Université Libre de Bruxelles

Master Thesis

Two approaches to reduce the rank reversal occurrences in the PROMETHEE II method

Author:
Gilles Dejaegere

Supervisor:
Yves De Smet

Mémoire présenté en vue de l'obtention du diplôme
d'Ingénieur Civil Informaticien à finalité spécialisée
Ecole Polytechnique de Bruxelles

year 2016-2017

Acknowledgements

I would first of all like to express my gratitude to my supervisor, Professor Yves De Smet, whose availability and advice throughout the year has been an extremely valuable help for the accomplishment of this work as well as for my personal development.

I would also like to thank all the members of the SMG unit for their helpfull feedback and their constructive comments.

I address my warm thanks to my familly, particularily my father, for his multiple rereading, and my mother, which is probabily having a hard time supporting me but espacilly my brothers.

Finally, I would like to extend my gratitude to all people who, closely or remotely, have been involved in this work, in this year, or even in my studies.

Abstract

Some practical decision problems are complex and need to be formalised to be efficiently managed. The decision aid discipline consists in modeling these problems. More specifically, multi-criteria decision aid methods model them using different conflicting criteria.

Different multi-criteria decision aid methods exist but the study performed in this master thesis focus on the PROMETHEE methods.

Like other multi-criteria decision aid methods, PROMETHEE II suffers from the rank reversal phenomenon. This phenomenon consists in the relative ordering between two alternatives being dependant on some other alternatives of the problem. The possibility of suffering from rank reversals can be undesirable due to manipulation threats. Therefore, two modifications of the PROMETHEE II method aimed at avoiding this phenomon have been studied.

The first modification analysed, ROBUST PROMETHEE, is based on the repetition of PROMETHEE II on a large number of samplings of the alternatives. The results of this analysis show that it seems to reduce the number of rank reversals, but its efficiency in this regard strongly depends on the decision problem and on the size of the subsets on which PROMETHEE II is applied. This method introduces two new parameters: the number of iterations and the size of the samplings. Some bounds on the minimal value of these parameters have been found. Finally, it has been shown that some irrational results could occur (such as the non respect of the dominance relation), but with a very low probability.

A second method, REFERENCED PROMETHEE has also been analysed. It is based on the comparison using PROMETHEE II of each alternative with a set of predefined reference profiles. Since each alternative is compared with the same fixed set, this method does not suffer from the rank reversal phenomenon. Nevertheless, it comes with the additional cost of having to find a set of fixed references. It has been seen in this work that the rankings obtained are highly dependent on this set of reference profiles.

Several properties of these sets have been found. First, it is shown that a small number of references is sufficient to discriminate large set of alternatives. Then, it is shown that it is generally even possible to replicate the rankings produced by PROMETHEE II with REFERENCED PROMETHEE using only this small quantity of reference profiles. Finally, two approaches for defining these sets of reference profiles are detailed and analysed. The efficiency of each of them however strongly depends on the decision problem and it is therefore not guaranteed that they will yield set of reference profiles which lead to a satisfactory ranking.

keywords : multi-criteria decision aid, PROMETHEE*, rank reversals,* ROBUST PROMETHEE*,* REFERENCED PROMETHEE*, reference profiles.*

Résumé

Certains problèmes d'aide à la décision sont trop complexes et doivent être formalisés pour pouvoir être appréhendés efficacement. La discipline de l'aide à la décision consiste à modéliser ces problèmes.
Plus précisément, les méthodes d'aide à la décision multicritères consistent à modéliser ces problèmes en utilisant plusieurs critères de natures souvent conflictuelles.

Différentes méthodes d'aide à la décision multi-critères existent, cependant seules les méthodes PROMETHEE seront abordées dans ce travail. L'une des méthodes de la famille PROMETHEE est PROMETHEE II qui a pour but d'ordonner l'ensemble fini des solutions du problème (ces solutions sont aussi appelées des alternatives).

Comme d'autres méthodes d'aide à la décision multi-critère, PROMETHEE II souffre de ce qui est appelé l'inversion de rang. Ce phénomène consiste en ce que l'ordre relatif entre deux alternatives dépende d'autres alternatives du problème. La possibilité pour une méthode de subir ces inversions de rang peut être indésirable. En effet, cela signifierait que celle-ci soit susceptible de subir des manipulations de rang via l'ajout ou la modification artificielle de données. C'est pourquoi, deux méthodes dérivées de PROMETHEE II, ayant pour but de réduire ces inversions de rang, sont analysées dans ce travail.

La première modification analysée, ROBUST PROMETHEE, est basée sur la répétition de PROMETHEE II sur une grande quantité de d'échantillons de solutions. Les résultats de cette analyse montrent que celle-ci semble réduire le nombre de d'inversions de rangs. Cependant, son efficacité à réduire cette quantité est fortement dépendante du problème de décision et de la taille des sous-ensembles sur lesquels PROMETHEE II est appliquée.
Cette méthode introduit deux nouveaux paramètres: le nombre d'itérations et la taille des échantillons. Certaines bornes sur la valeur minimale de ces paramètres ont été définies. Enfin, il a été démontré que des résultats irrationnels peuvent être produits, mais avec une très faible probabilité.

Une deuxième méthode, REFERENCED PROMETHEE, a ensuite été analysée. Celle-ci est basée sur la comparaison avec PROMETHEE II de chaque alternative avec un ensemble fixe de profils de référence. Étant donné que chaque alternative est comparée au même ensemble fixe de profils, cette méthode ne souffre pas du phénomène d'inversion de rang. Cependant, un coût additionnel consistant en la nécessité de devoir établir cet ensemble fixe de références est introduit. Il a été observé dans ce travail que le choix de cet ensemble a un impact considérable sur le rangement final obtenu.

Plusieurs propriétés de ces ensembles de profils de référence ont été trouvées. Tout d'abord, il a été constaté qu'un large ensemble d'alternatives pouvait être départagé en utilisant un petit nombre de profils de référence. Ensuite, il a été remarqué qu'il est en général possible de reproduire exactement le classement obtenu par PROMETHEE II avec REFERENCED PROMETHEE en utilisant cette petite quantité de profils de références.

Enfin, deux approches pour définir ces ensembles de profils de références sont détaillées et analysées. L'efficacité de chacune de ces deux approches dépend cependant du problème de décision, et aucune des deux ne garantit de fournir des profils de référence satisfaisants.

mots-clés: aide à la décision multicritères, PROMETHEE, *inversion de rang,* ROBUST PROMETHEE, REFERENCED PROMETHEE

Contents

List of symbols

This page contains a list of the symbols and acronyms used in this work.

