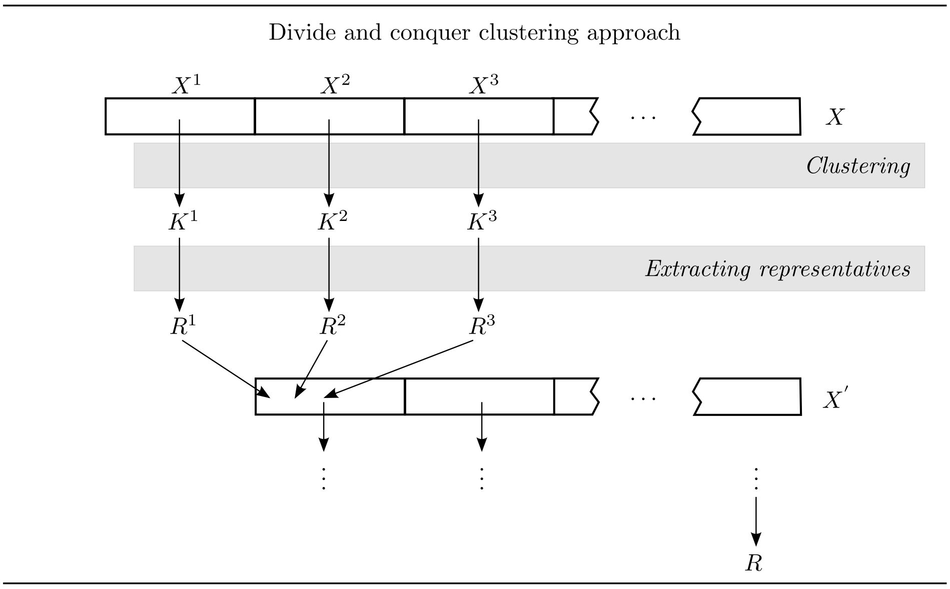


Figure 8.1: The divide and conquer approach;

Considering our clustering problem on a set of alternatives X containing n elements, we follow the algorithm described below:

1. **Split X into subsets of size $n^{'}$.**

   We denote these sets with $X^1, X^2, ..., X^p$, where $p = \lfloor \frac{n}{n^{'}} \rfloor$. The value for $n^{'}$ should be selected as an adequate dataset size for the considered clustering algorithm. In the case of the CLIP algorithm this values should be set around the value of 100, if the exact approach is used for the first step, while larger values of around 1000 may be considered if the approximative approach is used.

2. **Cluster each subset X^q into a set of partitions K^q, $\forall q \in 1..p$.**

   At this point the selected clustering algorithm is used on each subset of X.

3. **Construct a set of representatives R^q for each of the extracted sets of clusters K^q, $\forall q \in 1..p$.**

   We will use one of the methods presented in the previous chapter for constructing a central profile for each cluster. In the case of using an approach such the D-SOM algorithm, the cluster centroids can directly be used.

4. **Build a new dataset containing the all the generated representative alternatives.**

   X is substituted by $\bigcup\limits_{q \in 1..p} R^q$.

5. **Perform another sequence of the above steps or output the result.**

   If the size of the new dataset is larger than $n^{'}$ then we go back to Step 1. Otherwise X is clustered one more time using the selected algorithm, a central profile is constructed for each cluster and output as the final result.

The fitness of the clustering result, however, will only be reflected by the quality of the final representatives while considering one of the fitness functions describing a central profile from Equations (7.15), (7.16), (7.17).

If we would use the CLIP algorithm, with the objective of finding a partial tournament result at each stage, we denote this approach for handling large dataset with $\mathrm{CLIP}_{\mathrm{PT}}^{\mathrm{DC}_1}$. A similar denotation will be used for the other clustering objectives or clustering algorithms.

As the cluster sizes may vary, this approach can be further improved by placing a weight on each representative alternative equal in value to the number of alternatives it substitutes. For each subset $X^q, \forall q \in 1..p$ of X we therefore define a set $W^q = \{w_x \colon x \in K^q\}$ containing the weights of each alternative in the set. Initially all the weights are equal to 1. In Step 2, the clustering algorithm is slightly changed so that it takes into account these weights. Following this, after Step 3, where a representative is constructed for each of the generated clusters, a new weight is also computed for these representatives as follows:

$$w_{r_l^q} = \sum_{x \in K_l^q} w_x, \forall l \in 1..|K^q|, \forall q \in 1..p. \tag{8.1}$$

The algorithms using this approach will be denoted with a DC_2 superscript.

A further improvement can be made by also using the degree of approximation that each representative is able to achieve over the alternatives of the cluster to which it is assigned. As such, the weights for the cluster representatives are generated in the following way:

$$w_{r_l^q} = f(r_l^q, K_l^q) \cdot \sum_{x \in K_l^q} w_x, \forall l \in 1..|K^q|, \forall q \in 1..p. \tag{8.2}$$

The function f is selected as one of the fitness functions describing a central profile from Equations (7.15), (7.16), (7.17), which are further brought from the $[-1, 1]$ interval to the $[0, 1]$ one. The algorithms using this approach will be denoted with a DC_3 superscript.

The degree of quality of the final set of representatives can be computed as:

$$f(R) = \frac{\sum\limits_{x \in R} w_x}{n}. \tag{8.3}$$

8.2.2 Incremental clustering

The second approach that we will extend to clustering in the MCDA context is incremental clustering. Generally, only a model of the clustering result is stored in memory by this approach, and updated through considering incrementally each object in the dataset. The main steps of this approach are illustrated in Figure 8.2.

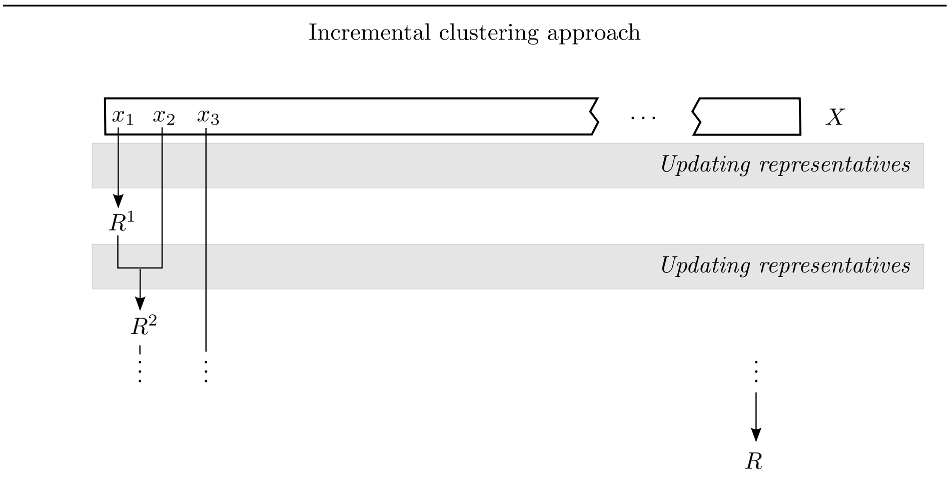


Figure 8.2: Incremental clustering approach;

We again consider a clustering problem on a set of n alternatives X and follow the algorithm described below:

1. **Initialization of the clustering model.**

   The first alternative x_1 is set as the first cluster representative r_1^1 and placed in R^1.

2. **For each alternative x_j, where $j \in 2..n$, update the clustering model.**

   i **Find the closest cluster representative.**

   The representative r_p^{j-1} from R^{j-1}, to which x_j is most indifferent, is selected.

$$p = \underset{q \in 1..|R^{j-1}|}{\arg\max}\ r(x_j \, \mathrm{I} \, r_q^{j-1}). \tag{8.4}$$

   ii **If x_j and r_p^{j-1} are indifferent and they can be merged, then build a new representative and go back to Step i**

   If x_j is indifferent to r_p^{j-1}, *i.e.* $r(x_j \, \mathrm{I} \, r_p^{j-1}) \geqslant \lambda$, then r' is built as the central profile for the set $\{r_p^{j-1}, x_j\}$. If the fitness of r' is high enough then remove the old representative from R^{j-1} and update x_j as r'.

   iii **Otherwise update the model and proceed to the next alternative.**

   $R^j = R^{j-1} \cup \{x_j\}$.

3. **Output the result.**

   R^n contains the final set of representative alternatives for the clusters in X.

We can improve the algorithm above by, like in the case of the previous approach, weighting each representative based on the number of alternatives that were used in its construction. Initially this weight will be 1 and it will be increased with 1 by each merging operation between a new alternative and an old representative. Furthermore, when merging two representatives that have come close to each other, the weight of the new representative will be equal to the sum of the two weights.

Another improvement may be to consider also the fitness with which a new representative is generated and to factor this into the weight given to it, just like in the case of the previous approach. We will denote the algorithms using the three variants of this approach with a INC$_1$, INC$_2$ or INC$_3$ superscript.

A second type of incremental clustering may be considered. This approach, whose steps are illustrated in Figure 8.3, takes increments of several alternatives at a time. We call this approach as incremental clustering using batches.

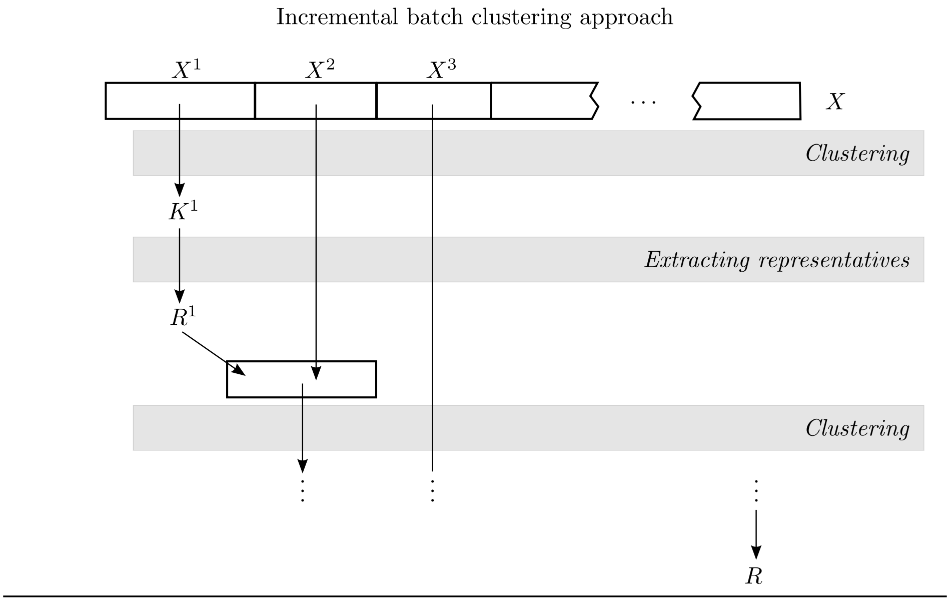

Incremental batch clustering approach
X^1
X^2
X^3
...
X
Clustering
K^1
Extracting representatives
R^1
Clustering
R

Figure 8.3: Incremental clustering approach using batches;

Considering the clustering problem on the set of n alternatives X, a batch size parameter $n^{'}$ is further given and the following sequence of steps is followed:

1. **Start with an empty model.**

   The set of initial representatives R^0 is defined as an empty set.

2. **Take a new batch.**

   A subset X^j is built from the old set of representatives R^{j-1} and the next batch of $n^{'} - |R^{j-1}|$ alternatives in X.

3. **Cluster the batch.**

   X^j is clustered into K^j using a preselected clustering algorithm.

4. **Construct the new representatives.**

   R^j is constructed as the set of the central profiles of the clusters in K^j.

5. **Go to Step 2 or output the result.**

   If there are elements left in X that were not yet considered then repeat the steps above starting from Step 2. Otherwise the algorithm has finished and the representatives in R^j are the end result.

As with the previous approaches, the representatives can be extended to take into account the number of alternatives that they represent. We do not detail this further as it has been described thoroughly before. The second approach of factoring in the fitness of each representative can further be employed. The algorithms for the three variants of this approach can then be denoted with a INCB$_1$, INCB$_2$ or INCB$_3$ superscript.

8.2.3 Clustering through samples

The last clustering approach for large datasets, similarly to the divide and conquer one, reduces the initial problem to a smaller one. However, where the divide and conquer approach splits the original problem in as many smaller ones so that the entire original dataset is covered, the approach that we will describe only considers one such reduced problem. Therefore a sample is drawn and only on this sample a clustering algorithm is used. The steps involved are highlighted in Figure 8.4 and described below:

Clustering approach using samples

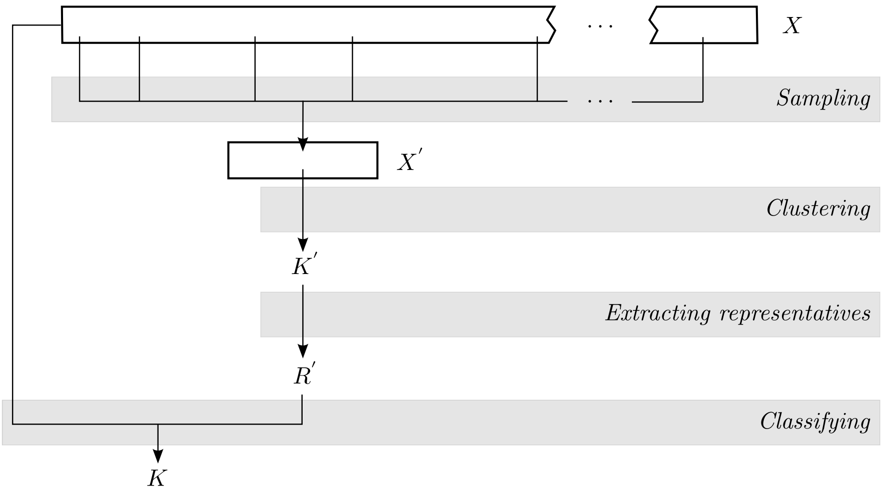


Figure 8.4: Approach of clustering using samples;

1. **Draw the sample.**

   A set of a given size $n^{'}$, $X^{'}$, is drawn from X.

2. **Cluster the sample.**

   $X^{'}$ is clustered using a given clustering algorithm into the partition $K^{'}$.

3. **Construct cluster representatives.**

   $R^{'}$ is constructed as the set of the central profiles of the clusters in $K^{'}$.

4. **Classify the original dataset.**

   The central profiles in $R^{'a}$ can be used in a simple classifier in order to split the original dataset X into $|R^{'}|$ clusters, which represent the end result.

It is rather easy to see that the efficiency of such an approach lies strongly in the way in which the sample is drawn. From the many existing sampling techniques, several that are suited to our approach are:

- simple random sampling;
- systematic sampling;
- stratified sampling.

The simple random sampling technique is straightforward, each element in the dataset having the same probability of being selected as any other. This approach is easy to implement and requires little computational effort, however, the degree of representation that the sample will hold over the entire dataset may be less than desired.

The systematic sampling approach initially orders based on some criteria the original data and then proceeds to drawing elements over fixed intervals. This approach is much more complex than the previous one, and so, building a criterion for globally assessing the quality of an alternative and then ordering them according to it may be infeasible for large datasets. Nevertheless, we will consider this sampling technique in the experiments that follow through the use of a simple utility function which initially scales all the alternatives evaluations on each criterion so that they are defined on the same interval and which then uses one-to-one trade-off between them. In other words, the score of an alternative will be the average of its evaluations on the scaled criteria.

The stratified sampling technique initially splits the data into categories, following certain criteria, and then draws the objects from each category with a probability proportional to the category size. Building such categories, which are furthermore disjoint, may prove however difficult, especially in our context. We will nevertheless use this approach by considering the following algorithm. Each criterion is divided into a given number of intervals which are proportional in size to the indifference and preference thresholds of the DM. Then one pass over the dataset is performed in order to count the number of evaluations that fall into each interval. Every alternative is then taken into consideration for taking part in the sample with a probability equal to the average percentage of the categories in which its evaluations are in, for each criterion.

The steps of clustering and building the representatives, in the algorithm described above, do not require too much consideration. One remark can however be made with respect to the nature of the representatives. We may additionally choose to use either bounding or separating profiles, the latter being used for cases where the clustering result is desired to be in an order structure.

The final classification step depends on the type of representatives used. In the case of central profiles, each alternative in the original dataset will be assigned to the cluster towards who's central profile the alternative is indifferent:

$$K_l \leftarrow K_l \cup \{x\}\colon\ r(r_l^{'}\,\mathrm{I}\,x) \geqslant \lambda, \forall l \in 1..|R^{'}|, \forall x \in X. \tag{8.5}$$

In this case, a fuzzy clustering may be produced, as an alternative may potentially be indifferent to several central profiles. Furthermore, the clustering may not be complete, as alternatives that are not indifferent to any profile will be considered as outliers.

In order to produce a complete partition, the assignment rule proposed above may be changed so that each alternative is placed in the cluster to who's central profile it is most indifferent to:

$$K_l \leftarrow K_l \cup \{x\}\colon\ l = \underset{l \in 1..|R^{'}|}{\arg\max}\, r(r_l^{'}\,\mathrm{I}\,x), \forall x \in X. \tag{8.6}$$

In the case of bounding profiles, an alternative will be assigned to a cluster through the following rule:

$$K_l \leftarrow K_l \cup \{x\}\colon\ r(r_l^{'+}\,\mathrm{S}\,x) \geqslant \lambda \text{ and } r(x\,\mathrm{S}\,r_l^{'-}) \geqslant \lambda, \forall l \in 1..|R^{'}|, \forall x \in X. \tag{8.7}$$

This approach will also produce a fuzzy partial clustering. Producing a complete partition may be achieved through the following:

$$K_l \leftarrow K_l \cup \{x\}\colon\ l = \underset{l \in 1..|R^{'}|}{\arg\max} \min\big(r(r_l^{'+}\,\mathrm{S}\,x), r(x\,\mathrm{S}\,r_l^{'-})\big), \forall x \in X. \tag{8.8}$$

Finally, using separating profiles will always produce an ordered partition over the set of alternatives using the rule below:

$$K_l \leftarrow K_l \cup \{x\}\colon\ l = \underset{l \in 1..|R^{'}|-1}{\arg\max}\ \min\big(r(r_l^{'\pm}\,\mathrm{S}\,x), r(x\,\mathrm{S}\,r_{l-1}^{'\pm})\big), \forall x \in X. \tag{8.9}$$

In the equation above, we consider that the $R^{'}$ set of separating profiles has been augmented with two fictive profiles, in order for the above formula to be properly defined. Therefore, a profile has been added at on the first position, containing the minimum values on each criterion, in order to bound below the bottom cluster, while another profile has been added on the last position, containing the maximum values on each criterion, in order to bound above the top cluster. This has been done only for notation purposes, and may be handled differently in the algorithm.

8.3 Validating the approaches

In this section we will present the results from several experiments performed over a set of artificially constructed benchmarks. These benchmarks are built in the same way as those from Chapter 6, but contain a larger number of alternatives.