PROMETHEE:

A	set of alternatives of the decision problem
n	quantity of alternatives considered
a_i	ith alternative
k	quantity of criteria on which the alternatives are evaluated
$f_c()$	evaluation function of the cth criterion
w_c	weight of the cth criterion
$\mathscr{P}$	set of the preference functions
$P_c()$	preference function of the cth criterion
q, p	indifference and strict preference thresholds of the preference functions

ROBUST PROMETHEE:

R	quantity of repetitions of PROMETHEE II on the samplings
m	size of the samplings

REFERENCED PROMETHEE:

$\mathscr{R}$	set of reference profiles (also called set of references)
p	size of the set of reference profiles
r	one of the reference profiles

Other:

τ_k	Kendall rank correlation coefficient

Chapter 1

Introduction to Multi-Criteria Decision Aid

1.1 Introduction to Decision aid

Every day, any human being has to face numerous choices (or decision problems). Most of them are rational and therefore try to make the choices leading to the most favorable consequences. All these decision problems appear in specific contexts that will be called the *decision problem reality* or more simply, the problem reality. The contexts of decision problems can have very different natures [18][22]: economical, political, industrial, military or even familial or personal.

The decision is not always taken by a unique person. It can as well be taken by larger entities. These can be well defined, such as a board of directors, a council of ministers, a jury, or even a family or can be more vaguely defined such as professional pressure groups, employees of a company or the public opinion [22].

In any cases, the entity, composed by one or more people, in charge of the final decision will be called the *decision maker*.

Most of the decision problems that we encounter in our daily lives are straightforward or of relatively small importance (what should we eat tonight? what movie should I watch?). They can generally be solved naturally and in a *qualitative* way [18]. As the problem is simple and the consequences of a non optimal decision are not dramatic, it is sufficient for the decision maker to use its own experience and its assessments of the problem reality in order to choose a solution that seems optimal to him. These kind of decision problems take most of the time place in a personal or familial context.

Some problems, on the other hand, require a more in depth reflection (should this old nuclear power plant be closed? should I invest my savings in this particular company?).
The decision problem reality in economical or industrial contexts have become too complex to be apprehensible in a purely qualitative manner [18]. Furthermore, the choice of a good or even of the best solution for these problems can be crucial since the consequences of a bad decision can be disastrous. In such cases, the decision maker could want to use a *quantitative* approach, building a model [18] [27]. In this approach, a mathematical model needs to be built. This model, which is an abstraction of the reality, will be used as support for the investigation and the communication between the different actors of the problem [27] [22]. To build this mathematical model, the consequences of each possible decision must be foreseen and quantified.

The discipline dealing with the analysis of the decision realities and the elaboration of the decision models is called operational research. As done in [18], the expression *decision aid* will be used as its signification is closer to the problem described.

It should also be noted that in decision aid, the quantitative and the qualitative approaches are not mutually exclusive. The decision problem reality being generally too complex and chaotic to be entirely modelable, the quantitative approach must be combined with the qualitative one. The quantitative approach will be used to give some insight into the problem to the decision maker, which will always use his own qualitative perception to use these insights.

In the rest of the chapter, the traditional mono-criterion approach will first be detailed. Some of it's drawbacks will then be highlighted, motivating the introduction of the multi-criteria approach that will then be detailed.

1.2 Mono-criterion approach

1.2.1 Formalisation and properties of mono-criterion problems

A mono-criterion decision problem can be formulated in the following way [18]:

$$Opt\{f(a_i) \mid a_i \in A\} \tag{1.1}$$

These classical models where the only one used until the end of the sixties [23]. They cover numerous families of problems and are formalised according to the three following properties.

1. A forms a well-defined set of possible alternatives [23]

The set A of alternatives (also called actions) forms a well-defined set. These are all the possible choices of the decision maker. This set of alternatives can be finite and enumerable, finite but not enumerable or infinite. Here under are examples of decision problem with these kinds of alternative sets:

- **finite and enumerable:** the decision maker must hire one person from a set of n applicants,
- **finite but not enumerable:** the decision maker is facing a *Travelling Salesman Problem*. As the number of possible path increases exponentially, the number of alternatives often becomes far too large to be enumerated,
- **infinite:** the decision maker must choose the amount of money invested in a project.

If the alternatives are not enumerable, the set must be defined by stating the properties (such as linear constraints) which characterise its elements [30].

The set A is also characterized by the following two properties [30]:

- A can be either *stable*, if the set does not change during the decision procedure, or *evolutive*, if, on the other hand, it can change.
  There can be different reasons leading to some changes of the alternatives considered such as for example an evolving decision reality or the elimination during the intermediate steps of the procedure of alternatives that are considered not to be relevant anymore. Consider the already mentioned example of a decision maker which has to hire one candidate from a set of applicants. It is very plausible that this set could vary during the decision process as some candidates could have found a better job elsewhere.
- A can be either *globalised*, if the decision problem consists in the selection of one unique alternative (we say that the alternatives are mutually exclusive), or *fragmented* if it consists in the selection of a subset of one or more alternatives.
  Consider for instance the problem of having to choose two locations from a set of n possible locations where new warehouses would be built. This problem can be seen as fragmented if the alternatives consist in the set of n locations and that two of them must be chosen, but it can also be seen as a globalised problem if the alternatives consist in the $\frac{n(n-1)}{2}$ possible couples of location and that only one must be chosen.

2. The real-valued function $f(.)$ correctly reflects the preferences of the decision maker [23]

The preferences of the decision maker are considered to be correctly represented by the unique criterion $f(.)$. This criterion must therefore synthesise on its own all the objectives of the decision maker and the consequences of each of the alternatives [23] [18].

Of course, this assumption can not always entirely be respected in real life decision problems. For example, a function $f(.)$ representing the profit made with the commercialisation of a product can often only be built by using estimations of the number of products that would be sold. In such cases $f(.)$ could be subject to some subjectivity. This specific kind of subjectivity should be avoided as much as possible as it introduces some imprecisions. The reality should be represented as accurately possible by the evaluation function $f(.)$. Some other kind of subjectivity, which is essential and should therefore not be avoided, will be introduced in the section concerning the multi-criteria approach.

The problem could consist in minimising this criterion (e.g. $f(.)$ represents a cost) or to maximise the criterion (e.g. $f(.)$ represents a profit). Without loss of generality, for the rest of this section we will consider that $f(.)$ is a criterion to maximise.

$$Max\{f(a_i) \mid a_i \in A\} \tag{1.2}$$

3. The decision problem forms a well-formulated mathematical problem [23]

If a mono-criterion decision problem is modeled as (1.2) with A being a finite set, then there exists at least one solution such that:

$$f(a_i^*) \geq f(a_j), \forall a_j \in A \tag{1.3}$$

Such solutions are said to be optimal for the decision problem. These are entirely determined by the modelling of the problem. The process of finding such an optimal solution is a classical optimisation problem, leading to the discovery of some “hidden truth” [23][30]. Indeed, if the decision maker approves the model, he should adopt one of the proposed optimal solutions.