8.3.1 Describing the experiments

The set of benchmarks that we will consider contains problem instances that have been built using one of the 12 different evaluations generators from Chapter 6, one of the 5 cluster sizes scenarios, and contain either $1,000$ or $5,000$ alternatives.

Following from the presented formulation of the approaches, we are only looking at this stage for a non-relational clustering result. Therefore we will only be analysing the three fitness measures for this clustering objective.

Since each approach divides the original problem into several smaller sub-problems, the clustering approach that will be used for solving these sub-problems is the CLIP_{NR} approach. We have selected this approach following the good results from Chapter 6. For the moment, we will only consider that the size of theses sub-problems is fixed to 100 alternatives, although we wish to explore several approaches for selecting this parameter in the future. The actual value selected in this case follows from the experiments from Chapter 6.

In order to limit the execution time of the algorithms, we set the first step of the CLIP approach to the approximative variant, while setting a total limit of 4 seconds to this algorithm for each clustering sub-problem. This has been done following the experiments from Chapter 6, where this time limit was observed to be more than sufficient for the algorithms to converge to a final solution. The second step of the CLIP approach will look to improve either one of the three fitness measures defined for non-relational clustering. The representatives that are being used in each of the four extensions to clustering large datasets will be constructed as central profiles. In this case too, we will be using an approximative approach to constructing them. A simulated annealing meta-heuristic, that aims at improving the fitness of the central profile, is used and limited to a 2 second execution time for each central profile. The fitness of the central profiles will depend on the selected fitness of the clustering result. The exact tuning of the time limits for the sub-problem clustering, and those for constructing central profiles will be explored in the future.

Considering the time limits, however, we consider that the approach of using samples will be the quickest, followed by the incremental approach using batches and the divide and conquer approach, while the slowest approach will most likely be the standard incremental approach. After analysing the execution times required by the four approaches, as previously defined, we may then adjust the time limits for the clustering sub-problems or the construction of the central profiles, so that they will all require approximatively the same time to execute.

8.3.2 Initial analysis

Let us begin by analysing the average execution times of the considered approaches. We will analyse each of the four types of approaches, and their three different variants, additionally considering looking for non-relational clustering results that are characterised by the f^*_{NR}, f_{NR} and f^{min}_{NR} fitness measures.

Table 8.1: Execution times in seconds over the benchmarks with 1,000 alternatives;

	SAMP$_1$	SAMP$_2$	SAMP$_3$	DC$_1$	DC$_2$	DC$_3$	INCB$_1$	INCB$_2$	INCB$_3$	INC$_1$	INC$_2$	INC$_3$
CLIP$^*_{\text{NR}}$	164.7 (5.6)	164.0 (6.1)	167.0 (5.4)	240.4 (39.5)	240.0 (40.1)	281.4 (54.1)	243.0 (38.6)	311.4 (40.1)	301.7 (44.5)	2031.1 (28.6)	184.9 (14.4)	112.5 (7.8)
CLIP$_{\text{NR}}$	351.8 (22.4)	193.2 (12.1)	183.6 (23.9)	760.8 (100.3)	637.8 (114.2)	636.6 (115.1)	752.2 (103.4)	635.8 (116.8)	633.0 (116.7)	2104.5 (22.5)	2114.3 (18.9)	2087.7 (33.8)
CLIP$^{min}_{\text{NR}}$	180.6 (7.8)	180.8 (8.3)	181.9 (7.7)	489.4 (88.4)	481.2 (81.5)	401.4 (38.9)	478.7 (85.3)	512.6 (84.4)	490.7 (71.4)	2047.2 (21.7)	2040.3 (24.9)	2040.7 (24.7)

When looking at the results from Table 8.1, which illustrate the average running times of the 36 approaches which were executed 20 times over each of the 300 benchmarks containing 1,000 alternatives, we find that indeed, the approaches of sampling the original dataset are much quicker than the others. The divide and conquer approaches and the incremental ones that cluster batches of the original dataset follow, while the classical incremental approach is last. These last approaches take the longest time to finish, which is due to the fact that they only work with central profiles, and therefore have more operations of building them than the other approaches. However, these execution times are rather stable, as noticed through the small standard deviations. We may also notice a general trend, over the first three types of approaches, of having the approaches that focus on the f_{NR} fitness measure take a longer time to finish. The approaches that focus on the f^{min}_{NR} fitness measure also take slightly longer than those that focus on f^*_{NR}.

Table 8.2: Execution times in seconds over the benchmarks with 5,000 alternatives;

	SAMP$_1$	SAMP$_2$	SAMP$_3$	DC$_1$	DC$_2$	DC$_3$	INCB$_1$	INCB$_2$	INCB$_3$	INC$_1$	INC$_2$	INC$_3$
CLIP$^*_{\text{NR}}$	3784.1 (48.0)	3777.5 (73.5)	3777.0 (8.7)	4181.9 (582.1)	3050.0 (214.2)	3126.2 (235.1)	3117.9 (207.4)	4562.0 (230.6)	4251.4 (264.3)	3874.2 (185.7)	3980.9 (135.4)	2113.7 (402.4)
CLIP$_{\text{NR}}$	4858.6 (1704.1)	3922.7 (164.0)	3245.6 (793.3)	6434.7 (1091.5)	6147.9 (589.2)	5945.5 (575.3)	6527.2 (1125.4)	6211.4 (607.4)	5960.9 (606.7)	13930.7 (182.5)	13485.8 (64.0)	12332.7 (15.5)
CLIP$^{min}_{\text{NR}}$	3771.2 (35.5)	3767.7 (64.2)	3030.8 (701.9)	5084.0 (228.9)	4075.0 (380.2)	4264.2 (326.2)	4171.9 (427.4)	5366.6 (343.0)	4783.8 (384.9)	12631.0 (21.4)	12324.5 (18.6)	12330.3 (18.4)

When considering larger benchmarks, we find in Table 8.2 that all of the remarks made over the smaller benchmarks hold. Furthermore, we notice that the execution times of the considered approaches are now closer to each other, which is due to the fact that the final step of all these clustering approaches takes the longest time to finish. This step takes the representative alternatives constructed by each approach and uses them to classify the original dataset. The very large number of outranking relations that need to be constructed at that point are the reason for the high execution times. For example, the first sampling approach only needs 4 seconds to cluster a randomly selected subset of alternatives, while the representatives of these clusters require 2 seconds each in order to be built. Therefore, in the worst case scenario where the sample of 100 alternatives yields 100 clusters, the execution time of the algorithm until the final classification step is only 204 seconds.

We continue by looking at the average number of clusters that are found by each method across all the considered benchmarks, in Table 8.3.

Table 8.3: Average number of clusters over the benchmarks with 1, 000 alternatives;

	SAMP$_1$	SAMP$_2$	SAMP$_3$	DC$_1$	DC$_2$	DC$_3$	INCB$_1$	INCB$_2$	INCB$_3$	INC$_1$	INC$_2$	INC$_3$
CLIP$^*_{\mathrm{NR}}$	19.6 (4.5)	19.9 (4.4)	19.5 (4.4)	24.6 (7.3)	24.7 (7.2)	20.9 (6.1)	21.4 (6.2)	20.0 (5.7)	20.1 (5.9)	40.9 (16.9)	39.5 (16.5)	38.9 (16.2)
CLIP$_{\mathrm{NR}}$	15.9 (3.9)	19.2 (4.5)	19.5 (5.6)	18.3 (7.1)	20.0 (7.4)	20.0 (7.3)	17.6 (5.3)	19.8 (6.3)	19.8 (6.4)	40.6 (16.7)	39.8 (16.5)	39.3 (16.6)
CLIP$^{min}_{\mathrm{NR}}$	13.2 (4.2)	13.5 (4.3)	13.2 (4.2)	18.0 (8.1)	17.9 (8.1)	10.4 (2.6)	18.3 (6.7)	16.2 (6.4)	15.6 (5.9)	31.3 (12.7)	30.0 (11.8)	29.4 (11.2)

We notice that the approaches that consider only a sample from the original dataset find overall the fewest clusters. This behaviour is expected due to the fact that the the approaches of drawing the sample may miss completely the alternatives from certain clusters. The divide and conquer approaches behave similarly to the incremental approaches that cluster batches of alternatives. Finally, the approaches that take increments of one alternative at a time find the highest number of clusters across their executions.

Although the benchmarks have been generated so that 10 clusters would be formed, the inclusion of the large performance differences, by certain benchmark generators, led to many of the alternatives inside these problems instances becoming incomparable to many others. As alternatives that are incomparable are generally not grouped together, this has led to an increase of the number of clusters.

8.3.3 Comparative analysis

We continue by focusing on the fitness of the results found by all of the considered approaches. We present a series of tables which display the results of each of the four types of clustering approaches for large datasets: SAMP, DC, INCB and INC. Inside these tables, the results of the nine different variants of each of the four major approaches are are highlighted using each of the three fitness measures for non-relational clustering: f^*_{NR}, f_{NR} and f^{min}_{NR}.

Only the results over the benchmarks containing 1, 000 alternatives are presented, although the results over those containing 5, 000 alternatives come very close to the ones below.

Table 8.4: Non-relational fitness results for all approaches over the benchmarks with 1,000 alternatives;

	f^*_{NR}			f_{NR}			f^{min}_{NR}		
	Samp$_1$	Samp$_2$	Samp$_3$	Samp$_1$	Samp$_2$	Samp$_3$	Samp$_1$	Samp$_2$	Samp$_3$
CLIP$^*_{\mathrm{NR}}$	90.5 (3.7)	90.6 (3.7)	90.5 (3.7)	81.9 (4.7)	81.9 (4.7)	81.9 (4.7)	0.9 (2.5)	1.2 (3.0)	0.8 (2.5)
CLIP$_{\mathrm{NR}}$	90.2 (3.9)	90.6 (3.8)	90.6 (3.8)	81.6 (4.8)	81.9 (4.8)	81.9 (4.8)	0.7 (2.5)	1.2 (3.3)	0.9 (2.8)
CLIP$^{min}_{\mathrm{NR}}$	89.1 (4.4)	89.2 (4.4)	89.1 (4.4)	81.0 (5.0)	81.0 (5.0)	81.0 (5.0)	0.7 (2.4)	0.8 (2.7)	0.6 (2.4)
	DC$_1$	DC$_2$	DC$_3$	DC$_1$	DC$_2$	DC$_3$	DC$_1$	DC$_2$	DC$_3$
CLIP$^*_{\mathrm{NR}}$	90.8 (3.7)	90.8 (3.7)	91.3 (4.0)	82.1 (4.8)	82.1 (4.8)	82.6 (4.7)	1.5 (3.5)	1.5 (3.5)	2.0 (4.4)
CLIP$_{\mathrm{NR}}$	90.3 (4.1)	90.7 (3.9)	90.7 (3.9)	81.7 (4.9)	81.9 (4.9)	81.9 (4.9)	1.7 (4.4)	1.7 (4.3)	1.7 (4.2)
CLIP$^{min}_{\mathrm{NR}}$	89.7 (4.5)	89.6 (4.5)	89.1 (7.3)	81.3 (5.1)	81.3 (5.1)	80.7 (7.3)	1.3 (3.8)	1.3 (3.9)	5.0 (7.1)
	IncB$_1$	IncB$_2$	IncB$_3$	IncB$_1$	IncB$_2$	IncB$_3$	IncB$_1$	IncB$_2$	IncB$_3$
CLIP$^*_{\mathrm{NR}}$	90.5 (3.8)	90.4 (3.8)	90.8 (3.8)	81.9 (4.8)	81.8 (4.8)	82.2 (4.8)	1.3 (3.2)	1.1 (3.0)	1.2 (3.2)
CLIP$_{\mathrm{NR}}$	90.3 (3.9)	90.5 (3.9)	90.5 (3.9)	81.7 (4.8)	81.9 (4.9)	81.9 (4.8)	1.4 (3.9)	1.4 (3.6)	1.4 (3.6)
CLIP$^{min}_{\mathrm{NR}}$	89.9 (4.2)	89.6 (4.4)	89.6 (4.7)	81.5 (4.9)	81.3 (5.0)	81.3 (5.1)	1.2 (3.2)	1.2 (3.4)	1.6 (3.9)
	Inc$_1$	Inc$_2$	Inc$_3$	Inc$_1$	Inc$_2$	Inc$_3$	Inc$_1$	Inc$_2$	Inc$_3$
CLIP$^*_{\mathrm{NR}}$	90.8 (3.2)	90.8 (3.2)	90.8 (3.9)	82.3 (4.8)	82.3 (4.6)	82.3 (4.6)	1.7 (3.5)	2.2 (4.3)	2.2 (4.1)
CLIP$_{\mathrm{NR}}$	90.9 (3.1)	90.8 (3.6)	90.9 (3.2)	82.3 (4.7)	82.2 (5.1)	82.3 (4.7)	1.7 (3.6)	2.7 (0.0)	1.8 (3.5)
CLIP$^{min}_{\mathrm{NR}}$	91.0 (3.5)	91.1 (3.5)	90.6 (3.1)	82.3 (5.1)	82.2 (4.7)	82.2 (4.7)	2.8 (4.1)	0.8 (2.3)	0.9 (2.4)

We begin by considering each of the four types of approaches independently of the rest, and finally conclude with several remarks across all of them.

In the case of the approach that first draws a sample of 100 alternatives, which it clusters and then extracts central profiles that are used for building the final result, we find that the results with respect to the f^*_{NR} fitness measure are quite high for all the considered variants of this approach. Although the approaches that focus on the f^{min}_{NR} fitness measure have a slightly smaller support reflected by the crisp credibility degrees of the relations between the alternatives, the difference in performance is only marginal. This statement may also be made when considering the non-relational fitness measures that takes into account the valued credibility degrees of these relations, either using an average or a min operators.

We may therefore conclude that no significant performance differences occur when using the three different sampling techniques, nor when clustering the sample and constructing the central profiles from it following either of the f^*_{NR}, f_{NR} or f^{min}_{NR} fitness measures. Taking into account the average execution times of these approaches, we may choose to select the CLIP$^{*\mathrm{Samp}_1}_{\mathrm{NR}}$ approach when looking for a non-relational clustering result over a large dataset in practice.

The divide and conquer approach splits the original dataset into subsets of 100 alternatives, which it clusters and reduces to the central profiles of these clusters. The results with respect to the first fitness measure highlight also in this case that the approaches looking to optimize the f^*_{NR} and f_{NR} fitness measures perform slightly better than the rest. Furthermore, the $\text{CLIP}^{*\text{DC}_3}_{\text{NR}}$ variant may be seen to perform slightly better than all the others. When considering the results with respect to the f_{NR} fitness measure, we may draw the same conclusions as before, with the addition that the approaches looking to optimize the f^*_{NR} perform slightly better than the ones that try optimize the illustrated fitness measure. $\text{CLIP}^{*\text{DC}_3}_{\text{NR}}$ performs better than the other approaches when considering this fitness measure too. Finally, when considering the f^{min}_{NR} fitness measure, the $\text{CLIP}^{min\text{DC}_3}_{\text{NR}}$ perform better than the others on average, although its standard deviation is higher.

The approach of incrementally clustering a batch of 100 alternatives, along with the central profiles of the clusters from the previous batch, may be seen to perform very similarly to the divide and conquer approach. This is due to the similarities between the two approaches. All of the remarks from the divide and conquer approaches apply in this case too.

The last type of approaches, those that consider each alternative at a time incrementally, may be seen to perform marginally better than the others, in most cases, with respect to every fitness measure. Between themselves, we again find very small performance differences.

Looking at all of the presented results, we may conclude that, although several approaches may be seen to perform slightly better than others, all of them show rather good overall results with respect to all of the presented fitness measures.

The illustrated experimental tests of these approaches should be complemented by further research. The size of the samples, as well as the execution times given to each sub-problem should be analysed further. Additionally, larger problem instances could easily be considered by making the sub-problems tackled by the presented approaches run in parallel on multiple processing units. Furthermore, making the final step of classifying the entire dataset using the representatives extracted by each approach also parallel would greatly reduce the required execution time of the algorithms. The analysis of the quality of the results could also benefit from splitting it into several independent parts and using multiple processing units in order to speed it up. At the moment, the final classification and the evaluation of the results are responsible for the largest overheads in the presented experiments.

Summary

In this chapter, we have presented several extensions of the algorithms from the second part of the thesis to clustering large sets of alternatives. Although we have only scratched the surface of this topic, we have been able to illustrate an overall good performance of the proposed approaches when a non-relational clustering result is desired.

We consider that the different variants of the algorithms may prove to perform differently when stricter constraints are placed on the execution times of the algorithms for each sub-problem. At the moment, it seems that a sample of 10% of the original dataset is sufficient in order to construct a good cluster representation from it, which models the structure of the entire dataset.

The topic of relational and ordered clustering will be addressed in the future, which we consider that will benefit from the use of the other types of profiles that we have presented in the previous chapter.

With the foundation for clustering large MCDA datasets laid, we continue to analysing a large dataset containing real-life data.

Chapter 9

Case Study: Mining Toxics Release Inventory data

Contents

Abstract

We present a case study, containing real-life data over a large number of toxic chemical release practices of facilities from the United States of America. This data is freely and easily accessible.

The purpose of this study is to apply some of the proposed approaches on real data, and extract important information on the usefulness of the presented work but also with respect to issues that might need to be addressed further. Therefore, we construct a fictitious problem around this data, apply some of the proposed MCDA clustering approaches, and analyse in detail one of the results.

9.1 Describing the available data

In order to test the approaches presented in this paper, we have selected to use as a case study data procured from the U.S. Environmental Protection Agency's (EPA) Toxics Release Inventory (TRI) program, in order to asses the practices of different facilities that handle toxic chemicals. The TRI program has collected and made publicly available data on more than 53,000 facilities with respect to the handling of different chemicals that are considered hazardous, during more than 25 years of activity. This data is easily accessible though a variety of tools found at [USEPAa].

Before detailing the selected data sources, we first need to consider the following factors. Most of the available data is in the form of annual quantities of certain chemicals that are released from a facility into the environment, or managed in one way or another. This information may be used to determine the different priorities that facilities need to take based on the quantities of their emissions of toxic chemicals. Also, the trends across different years can be used for analysing the evolution of their practices with respect to these chemicals.

In order to properly analyse the TRI data, the toxicity of the covered chemicals needs to be accounted for. Small quantities of highly toxic chemicals might, in some cases, be considered more hazardous than large quantities of less toxic chemicals, or vice versa. Furthermore, the environment around the facility and the type of exposure of the considered chemical need to be factored in. This information is not readily accessible through the previously mentioned release forms and therefore needs to be extracted from a different source.

9.1.1 Selecting the data on chemicals toxicity

As of the beginning of 2013, 593 chemicals and 30 chemical categories are monitored by the TRI program. From the 30 chemical categories, three of them contain 62 specifically listed chemicals, therefore totalling 682 entries. In the case of each of them, a considerable amount of information is available from different sources. The TRI-CHIP tool [USEPAb] may be used to easily gather this information from several sources at once.

The chemicals may initially be evaluated considering four major classes of effects that they may produce:

- **Acute**: the chemical is associated to acute diseases in humans;
- **Chronic**: the chemical is associated to chronic diseases in humans;
- **Cancer**: the chemical is associated to cancerous diseases in humans;
- **Environmental**: the chemical is hazardous for the environment.

The first three correspond to effects pertaining to humans, while the last towards the environment.

In the case of human-related diseases, for each chemical different experiments on lab animals are listed, detailing the dosage used, administration type and effects. Measurements linking each chemical with a particular human affection are also documented.

In the case of the environmental effects, information on the amounts of toxic chemicals found in different animal species are also gathered.

As it is rather difficult to asses the toxicity of a chemical from this information, in the absence of one or several experts in the field, an indicator is also provided for each of the four classes of effects. These indicators are binary, and denote the fact that a particular chemical is associated with either one of the four classes of effects.

We will be using these indicators further in assessing the level of toxicity of each of the chemicals monitored by TRI.

9.1.2 Selecting the data on facility practices

The data gathered by the TRI program on the practices of the facilities with respect to the toxic chemicals monitored is generally in the form of quantities that were disposed, recycled, converted to energy or treated. In Figure 9.1 we have an overview of the different types of information available.

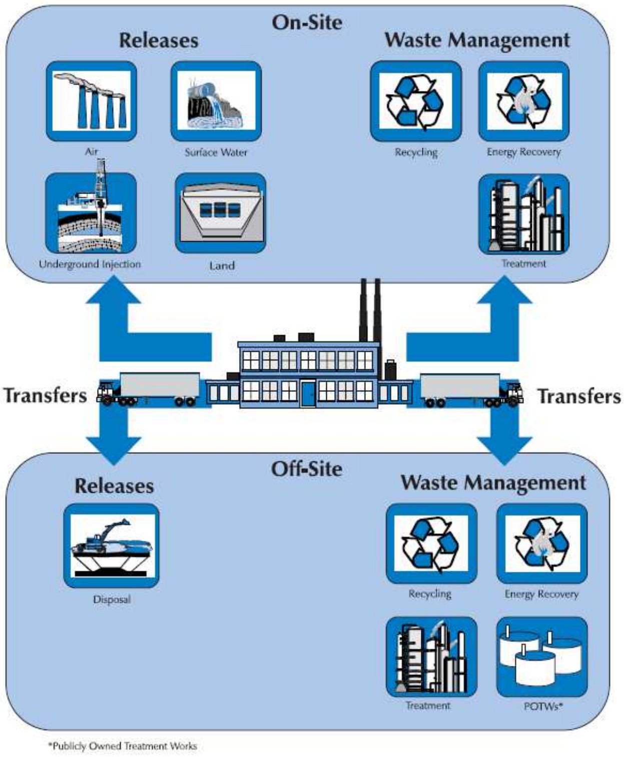


Figure 9.1: Release and waste management data available in TRI [USEPAa];

We find that there are two ways in which this information is divided. Firstly, the information is split between on-site practices and those that are performed off-site. Secondly, both the on-site and off-site information is divided between releases of the toxic chemical into the environment, in different ways, and the waste management operations.

The data related to the release and waste management of TRI reporting facilities is gathered by means of forms that are filled in annually by each facility. The form is divided in two parts, one containing the facility identification information and the second containing the chemical specific information. We will focus only on the second part which contains the information we are more interested in. The structure is as follows:

Form R, Section 2:

1. **Toxic Chemical Identity**
   - (a) **CAS Number** - chemical identifier;
   - (b) **Toxic Chemical or Chemical Category Name**
   - (c) **Generic Chemical Name**
   - (d) **Distribution of Each Member of the Dioxin and Dioxin like Compounds Category**: indicators for Dioxin and Dioxin-like compounds from 1 to 17;
2. **Mixture Component Identity**
   - (a) **Supplier Provided Generic Chemical Name**: indicator for use of generic chemical name;
3. **Activities and Uses of the Toxic Chemical**
   - (a) **Manufacture the Toxic Chemical**
     - **Produce**: indicator stating that the chemical is produced on-site;
     - **Import**: indicator stating that the chemical is imported;
     - **On-Site Use/Processing**: indicator stating that the chemical used or processed on-site;
     - **Sale/Distribution**: indicator stating that the chemical is sold or distributed;
     - **By-product**: indicator stating that the chemical is a by-product;
     - **Impurity**: indicator stating that the chemical is a manufacturing impurity;
   - (b) **Process the Toxic Chemical**
     - **Reactant**: indicator stating that the chemical is used as reactant;
     - **Formulation Component**: indicator stating that the chemical is a formulation component;
     - **Article Component**: indicator stating that the chemical is an article component;
     - **Repackaging**: indicator stating that the chemical is repackaged;
     - **Impurity**: indicator stating that the chemical is a processing impurity;
   - (c) **Otherwise Use the Toxic Chemical**
     - **Chemical Processing Aid**: indicator stating that the chemical is used as processing aid;
     - **Manufacturing Aid**: indicator stating that the chemical is used as manufacturing aid;
     - **Ancillary or Other Use**: indicator stating that the chemical is used in other ways;
4. **Maximum Amount of the Toxic Chemical On-site During the Calendar Year**
5. **Quantity of the Toxic Chemical Entering each Environmental Medium Onsite**
   - (a) **Fugitive or Non-Point Air Emissions**: amount released as fugitive emissions;
   - (b) **Stack or Point Air Emissions**: amount released as stack emissions;
   - (c) **Discharges to Receiving Streams or Water Bodies**: amount released in surface water;
   - (d) **Underground Injection On-site to Class I Wells**: amount released in Class I Underground Injection Wells;
   - (e) **Underground Injection On-site to Class II-V Wells**: amount released in Class II-V Underground Injection Wells;
   - (f) **Disposal to Land On-site**
     - **RCRA Subtitle C Landfills**: amount disposed in regulated landfills;
     - **Other Landfills**: amount disposed in other landfills;
     - **Land Treatment/Application Farming**: amount used in land treatment/farming;
     - **RCRA Subtitle C Surface Impoundments**: surface impoundments for regulated landfills;
     - **Other Surface Impoundments**: surface impoundments for other landfills;
     - **Other Disposal**: other land releases;
6. **Transfers of the Toxic Chemical in Wastes to Off-Site Locations**
   - (a) **Discharges to Publicly Owned Treatment Works (POTWs)**
     - **Total Quantity Transferred to POTWs**: amount transferred to POTWs;
     - **POTW Locations**;
   - (b) **Transfers to other Off-Site Locations**: locations of other off-site locations and amounts transferred;

7A **On-Site Waste Treatment Methods and Efficiency**: processes used for waste treatment;
   (a) **Waste Stream** : type of waste;
   (b) **Waste Treatment Method**;

7B **On-Site Energy Recovery Processes**: processes used for energy recovery;

7C **On-Site Recycling Processes**: processes used for recycling;

8 **Source Reduction and Recycling Activities**
   (a) **Disposals**
      – **Total on-site disposal to Class I Underground Injection Wells, RCRA Subtitle C landfills, and other landfills**
      – **Total other on-site disposal or other releases**
      – **Total off-site disposal to Class I Underground Injection Wells, RCRA Subtitle C landfills, and other landfills**
      – **Total other off-site disposal or other releases**
   (b) **Quantity Used for Energy Recovery On-site**
   (c) **Quantity Used for Energy Recovery Off-site**
   (d) **Quantity Recycled On-site**
   (e) **Quantity Recycled Off-site**
   (f) **Quantity Treated On-site**
   (g) **Quantity Treated Off-site**
   (h) **One-Time Event Release**: one-time release amount;
   (i) **Production Ratio**
   (j) **Source Reduction Activities**: list of activities for reducing chemical production;
   (k) **Additional Data Indicator**: indicator for additionally provided information;

As the information provided in these forms is rather extensive, we will extract the relevant information following the schema in Figure 9.1. We will consider that both on-site and off-site releases and waste management operations may be used together in order to characterise the quality of the practices of the facility with respect to the handled chemicals. Furthermore, the amounts released into the environment may characterise negatively these practices, while the waste management operations may be used to characterise them positively. Hence we perform this division and select the following information from the reporting forms:

1. **Releases**:
   - **Air Total Release**: the release estimate (in pounds) reported by the facility as air releases;
   - **Water Total Release**: the release estimate (in pounds) reported by the facility as water releases;
   - **Land Total Release**: the release estimate (in pounds) reported by the facility as landfill releases;
   - **Underground Injection Total Release**: the release estimate (in pounds) reported by the facility as underground injection releases;
   - **Off-site Release**: the release estimate (in pounds) reported by the facility as off-site releases.
2. **Waste management**:
   - **Recycling on and off site**: the estimate (in pounds) reported by the facility as recycled on and off site;
   - **Energy conversion on and off site**: the estimate (in pounds) reported by the facility as converted to energy on and off site;
   - **Treated on and off site**: the estimate (in pounds) reported by the facility as treated on and off site;
   - **POTW Total Transfer**: the total amount (in pounds) of the toxic chemical transferred from the facility to an off-site location or to POTWs.

We have decided to consider only the total amounts that are released via each of the documented mediums, in order to reduce the complexity of our model. At a later stage we may be able to complement the model with additional information that is present in the original data, in order to perform a more detailed analysis.

We have also chosen in the case of waste management practices to consider both the on-site and off-site amounts of materials handled in each of the four mentioned ways, as this differentiation does not provide any additional qualitative information. We consider that there is no basis for stating that a facility that sends toxic material to be recycled off-site, for instance, has a worse practice than one that recycles the same amount of material on-site, or vice versa.

As the information described above contains annual summaries with respect to the facility practices, we will also restrict our analysis to the data coming from only one year at a time. We have selected the year 2010, so that the information that we will analyse is recent, but also ensuring that the reporting stage has finalized.

9.1.3 Analysing the selected data

The chemicals toxicity data

Starting with the analysis of the data on the toxicity of the chemicals monitored by TRI, we find that from the total of 682 entries, only 396 contain complete information across all of the four indicators (Acute, Chronic, Cancer and Environmental). The vast majority of the chemicals containing missing data hold no information on any indicator, while three of them have missing data on only one. Additional data may be gathered from other data sources, however, in the absence of an expert, we have decided to consider further only those chemicals on which the TRI database contains complete information related to the four indicators.

As this list of chemical has been gathered at the beginning of 2013, while the data on the facility practices is taken from 2010, we have further reduced the set of chemicals to those on which reports have been made in 2010, which are 242 in number.

In Table 9.1 the distributions of the 4 hazard indicators is displayed.

Table 9.1: Distribution of the 242 chemicals reported in 2010 over the 4 hazard indicators;

Indicators	Frequency	
Acute	18	7.4%
Chronic	182	75.2%
Cancer	64	26.4%
Environmental	83	34.3%

It may be noticed that a very large majority of three quarters the chemicals that were reported in 2010 are associated with chronic affections in humans, followed by one third which are known to cause environmental issues, a quarter being associated with different types of cancers while the fewest of them being known to cause acute affections.

We follow with a more detailed analysis of the chemicals with Table 9.2 showing the distributions of all the combinations between the four indicators found in the 2010 reported chemicals data set.

Table 9.2: Distribution of the 242 chemicals reported in 2010 over the existing combinations of the 4 hazard indicators;

Indicators	Frequency	
Chronic	110	45.4%
Chronic + Environmental	35	14.4%
Chronic + Cancer + Environmental	29	11.9%
Cancer	26	10.7%
Environmental	15	6.1%
Acute	15	6.1%
Chronic + Cancer	7	2.8%
Cancer + Environmental	2	0.8%
Acute + Environmental	2	0.8%
Acute + Chronic	1	0.4%
	242	100%

We notice that almost half of the chemicals are associated only with chronic diseases. Then we have almost a sixth of the chemicals being associated with both chronic diseases and environmental hazards, an eight which add cancer on the list, while the following 10% are associated only with cancer. Chemicals that produce only environmental hazards and no human related diseases are then tied at 6% with those chemicals that produce only acute diseases for humans. Finally, very few chemicals combine the chronic and cancer indicators, cancer and environmental, acute and environmental and lastly acute and chronic. A few combinations of the 4 indicators do not appear at all among the studied chemicals.

The release and waste management data

We continue with the analysis of the data extracted from the reporting forms from 2010. A total of 80,366 reports have been pulled from the TRI database. As each report provides information on only one chemical managed by a particular facility, each facilities may appear in several reports at once.

A total of 530 reports were found to contain duplicate entries, in other words, certain facilities filled in multiple reports during 2010 for the same chemical. As the reported information from them was considerably different between the duplicate reports, and due to the lack of any additional knowledge on the reason for their existence, all these entries were removed, leaving a total of 79,105 entries.

We have additionally removed all the reports pertaining to the chemicals for which we were not able to find complete toxicity information, as described previously. Furthermore, those entries where the total amount of both released and managed chemical material was below 0.5 pounds were removed from the dataset as they do not provide sufficient information on the quality of the practices of the facilities.

The final dataset thus contains a total of 22,676 entries and will be analysed further.

We present in Table 9.3 information on the distribution of each of the Air, Water, Land, Underground Injection and Off-site release attributes.

Table 9.3: Distribution of the values of the Release Attributes;

Attributes	Measures (pounds)					
	Minimum	First quartile	Median	Third quartile	Maximum	Average
Air	0	0	11	320.5	7,500,000	8,417.2
Water	0	0	0	0	23,517,962	9,236.5
Land	0	0	0	0	453,658,500	47,177.7
Und. Inj.	0	0	0	0	17,000,000	3,378.1
Off-site	0	0	0	255	13,507,700	14,675.1

We find that for each of the considered attributes, there are many facilities that do not release chemicals via that particular medium.

In the case of the releases to air, at least a quarter of the reports record no such releases, while half of them, between the first quartile and the third, send out amounts of a few hundred pounds per year at most. However, a few reports show very large amounts of toxic chemicals that were released into the atmosphere in 2010, with a maximum amount of 7.5 million pounds.

Looking at the second line in the table above, we find that three quarters of the reports do not describe any releases to the surface water, although again, in some of them very large quantities are again reported. This is also the case for landfill releases where the largest reported amount describe almost half a million pounds of chemicals being disposed in such a manner, but also for underground injections, which record fewer overall quantities as in the previous case.

At least half of the reports do not document any off-site releases, though we find that a large number of the rest document such disposals, as seen through the third quantile of this attribute.

Due to the fact that many reports document zero releases of certain chemicals in each of the 5 mediums that are considered, we extend the analysis to highlighting the number of reports which display non-zero releases for each of them.

Looking at Table 9.4 we notice that many reports document releases to air. The releases to other sites follow, then those to water bodies and landfills, while the releases to underground injection wells are the fewest.

If we would also be interested in seeing the interactions between the five attributes, when non-zero emissions are considered we can find these in Table 9.5.

From this table we notice that slightly more than a quarter of the reports contain information on chemicals that are solely released into the atmosphere, while those

Table 9.4: Distribution of the reports over the existing combinations of the 5 Release Attributes with non-zero amounts reported;

Non-zero release attributes	Frequency	
Air	15479	68%
Off-site	9890	43.4%
Water	4773	21%
Land	3372	14.8%
Underground Injection	177	0.8%

Table 9.5: Distribution of the reports over the existing combinations of the 5 Release Attributes with non-zero amounts reported;

Non-zero release attributes	Frequency	
Air	6313	27.7%
Air + Off-site	4073	17.9%
-	3169	13.9%
Off-site	2529	11.1%
Air + Water + Off-site	1674	7.3%
Air + Land	961	4.2%
Air + Water + Land	816	3.6%
Water	588	2.6%
Air + Water + Land + Off-site	560	2.4%
Air + Water	519	2.3%
every other combination	< 500	< 2%

that are released both into air and to other sites follow. The third line depicts the number of reports where the toxic chemical considered is not at all released into the environment. However, in these cases, the facilities do in fact manage the chemicals through waste management approaches. The reports which document chemical releases off-site are next, followed by reports documenting a few combinations between air, water, land and off-site releases, which are less frequent than the ones already mentioned.

Turning our attention to the attributes involving waste management, we find the distribution of the amounts on each of the four attributes in Table 9.6.

Table 9.6: Distribution of the values of the Waste Management Attributes;

Attributes	Measures (pounds)					
	Minimum	First quartile	Median	Third quartile	Maximum	Average
Recycling	0	0	0	6,412.5	526,355,320	102,120
POTW	0	0	0	0	99,786	37.5
Energy	0	0	0	0	45,378,000	17,761.5
Treatment	0	0	0	0	80,000,000	71,525.1

Just like in the case of the attributes involving chemical releases, the ones revolving around the waste management approaches have a large distribution of values equal to 0. In the case of the recycling attribute, the distribution is not as steep as for the others, but also the maximum amount of reported recycled material is larger than the maximum amounts handled in the other manners. The maximum amounts of chemicals sent to publicly owned waste treatment facilities is the smallest among all the attributes.

Table 9.7: Distribution of the reports over the existing combinations of the 4 Waste Management Attributes with non-zero amounts reported;

Non-zero release attributes	Frequency	
Recycling	9099	40%
Treatment	4966	21.8%
POTW	3018	13.2%
Energy	1313	5.7%

When considering Table 9.7, we find that most toxics release activities are also done in conjunction with recycling practices. Procedures of treatment follow for roughly 20% of the release activities, while energy conversion is present in the fewest number of releases.

Table 9.8: Distribution of the reports over the existing combinations of the 4 Waste Management Attributes with non-zero amounts reported;

Non-zero release attributes	Frequency	
-	7904	34.7%
Recycling	6372	28%
Treatment	3714	16.3%
Recycling + POTW	1946	8.5%
POTW	1018	4.5%
Energy + Treatment	543	2.4%
every other combination	< 500	< 2%

If we additionally consider the distribution of all the combinations of waste management practices, we notice, in Table 9.8, that one third of the release activities are not accompanied by any waste management action. Around 28% of the toxic chemicals release activities are accompanied by measures of recycling, while 16% only by measures of prior treatment of the chemicals involved. Roughly 8% of the release activities combine recycling measures with disposal through Publicly Owned Treatment Works, while all the remaining combinations of waste management activities are found in less than 5% of the release activities.

9.2 Defining the problem

Having completed a detailed analysis of the available data on the toxic release activities of 21,064 facilities from the U.S.A. during the year 2010, we proceed to defining a problem which would require the use of MCDA clustering approaches in order to be solved.

For the moment, the problem is a fictive one, although we may easily imagine similar problems to occur in real-life.

9.2.1 Formulating the problem

We consider the problem of exploring the practices of facilities that release toxic chemicals, with respect to the potential negative impact that these practices may have on the environment and on the human population.