4. Additional properties of the mono-criterion problem

With a mono-criterion problem, more can be done than finding an optimal solution. Indeed, $f(.)$ implies a natural dominance relation (P,I) on every pair of elements of A [18]:

$$\forall a_i, a_j \in A : \begin{cases} f(a_i) & > & f(a_j) & \Leftrightarrow & a_i P a_j \\ f(a_i) & = & f(a_j) & \Leftrightarrow & a_i I a_j \\ f(a_i) & < & f(a_j) & \Leftrightarrow & a_j P a_i \end{cases} \tag{1.4}$$

where P and I respectively stand for preference and indifference.

The preference structure of this relation forms a *complete preorder* [30]. All the alternatives can be ranked from the best one to the worst one (complete) with eventual ties between two alternatives (preorder).

By nature, the following properties of P and I should also be noted [30]:

$$\forall a_i, a_j \in A \begin{cases} a_i P a_j \Rightarrow a_j \neg P a_i : & \text{P is asymmetric} \\ a_i I a_i : & \text{I is reflexive} \\ a_i I a_j \Rightarrow a_j I a_i : & \text{I is symmetric} \end{cases} \tag{1.5}$$

1.2.2 Drawback of the mono-criterion approach

Two main drawbacks of the mono-criterion approach are underlined in [18]. The first one is that, if the decision maker wants to take a decision according to several points of view (e.g. cost, sustainability, equity, ...), a single criterion can generally not synthesise the decision problem reality. This is illustrated with the following example taken from [22].

Suppose you are working in a rubber manufacturing firm and that you need to design a new rubber to respond to the needs of a customer. Your task will be to choose, between the different possible compositions of the rubber, one that best fulfills the specified requirements. Depending on these possible requirements you will probably have to choose a composition with a high breaking strength, a limited thermal conductivity, a certain elasticity, and that all these characteristics are valid for a range of temperature as large as possible. The client will probably also require that these properties do not deteriorate too quickly with time. He could also require that the rubber does not react with a specific paint with which it will be painted. Finally, you will also have to find a composition that minimises the cost of production.

It can be concluded from this example that it is not always realistic to hope to find a unique criterion synthesising correctly all the aspects of the decision problem reality.

The second drawback of the mono-criterion approach is that the classic notion of a criterion is only used in order to compare if the evaluation of an alternative a_i is greater, smaller, or equal to the evaluation of an alternative a_j. This information could be misleading.
If we consider again the problem of choosing a composition for the new rubber but this time only focusing on the expected lifetime criterion. A composition which is expected to maintain his required properties for 45 years will be preferred over a composition maintaining its properties for 40 years in the same way that it would be preferred over a third composition maintaining its properties for 5 years.
This often does not represent the reality accurately. Suppose that the rubber is aimed to be used to manufacture parts of a car. Both 40 year and 45 year would therefore be considered as perfectly fine expected lifetimes and the difference between those two compositions would generally be considered as neglectable. The two compositions would then be equally preferred. On the other hand, an expected lifetime of 5 years is insufficient. In the classical mono-criterion approach, however, the rubber compositions whose expected lifetime is 45 years will be preferred over the one of 40 years exactly in the same that it will be preferred over the one whose expected lifetime is 5 years. This is misleading.

For these reasons, the mono-criterion approach is not always adapted and some decision problems require a multi-criteria one. As it can easily be guessed, this second approach tackles the drawbacks mentioned above by introducing several evaluation criteria. This approach will be detailed in the next section.

1.3 Multi-Criteria Approach

A multi-criteria decision problem can be modeled in the following way [18]:

$$Opt\ \{f_1(a_i), f_2(a_i), \cdots, f_c(a_i), \cdots, f_k(a_i) | a_i \in A\} \tag{1.6}$$

With A, the set of alternatives, and $f_c(.), c = 1, \cdots, k$, a set of k evaluation criteria which are applications of A on the set of real numbers.

The definition of the set of alternatives does not change from the mono-criterion case (section 1.2.1).
The set of alternatives can be defined by an enumeration of each of the alternatives if these are enumerables. If the set is too large, it must be defined by the properties characterizing its alternatives. The set of alternatives can be stable or evolutive and globalized or fragmented.

In the rest of this work, all multi-criteria decision aid methods that will be considered must be applied on a finite set of alternatives with a finite family of criteria. Without loss of generality, we will therefore only consider decision problems consisting in a set A of n alternatives evaluated on k criteria that must be maximised.

$$\max \{f_1(a_i), f_2(a_i), \cdots, f_c(a_i), \cdots, f_k(a_i) | a_i \in A\} \tag{1.7}$$

1.3.1 Mathematically ill-defined problem

In opposition to the mono-criterion decision problem, the concept of optimal solution does not make sense anymore in a multi-criterion context. Indeed, due to the usually conflicting nature of the different criteria considered in a decision problem reality, it is generally not possible to find an alternative a_i such that:

$$f_c(a_i) \geq f_c(a_j), \forall a_j \in A, \forall c = 1, \ldots, k \tag{1.8}$$

The problem in this case does not consist anymore in the discovering of some hidden truth. It will consist in finding one or more alternatives that consist in a good compromise on all the criteria. The quality of a compromise is of course strongly dependent on the decision maker's perception of the problem reality. Some subjectivity must therefore explicitly be introduced. This subjectivity can be associated with the qualitative approach presented in section 1.1.

It has already been seen in the mono-criterion approach that some subjectivity could be involuntarily introduced when building the evaluation function $f(.)$. This kind of subjectivity however should still be avoided in the multi-criteria approach.

1.3.2 Types of multi-criteria decision problems

As shown by B. Roy, there exist different types of decisions problems [18]. Here under are three examples of common types:

- *Choice Problems*: these consist in choosing one or several alternatives from A which could be considered as the best alternatives. One example of a choice problem could be the already mentioned problem of choosing the two most adapted locations to build new warehouses.
- *Ranking Problems*: these consist in the partially ordering all the alternatives from the worst to the best one. One example of an ordering problem could be the following one: n nuclear power station must be

closed. However, it is not possible to close all these stations simultaneously. In which order should they be closed to minimise the risk and the cost.

- *Sorting Problems*: these consist in sorting (or partially sorting) all the alternatives in different categories. One example of sorting problem is given hereafter. Consider n patients that are sent to a liver specialist to assess whether or not these patients need a liver transplant. The specialist could have to classify these patients in three categories, the ones who do not need a transplant, the one who need a transplant, and the ones who need a transplant with urge.

More examples of decision problem types with some applications of these problems can be found in [27].