Due to the fact that certain facilities release more than one toxic chemical, we will consider each release procedure of one toxic chemical by a certain facility as the fundamental objects of the problem. These objects will also be referred to as alternatives, and are denoted with X. Each facility may then later be assessed, if necessary, based on the evaluation of the release processes in which it was involved.

We consider that grouping the facilities, according to potential impact their chemical release practices may have on the environment, could be used when certain measures are desired to be taken, depending on this level of impact. We also assume that no clear guidelines exist for taking these measures, therefore we cannot define initially several characteristics for the different chemical release practices, which may then be used in conjunction with classification approaches. Furthermore, if classification approaches would be used, without sufficient a priori information certain classes may be empty.

In this context, we imagine that the interested authorities would initially analyse several ways in which the chemical release practices may be grouped, furthermore considering certain preferential information they may have on them. Following this, the different classes of facility practices may be identified, along with the measures needed in order to reduce the impact they have on the environment. Furthermore, depending on the number of elements in each class, the measures may change, *e.g.* implementing measures of a particular type for only a few recorded practices may not be well motivated, as opposed to having larger numbers of these practices recorded.

Considering the information that is available on each of the chemical release practices, as presented in the first section, we define the following three objectives that we consider that a facility should aim for, in order to lower the potential negative impact on the environment:

- handle less toxic chemicals;
- release fewer amounts of toxic chemicals;
- process larger amounts of toxic chemicals through waste management procedures.

It is straightforward that a facility that handles chemicals that are more toxic than those handled by another, poses a higher risk of having a negative impact on the environment. Although this is linked with a certain degree to the amount of the toxic chemical that is release, and furthermore, depending on the type of chemical, it is also linked to the medium into which this substance is release. Due to the unavailability, for the moment, of such information, we will not consider these interactions in our problem. Nevertheless, the potential exposure of the facility personnel, the risk of exposure as a result of force-of-nature event, make the toxicity of the handled chemical independent of the other two objectives.

The amount of toxic chemical that is released is also directly proportional to the negative environmental impact that the practice of releasing it has. We may again argue that the added risk of these releases is dependent on the toxicity of the chemical, and on the release medium, however, as this information is not available to us for the moment, we will consider it independent of the rest.

Finally, the amount of toxic chemical that is processed in order to neutralize it, or reduce its environmental risk, is indirectly proportional to the potential negative environmental impact. The type of process used is furthermore important in assessing the mitigating effect of these practices, however, it is again heavily dependent on the type of chemical processed. Processing larger quantities may also be viewed as a commitment of the facility to mitigating against the negative effect that their practices may produce to the environment, therefore, this objective may be considered as independent of the others from this perspective.

Therefore, a process of releasing a small amount of a weakly toxic chemical into the environment, while further processing large amounts of it in order to mitigate against its potential negative effects on the environment, may be considered as a good practice with respect to the potential negative impact it may have on the environment.

A process of releasing a large amount of a highly toxic chemical, while little processing is done in order to mitigate against the environmental risks, may be considered as a bad practice with respect to the potential negative impact it may have on the environment.

9.2.2 Structuring the problem

We follow with structuring the available data on the toxic chemicals release practices, according to the formulation of the problem.

We begin by assigning each of the 4 chemical toxicity degrees indicators, and the 9 selected chemical release amounts, to the 3 previously defined objectives. We name these objectives as **Toxicity**, **Releases** and **Mitigation**, and define for each several criteria, which are constructed from the 13 attributes. We present this structure in Table 9.9

Only the toxicity indicators have been split in two criteria, one indicator the degree of toxicity to humans, and the other to the environment, which in turn are assigned to the **Toxicity** objective. The released amounts, vie different mediums, have been assigned to a single criterion under the **Releases** objective, while the mitigated amounts have been similarly grouped under a single criterion under the **Mitigation** objective. These assignments are straightforward.

Although the three objectives are considered to be independent of each other, when considering the overall objective of minimizing the negative impact of the toxic chemical releases into the environment, the attributes that have been assigned to

Table 9.9: Structuring the problem;

Objectives	Criteria	Attributes
Toxicity (T)	Toxicity to Humans (TH)	Cancer Indicator (CAN)
		Chronic Indicator (CHR)
		Acute Indicator (ACU)
	Toxicity to Environment (TE)	Environmental Indicator (ENV)
Releases (R)	Release Amounts (RA)	Air Release Amounts (AIR)
		Water Release Amounts (WAT)
		Land Release Amounts (LAN)
		Underground Injection Amounts (UND)
		Off-site Release Amounts (OF)
Mitigation (M)	Mitigated Amounts (MA)	Recycled Amounts (REC)
		Transferred to POTW Amounts (POT)
		Converted to Energy Amounts (ENE)
		Treated Amounts (TRE)

each of them are not. The criteria we have highlighted group the attributes that are not independent of each other, but that are independent of the rest, so that the criteria are also independent of each other.

For the purpose of reducing the number of assumptions that we make on the considered problem, we wish to construct the set of criteria so that potential interactions between them would not be required to be considered.

As the first three indicators, CAN, CHR and ACU, together refer to different human ailments, they each express a certain degree of health risk to the human population. We construct the **Toxicity to Humans** criterion, by considering the following order between them:

$$\text{CAN} \succ \text{CHR} \succ \text{ACU}$$

This order is motivated by the fact that we consider toxic chemicals that are known to cause cancerous diseases to be more dangerous than those that cause chronic diseases, while in turn both of them are considered more dangerous than substances that lead to acute diseases.

The three attributes, and the order between them, form a lexicographic order, therefore two alternatives are first compared using the CAN indicator. If this indicator is not sufficient to discriminate between the two alternatives, *i.e.* the evaluations are indifferent, then the following indicator, CHR, is considered. This process is repeated if the second indicator is also not sufficient for discriminating between the alternatives.

In the case of the **Toxicity to the Environment** criterion, he process of building it is straightforward, as only one attribute is assigned to it.

The **Release Amounts** criterion is constructed in the same way as the **Toxicity to Humans** criterion, considering the following order between the attributes that are assigned to it:

$$\text{AIR} \succ \text{WAT} \succ \text{LAN} \succ \text{UND} \succ \text{OFF}$$

The order above is motivated by the spread possibilities that each medium offers to the released toxic chemical. Chemicals that are release into the atmosphere have the largest spread potential, with very little possibilities for containment. Releases to water bodies also have a good spread potential, although slightly less air releases. Releases to landfills are next, while underground injection practices may be considered to restrict the spread of the toxic chemical better. Finally, off-site releases are last, due to the fact that through this approach, the released chemical is potentially further processed in order to reduce environmental impact.

Finally, the last four attributes are used to construct the **Mitigated Amounts** criterion, which is built in the same way as above, considering the following order between the attributes:

$$\text{REC} \succ \text{POT} \succ \text{ENE} \succ \text{TRE}$$

Recycling can be clearly seen as one of the best mitigation practice, as the indicated amounts of toxic chemicals are completely reused. The amounts transferred to Publicly Owned Treatment Works are also a good practice, while the amounts converted to energy leave only a few percentages of the original amount to be released into the environment. Finally, the last attribute reflects the amounts that have been treated in order to reduce their negative environmental effects, however, some negative effects may remain.

9.2.3 The preference model

Following the construction of 4 criteria, $F = \{\text{TH}, \text{TE}, \text{RA}, \text{MA}\}$, under the 3 objectives, we proceed to selecting a preference model, which will reflect the perspective of a fictive DM on the quality of the practices of releasing toxic chemicals into the environment.

In our case, as the objectives and criteria are considered independent of each other, we select the bipolar-valued outranking relation with large performance differences polarization from [Bisdorff 2012] in order to model the perspective of a DM on the set of alternatives.

We remind below the parameters on each criterion of the chosen preference model:

- preference direction ($\uparrow$ or $\downarrow$);
- significance weights: $w_i \in (0,1), \forall i \in F\colon \sum\limits_{i \in F} w_i = 1$;
- indifference thresholds: q_i $\forall i \in F$;
- preference thresholds: p_i $\forall i \in F$;
- veto thresholds: v_i $\forall i \in F$.

We consider that lower values on all the attributes of the TH, TE, and RA criteria are preferred to higher ones, while higher values on all the attributes of the MA criterion are preferred to lower ones.

We further divide the significance weights equally between the three objectives, while giving more importance to the TH criterion than to TE. Therefore: $w_{\text{TH}} = 0.(2)$, $w_{\text{TE}} = 0.(1)$, $w_{\text{RA}} = 0.(3)$ and $w_{\text{MA}} = 0.(3)$.

As certain criteria contain several attributes, which have not been aggregated into a single one, we define an indifference, a preference and a veto threshold for each of these attributes. In the case of the binary attributes CAN, CHR, ACU and ENV, the indifference thresholds are all equal to 0, while the preference thresholds are equal to 1. Furthermore, no veto thresholds are defined on them.

All the other attributes are defined on ratio scales with large positive value ranges. Hence, we consider that thresholds that are proportional to the evaluations of each alternative are better suited in this context. In all of these cases, the indifference thresholds are set at 10% of the considered alternatives evaluations, while the preference thresholds are set at 20%. In this case too, no veto thresholds are considered.

The final parameter of the selected preference model, the cut threshold, is set at the standard median level of 0, as we consider that there is no uncertainty with respect to the selected criteria significance weights.

9.3 Solving the problem

9.3.1 Finding the result

In order to find the best result for this problem, we have executed several times each of the proposed extensions of the CLIP algorithm for handling large datasets. Since these approaches are, for the moment, oriented to finding a non-relational clustering result, and due to the size of the problem, we consider the f^*_{NR} fitness measure as the criterion for selecting our result.

Therefore, it is sufficient for us that as many of the alternatives inside a cluster should be considered indifferent to each other, while as many of the alternatives from different cluster should be considered not indifferent, but without taking into account the credibility degrees attached to these relations. Indeed, it makes sense that we should initially try to find a good result with respect to the weakest of the proposed fitness measures, and then afterwards determine if the more restrictive fitness measures should also be considered.

The CLIP approach has been selected due to the fact that it has been shown, through the experimental results from Chapter 6, to consistently perform better than the other considered approaches. A deeper analysis of the clustering approaches for handling large datasets containing other clustering approaches will be considered in the future.

Following the executions of the considered approaches, several of them performed very well with respect to the f^*_{NR} fitness measure. Among them we mention the $\text{CLIP}^{*\textsc{Samp}_1}_{\text{NR}}$, $\text{CLIP}^{*\text{DC}_2}_{\text{NR}}$ and $\text{CLIP}^{*\textsc{IncB}_2}_{\text{NR}}$ approaches. As the results were very close with respect to the fitness measure, within a 1% margin, we have selected the $\text{CLIP}^{*\textsc{Samp}_1}_{\text{NR}}$ approach, due to the fact that it contained the lowest number of clusters and therefore reduced the difficulty of presenting the result.

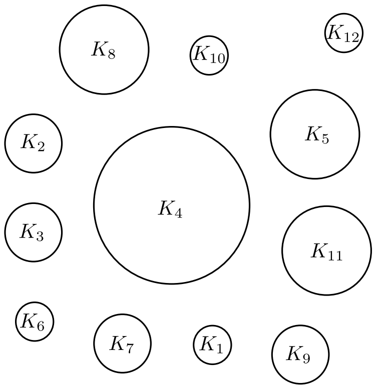
This result is characterised through the fitness values illustrated in Table 9.10.

Table 9.10: Fitness values of the clustering result;

Fitness (%)					
f^*_{NR}	f_{NR}	f^{min}_{NR}	f^*_{R}	f_{R}	f^{min}_{R}
80.0	62.7	0.0	64.0	55.2	0.0

We may easily observe that the clustering result is supported by a high number of relations, as seen through the first fitness measure which the approach directly tries to improve. When considering a non-relational setting, the ideal case of having all the alternatives inside the same cluster indifferent to each other, and all the alternatives from different clusters not indifferent, is reflected by 80% of these relations. The average degree of credibility of the support of this non-relational clustering result is also quite high, around 62%. The third fitness function is at a minimum value, which we attribute to the size of the problem and the approximative approach of extracting the result.

The relational clustering fitness measures are naturally lower than the non-relational ones, which is also due to the approach of building the clustering result being oriented towards finding a non-relational clustering result.



Cluster sizes	
K_1	296
K_2	1,429
K_3	1,632
K_4	6,237
K_5	2,973
K_6	167
K_7	1,316
K_8	2,615
K_9	1,688
K_{10}	540
K_{11}	3,518
K_{12}	356

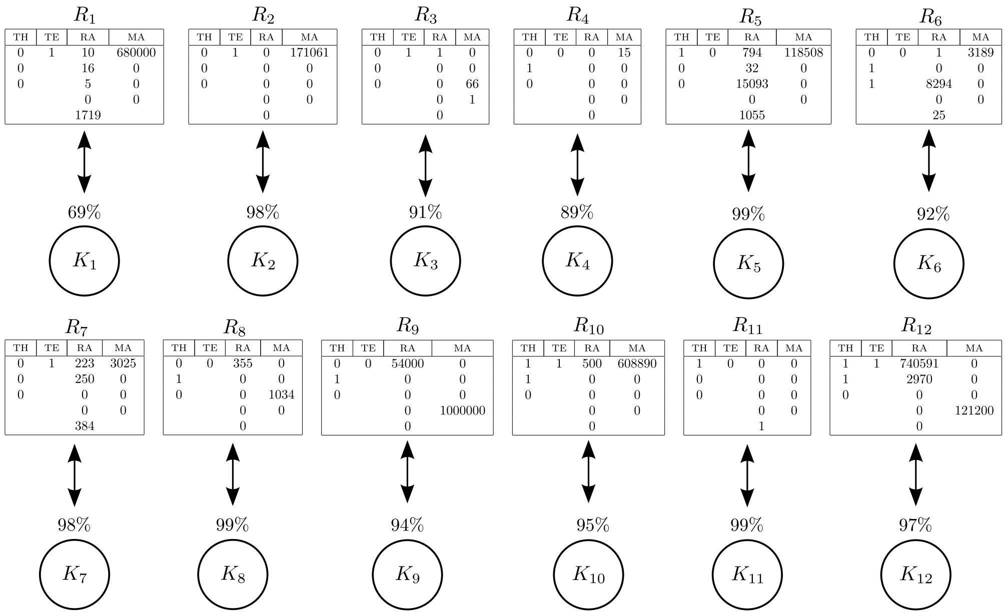
Figure 9.2: Clusters and their sizes;

In Figure 9.2 an initial representation of the clustering result is presented, on the left, along with the number of alternatives inside each cluster, on the right.

We find that 12 clusters have been retrieved. One of them, K_4, contains roughly a quarter of the entire dataset, three of them contain around $3,000$ alternatives, while four hold around 1, 500 alternatives each, with the rest containing only a few hundreds of alternatives.

9.3.2 Analysis using representatives

We continue by describing the clusters of alternatives through their representatives, in Figure 9.3. The evaluations of these representatives have been grouped into four columns, one for each criterion, while on each row we illustrate the evaluations on each attribute contained by a particular criterion. The order of the attributes is that presented in the previous section. Hence, for the TH criterion, the first evaluation is on the CAN attribute, the second on CHR, while the last one is that of ACU. In the case of the TE criterion we have only the ENV attribute. The evaluations of the RA criterion are listed from the AIR criterion at the top, to WAT, LAN, UND and finally OFF at the bottom. In the case of MA we list the evaluations on REC, then on POT, ENE and lastly on TRE. We have additionally illustrated the percentage of alternatives in each cluster that are considered to be indifferent to the cluster representative.



R_1

TH	TE	RA	MA
0	1	10	680000
0		16	0
0		5	0
		0	0
		1719	

R_2

TH	TE	RA	MA
0	1	0	171061
0		0	0
0		0	0
		0	0
		0	

R_3

TH	TE	RA	MA
0	1	1	0
0		0	0
0		0	66
		0	1
		0	

R_4

TH	TE	RA	MA
0	0	0	15
1		0	0
0		0	0
		0	0
		0	

R_5

TH	TE	RA	MA
1	0	794	118508
0		32	0
0		15093	0
		0	0
		1055	

R_6

TH	TE	RA	MA
0	0	1	3189
1		0	0
1		8294	0
		0	0
		25	

R_7

TH	TE	RA	MA
0	1	223	3025
0		250	0
0		0	0
		0	0
		384	

R_8

TH	TE	RA	MA
0	0	355	0
1		0	0
0		0	1034
		0	0
		0	

R_9

TH	TE	RA	MA
0	0	54000	0
1		0	0
0		0	0
		0	1000000
		0	

R_{10}

TH	TE	RA	MA
1	1	500	608890
1		0	0
0		0	0
		0	0
		0	

R_{11}

TH	TE	RA	MA
1	0	0	0
0		0	0
0		0	0
		0	0
		1	

R_{12}

TH	TE	RA	MA
1	1	740591	0
1		2970	0
0		0	0
		0	121200
		0	

Figure 9.3: Cluster representatives;

Taking into account the number of alternatives that could be replaced by a cluster representative, we find that a total of only 5.2% of the alternatives in our dataset are outliers. This means that, given the model containing 12 representative alternatives, these outliers are not indifferent to any one of them and therefore do not fit in this model. It is not uncommon for outliers to emerge from clustering approaches, which in this case is also due to the approximative nature of the considered approach.

A real DM may analyse these representatives, and label them with respect to the quality of the chemical release practices that they reflect. The advantage of grouping indifferent alternatives lies in the extraction of a reduced number of clusters, as opposed to grouping similar alternatives. When analysing each cluster representative, the DM should take into account that alternatives that are rather different from the representative, which is depicted through its evaluations on each criterion, may exist in its cluster.

For instance, almost 70% of the alternatives in the first cluster may be replaced by an alternative which describes a toxic chemical release practice which involves a substance that has no impact on human health, but has an effect on the environment. Furthermore, small quantities are released into the atmosphere and water bodies, which are considered by us to be mediums with a high degree of spread. Moderate amounts are also released off-site. Finally, good mitigation practices may be involved, which are reflected by high amounts of the considered chemical being recycled for reuse. We may therefore consider the elements from the first cluster to contain good toxic chemical release practices. Furthermore, alternatives that describe chemical release practices of more toxic chemicals may be exist in this cluster, however, in their case, the release amounts would be less than the ones of the representative, of the recycled amounts would be larger. Similar statements may be made when considering a significant increase or decrease of the evaluations of the representative on the remaining criteria. We still consider that such alternatives would describe good chemical release practices, as this is in accordance with the models used to represent the DMs preferences.

The representative of the second cluster may be seen to also characterise good chemical release practices. In comparison to the first representative, the release amounts are lower, although the amounts recycled are also lower, while the toxicity of the released chemical is the same as in the case of the first. We may therefore consider that the alternatives in the second cluster also represent good chemical release practices.

In regard to the third cluster representative, we may see that it is quite similar to the second, although no recycling procedures are performed. Although we mat consider it weakly preferred by the other two cluster representative, we may still label the alternatives in this cluster as good chemical release practices.

The fourth cluster representative may also be seen as a good chemical release practice. Although it reflects the release of a more toxic chemical than the first three cluster representatives, there are no releases into the environment, while certain small recycling practices are put into place.

The fifth cluster representative, however, may be seen to reflect poorer practices with respect to the release of toxic chemicals. The handled chemical is more toxic than the ones discussed so far, while significant quantities are released into the environment, although good recycling practices are also put into place. We nevertheless consider the alternatives in K_5 to reflect poor practices.

On the other hand, the sixth cluster representative illustrates the release of less toxic chemicals, which may produce chronic or acute diseases to humans. The release amounts are not so large as in the previous case, although neither are the amounts that are recycled. The alternatives in K_6 are therefore considered to be of a medium quality with respect to the quality of their release practices.

The representative of the seventh cluster may be seen to reflect practices of chemicals that are similar to those of the clusters labelled as good, although the release amounts are larger and the recycled amounts are lower. Therefore, this representative will not be considered as a good release practice. However, as the toxicity of the handled chemical is lower than that of R_5 and R_6, and as the released amount into air is larger than that of R_6, but lower than that of R_5, with the amount recycled similar to that of R_6, R_7 may be considered to be a medium release practice, and so are the alternatives in K_7.

When analysing R_8, we find that it handles more toxic chemicals than R_7, the amounts released into air are larger while no recycling of this chemical is performed. Hence, we cannot place R_8 at the same level as R_7. However, we may consider R_8 on the same level as R_5, since it handles a slightly less toxic chemical which is released in lower amounts into the air, although the recycling practices are more restricted. Therefore we label the alternatives in cluster K_8 as poor.

It is quite easy to consider the practices described by the ninth cluster representative as very poor, due to the very large amounts of the released chemical, and the very poor mitigation practices which only consider the treatment of the toxic chemical which may produce chronic diseases in humans.

The representative of K_{10}, however, may not be placed on the bottom level, since fewer amounts are released into the air, while more amounts of the highly toxic chemical are recycled. Nevertheless, R_{10} illustrates poorer release practices than the representative from the second level, therefore we label the the alternatives in cluster K_{10} as poor.

The representative of cluster K_{11} raises above the previous one, due to a slight decrease in the toxicity of the release chemical, and the fact that no amount is actually released, although no mitigation practices are in effect either. The alternatives in K_{11} are labelled as medium.

Finally, the last cluster representative highlight clearly very poor release practices, being placed along with K_9 on the bottom level.

Denoting with $\{\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}\}$ the four ordinal labels used to assess the quality of the release practices of the alternatives in each of the 12 clusters, we find a summary of these results in Tables 9.11 and 9.12.

Table 9.11: Cluster labels;

K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}	K_{11}	K_{12}
A	A	A	A	C	B	B	C	D	C	B	D

We may deduce that a majority of the release practices that we have considered

Table 9.12: Distribution of labels in the dataset;

A	B	C	D
9,594	5,001	6,128	2,044

from 2010 in the U.S., are of good quality. We may at any point substitute this with the representatives of the four clusters that have been labelled in this way, in order to illustrate how such practices may appear.

A more or less equal number of practices may be placed in the two intermediary levels, forming a total of half of the considered dataset, while a few over $2,000$ practices may be considered very poor. A DM may, at this stage, extract overall judgements on the release practices of facilities that handle toxic chemicals, and may consider applying certain measures, especially for the ones labelled with D.

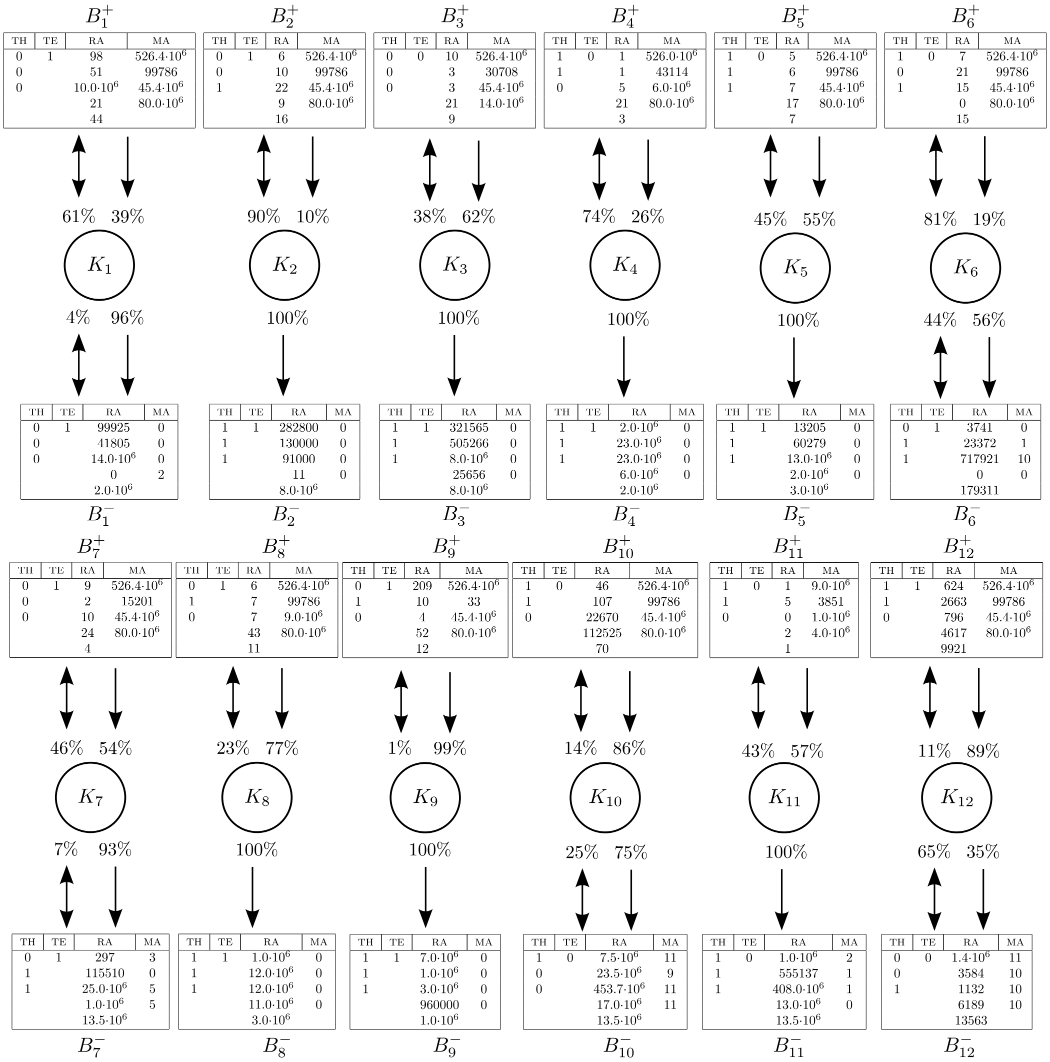
We acknowledge the fact that a sorting approach may have been considered in order to extract a set of ordered classes of alternatives, therefore reaching similar conclusions over the quality of the toxic chemical release practices. The methods that we have illustrated above may be seen as an alternatives approach to finding such a result. Furthermore, the applications of the clustering approaches are not limited to this problem, which has been used in order to illustrate one of its potential uses.

9.3.3 Analysis using bounding profiles

Although the alternatives in each cluster are most of them indifferent to the cluster representative, this does not give any guarantees with respect to them being indifferent to each other. For instance, we may have a cluster of alternatives that are divided in two sub-clusters, where the alternatives in the first are strictly preferred to those from the second. However, the cluster representative may be built in such a way that both the alternatives inside the first sub-cluster and those from the second are indifferent to it. This representative may be seen to bridge the gap between the two sub-clusters, being able to represent both of them, but not necessarily affirming that the cluster is compact.

For assessing this aspect, due to the fact that we are dealing with a problem of a large cardinality, we cannot consider the relations between all the alternatives, or else we would have used one of the clustering approaches proposed in the second part of this thesis. In order to gain an insight on this aspect, we may consider building bounding profiles around each cluster in order to understand better how much the alternatives inside them are spread apart.

For the selected clustering result, we have used an approach of constructing upper and lower bounding profiles that, initially, used the best and worst evaluations of the alternatives inside each cluster for constructing these profiles, and then a simulated annealing meta-heuristic, which looked to optimize the f^{*+} and f^{*-} fitness functions for a bounding profile from Equations 7.28 and 7.31. This approach is motivated by the results from Chapter 7. Furthermore, we have given the meta-heuristic sufficient time in order to perform a significant number of iterations for each profile. The resulting bounding profiles are depicted in Figure 9.4. The percentage of alternatives from each cluster that are indifferent, strictly preferred to, or strictly preferred by the profiles are also illustrated.



B_1^+

TH	TE	RA	MA
0	1	98	$526.4\cdot10^6$
0		51	99786
0		$10.0\cdot10^6$	$45.4\cdot10^6$
		21	$80.0\cdot10^6$
		44	

B_2^+

TH	TE	RA	MA
0	1	6	$526.4\cdot10^6$
0		10	99786
1		22	$45.4\cdot10^6$
		9	$80.0\cdot10^6$
		16	

B_3^+

TH	TE	RA	MA
0	0	10	$526.4\cdot10^6$
0		3	30708
0		3	$45.4\cdot10^6$
		21	$14.0\cdot10^6$
		9	

B_4^+

TH	TE	RA	MA
1	0	1	$526.0\cdot10^6$
1		1	43114
0		5	$6.0\cdot10^6$
		21	$80.0\cdot10^6$
		3	

B_5^+

TH	TE	RA	MA
1	0	5	$526.4\cdot10^6$
1		6	99786
1		7	$45.4\cdot10^6$
		17	$80.0\cdot10^6$
		7	

B_6^+

TH	TE	RA	MA
1	0	7	$526.4\cdot10^6$
0		21	99786
1		15	$45.4\cdot10^6$
		0	$80.0\cdot10^6$
		15	

TH	TE	RA	MA
0	1	99925	0
0		41805	0
0		$14.0\cdot10^6$	0
		0	2
		$2.0\cdot10^6$	

B_1^-

TH	TE	RA	MA
1	1	282800	0
1		130000	0
1		91000	0
		11	0
		$8.0\cdot10^6$	

B_2^-

TH	TE	RA	MA
1	1	321565	0
1		505266	0
1		$8.0\cdot10^6$	0
		25656	0
		$8.0\cdot10^6$	

B_3^-

TH	TE	RA	MA
1	1	$2.0\cdot10^6$	0
1		$23.0\cdot10^6$	0
1		$23.0\cdot10^6$	0
		$6.0\cdot10^6$	0
		$2.0\cdot10^6$	

B_4^-

TH	TE	RA	MA
1	1	13205	0
1		60279	0
1		$13.0\cdot10^6$	0
		$2.0\cdot10^6$	0
		$3.0\cdot10^6$	

B_5^-

TH	TE	RA	MA
0	1	3741	0
1		23372	1
1		717921	10
		0	0
		179311	

B_6^-

B_7^+

TH	TE	RA	MA
0	1	9	$526.4\cdot10^6$
0		2	15201
0		10	$45.4\cdot10^6$
		24	$80.0\cdot10^6$
		4	

B_8^+

TH	TE	RA	MA
0	1	6	$526.4\cdot10^6$
1		7	99786
0		7	$9.0\cdot10^6$
		43	$80.0\cdot10^6$
		11	

B_9^+

TH	TE	RA	MA
0	1	209	$526.4\cdot10^6$
1		10	33
0		4	$45.4\cdot10^6$
		52	$80.0\cdot10^6$
		12	

B_{10}^+

TH	TE	RA	MA
1	0	46	$526.4\cdot10^6$
1		107	99786
0		22670	$45.4\cdot10^6$
		112525	$80.0\cdot10^6$
		70	

B_{11}^+

TH	TE	RA	MA
1	0	1	$9.0\cdot10^6$
1		5	3851
0		0	$1.0\cdot10^6$
		2	$4.0\cdot10^6$
		1	

B_{12}^+

TH	TE	RA	MA
1	1	624	$526.4\cdot10^6$
1		2663	99786
0		796	$45.4\cdot10^6$
		4617	$80.0\cdot10^6$
		9921	

TH	TE	RA	MA
0	1	297	3
1		115510	0
1		$25.0\cdot10^6$	5
		$1.0\cdot10^6$	5
		$13.5\cdot10^6$	

B_7^-

TH	TE	RA	MA
1	1	$1.0\cdot10^6$	0
1		$12.0\cdot10^6$	0
1		$12.0\cdot10^6$	0
		$11.0\cdot10^6$	0
		$3.0\cdot10^6$	

B_8^-

TH	TE	RA	MA
1	1	$7.0\cdot10^6$	0
1		$1.0\cdot10^6$	0
1		$3.0\cdot10^6$	0
		960000	0
		$1.0\cdot10^6$	

B_9^-

TH	TE	RA	MA
1	0	$7.5\cdot10^6$	11
0		$23.5\cdot10^6$	9
0		$453.7\cdot10^6$	11
		$17.0\cdot10^6$	11
		$13.5\cdot10^6$	

B_{10}^-

TH	TE	RA	MA
1	0	$1.0\cdot10^6$	2
1		555137	1
1		$408.0\cdot10^6$	1
		$13.0\cdot10^6$	0
		$13.5\cdot10^6$	

B_{11}^-

TH	TE	RA	MA
0	0	$1.4\cdot10^6$	11
0		3584	10
1		1132	10
		6189	10
		13563	

B_{12}^-

Figure 9.4: Bounding profiles for the first six clusters;

The resulting bounding profiles are able to fully represent all the alternatives in the dataset. Therefore, each alternative inside a cluster will be either indifferent or strictly preferred by the upper bounding profile, while it will be either indifferent or strictly preferred to the lower bounding profile.

Bounding the alternatives of a cluster between two such profiles may be easily done through the construction of the absolute best and absolute worst alternatives that may be built on the considered set of criteria. However, we are also interested in bringing these profiles as close to the alternatives inside the cluster as possible, *i.e.* to be indifferent to them. Therefore, the percentage of alternatives from each cluster that are indifferent to either profile give an indication with respect to this aspect.

For instance, when looking at the first cluster we notice that the upper bounding profile comes very close to a large percentage of the alternatives inside this cluster. A total of 61% of the alternatives in this cluster are indifferent to the upper bounding profile, while the rest are strictly preferred by it. Furthermore, a few alternatives are also indifferent to the lower bounding profile. Due to the large number of alternatives in this cluster that are indifferent to the bounding profiles, we may conclude that the approach of building these profiles performed rather well.

The DM would normally consider each cluster of alternatives through the two bounding profiles and the distributions of the alternatives around them, in order to label each cluster as we have done in the previous section. As the number of bounding profiles is considerably larger than in the case of central profiles, we will be using another approach of labelling the clusters of alternatives. This approach highlights certain intuitions that might be extracted from the bounding profiles, which may additionally be used in an algorithmic framework in order to construct results with better fitness values, or even relational and ordered clustering results. For this purpose, we will consider the information given by the bounding profiles with respect to the spread of each cluster, considering the scale of the overall preferences of the DM.

The representations of the bounding profiles from Figure 9.3 are not sufficient for determining how much the alternatives inside a cluster are spread apart, but only indicate how many are concentrated around the upper and lower profiles. For example, most of the alternatives in the first cluster are distributed towards the upper bounding profile, while only a few towards the lower one. In the case of the second cluster, an even larger percentage of alternatives are close to the upper bounding profile, however, the lower bounding profile is not indifferent to any of the alternatives in this cluster. This means that the approach for constructing this profile was not able to bring it close enough to the alternatives that are worse than the others from this cluster. Either the approach required a longer execution time, or it was impossible to construct the profile to be indifferent to one of these alternatives without becoming strictly preferred to others inside the cluster.

In order to illustrate the relative spread of the alternatives in each cluster, when compared to the those from the others, we present the representation from Figure 9.5.

In this figure, we have ordered the bounding profiles of each cluster over a vertical scale. We have additionally drawn the relations of strict preference between any pair of bounding profiles. We must state, however, that having illustrated a bounding profile on a level above another does not imply that it is strictly preferred to it, only that the latter is not strictly preferred to the former. The bounding profiles that are placed at the top are neither strictly preferred to each other, *i.e.* they are indifferent, while the profiles from the second level are strictly preferred by at least one of these profiles. This rule is applied further down, as a bounding profile that is strictly preferred by another will always be illustrated on a level below it.

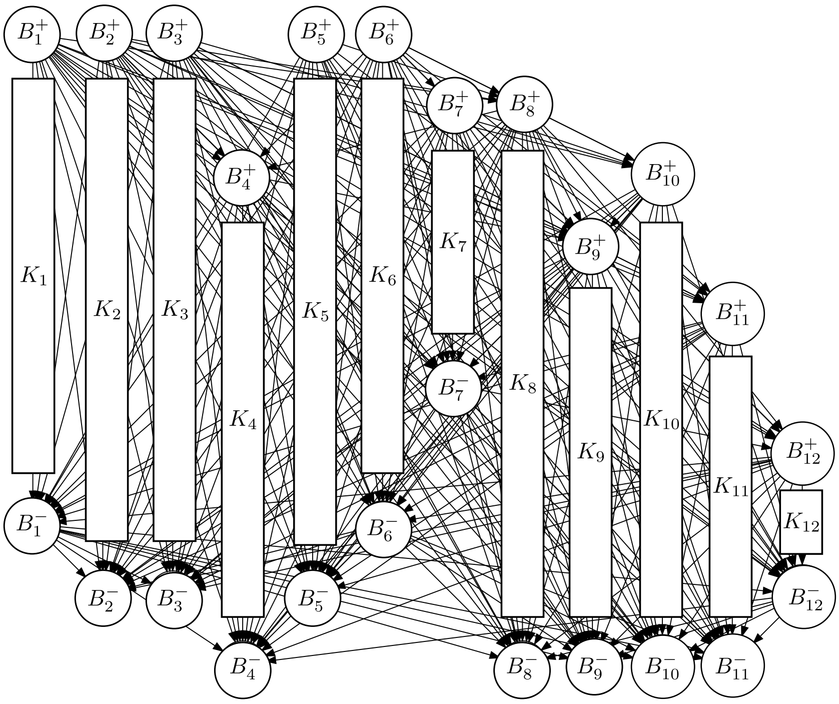


Figure 9.5: Relative ordinal placement of bounding profiles;

Only the relative placement of the bounding profiles with respect to a scale of the overall preferences of the DM over them is illustrated in this image. The actual spread of the clusters of alternatives over this scale cannot be deduced from this illustration. Although we have spaced the different ordinal levels equally apart, the actual spacing between them may be very contrasting.