1.3.3 Dominance relation

The natural dominance relation defined in a mono-criterion context (equation 1.4) is generalized to a multi-criterion context as follow [18]:

$$\forall a_i, a_j \in A : \begin{cases} a_i P a_j & \Leftrightarrow \begin{cases} f_c(a_i) \geq f_c(a_j) & \forall c = 1 \ldots k \\ \exists h : f_h(a_i) > f_h(a_j) \end{cases} \\ a_i I a_j & \Leftrightarrow \quad f_c(a_i) = f_c(a_j) \quad \forall c = 1 \ldots k \\ a_i R a_j & \Leftrightarrow \begin{cases} \exists h : f_h(a_i) > f_h(a_j) \\ \exists h' : f_{h'}(a_i) < f_{h'}(a_j) \end{cases} \end{cases} \tag{1.9}$$

where P and I still stand for preference and indifference and R stands for incomparability. The preference structure of this relation is a *partial preorder* structure [30] as only some subsets of alternatives can be ranked from “best” to “worst” with eventual ties.

As the preference and indifference relation require unanimity on all the criteria, and the incomparability relation only needs two conflicting criteria, one can easily convince himself that the dominance relation is generally very poor, and that most pairs of alternatives will be incomparable.
The main objective of the multi-criteria decision aid methods will be to enrich this dominance relation by reducing the number of incomparable pairs of alternatives. The final ordering of the alternatives is, as it will be seen here under, not only dependent on the decision maker, but also on the decision method used to enrich the dominance relation.

1.3.4 Types of multi-criteria decision aid methods

Multi-criteria decision aid methods can be divided in three families of methods [18]:

1. Aggregating methods (Multi-attribute utility theory)
2. Outranking methods
3. Interactive methods

Aggregation methods (Multi-attribute utility theory)

Aggregating methods consist in the substitution of a multi-criterion decision problem into a mono-criterion one. This is done by synthesising all of the k criteria of the multi-criteria problem, into a unique utility function $U(a_i)$:

$$U(a_i) = U[f_1(a_i), \ldots, f_k(a_i)] \tag{1.10}$$

The problem is therefore summarised to:

$$Max\{U(a_i) \mid a_i \in A\} \tag{1.11}$$

A usual way to select an utility function is to build it as a sum of all the criteria evaluations [30]:

$$U(a_i) = \sum_{c=1}^{k} U_c(f_c(a_i)) \tag{1.12}$$

This kind of utility function will constitute the additive model. The role of the different $U_c(.)$ functions must be at least to normalise all the criteria to get rid of all scaling effects introduced by the measuring scale in which the criteria are expressed. This is not the only role of these functions. They could for example also be used to manage criteria evaluations that should neither be maximised nor minimised but which should be as close as possible to a desired value. $U_c(.)$ could then be maximal when $f_c(a_i)$ is equal to that desired value.

Another formulation can also be used to compute the utility function of the alternatives:

$$U(a_i) = \sum_{c=1}^{k} w_c \cdot U_c(f_c(a_i)) \tag{1.13}$$

Here, some weight factors w_c are introduced to express the relative importance of each criterion according to the decision maker. In equation 1.12, the effect of these coefficients were obtained by selecting an adequate $U_c(.)$ function but the two are kept distinct in equation 1.13 to emphasise their respective meaning.

It is quite obvious that the preference structure of such a problem will be the same as in a mono-criterion context. The dominance relation will therefore form a complete preorder.

This seems at first sight to solve all the problem introduced by the different criteria. Unfortunately, these methods have some drawbacks. First of all, in such additive models, a very bad evaluation on one criterion can be completely compensated by better performances on the other criteria.

Let's consider once again, the example of having to choose a rubber composition. Let's suppose we are evaluating the two only possible alternatives, $\mathscr{R}_1$ and $\mathscr{R}_2$, according to four following criteria: *breaking strength, price, thermal conductivity and expected lifetime.*
To get rid of any scaling effect, the $U_c(.)$ functions will assign to each criterion a score between 0 and 100 according to the preferences of the decision maker.

Suppose, once again, that the rubber is aimed at manufacturing car parts and that one of the two alternatives ($\mathscr{R}_1$) has an expected lifetime of 5 years. The decision maker will therefore give its "expected lifetime" evaluation a value of 0.

The evaluation table of the two alternatives could be similar to the following one:

	$U_c(.)$			
Alternatives	Breaking strength	Price	Conductivity	Lifetime
$\mathscr{R}_1$	100	40	90	0
$\mathscr{R}_2$	50	50	50	50
Weights	0.5	0.2	0.15	0.15

Table 1.1: evaluation table for the rubber composition problem using an MAUT method

From this table it can easily be seen that even if the evaluation on the "expected lifetime" criterion of the first alternative is the worst possible, the total utility function $U(\mathscr{R}_1)$ will be equal to $0.5{\cdot}100{+}0.2{\cdot}40{+}0.15{\cdot}90 = 71.5$. This is greater than the utility function $U(\mathscr{R}_2)$ which is equal to 0.5.

Using this additive model, a good rubber composition, but with an unacceptable expected lifetime, could be preferred over a compromise composition.

Another possible problem of the additive model is that not all preferences of a decision maker can be represented.

Let's for example consider the following example proposed by Vincke [30]:

a_i :	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
$f_1(a_i)$	1	1	1	2	2	2	3	3	3
$f_2(a_i)$	1	3	5	1	3	5	1	3	5

Table 1.2: evaluation for a potential decision maker

Suppose that the preferences of the decision maker are the following ones:

$$a_9 P a_6 P a_8 P a_5 P a_3 I a_7 P a_2 I a_4 P a_1 \tag{1.14}$$

This global preference can not be represented using an additive function. Indeed by the definition of the indifference relation in a mono-criterion context (section 1.4), and therefore valid for a aggregating method:

$$\begin{aligned}
a_2 I a_4 &\Rightarrow U(a_2) = U(a_4) \\
&\Rightarrow \sum_{c=1}^{k} w_c \cdot U_c(f_c(a_2)) = \sum_{c=1}^{k} w_c \cdot U_c(f_c(a_4)) \\
&\Rightarrow w_1 U_1(1) + w_2 U_2(3) = w_1 U_1(2) + w_2 U_2(1) \\
a_3 I a_7 &\Rightarrow U(a_3) = U(a_7) \\
&\Rightarrow \sum_{c=1}^{k} w_c \cdot U_c(f_c(a_3)) = \sum_{c=1}^{k} w_c \cdot U_c(f_c(a_7)) \\
&\Rightarrow w_1 U_1(1) + w_2 U_2(5) = w_1 U_1(3) + w_2 U_2(1)
\end{aligned} \tag{1.15}$$

If we subtract the second equation to the first one, we obtain the following equality:

$$\begin{aligned}
w_1 U_1(3) + w_2 U_2(3) &= w_1 U_1(2) + w_2 U_2(5) \\
\sum_{c=1}^{k} w_c \cdot U_c(f_c(a_8)) &= \sum_{c=1}^{k} w_c \cdot U_c(f_c(a_6))
\end{aligned} \tag{1.16}$$

which is in contradiction with $a_6 P a_8$.