Nevertheless, we may for instance observe that cluster K_7 is definitely strictly preferred to K_{12}, as its lower bounding profile is strictly preferred to the upper bounding profile of the latter cluster. We may also notice that these two clusters seem more compact than the others.

Taking into account the percentages of alternatives that are indifferent to a bounding profile, we may further illustrate this limited information with respect to the distribution of alternatives inside each cluster, as depicted in Figure 9.6.

Each cluster is represented through a shape which reflects the percentage of alternatives that are distributed close to the upper bounding profile (at the top), that of the alternatives that are not indifferent to either bounding profile (in the middle), and that of the alternatives that are indifferent to the lower bounding profile (at the bottom).

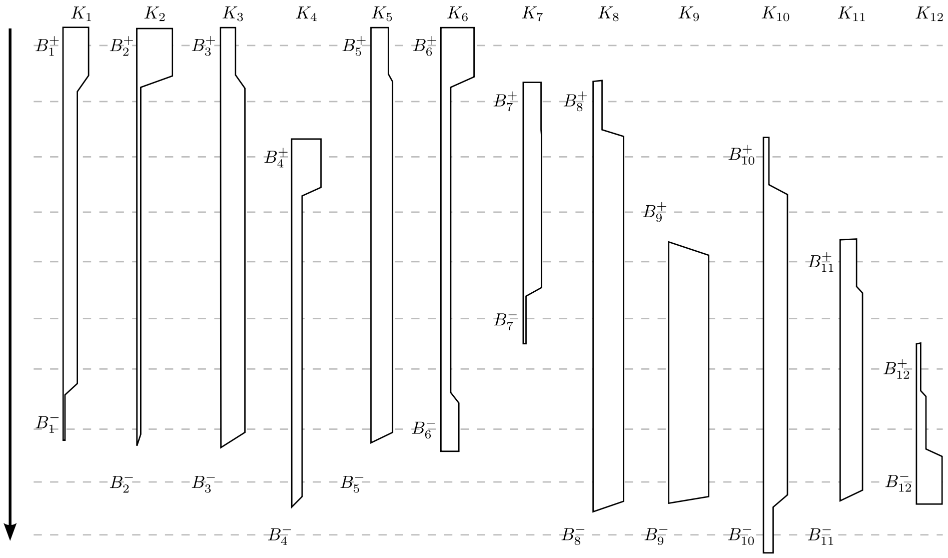


Figure 9.6: Distribution of alternatives and relative ordinal placement with respect to bounding profiles;

This information greatly enhances our understanding of the division between these clusters of alternatives, and potentially also that of the relations between them.

We find that the first two clusters should be rightfully considered to contain good chemical release practices, due to the large concentrations of alternatives around their upper bounding profiles. Looking back at Figure 9.4, we may indeed notice that these two profiles reflect good chemical release practices. The first illustrates the handling of a low toxic chemical, with moderate amounts being released into the environment, while good mitigation practices are put into place. The second upper bounding profile balances the handling of a slightly more toxic chemical, although the released amounts are lower than in the previous case.

Although the upper bounding profiles of clusters K_3, K_5 and K_6 are indifferent to each other and to the upper bounding profiles of the first two clusters, due to the fact that many alternatives are not centred around them, we consider them to contain release practices of medium quality.

Clusters K_4, K_7, K_8, K_{10} and K_{11} may be considered to contain poor chemical release practices. K_4 is placed in this class due to the fact that many of the alternatives inside it are centred around the upper bounding profile of this cluster. This profile is strictly preferred by the upper bounding profiles of K_8, which contains 20% of the alternatives in this cluster around it. Therefore K_4 cannot be placed above K_8. K_8 contains a rather wide distribution, considering the overall preference scale on which we have illustrated the placement of the upper and lower bounding profiles. Nevertheless, no alternatives from this cluster are centred around the lower bounding profiles. Cluster K_7 is also placed in this class due to the lower bounding profile being strictly preferred to all the others, but also to the upper bounding profile of K_{12}. Although clusters K_{10} and K_{11} may be seen to have certain alternatives distributed lower with respect to the overall preference scale, they may be considered better than clusters K_9 and K_{12}, therefore they are placed in this category.

Finally, clusters K_9 and K_{12} may be considered to contain very poor chemical release practices.

We denote with {A', B', C', D'} the four ordinal labels used to assess the quality of the considered release practices. Since we have used a different approach for assigning these labels, they are not the same as those used in conjunction with the central profiles. The summary of these results in depicted in Tables 9.13 and 9.14.

Table 9.13: Cluster labels;

K_1	K_2	K_3	K_4	K_5	K_6	K_7	K_8	K_9	K_{10}	K_{11}	K_{12}
A'	A'	B'	C'	B'	B'	C'	C'	D'	C'	C'	D'

Table 9.14: Distribution of labels in the dataset;

A'	B'	C'	D'
1,725	4,772	14,226	2,044

We notice that in this case a larger number of the alternatives have been assigned to the third category, and only a few to the first. A significant number of alternatives may be found in the second category, while the same alternatives as in the case of using central profiles, have been considered to contain very bad release practices in this case too.

Summary

In this final chapter, we have turned our attention towards the application of the proposed MCDA clustering approaches, and the measures for describing such a clustering result, to a problem which is based on real information.

The illustrated problem is fictitious, due to the fact that its purpose is to highlight the potential use of the work presented in this thesis, but also to identify issues that may need to be addressed in the future. Using artificially constructed data may not have been able to highlight some of the apparent issues which will need to be considered.

Nevertheless, the information that we have used is real, and is easily accessible in large amounts. The perspective that we have defined over this information, which is that of preferences over the toxic release practices of a large number of facilities, may be easily imagined by many of us. Therefore, certain decision problems may be formulated over this type of information, in the future.

Due to the large volume of information, the need to analyse and retrieve information on the way the data is structured becomes very important. In the imagined context, it is clear that this analysis needs to also consider preferential information that one might have over this data, and so an MCDA-oriented approach should be used. Considering also that little a priori information is available over this data, as is the case in our imagined problem, the use of an MCDA clustering approach becomes very well-motivated.

The result that we have illustrated, which was retrieved through one of the proposed algorithmic approaches to clustering in MCDA, was supported by a large percentage of preferential relations that exist between the decision alternatives. At this point, we have only considered looking for a non-relational clustering result, using the crisp credibility degrees of the relations between the alternatives. We have used the central profiles, or cluster representatives, in order to provide an initial description of the clustering result. We were, in this way, able to reduce the entire dataset of 22 thousand alternatives to only 12, which were able to model the original set in a proportion of 95%.

The central profiles were used to label the alternatives in the dataset using four ordinal levels. We found that many chemical release practices appear to be of good quality considering the negative effect that they might produce in the environment. Furthermore, around $2,000$ of these releases may be considered to be of very poor quality, and certain investigations around them might be necessary.

Through the construction of bounding profiles for each cluster, we were able to completely model all the alternatives in the dataset, while further finding information with respect to how the alternatives are distributed inside each cluster. Instead of labelling each cluster based on the representations of the bounding profiles, we have used the information given by a weak order on them, as well as the percentages of alternatives from each cluster that were indifferent to them, in order to extract information on the relations between them. A labelling was then performed following this information, although, we consider that further investigation is required. A slightly different results as before was found, although the clusters that were previously labelled as the worst, were in this case too labelled in this way.

Further work will be considered in the future, in order to explore the other clustering objectives and structures, but also the more restrictive fitness measures. The bounding profiles, as well as the separating ones may be used to this end. Nevertheless, we have been able to find significant amounts of useful information from this case study in order to motivate further work on this topic.

Conclusions and perspectives

Summary of the main achievements

In the introduction of this thesis we have asked ourselves whether the notion of similarity is sufficient in order to group alternatives that a DM cannot distinguish between, and to separate those that he can. Following the introduction to the problem of clustering from a data analysis perspective, and to the field of multiple criteria decision aid, we have addressed this question by analysing the differences between several measures of similarity and another of indifference. Our conclusions are that, indeed, we may find cases where alternatives that are dissimilar may still not be distinguished by the DM, due to the fact that they are considered indifferent with respect to his preferences. Furthermore, when considering several scenarios, we have been able to observe that the percentage of cases where different measures of similarity and another of indifference discriminate between the alternatives differently is significant. Following from this, the redefinition of the problem of clustering in MCDA, as an approach that groups alternatives that are indifferent and separates those that are not, is well-motivated. We have additionally extended this definition to relational and ordered clustering.

In the second part of the thesis we have presented several models for the proposed definitions of clustering in MCDA. Due to the fact that only a few existing approaches fit with these definitions, we have extended the classical K-MEANS algorithm and the self-organizing maps to this case, but also proposed a new graph theoretic approach. The question related to the validation of the proposed approaches, through the construction of a large number of benchmarks, was addressed here. We have considered constructing a wide array of structures, ranging from a clear order over sets of indifferent alternatives to difficult problems instances containing alternatives with very contrasting evaluations. These benchmarks have been used in order to validate the proposed approaches, but also to compare them with other existing methods. We may clearly conclude that the proposed approaches, which group alternatives that are indifferent and separate those that are strictly preferred or incomparable, bring significant improvements in the results they proposed with respect to approaches that group similar alternatives and separate those that are dissimilar, considering the different fitness measures of the presented models.

The last questions, that we have raised in the beginning, revolved around the topic of the existence and handling of large MCDA datasets. These questions have been addressed in the last part of the thesis. Although in data analysis we may easily find many datasets containing very large numbers of objects, the decision problems generally tackled in MCDA have a considerably smaller cardinality. However, we have been able to find one data source from which we have extracted a large number of objects over which certain preferences can easily be expressed. We are talking about a dataset containing information on $22,000$ toxic chemical release practices in the United States of America, only from the year 2010, although much larger amounts of information are available from the considered data source. Although the extracted data was not directly used in a real decision problem, preferential information over the toxic chemical release practices could easily be considered. Due to the problems related to the recent environmental changes, the issue of assessing the quality of the toxic chemical release practices of different facilities may potentially be formulated in the near future, so that different actions in order to reduce these changes may be taken.

Over the mentioned dataset, we have considered the topic of exploring it and extracting concise information related to the overall quality of the toxic chemical release practices of each facility. Due to the exploratory nature of clustering, the choice of applying it to this problem followed naturally. In order to be able to handle this dataset, and others of its size, we have defined in the beginning of the third part of the thesis several measures that may be used to characterise sets of alternatives. Different fitness measures and approaches for extracting these measures have been additionally proposed and validated empirically, over the same set of proposed benchmarks. Furthermore, we have addressed in part the problem of extending the proposed clustering approaches to handling large datasets, although further exploration of this topic is needed. Finally, we have applied the proposed clustering approaches over the toxic chemical releases dataset and have been able to reduce it to only 12 representative alternatives which were able to summarize almost all of the original $22,000$.

As a practical application, we have considered the use of the 12 representative alternatives in the context of an ordinal labelling problem. While a similar result may be achieved using a sorting approach and asking the DM for assignment examples, we consider this illustrated application to be a alternative approach. Furthermore, the potential applications of clustering in MCDA are not limited to it, a topic that we will address in the perspectives for the presented work, which follow next.

Perspectives and future work

Having tackled the topic of an entire problem from the field of data mining, that of clustering, which we redefined within the context of the field of MCDA, it comes to no surprise that certain aspects have not been completely explored, some of them have been only briefly considered, while plenty others could further be imagined.

In the following, we provide a non-exhaustive list of perspectives that could be explored in the future, around the work that has been presented in this thesis.

Similarity, indifference, strict preference and incomparability

The analysis on the similarity and indifference measures, which we have begun exploring in the third chapter, could definitely be complemented by the addition of other measures. We may consider other measures of similarity but also measures of indifference derived from other types of outranking relations, or even value functions. Furthermore, the relations of strict preference and incomparability could also be analysed. Axiomatic characterisations of value functions, concordance and outranking relations have already been explored, although to our knowledge, this has not been done when considering the relations of indifference, strict preference and incomparability.

Clustering structures, their fitness and the algorithms

For the moment, we have considered each type of structure that may appear over the clusters of alternatives, given that any relation of indifference, strict preference or incomparability may be expressed between them. We have defined fitness measures, for each of them, that express different degrees with which the proposed clustering result is supported by the actual relations that appear between the alternatives inside them, but also fitness measures that reflect the degree with which a set of representatives is able to reflect the existing structure in the data. Following the analysis on the relations of indifference, strict preference and incomparability, we may consider that only a few of the different types of structures may appear over a clustering result. Furthermore, new fitness measures may be proposed.

Following the exploration of the existing and the new measures that model preferences, but also the definition of new clustering structures and their fitness measures, new clustering approaches could also be explored. At the moment, we consider focusing on developing hierarchical approaches.

At the moment we have explored only the case of non-relational clustering when large MCDA datasets are considered. The proposed approaches have also the potential of being extended in order to provide relational and ordered clustering results. The use of bounding and separating profiles may be motivated in these cases. This follows as well from the last chapter of the thesis, where we have shown certain potential uses of bounding profiles.

Applications

The characterising measures for MCDA clustering results could further be explored and extended, in order to improve their application when exploring MCDA datasets. Considering that other preference models may be used for constructing the indifference, strict preference and incomparability relations, the characterising measures proposed in the seventh chapter could also be extended. Furthermore, we have already illustrated the potential use of several of these measures in the last chapter of this thesis. However, we have only focused on their use over a non-relational clustering result that has been built using the crisp credibility degrees of the relations between alternatives.

The application of clustering in MCDA to real problems, concerning real people and real decisions, is also something that we would like to explore in the future. Although we have used in our case study real data on which preferences could easily be expressed, we have only constructed a fictive problem around this data, in order to illustrate the potential application of the presented work.

We have also taken into consideration the application of the problem of clustering in MCDA for aiding in the elicitation of the parameters of a given preference model. As through clustering we are able to structure the existing data, without taking into account a large amount of a priori information, using such an approach when the parameters of the underlying preference model are not fixed, may provide certain insights on the pairs of alternatives that we would need to confront a DM with. This might lead to a decrease in the number of questions that we would need to ask a DM so that the parameters of the model converge to their final values.

There is also a great potential of applying the presented work and the MCDA approaches of modelling preferences in the context of other fields which handle large volumes of data and which would benefit from the exploitation of this preferential information. A clear identification of these fields and the ways in which they could benefit from the work we have presented is something that we wish to focus in the near future.

Concluding remarks

We conclude this work with the hope that we have been able to convince the reader not only on the differences between clustering in data analysis and in MCDA, and the motivation for redefining it as we have proposed, but also on the fact that it may bring a significant amount of advantages to the exploration, the analysis and the understanding of decision problems in MCDA.

We consider that the reduced interest in the clustering problem in MCDA, in comparison to the choice, sorting and ranking problems, is mostly due to the size of most of the decision problems considered in the literature. As many of these problems contain sets of alternatives that may be explored without too much difficulty by the DM, the benefits that clustering would bring over them are limited. The field of MCDA continues to grow and as it does so, it is only natural that larger decision problems will appear, a perspective which we have additionally illustrated that is very probable. The problem of clustering may therefore bring significant benefits especially when such large decision problems come into play, which is mainly due to the fact that a DM may no longer be able to consider every individual alternative.

The work that has been presented may be regarded as a first step in the exploration of the presented problem. Plenty of directions for future work have been laid before us, from the further exploration of the measures of indifference, strict preference and incomparability, to the proposal of new models for clustering, from the construction of benchmarks for testing and comparing different algorithms, to the extension of these approaches to handling large datasets. Nevertheless the real potential of clustering using preferential information is much larger, therefore we invite anyone interested to consider any future directions for clustering in MCDA.

Bibliography

[Agrawal et al. 1998] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan (1998). *Automatic subspace clustering of high dimensional data for data mining applications*. SIGMOD Rec., 27(2), 94–105. (Cited on page 17.)

[Anderberg 1973] M. R. Anderberg (1973). *Cluster Analysis for Applications*: Academic Press. (Cited on pages v, 4, 8 and 11.)

[Angilella et al. 2010] S. Angilella, S. Greco, and B. Matarazzo (2010). *Non-additive robust ordinal regression: A multiple criteria decision model based on the Choquet integral*. European Journal of Operational Research, 201(1), 277–288. (Cited on page 29.)

[Arrow 1951] K. J. Arrow (1951). *Social choice and individual values*. New York: J. Wiley. 2nd edition, 1963. (Cited on page 21.)

[Asuncion and Newman 2007] A. Asuncion and D. J. Newman (2007). *UCI Machine Learning Repository*. (Cited on page 13.)

[Ball and Hall 1965] G. H. Ball and D. J. Hall (1965). *ISODATA. A novel method of data analysis and pattern classification.*. Technical Report., Menlo Park: Stanford Research Institute. (Cited on pages 12 and 13.)

[Bana e Costa et al. 2001] C. A. Bana e Costa, F.N. da Silva, and J.-C. Vansnick (2001). *Conflict dissolution in the public sector: A case-study*. European Journal of Operational Research, 130, 388–401. (Cited on page 20.)

[Banerjee et al. 2005] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh (2005). *Clustering with Bregman Divergences*. J. Mach. Learn. Res., 6, 1705–1749. (Cited on page 13.)

[Baroudi and Safia 2010] R. Baroudi and N.B. Safia (2010). *Towards multicriteria analysis: A new clustering approach*. in. *Machine and Web Intelligence (ICMWI), 2010 International Conference on*, pages. 126 –131. (Cited on page 36.)

[Basak and Krishnapuram 2005] J. Basak and R. Krishnapuram (2005). *Interpretable Hierarchical Clustering by Constructing an Unsupervised Decision Tree*. IEEE Trans. on Knowl. and Data Eng., 17(1), 121–132. (Cited on page 16.)

[Belacel 2000] N. Belacel (2000). *Multicriteria assignment method PROAFTN: Methodology and medical application*. European Journal of Operational Research, 125(1), 175–183. (Cited on pages viii, 31 and 142.)

[Belkin and Niyogi 2001] M. Belkin and P. Niyogi (2001). *Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering*. Advances in Neural Information Processing Systems, 14, 585–591. (Cited on page 17.)

[Bisdorff 2000] R. Bisdorff (2000). *Logical foundation of fuzzy preferential systems with application to the electre decision aid methods*. Computers & Operations Research, 27(7–8), 673–687. (Cited on page 60.)

[Bisdorff 2002] R. Bisdorff (2002). *Electre-like clustering from a pairwise fuzzy proximity index*. European Journal of Operational Research, 138(2), 320–331. (Cited on page 36.)

[Bisdorff 2012] R. Bisdorff (2012). *On polarizing outranking relations with large performance differences*. Journal of Multi-Criteria Decision Analysis, DOI: 10.1002/mcda.1472, 1–20. (Cited on pages 46, 62, 64, 136, 137 and 204.)

[Bisdorff et al. 2011] R. Bisdorff, P. Meyer, and A.-L. Olteanu (2011). *A Clustering Approach Using Weighted Similarity Majority Margins*. in. J. Tang, I. King, L. Chen, and J. Wang (Editors.): *Proceedings of the 7th International Conference on Advanced Data Mining and Applications*, vol. 7120 of Series. *Lecture Notes in Computer Science*, pages. 15–28: Springer. (Cited on pages 36 and 46.)

[Bisdorff et al. 2008] R. Bisdorff, P. Meyer, and M. Roubens (2008). *RUBIS: a bipolar-valued outranking method for the choice problem*. 4OR, A Quarterly Journal of Operations Research, 6(2), 143–165. (Cited on pages 26, 30 and 102.)

[Blei et al. 2003] D. M. Blei, A. Y. Ng, and M. I. Jordan (2003). *Latent dirichlet allocation*. J. Mach. Learn. Res., 3, 993–1022. (Cited on page 17.)

[Bouyssou 1990] D. Bouyssou (1990). *Building criteria: A prerequisite for MCDA*. in. C.A. Bana e Costa (Editor.): *Readings in Multiple Criteria Decision Aid*, pages. 58–80: Springer-Verlag, Berlin. (Cited on page 21.)

[Bouyssou et al. 2006] D. Bouyssou, T. Marchant, M. Pirlot, A. Tsoukiàs, and P. Vincke (2006). *Evaluation and decision models with multiple criteria: Stepping stones for the analyst*. International Series in Operations Research and Management Science, Volume 86. Boston: Springer, 1st ed. (Cited on page 22.)

[Bradley et al. 1998] P. S. Bradley, U. M. Fayyad, and C. Reina (1998). *Scaling Clustering Algorithms to Large Databases*. in. *Knowledge Discovery and Data Mining*, pages. 9–15. (Cited on page 13.)

[Brans 1982] J. P. Brans (1982). *L'ingénièrie de la décision; Elaboration d'instruments d'aide à la décision. La méthode PROMETHEE*. in. R. Nadeau and M. Landry (Editors.): *L'aide à la décision: Nature, Instruments et Perspectives d'Avenir*, pages. 183–213, Québec, Canada: Presses de l'Université Laval. (Cited on page 32.)

[Brans and Mareschal 1992] J.P. Brans and B. Mareschal (1992). *Promethee-V – MCDM Problems with Segmentation Constraints.*. INFOR, 30(2), 85–96. (Cited on page 32.)

[Brans and Mareschal 1994] J.P. Brans and B. Mareschal (1994). *The PROMCALC and GAIA Decision Support System for Multicriteria Decision Aid*. Decision Support Systems, 12, 297–310. (Cited on page 32.)

[Brans and Mareschal 2002] J.P. Brans and B. Mareschal (2002). *PROMETHEE-GAIA. Une Méthodologie d'Aide à la Décision en Présence de Critères Multiples*. Paris, France: Ellipses. (Cited on pages 26, 31 and 102.)

[Cailloux et al. 2007] O. Cailloux, C. Lamboray, and P. Nemery (2007). *A taxonomy of clustering procedures*. in. *Proceedings of the 66th Meeting of the European Working Group on MCDA*. (Cited on pages vii and 36.)

[Castro et al. 2004] R.M. Castro, M.J. Coates, and R.D. Nowak (2004). *Likelihood based hierarchical clustering*. Trans. Sig. Proc., 52(8), 2308–2321. (Cited on page 16.)

[Chapelle et al. 2006] O. Chapelle, B. Schölkopf, and A. Zien (2006). *Semi-Supervised Learning*. Cambridge, MA: MIT Press. (Cited on page 5.)

[Charon and Hudry 2010] I. Charon and O. Hudry (2010). *An updated survey on the linear ordering problem for weighted or unweighted tournaments*. Annals OR, 175(1), 107–158. (Cited on pages 32 and 54.)

[Cherng and Lo 2001] J.-S. Cherng and M.-J. Lo (2001). *A Hypergraph Based Clustering Algorithm for Spatial Data Sets*. in. *Proceedings of the 2001 IEEE International Conference on Data Mining*, ICDM '01, pages. 83–90, Washington, DC, USA: IEEE Computer Society. (Cited on page 16.)

[Choquet 1953] G. Choquet (1953). *Theory of capacities*. Annales de l'Institut Fourier, 5, 131–295. (Cited on page 25.)

[Condorcet 1785] M. Condorcet (1785). *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Paris: Imprimerie Royale. (Cited on page 21.)

[Bana e Costa 1996] C. A. Bana e Costa (1996). *Les problématiques de l'aide à la décision : vers l'enrichissement de la trilogie choix-tri-rangement*. RAIRO/Recherche Opérationnelle, 30(2), 191–216. (Cited on page 22.)

[Dahlhaus 2000] E. Dahlhaus (2000). *Parallel algorithms for hierarchical clustering and applications to split decomposition and parity graph recognition*. J. Algorithms, 36(2), 205–240. (Cited on page 16.)

[Dantzig and Orchard-Hays 1953] G.B. Dantzig and W. Orchard-Hays (1953). *The Product Form for the Inverse in the Simplex Method*: Defense Technical Information Center. (Cited on page 28.)

[Dempster et al. 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin (1977). *Maximum likelihood from incomplete data via the EM algorithm*. Journal of the Royal Statistical Society, Series B, 39(1), 1–38. (Cited on page 17.)

[Despotis et al. 1990] D. K. Despotis, D. Yanacopoulos, and C. Zopounidis (1990). *A review of the UTA multicriteria method and some improvements*. Foundations of Computing and Decision Sciences, 15(2), 63–76. (Cited on page 29.)

[Devaud et al. 1980] J. M. Devaud, G. Groussaud, and E. Jacquet-Lagreze (1980). *UTADIS: Une methode de construction de fonctions d'utilite additives rendant compte de jugements globaux*. in. *European working group on MCDA, Bochum , Germany*. (Cited on page 29.)

[Dias and Mousseau 2003] L. C. Dias and V. Mousseau (2003). *IRIS: A DSS for Multiple Criteria Sorting Problems*. Journal of Multi-Criteria Decision Analysis, 12, 285–298. (Cited on page 29.)

[Dias and Mousseau 2006] L. C. Dias and V. Mousseau (2006). *Inferring Electre's veto-related parameters from outranking examples*. European Journal of Operational Research, 170(1), 172–191. (Cited on pages 29 and 69.)

[Dice 1945] L. Dice (1945). *Measures of the amount of ecologic association between species*. Journal of Ecology, 26, 297–302. (Cited on page 11.)

[Dixon 1979] J. Dixon (1979). *Pattern recognition with partly missing data*. IEEE Transactions on Systems, Man and Cybernetics, 9, 617–621. (Cited on page 9.)

[Doumpos and Zopounidis 1998] M. Doumpos and C. Zopounidis (1998). *The use of the preference disaggregation analysis in the assessment of financial risks*. Fuzzy Economic Review, 3(1), 39–57. (Cited on page 31.)

[Duda et al. 2001] R. O. Duda, P. E. Hart, and D. G. Stork (2001). *Pattern Classification*. New York: Wiley, 2. ed. (Cited on pages 4 and 9.)

[Dunn 1973] J. C. Dunn (1973). *A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters*. Journal of Cybernetics, 3(3), 32–57. (Cited on page 13.)

[Eisen et al. 1998] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein (1998). *Cluster analysis and display of genome-wide expression patterns*. Proc. National Academy of Science USA, 95(25), 14863–14868. (Cited on page 10.)

[Eschrich et al. 2003] S. Eschrich, J. Ke, L. O. Hall, and D. B. Goldgof (2003). *Fast accurate fuzzy clustering through data reduction*. Fuzzy Systems, IEEE Transactions on, 11(2), 262–270. (Cited on page 13.)

[Ester et al. 1996] M. Ester, H.-P. Kriegel, S. Jörg, and X. Xu (1996). *A density-based algorithm for discovering clusters in large spatial databases with noise*. in. *KDD*, pages. 226–231. (Cited on page 16.)

[Everitt et al. 2001] B.S. Everitt, S. Landau, and M. Leese (2001). *Cluster Analysis*. A Hodder Arnold Publication: Hodder Arnold. (Cited on pages v, 4, 11, 14 and 42.)

[Fernandez et al. 2010] E. Fernandez, J. Navarro, and S. Bernal (2010). *Handling multicriteria preferences in cluster analysis*. European Journal of Operational Research, 202(3), 819–827. (Cited on pages vii, 36, 46, 51, 62 and 79.)

[Figueira et al. 2005] J. Figueira, V. Mousseau, and B. Roy (2005). *ELECTRE methods*. Multiple criteria decision analysis: State of the art surveys, pages. 133–153. (Cited on page 31.)

[Figueira et al. 2004] J. R. Figueira, Y. de Smet, and J.-P. Brans (2004). *MCDA methods for sorting and clustering problems: Promethee TRI and Promethee CLUSTER*. Technical Report. TR/SMG/2004-002, SMG, Université Libre de Bruxelles. (Cited on page 36.)

[Fishburn 1964] P. C. Fishburn (1964). *Decision and value theory*. no. 10 in-series. *Publications in operations research*. New York: Wiley. (Cited on page 24.)

[Fishburn 1965] P. C. Fishburn (1965). *Independence in Utility Theory with Whole Product Sets*. Operations Research, 13, 28–45. (Cited on page 24.)

[Fishburn 1970] P. C. Fishburn (1970). *Utility Theory for Decision Making*: Wiley, New York. (Cited on page 24.)

[Fisher 1987] D. Fisher (1987). *Knowledge Acquisition via Incremental Conceptual Clustering*. Machine Learning, 2, 139–172. (Cited on page 176.)

[Fisher 1958] W. Fisher (1958). *On Grouping for Maximum Homogeneity*. Journal of the American Statistical Association, 53(284). (Cited on page 78.)

[Forgy 1965] E. Forgy (1965). *Cluster analysis of multivariate data: efficiency versus interpretability of classifications*. Biometrics, 21, 768–780. (Cited on page 13.)

[Frank and Todeschini 1994] I.E. Frank and R. Todeschini (1994). *The Data Analysis Handbook*. Data Handling in Science and Technology: Elsevier Science. (Cited on page 16.)

[Gantz and Reinsel 2011] J. Gantz and D. Reinsel (2011). *Extracting Value from Chaos*. Research report, IDC, Framingham, MA. (Cited on page 4.)

[Garey and Johnson 1979] M.R. Garey and D.S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of Np-Completeness*. A Series of Books in the Mathematical Sciences: W. H. Freeman. (Cited on page 12.)

[Glover 1989] F. Glover (1989). *Tabu Search - Part I*. INFORMS Journal on Computing, 1(3), 190–206. (Cited on pages 39, 96 and 98.)

[Glover 1990] F. Glover (1990). *Tabu Search - Part II*. INFORMS Journal on Computing, 2(1), 4–32. (Cited on pages 39, 96 and 98.)

[Gonzalez 1982] T. Gonzalez (1982). *On the computational complexity of clustering and related problems*. in. R.F. Drenick and F. Kozin (Editors.): *System Modeling and Optimization*, vol. 38 of Series. *Lecture Notes in Control and Information Sciences*, pages. 174–182: Springer Berlin Heidelberg. (Cited on pages 12 and 78.)

[Goodall 1966] D. Goodall (1966). *A new similarity index based on probability*. Biometrics, 22, 882–907. (Cited on page 16.)

[Gower 1971] J. Gower (1971). *A general coefficient of similarity and some of its properties*. Biometrics, 27, 857–872. (Cited on page 11.)

[Grabisch et al. 2008] M. Grabisch, I. Kojadinovic, and P. Meyer (2008). *A review of methods for capacity identification in Choquet integral based multi-attribute utility theory: Applications of the Kappalab R package*. European Journal of Operational Research, 186(2), 766–785. (Cited on page 29.)

[Greco et al. 2010] S. Greco, J. Figueira, R. Slowinski, and B. Roy (2010). *ELECTRE Methods: Main Features and Recent Developments*, pages. 51–89. Berlin: Springer. (Cited on pages 24 and 41.)

[Guha et al. 1998] S. Guha, R. Rastogi, and K. Shim (1998). *CURE: an efficient clustering algorithm for large databases*. SIGMOD Rec., 27(2), 73–84. (Cited on pages 15 and 176.)

[Guha et al. 2000] S. Guha, R. Rastogi, and K. Shim (2000). *ROCK: a robust clustering algorithm for categorical attributes*. Inf. Syst., 25(5), 345–366. (Cited on page 16.)

[Hagen and Kahng 1992] L. W. Hagen and A. B. Kahng (1992). *New spectral methods for ratio cut partitioning and clustering.*. IEEE Trans. on CAD of Integrated Circuits and Systems, 11(9), 1074–1085. (Cited on page 17.)

[Hertszprung 1911] E. Hertszprung (1911). *Uber die Verwendung photographischer effektiver Wellenlaengen zur Bestimmung von Farbenaequivalenten*. Publikationen des Astrophysikalischen Observatoriums zu Potsdam, 63. (Cited on page 6.)

[Hugonnard and Roy 1982] J. C. Hugonnard and B. Roy (1982). *Le plan d'extension du métro en banlieue parisienne, un cas type d'application de l'analyse multicritère*. Les Cahiers Scientifiques de la Revue Transports, (6), 77–108. 1er trimestre. (Cited on page 32.)

[Jaccard 1908] P. Jaccard (1908). *Nouvelles recherches sur la distribution florale*. Bulletin de la Sociète Vaudense des Sciences Naturelles, 44, 223–270. (Cited on page 11.)

[Jacquet-Lagrèze 1995] E. Jacquet-Lagrèze (1995). *An application of the UTA discriminant model for the evaluation of sc R&D projects*. in. P. Pardalos, Y. Siskos, and C. Zopounidis (Editors.): *Advances in Multicriteria Analysis*, Nonconvex Optimization and its Applications, pages. 203–211. Dordrecht: Kluwer Academic. (Cited on page 31.)

[Jacquet-Lagrèze and Siskos 1982] E. Jacquet-Lagrèze and Y. Siskos (1982). *Assessing a set of additive utility functions for multicriteria decision making: the UTA method*. European Journal of Operational Research, 10, 151–164. (Cited on pages 28 and 29.)

[Jain 2010] A. Jain (2010). *Data clustering: 50 years beyond K-means*. Pattern Recognition Letters, 31(8), 651–666. (Cited on pages vi, ix, 5, 40, 41, 53, 63 and 175.)

[Jain et al. 1999] A. Jain, M. Murty, and P. Flynn (1999). *Data clustering: a review*. ACM Computer Survey, 31(3), 264–323. (Cited on page 174.)

[Jain and Dubes 1988] A. K. Jain and R. C. Dubes (1988). *Algorithms for clustering data*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc. (Cited on pages 6, 7, 8, 14 and 68.)

[Karger 1993] D. R. Karger (1993). *Global min-cuts in RNC, and other ramifications of a simple min-cut algorithm*. in. *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, SODA '93, pages. 21–30, Philadelphia, PA, USA: Society for Industrial and Applied Mathematics. (Cited on page 17.)

[Karypis et al. 1999] G. Karypis, E.-H. Han, and V. Kumar (1999). *Chameleon: hierarchical clustering using dynamic modeling*. Computer, 32(8), 68–75. (Cited on page 16.)

[Kashima et al. 2003] H. Kashima, K. Tsuda, and A. Inokuchi (2003). *Marginalized Kernels Between Labeled Graphs*. in. T. Fawcett and N. Mishra (Editors.): *Proceedings of the 20th International Conference on Machine Learning (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages. 321–328: AAAI Press, Chicago, IL, USA. (Cited on page 13.)

[Kaufman and Rousseeuw 1990] L. Kaufman and P. J. Rousseeuw (1990). *Finding groups in data: an introduction to cluster analysis*. New York: John Wiley and Sons. (Cited on pages 10, 11, 13, 15 and 42.)

[Kaufman and Rousseeuw 2005] L. Kaufman and P. J. Rousseeuw (2005). *Finding Groups in Data: An Introduction to Cluster Analysis (Wiley Series in Probability and Statistics)*: Wiley-Interscience. (Cited on page 14.)

[Keeney and Raiffa 1976] R. L. Keeney and H. Raiffa (1976). *Decisions with multiple objectives: Preferences and value tradeoffs*: J. Wiley, New York. (Cited on pages vi, 21, 24, 26, 31 and 102.)

[Kemeny 1959] J. G. Kemeny (1959). *Mathematics without numbers*. Daedalus, 88, 575–591. (Cited on page 32.)

[Kirkpatrick et al. 1983] S. Kirkpatrick, C. Gelatt, and M. Vecchi (1983). *Optimization by Simulated Annealing*. Science, 220(4598), 671–680. (Cited on pages 96 and 98.)

[Kiss et al. 1994] L. N. Kiss, J.-M. Martel, and R. Nadeau (1994). *ELECCALC-an interactive software for modelling the decision maker's preferences*. Decis. Support Syst., 12(4-5), 311–326. (Cited on page 29.)

[Kohonen et al. 2001] T. Kohonen, M. R. Schroeder, and T. S. Huang (Editors.) (2001). *Self-Organizing Maps*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 3rd ed. (Cited on pages viii and 80.)

[Krishnanand and Ghose 2005] K. Krishnanand and D. Ghose (2005). *Multimodal Function Optimization using a Glowworm Metaphor with Applications to Collective Robotics*. in. B. Prasad (Editor.): *Proceedings of the 2nd Indian International Conference on Artificial Intelligence, Pune, India, December 20-22, 2005*, pages. 328–346. (Cited on pages 91 and 114.)

[Lamboray 2007a] C. Lamboray (2007a). *A comparison between the prudent order and the ranking obtained with Borda's, Copeland's, Slater's and Kemeny's rules*. Mathematical Social Sciences, 54(1), 1–16. (Cited on page 33.)

[Lamboray 2007b] C. Lamboray (2007b). *Prudent ranking rules: theoretical contributions and applications*. phdthesis. (Cited on page 32.)

[Lance and Williams 1967] G. N. Lance and W. T. Williams (1967). *A general theory of classificatory sorting strategies 1. Hierarchical systems*. The Computer Journal, 9(4), 373–380. (Cited on page 15.)

[Leung et al. 2000] Y. Leung, J.-S. Zhang, and Z.-B. Xu (2000). *Clustering by Scale-Space Filtering*. IEEE Trans. Pattern Anal. Mach. Intell., 22(12), 1396–1410. (Cited on page 16.)

[Li and Biswas 2002] C. Li and G. Biswas (2002). *Unsupervised Learning with Mixed Numeric and Nominal Data*. IEEE Trans. on Knowl. and Data Eng., 14(4), 673–690. (Cited on page 16.)

[Li and McCallum 2006] W. Li and A. McCallum (2006). *Pachinko allocation: DAG-structured mixture models of topic correlations*. in. *Proceedings of the 23rd international conference on Machine learning*, ICML '06, pages. 577–584, New York, NY, USA: ACM. (Cited on page 17.)