Of course, not being able to model all possible preferences of a decision maker is not a problem since all possible preferences are not necessarily rational. A decision maker could for example prefer an alternative that is strictly worse than another for any concerned criterion. Useful decision aid methods could not be able to produce such outputs as they are expected to produce rankings which should be considered as rational for the given decision problem modelisation.
However, this is not the case in this example where it is not possible to reproduce the preferences of a decision maker which seem rational and coherent with the problem modelisation. This is, in fact, not a particularity of aggregating methods. There are often preferences which are not compatible with how the multi-criteria decision aid method models the problem reality.

There exist other models than the additive model, but, as already stated, the additive model is the most commonly used in practice [30].

The result of the application of a multiple attribute theory method leads to a complete preorder of all the alternatives. This preorder generally contains far more information than the natural dominance relation defined on the initial multi-criteria problem. This abundant quantity of information is due to the theory's strong assumptions (e.g. existence of a function U, additivity) and to the vast amount of information asked to the decision maker (e.g. preference intensities, ...) [30].

Building a complete preorder may seems too radical as the data available with the problem is not always sufficient to compare all alternatives. Furthermore, keeping some alternatives incomparable can give some insights to the decision maker about the conflicting nature of the criteria in a specific multi-criteria problem.

To deal with this problem, B. Roy introduced the concept of *outranking relations* and *outranking methods*. These concepts will be detailed here under.

Outranking methods

Outranking relations are relations aimed at enriching the dominance relation which is considered too poor, but in a realistic way. This is, not as much as with an utility function which is considered too rich to be reliable [18].

B. Roy defines the outranking relation as [22]:

> *"a binary relation S defined in A such that aSb if, given what is known about the decision-maker's preferences and given the quality of the valuations of the actions and the nature of the problem, there are enough arguments to decide that a is at least as good as b, while there is no essential reason to refute that statement."*

The outranking relations have a different mathematical formalization in each outranking method. For instance, Roy's definition given here above describes a binary relation, meaning that the alternative a_i either outranks another alternative a_j either does not outranks this alternative, but an outranking relation could also be valued and give an indication of how a_i outranks a_j.

These relations must not be complete (as already suggested) and must neither be transitive.

The PROMETHEE methods are outranking methods using an valued relation and will be explained in details in the following chapter.

Here under is a brief description of the ELECTRE I method.

ELECTRE I

ELECTRE is a family of multi-criteria decision aid methods whose name stand for "ELimination Et Choix Traduisant la REalité" (elimination and choice expressing reality).

ELECTRE I is a multi-criteria decision aid method of the ELECTRE family. It is based on the construction of an binary outranking relation a_iSa_j, meaning that a_i is at least as good as a_j. This relation is built according to two indices [21]: the *concordance* index and the *discordance* index. These indices are computed as follows.

Given two alternatives a_i and a_j, the set of criteria $C(a_i, a_j)$ is the subset of all the criteria composed of the criteria c where the evaluation of a_i is better or equal than the evaluation of a_j.

If we suppose once again that the multi-criteria decision problem is composed of k criteria to maximise, we can define $C(a_i, a_j)$ as:

$$C(a_i, a_j) = \{c = 1 \dots k \mid f_c(a_i) \geq f_c(a_j)\} \tag{1.17}$$

This set of criteria includes all the criteria in concordance with a_iSa_j [21]. The other criteria, that are not included in $C(a_i, a_j)$, form another subset $D(a_i, a_j)$. This subset includes all the criteria that are in discordance with a_iSa_j.

To be able to use these two sets consistently, some relative importance factor must be assigned to each criterion in order to be able to measure the importance of the two sets.
With these factors w_c, we can compute a concordance index $c(a_i, a_j)$ [21]:

$$c(a_i, a_j) = \frac{1}{W} \sum_{\substack{c \in \\ C(a_i, a_j)}} w_c \quad \text{with } W = \sum_{c=1}^{k} w_c \tag{1.18}$$

There are different ways of computing the discordance index $d(a_i, a_j)$, such as the one described in [21]:

$$d(a_i, a_j) = \begin{cases} 0 & \text{if } D(a_i, a_j) = \emptyset \\ \dfrac{1}{d} \max\limits_{\substack{c \in \\ D(a_i, a_j)}} [f_c(a_j) - f_c(a_i)] & \text{if } D(a_i, a_j) \neq \emptyset \end{cases} \tag{1.19}$$

with

$$d = \max_{c,i,j} \mid f_c(a_i) - f_c(a_j) \mid \tag{1.20}$$

being the maximal difference of two alternatives on any criterion.

The concordance index and the discordance index both have values in the range $[0, 1]$. The concordance index $c(a_i, a_j)$ increases as the number of criteria for which a_i is preferred or indifferent over a_j increases. It represents the relative importance of the coalition of criteria for which a_i is preferred. The discordance index increases as the maximal gap between an evaluation of a_j and an evaluation of a_i (where a_j is preferred) increases. This index can be seen as the veto power to reject that a_i outranks a_j.

Using these two indices, the outranking relation S will be built as follow:

$$\forall a_i, a_j \in A, a_i S a_j \Leftrightarrow \begin{cases} c(a_i, a_j) \geq p \\ d(a_i, a_j) \leq q \end{cases} \tag{1.21}$$

with p and q being the concordance and discordance thresholds defined by the decision maker.

Once the outranking relation is built, it must still be exploited. We can represent the problem as a graph having as vertices the alternatives of the problem. The graph will have a directed edge from the vertice a_i to the vertice a_j if and only if $a_i S a_j$. We will therefore denote it as the outranking graph of the decision problem.

The ELECTRE I method was designed for the multi-criteria choice problems (see section 1.3.2). It is aimed at finding a subset N of alternatives such that no alternative in N is outranked by any other alternative in N and all alternatives in $A \setminus N$ are outranked by at least one alternative of N.

$$\begin{cases} \forall a_j \in A \setminus N, \ \exists a_i \in N : a_i S a_j \\ \forall a_i, a_j \in N : a_i \neg S a_j \end{cases} \tag{1.22}$$

The problem of finding a subset N of alternatives fulfilling these conditions is the problem of finding a kernel of the outranking graph.

1.4 Conclusion

As it has been seen, some decision problem can be more efficiently handled using a multi-criteria approach. There exist different type of multi-criteria decision aid methods, each having their advantages and their drawbacks.

The next chapter will introduce the PROMETHEE family of methods and the main focus of the rest of this work will be the study of two variations of PROMETHEE II.

Chapter 2

Introduction to the PROMETHEE methods

The PROMETHEE (Preference Ranking Organisation METHod for Enrichment Evaluations) methods are outranking methods that have been developed at the Université Libre de Bruxelles and the Vrije Universiteit Brussel since the beginning of the 80'. Initially proposed by J.P. Brans in 1982, they are used to address multi-criteria problems.

The PROMETHEE family is relatively young compared to some other multi-criteria decision aid methods but have experienced a rapid spread as it can be seen from the increasing amount of papers that are written each year about them [5].

The PROMETHEE methods must be applied on problems with a finite set of alternatives and a finite family of criteria. The information about each alternative and its evaluations for each criterion can be summarized in a table such as Table 2.1.

a	$f_1(\cdot)$	$\ldots$	$f_c(\cdot)$	$\ldots$	$f_k(\cdot)$
a_1	$f_1(a_1)$	$\ldots$	$f_c(a_1)$	$\ldots$	$f_k(a_1)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_i	$f_1(a_i)$	$\ldots$	$f_c(a_i)$	$\ldots$	$f_k(a_i)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_n	$f_1(a_n)$	$\ldots$	$f_c(a_n)$	$\ldots$	$f_k(a_n)$

Table 2.1: example of evaluation table

This raw information is generally not directly usable by the decision maker. PROMETHEE offer some tools to help him in his ordering, choice, sorting problem, etc.

These methods are based on the three followings concepts [18]:

1. **Enrichment of the preference structure**
   For each criterion, a preference function is introduced. These preference functions are aimed at indicating the degree at which one alternative is prefered over another for that specific criterion. This is done by taking into account the deviation's amplitude between the evaluations of the two alternatives regarding the concerned criterion. This leads to an enriched intra-criterion preference structure of the alternatives. This procedure also allows to get rid of all the scaling effects due to the measuring system used to express the different criteria.
2. **Enrichment of the dominance relationship**
   For each couple of alternatives, a global preference index of the one over the other will be computed using the previously defined preference functions on all the criteria. Using these global preference indices, a quantitative outranking score is computed.
3. **Decision support**
   An outranking relation is used to rank the alternatives according to their outranking scores. This relation is used to provide some insights to the decision maker.

2.1 Enrichment of the preference structure

First of all, let us introduce the notation $d_c(a_i, a_j)$ that will represent the difference between $f_c(a_i)$ and $f_c(a_j)$:

$$d_c(a_i, a_j) = f_c(a_i) - f_c(a_j) \tag{2.1}$$

As pointed out by [7], the natural dominance relation defined in section 1.3.3 is often very poor. An alternative a_i is dominating another alternative a_j only if there is unanimity on all the criteria. This means that the evaluation for every criterion of the alternative a_i must be at least as good as the evaluations on the corresponding criterion of a_j. This is generally not the case due to the conflicting nature of the criteria (for instance, a smartphone with a better battery will generally be heavier than one with a poor battery). Furthermore only the signs of the differences between the evaluations are taken into account and not their amplitudes. A difference of amplitude d_c of 1% between $f_c(a_i)$ and $f_c(a_j)$ should generally (but not always) not have the same impact as a difference of 200% (see the rubber composition choosing problem on page 5).

To cope with this disadvantage, the preference structure will be enriched with the new notion of *preference functions*. These preference functions $P_c(a_i, a_j)$ will give the preference degree of alternative a_i over a_j on the criterion c in function of the difference d_c between $f_c(a_i)$ and $f_c(a_j)$:

$$P_c(a_i, a_j) = P_c[d_c(a_i, a_j)] \tag{2.2}$$

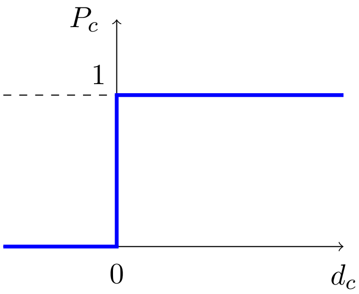
The preference degree will be between 0 and 1 and will be valued as follow [18]:

$$\begin{cases} P_c(a_i, a_j) = 0 & if \quad d_c(a_i, a_j) & \leq & 0 & \text{(no preference)} \\ P_c(a_i, a_j) \approx 0 & if \quad d_c(a_i, a_j) & > & 0 & \text{(weak preference)} \\ P_c(a_i, a_j) \approx 1 & if \quad d_c(a_i, a_j) & >> & 0 & \text{(strong preference)} \\ P_c(a_i, a_j) = 1 & if \quad d_c(a_i, a_j) & >>> & 0 & \text{(strict preference)} \end{cases} \tag{2.3}$$

To be compatible with the meaning of preference, the preference functions must be nondecreasing and must be equal to 0 for any negative difference of evaluation ($d_c(a_i, a_j) \leq 0$).

The decision of a preference function satisfying these conditions is let to the decision maker. However, a set of 6 different types are proposed in [18] which should satisfy the majority of them:

Type 1: the usual criterion

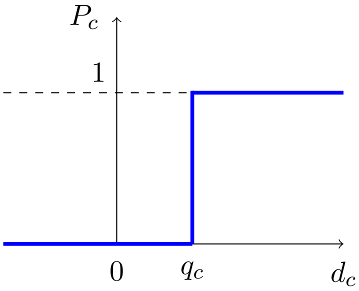


$$P_c = \begin{cases} 0 & \text{if} \quad d_c \leq 0 \\ 1 & \text{if} \quad 0 < d_c \end{cases} \tag{2.4}$$

Figure 2.1: Preference function of the usual criterion

This criterion does not imply any extension of the classical notion of preferences. As soon as the evaluation between the two alternatives is not identical, there will be a strict preference for the alternative having the highest evaluation. This criterion is therefore not enriching the preference structure but allows the decision maker to use the criteria in their usual sense, without having to define any parameter. This preference function is also useful when using the amplitude between two evaluations does not make sense.

Type 2: the quasi-criterion

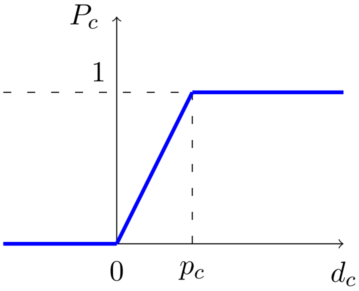


$$P_c = \begin{cases} 0 & \text{if} \quad d_c \leq q_c \\ 1 & \text{if} \quad q_c < d_c \end{cases} \tag{2.5}$$

Figure 2.2: Preference function for the quasi-criterion

This criterion adds an indifference threshold q_c. This means that as long as d_c does not exceeds that threshold the difference will be neglected and the two evaluations will be equally prefered.

Type 3: the criterion with linear preference



$$P_c = \begin{cases} 0 & \text{if} \quad d_c \leq 0 \\ \frac{d_c}{p_c} & \text{if} \quad 0 < d_c < p_c \\ 1 & \text{if} \quad d_c \geq p_c \end{cases} \tag{2.6}$$

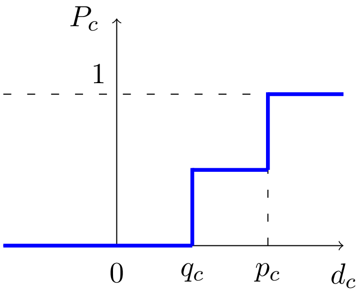
Figure 2.3: Preference function for criterion with linear preferences

The preference function for a criterion with linear preferences increases continuously with the difference of the evaluations between 0 and some threshold

p_c. If the difference is higher than this threshold, then the alternative with the highest evaluation is strictly prefered over the other.