[Li 1990] X. Li (1990). *Parallel Algorithms for Hierarchical Clustering and Cluster Validity*. IEEE Trans. Pattern Anal. Mach. Intell., 12(11), 1088–1092. (Cited on page 16.)

[Linde et al. 1980] Y. Linde, A. Buzo, and R. M. Gray (1980). *An Algorithm for Vector Quantizer Design*. IEEE Transactions on Communications, 28, 84–95. (Cited on page 13.)

[Little and Rubin 2002] R.J.A. Little and D.B. Rubin (2002). *Statistical analysis with missing data*. Wiley series in probability and mathematical statistics: Wiley. (Cited on page 9.)

[Lloyd 1982] S. P. Lloyd (1982). *Least squares quantization in pcm*. IEEE Transactions on Information Theory, 28, 129–137. (Cited on page 12.)

[Macqueen 1967] J. B. Macqueen (1967). *Some methods of classification and analysis of multivariate observations*. in. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages. 281–297. (Cited on pages viii, 12 and 13.)

[Mahajan et al. 2009] M. Mahajan, P. Nimbhorkar, and K. Varadarajan (2009). *The Planar k-Means Problem is NP-Hard*. in. *Proceedings of the 3rd International Workshop on Algorithms and Computation*, WALCOM '09, pages. 274–285, Berlin, Heidelberg: Springer-Verlag. (Cited on page 78.)

[Mao and Jain 1996] J. Mao and A. K. Jain (1996). *A self-organizing network for hyperellipsoidal clustering (HEC)*. Trans. Neur. Netw., 7(1), 16–29. (Cited on page 13.)

[Mareschal and Brans 1988] B. Mareschal and J. P. Brans (1988). *Geometrical representations for MCDA*. European Journal of Operational Research, 34, 69–77. (Cited on page 32.)

[Massaglia and Ostanello 1991] R. Massaglia and A. Ostanello (1991). *N-tomic: a support system for multicriteria segmentation problems*. in. P. Korhonen, A. Lewandowski, and J. Wallenius (Editors.): *Multiple Criteria Decision Support*, pages. 167–174: Springer Verlag, LNEMS 356, Berlin. (Cited on page 31.)

[Maystre et al. 1994] L. Y. Maystre, J. Pictet, and J. Simos (1994). *Méthodes multicritères ELECTRE. Description, conseils pratiques et cas d'application à la gestion environnementale*. Lausanne: Presses Polytechniques et Universitaires Romandes. (Cited on page 30.)

[Meila and Shi 2001] M. Meila and J. Shi (2001). *A Random Walks View of Spectral Segmentation*. in. *AI and STATISTICS (AISTATS)*. (Cited on page 17.)

[Meyer et al. 2008] P. Meyer, J.-L. Marichal, and R. Bisdorff (2008). *Disaggregation of Bipolar-Valued Outranking Relations.*. in. L. T. H. An, P. Bouvry, and T. P. Dinh (Editors.): *MCO*, vol. 14 of Series. *Communications in Computer and Information Science*, pages. 204–213: Springer. (Cited on pages 29 and 138.)

[Meyer and Olteanu 2012] P. Meyer and A.-L. Olteanu (2012). *Preferentially ordered clustering*. in. V. Torra, Y. Narukawa, B. López, and M. Villaret (Editors.): *Proceedings of the 9th International Conference on Modelling Decisions for Artificial Intelligence*, pages. 77–88. (Cited on page 36.)

[Meyer and Olteanu 2013] P. Meyer and A.-L. Olteanu (2013). *Formalizing and solving the problem of clustering in MCDA*. European Journal of Operational Research, 227(3), 494–502. (Cited on pages 36 and 88.)

[Moon and Moser 1965] J. Moon and L. Moser (1965). *On cliques in graphs*. Israel Journal of Mathematics, 3(1), 23–28. (Cited on page 90.)

[Moscarola and Roy 1977] J. Moscarola and B. Roy (1977). *Procédure automatique d'examen de dossiers fondée sur une segmentation trichotomique en présence de critères multiples*. RAIRO Recherche Opérationnelle, 11(2), 145–173. (Cited on page 31.)

[Mousseau 2005] V. Mousseau (2005). *A general framework for constructive learning preference elicitation in multiple criteria decision aid*. Cahier du Lamsade 223, Université Paris Dauphine. (Cited on page 28.)

[Mousseau and Dias 2004] V. Mousseau and L. Dias (2004). *Valued outranking relations in ELECTRE providing manageable disaggregation procedures*. European Journal of Operational Research, 156(2), 467–482. (Cited on page 29.)

[Mousseau et al. 2001] V. Mousseau, J. R. Figueira, and J.-P. Naux (2001). *Using assignment examples to infer weights for ELECTRE TRI method: Some experimental results.*. European Journal of Operational Research, 130(2), 263–275. (Cited on page 29.)

[Mousseau and Slowinski 1998] V. Mousseau and R. Slowinski (1998). *Inferring an ELECTRE TRI Model from Assignment Examples*. J. of Global Optimization, 12(2), 157–174. (Cited on page 29.)

[Nemery and de Smet 2005] P. Nemery and Y. de Smet (2005). *Multicriteria ordered clustering*. Technical Report. TR/SMG/2005-003, Universit'e Libre de Bruxelles/SMG. (Cited on pages 36, 38, 51 and 79.)

[Ng et al. 2001] A. Y. Ng, M. I. Jordan, and Y. Weiss (2001). *On spectral clustering: Analysis and an algorithm*. in. *Advances in Neural Information Processing Systems 14*, pages. 849–856: MIT Press. (Cited on page 17.)

[Okabe et al. 2000] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu (2000). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Series in Probability and Statistics: John Wiley and Sons, Inc., 2nd ed. ed. (Cited on page 16.)

[Olson 1995] C. F. Olson (1995). *Parallel algorithms for hierarchical clustering*. Parallel Comput., 21(8), 1313–1325. (Cited on page 16.)

[Orlov 2002] M. Orlov (2002). *Efficient Generation of Set Partitions*. (Cited on page 78.)

[Pampalk et al. 2003] E. Pampalk, S. Dixon, and G. Widmer (2003). *On the Evaluation of Perceptual Similarity Measures For Music*. in. *Proc. of the 6th Int. Conference on Digital Audio Effects (DAFx-03)*, London, UK. (Cited on page 7.)

[Peña et al. 1999] J. Peña, J. Lozano, and P. Larrañaga (1999). *An empirical comparison of four initialization methods for the K-Means algorithm*. Pattern Recognition Letters, 20(10), 1027–1040. (Cited on page 13.)

[Pelleg and Moore 2000] D. Pelleg and A. W. Moore (2000). *X-means: Extending K-means with Efficient Estimation of the Number of Clusters*. in. *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pages. 727–734, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. (Cited on page 13.)

[Perny 1998] P. Perny (1998). *Multicriteria Filtering Methods Based on Concordance/Non-Discordance principles*. Annals of Operations Research, 80, 137–167. (Cited on pages viii, 31 and 142.)

[Rajasekaran 2005] S. Rajasekaran (2005). *Efficient Parallel Hierarchical Clustering Algorithms*. IEEE Trans. Parallel Distrib. Syst., 16(6), 497–502. (Cited on page 16.)

[Rocha and Dias 2008] C. Rocha and L. C. Dias (2008). *An algorithm for ordinal sorting based on ELECTRE with categories defined by examples*. J. of Global Optimization, 42(2), 255–277. (Cited on page 29.)

[Rocha et al. 2012] Clara Rocha, Luis C. Dias, and Isabel Dimas (2012). *Multicriteria Classification with Unknown Categories: A Clustering–Sorting Approach and an Application to Conflict Management*. Journal of Multi-Criteria Decision Analysis. (Cited on page 36.)

[Rogers and Tanimoto 1960] D. Rogers and T. Tanimoto (1960). *A computer program for classifying plants*. Science, 132, 1115–1118. (Cited on page 11.)

[Roubens and Vincke 1985] M. Roubens and P. Vincke (1985). *Preference Modeling*: LNEMS 250, Springer Verlag, Berlin. (Cited on pages 27, 42 and 62.)

[Roy 1968] B. Roy (1968). *Classement et choix en présence de points de vue multiples: La méthode ELECTRE*. Revue Francaise d'Informatique et de Recherche Opérationnelle, 8, 57–75. (Cited on pages 21, 30, 42 and 44.)

[Roy 1971a] B. Roy (1971a). *La méthode ELECTRE II*. Note de Travail n. 142, METRA, Direction Scientifique. (Cited on page 31.)

[Roy 1971b] B. Roy (1971b). *Problems and methods with multiple objective functions*. Mathematical Programming, 1(2), 239–266. (Cited on page 24.)

[Roy 1978] B. Roy (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. (Cited on page 32.)

[Roy 1985] B. Roy (1985). *Méthodologie multicritère d'aide à la décision*. Paris: Economica. (Cited on pages vi, 22, 30, 31, 53 and 54.)

[Roy 1988] B. Roy (1988). *Des critères multiples en recherche opérationnelle: pourquoi ?*. in. *Operational research '87 (Buenos Aires, 1987)*, pages. 829–842. Amsterdam: North-Holland. (Cited on page 20.)

[Roy 1990] B. Roy (1990). *The Outranking Approach and the Foundations of ELECTRE methods*. in. C. A. Bana e Costa (Editor.): *Readings in Multiple Criteria Decision Aid*, pages. 155–183: Springer-Verlag, Berlin. (Cited on page 27.)

[Roy 1993] B. Roy (1993). *Decision science or decision-aid science?*. European Journal of Operational Research, 66, 184–203. (Cited on pages vi, 26, 102 and 136.)

[Roy 1996] B. Roy (1996). *Multicriteria Methodology for Decision Aiding*. Dordrecht: Kluwer Academic. (Cited on pages 24 and 41.)

[Roy and Bouyssou 1993] B. Roy and D. Bouyssou (1993). *Aide Multicritère à la Décision : Méthodes et Cas*. Paris: Economica. (Cited on pages viii, 30, 54, 143 and 157.)

[Roy and Skalka 1984] B. Roy and J. M. Skalka (1984). *ELECTRE IS: Aspects méthodologiques et guide d'utilisation*. Technical Report., Document du LAMSADE N^o 30, Université Paris-Dauphine, Paris. (Cited on pages 30 and 44.)

[Saaty 1980] T. L. Saaty (1980). *The Analytic Hierarchy Process, Planning, Piority Setting, Resource Allocation*. New york: McGraw-Hill. (Cited on page 28.)

[Samuelson 1938] Samuelson (1938). *A Note on the Pure Theory of Consumer's Behaviour*. Economica, 51(17). (Cited on page 21.)

[Schölkopf et al. 1998] B. Schölkopf, A. Smola, E. Smola, and K.-R. Müller (1998). *Nonlinear Component Analysis as a Kernel Eigenvalue Problem*. Neural Computation, 10, 1299–1319. (Cited on page 14.)

[Schwarz 1978] G. Schwarz (1978). *Estimating the dimension of a model*. Annals of Statistics, 6, 461–464. (Cited on page 7.)

[Selim and Ismail 1984] S. Z. Selim and M. A. Ismail (1984). *K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality*. IEEE Trans. Pattern Anal. Mach. Intell., 6(1), 81–87. (Cited on page 13.)

[Shi and Malik 2000] J. Shi and J. Malik (2000). *Normalized Cuts and Image Segmentation*. IEEE Trans. Pattern Anal. Mach. Intell., 22(8), 888–905. (Cited on page 17.)

[Simon 1957] H. A. Simon (1957). *A behavioural model of Rational Choice*. in. *Models of man: social and rational; mathematical essays on rational human behavior in a social setting*, pages. 241–260: J. Wiley, New York. (Cited on page 21.)

[Siskos and Despotis 1989] J. Siskos and D. K. Despotis (1989). *A DSS oriented method for multiobjective linear programming problems*. Decis. Support Syst., 5(1), 47–55. (Cited on page 29.)

[Siskos 1985] Y. Siskos (1985). *Analyses de regression et programmation lineaire*. Revue de Statistique Appliquée, 23(2), 41–55. (Cited on page 28.)

[Siskos et al. 2005] Y. Siskos, E. Grigoroudis, and N. F. Matsatsinis (2005). *UTA Methods*. in. J. Figueira, S. Greco, and M. Ehrgott (Editors.): *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages. 297–344. Boston, Dordrecht, London: Springer Verlag. (Cited on page 29.)

[Siskos and Yanacopoulos 1985] Y. Siskos and D. Yanacopoulos (1985). *UTA STAR - an ordinal regression method for building additive value functions*. Investigacao Operational, 5, 39–53. (Cited on pages 28 and 29.)

[Slater 1961] P. Slater (1961). *Inconsistencies in a schedule of paired comparisons*. Biometrika, 48(3-4), 303–312. (Cited on page 32.)

[de Smet and Eppe 2009] Y. de Smet and S. Eppe (2009). *Relational Multicriteria Clustering: The Case of Binary Outranking Matrices*. in. M. et al. Ehrgott (Editor.): *Evolutionary Multi-Criterion Optimization. Fifth international conference, EMO 2009. Proceedings*, vol. 5467 of Series. *Lecture Notes in Computer Science*, pages. 380–392: Springer Berlin. (Cited on pages viii, 36, 37, 64, 65, 79 and 125.)

[de Smet and Guzman 2004] Y. de Smet and L. Guzman (2004). *Towards Multicriteria Clustering: an extension of the k-means algorithm*. European Journal of Operational Research, 158(2), 390–398. (Cited on pages 36 and 79.)

[Smith and Jain 1984] S. P. Smith and A. K. Jain (1984). *Testing for Uniformity in Multidimensional Data.*. IEEE Trans. Pattern Anal. Mach. Intell., 6(1), 73–81. (Cited on page 7.)

[Sokal and Sneath 1963] R.R. Sokal and P.H.A. Sneath (1963). *Principles of numerical taxonomy*. Series of books in biology: W. H. Freeman. (Cited on page 11.)

[Steinbach et al. 2000] M. Steinbach, G. Karypis, and V. Kumar (2000). *A comparison of document clustering techniques*. in. *In KDD Workshop on Text Mining*. (Cited on pages 13 and 175.)

[Steinhaus 1956] H. Steinhaus (1956). *Sur la division des corp materiels en parties*. Bull. Acad. Polon. Sci, 1, 801–804. (Cited on page 12.)

[Stevens 1946] S. S. Stevens (1946). *On the Theory of Scales of Measurement*. Science, 103(2684), 677–680. (Cited on page 8.)

[Sugeno 1974] M. Sugeno (1974). *Theory of Fuzzy Integrals and its Applications*. phdthesis., Tokyo, Japan. (Cited on page 25.)

[Tabachnick and Fidell 2006] B. G. Tabachnick and L. S. Fidell (2006). *Using Multivariate Statistics (5th Edition)*: Allyn & Bacon, Inc. (Cited on page 4.)

[Talbi 2009] E. Talbi (2009). *Metaheuristics: From Design to Implementation*. Wiley Series on Parallel and Distributed Computing: Wiley. (Cited on pages 90, 92, 96, 118 and 149.)

[Tibshirani et al. 2001] R. Tibshirani, G. Walther, and T. Hastie (2001). *Estimating the number of clusters in a dataset via the Gap statistic*. Journal of the Royal Statistical Society, B, 411–423. (Cited on page 13.)

[Tsoukiàs 2007] A. Tsoukiàs (2007). *On the concept of decision aiding process: an operational perspective*. 154(1), 3–27. (Cited on pages 22, 23, 41 and 53.)

[Tukey 1977] J. W. Tukey (1977). *Exploratory Data Analysis*: Addison-Wesley. (Cited on page 4.)

[USEPAa] United States Environmental Protection Agency. *Toxics Release Inventory*. http://http://www.epa.gov/tri/. (Cited on pages ix, 192 and 193.)

[USEPAb] United States Environmental Protection Agency. *TRI-Chemical Hazard Information Profiles*. http://www.epa.gov/tri/tri-chip/. (Cited on page 192.)

[Valls 2002] A. Valls (2002). *ClusDM: a Multiple Criteria Decision Method for heterogeneous data sets*. phdthesis. (Cited on page 63.)

[Valls et al. 2009] A. Valls, M. Batet, and E. M. López (2009). *Using expert's rules as background knowledge in the ClusDM methodology*. European Journal of Operational Research, 195(3), 864–875. (Cited on page 36.)

[Valls and Torra 2000] A. Valls and V. Torra (2000). *Using classification as an aggregation tool in MCDM*. Fuzzy Sets Syst., 115(1), 159–168. (Cited on page 36.)

[Venn 1880] J. Venn (1880). *On the diagrammatic and mechanical representation of propositions and reasonings*. The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science, 10(58), 1–18. (Cited on page 14.)

[Von Neumann and Morgenstern 1944] J. Von Neumann and O. Morgenstern (1944). *Theory of Games and Economic Behavior*: Princeton University Press, 3 ed. (Cited on page 21.)

[Welling et al. 2004] M. Welling, M. Rosen-Zvi, and G. E. Hinton (2004). *Exponential Family Harmoniums with an Application to Information Retrieval*. in. *Advances in Neural Information Processing Systems 17*. (Cited on page 17.)

[Xu and Wunsch II 2008] R. Xu and D. Wunsch II (2008). *Clustering*. IEEE Press Series on Computational Intelligence: John Wiley & Sons. (Cited on pages 6 and 8.)

[Yu 1992] W. Yu (1992). *ELECTRE TRI : Aspects méthodologiques et manuel d'utilisation*. Document du LAMSADE no 74, Université Paris-Dauphine. (Cited on pages viii, 30, 54, 143 and 157.)

[Zhang et al. 1996] T. Zhang, R. Ramakrishnan, and M. Livny (1996). *BIRCH: an efficient data clustering method for very large databases*. in. Jennifer Widom (Editor.): *SIGMOD '96: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages. 103–114: ACM Press. (Cited on pages 15 and 175.)

[Zopounidis and Doumpos 1999] C. Zopounidis and M. Doumpos (1999). *A Multicriteria Decision Aid Methodology for Sorting Decision Problems: The Case of Financial Distress*. Computational Economics, 14(3), 197–218. (Cited on page 31.)

[Zopounidis and Doumpos 2000] C. Zopounidis and M. Doumpos (2000). *Building additive utilities for multi-group hierarchical discrimination: The M.H.DIS method*. Optimization Methods and Software, 14(3), 219–240. (Cited on page 31.)

[Zopounidis and Doumpos 2002] C. Zopounidis and M. Doumpos (2002). *Multicriteria classification and sorting methods: A literature review*. European Journal of Operational Research, 138(2), 229–246. (Cited on page 31.)

[Zubin 1938] T. Zubin (1938). *A technique for measuring like-mindedness*. Journal of Abnormal and Social Psychology, 33, 508–516. (Cited on page 11.)