This preference function is the first one that allows the decision maker to express valued preferences. If the difference of the evaluation lies between 0 and p_c the preference degree will be between $[0,1]$.

Type 4: the level-criterion

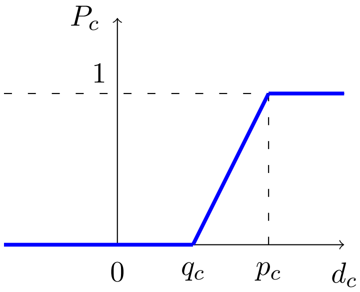


$$P_c = \begin{cases} 0 & \text{if} \quad d_c \leq 0 \\ \frac{1}{2} & \text{if} \quad q_c < d_c \leq p_c \\ 1 & \text{if} \quad p_c < d_c \end{cases} \qquad (2.7)$$

Figure 2.4: Preference function for level-criterion

This criterion uses two parameters, the indifference threshold q_c and the preference threshold p_c. If the difference of evaluations lies between these two thresholds, then the preference functions is equal to 0.5. This kind of function can be useful for criterions whose evaluations have discrete values (for example a criterion whose values can be “green”, “red”, “blue”). If the criterion’s evaluations can take more than three values, an intuitive generalisation of this preference function with more than three levels can be used.

Type 5: the criterion with linear preferences and indifference zone

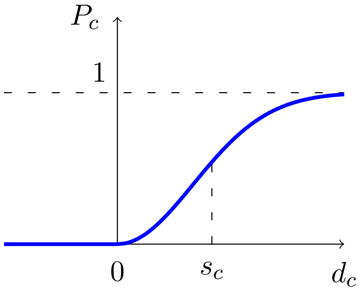


$$P_c = \begin{cases} 0 & \text{if} \quad d_c \leq q_c \\ \frac{d_c - q_c}{p_c - q_c} & \text{if} \quad q_c < d_c \leq p_c \\ 1 & \text{if} \quad p_c < d_c \end{cases} \qquad (2.8)$$

Figure 2.5: Preference function for the criterion with linear and indifference zone

This preference function is a more general version of the Type 3 preference function. Here a parameter q_c can be used to define an indifference threshold.

Type 6: the gaussian criterion



$$P_c = \begin{cases} 0 & \text{if} \quad d_c \leq 0 \\ 1 - e^{-d_c^2/2s_c^2} & \text{if} \quad 0 < d_c \end{cases} \tag{2.9}$$

Figure 2.6: Preference function for the gaussian criterion

This preference function leads to a preference increasing continuously with d_c. No threshold parameter is needed for this function but instead, the parameter s_c must be fixed. This parameter indicates the difference d_c needed to have a mean preference (0.39) of the criterion with the highest evaluation over the other. The advantage of this preference function is that it will never reach 1. There is therefore no threshold of strict preference.

It is important to note that at most two parameters (q_c, p_c or s_c) must be defined by the decision maker. As each of these parameters have a meaning in the real world, their assignation by the decision maker should not be an insurmountable task [8].
The choice of these parameters and of the preference functions consists in all the information *within a criterion* [8] that must be given by the decision maker. This choice should be done with care as these parameters can have a crucial impact on the final ranking. For example, choosing indifference thresholds too high for a specific criterion will have as consequence that this criterion will not have much impact as two alternatives will generally be equally preferred regarding to this criterion. On the other hand, choosing a preference threshold too low will impoverish the preference structure, as alternatives will generally be strictly preferred the ones over the others, and the situation will be similar as when the preference was computed only by taking into account the sign of d_c.

2.2 Enrichment of the dominance relationship

2.2.1 Global preference index

After having computed the preference degrees for a given pair of alternatives for each criterion, a set of weights w_c must still be defined in order to be able to compute the global preference index $\pi(a_i, a_j)$ of the alternative a_i over the alternative a_j. These weights w_c represent the relative importance of a criterion c compared to the others. The global preference index is given by:

$$\pi(a_i, a_j) = \sum_{c=1}^{k} P_c(a_i, a_j) \cdot w_c \tag{2.10}$$

The choice of the different w_c consists in the information *between criteria* [8] that must be given by the decision maker. Similarly as for the parameters giving information within a criterion, the choice of the w_c is of the highest importance. For example, a criterion with a weight equal to zero would simply have no influence on the global preference index.

These weights factors must be positive as criteria would have a negative influence otherwise:

$$w_c \geq 0 \qquad \forall c \tag{2.11}$$

As these parameters only consists of multiplicative constants, there is no inconvenient to choose them normalised:

$$\sum_{c=1}^{k} w_c = 1 \tag{2.12}$$

Some relations can be deduced from the equations (2.2) and (2.10):

$$\pi(a_i, a_i) = 0 \tag{2.13}$$

which states that an alternative is not preferred over itself, and

$$0 \leq \pi(a_i, a_j) \leq 1 \tag{2.14}$$

which holds since the weights are normalised and the P_c are lower or equal to 1. Furthermore, if $\pi(a_i, a_j)$ is strictly positive, then $\pi(a_j, a_i)$ will be equal to 0, leading to:

$$\pi(a_i, a_j) + \pi(a_j, a_i) \leq 1 \tag{2.15}$$

It should also be clear that [8]:

$$\left\{ \begin{array}{lcc} \pi(a_i, a_j) \approx 0 & \Leftrightarrow & \text{weak preference of } a_i \text{ over } a_j \\ \pi(a_i, a_j) \approx 1 & \Leftrightarrow & \text{strong preference of } a_i \text{ over } a_j \end{array} \right. \tag{2.16}$$

All the pairwise preference indices $\pi(a_i, a_j)$ can be seen as forming an $n \times n$ preference matrix Π.

2.2.2 Outranking flow

As just seen here above, $\pi(a_i, a_j)$ gives an indication of how a_i is dominating a_j on all the criterions. This information is only related to a pair of alternatives and does not give a sufficient indication of the dominance of a_i in the set A.

Such as for the ELECTRE I method (section 1.3.4), an outranking graph can be built. However, it will not be a binary outranking graph with arcs going from outranking alternatives to outranked alternatives anymore. The graph will be a complete digraph on n vertices representing the n possible alternatives. The valued arc going from an alternative a_i to an alternative a_j will be valued $\pi(a_i, a_j)$ [9]. This graph is therefore not a graph directly leading to an ordering, but will, for each pair of alternatives a_i and a_j, give a quantified information of how much a_i outranks a_j and vice-versa.

This graph can be used to compute an *outgoing outranking flow* and an *incoming outranking flow* for each alternative [18].
The outgoing flow is given by:

$$\phi^+(a_i) = \frac{1}{n-1} \sum_{j=1}^{n} \pi(a_i, a_j) \tag{2.17}$$

This flow indicates how much alternative a_i is dominating or outranking on average the remaining $n-1$ alternatives. The higher this flow, the more a_i should be preferred by the decision maker.

Similarly, the incoming flow is defined by [18]:

$$\phi^-(a_i) = \frac{1}{n-1} \sum_{j=1}^{n} \pi(a_j, a_i) \tag{2.18}$$

This flow indicates how much alternative a_i is outranked on average by the $n-1$ other alternatives. The lower this flow, the more alternative a_i should be preferred.

One can see in the expressions of the outranking flows that the summations of the preference index are divided by $n-1$. This is done to ensure that the outranking flows will be lower or equal to 1 (the summations consist of sums on n terms lower than 1 from which at least one is null by (2.13)).

The netflow score can be obtained by subtracting the incoming flow from the outgoing flow:

$$\phi(a_i) = \phi^+(a_i) - \phi^-(a_i) \tag{2.19}$$

2.3 Decision aid

2.3.1 PROMETHEE I

To help the decision maker in his task, the PROMETHEE methods will rank the alternatives according to their outranking flows.

The two complete preorders (P^+,I^+) and (P^-,I^-) obtained with the outranking flows are defined as follow [18]:

$$\forall a_i, a_j \in A \begin{cases} a_i P^+ a_j & \Leftrightarrow \quad \phi^+(a_i) > \phi^+(a_j) \\ a_i I^+ a_j & \Leftrightarrow \quad \phi^+(a_i) = \phi^+(a_j) \end{cases} \tag{2.20}$$

$$\forall a_i, a_j \in A \begin{cases} a_i P^- a_j & \Leftrightarrow \quad \phi^-(a_i) < \phi^-(a_j) \\ a_i I^- a_j & \Leftrightarrow \quad \phi^-(a_i) = \phi^-(a_j) \end{cases} \tag{2.21}$$

The PROMETHEE I ranking is made by taking the intersection of these two preorders. An alternative a_i is preferred over another alternative a_j according to PROMETHEE I if both its outgoing flow and its incoming flow are better (respectively greater and smaller) than the ones of a_j. If both the flows are equal for the two alternatives, then the alternatives will be indifferent, and if none of these two cases hold, then the alternatives will be said to be incomparable [18]:

$$\forall a_i, a_j \in A : \begin{cases} a_i P a_j & \Leftrightarrow & \begin{cases} a_i P^+ a_j \text{ and } a_i P^- a_j \\ a_i P^+ a_j \text{ and } a_i I^- a_j \\ a_i I^+ a_j \text{ and } a_i P^- a_j \end{cases} \\ a_i I a_j & \Leftrightarrow & a_i I^+ a_j \text{ and } a_i I^- a_j \\ a_i R a_j & \Leftrightarrow & \text{else} \end{cases} \tag{2.22}$$

2.3.2 PROMETHEE II

It can happen that the decision maker wishes to obtain a complete ordering of the alternatives (without any incomparabilities). This can be done using PROMETHEE II. This method builds a ranking based on the net flow score of each alternatives:

$$\forall a_i, a_j \in A \begin{cases} a_i P a_j \Leftrightarrow \phi(a_i) > \phi(a_j) \\ a_i I a_j \Leftrightarrow \phi(a_i) = \phi(a_j) \end{cases} \tag{2.23}$$

As it can be seen, using PROMETHEE II, two alternatives can be either prefered the one over the other (P) either indifferent (I). Two alternatives can therefore not be incomparable, leading to a complete preorder of the alternatives.

Furthermore, it should be pointed out that the outgoing, incoming or net flows are not just some artificial and arbitrary constructions made to aggregate the pairwise preferences. In fact, it can be shown that the net flow score is the centered score that minimizes the sum of the square deviations for the pairwise preferences of the alternatives [19].

In other words, if Q is the sum of the square deviations between a centered score s and the pairwise preferences, with s_i being the score of the alternative a_i:

$$Q = \sum_{i=1}^{n} \sum_{j=1}^{n} \left[(s_i - s_j) - (\pi(a_i, a_j) - \pi(a_j, a_i)) \right]^2 \tag{2.24}$$

then Q will be minimal when the score s_i of alternatives are proportionals to their net flows:

$$s_i = \frac{1}{n} \sum_{j=1}^{n} (\pi(a_i, a_j) - \pi(a_j, a_i)) = \frac{n-1}{n} \phi(a_i) \tag{2.25}$$

2.3.3 Conclusion

There exist other variations of PROMETHEE than the PROMETHEE I and PROMETHEE II methods but these will not be detailed here and the interested reader can refer to [8] for additional information.

There also exists an additional tool, named GAIA (Geometrical Analysis for Interactive Assistance), which is, as its name suggests, a tool aimed at performing an interactive visualisation of the data. Once again, this method will not be detailed here and the interested reader can refer to [8] or [18] for additional information.

Finally, it should also be noted that part the success of PROMETHEE is due to the existence of userfriendly software (such as *D-Sight*) that helps the decision maker in the visualisation of the alternatives and the influence of the different parameters of the method [27].

A criticism that is often addressed to the PROMETHEE methods is that it is subject to the *rank reversal* phenomenon. An introduction to this phenomenon will be presented in the next chapter. Then, two variations of PROMETHEE II which are aimed at reducing these rank reversals will be studied.

Chapter 3

Rank Reversal

The so called rank reversal phenomenon in a multi-criteria decision aid context is a phenomenon initially highlighted by Belton and Gear in 1983 [6][31].

There is not one unique definition of the rank reversal phenomenon but there rather exist several, which however generally share the same idea: “the relative ordering between two alternatives depends on the presence of one or several other alternatives”.
Some of the definitions that can be found in the literature are that the relative position of two alternatives can be influenced by the presence of [8]:

- a non discriminating criterion
- a copy of an alternative
- a dominated alternative
- any other alternative

In the rest of this work, unless stated otherwise, we will say that there is a “rank reversal occurrence” when the relative ordering between two alternatives is affected by the presence or absence of any third alternative.

This phenomenon can lead to some undesirable consequences. One of these obvious undesirable consequence is highlighted in the following example.
Suppose that a municipal administration of a city considers that it is necessary to build a new sports stadium. This could be due to the fact that the existing one is becoming too old or because its capacity is no longer sufficient to host enough supporters. The municipal administration would therefore organise a competition where architects and entrepreneurs would propose their projects. The administration could decide to rank all these projects using a multi-criteria decision aid method according to criteria such as the estimated price of the project, the estimated duration of the works,

the capacity of the stadium, the capacity of its parking, etc. The project arriving in first position of the ranking would be selected and built.
In such a situation, the possibility of a rank reversal between the first two alternatives would have important consequences as it would influence which stadium would be built, and which group of people would have the right to build it. There does not seem to be any rational reason why the order of the two first projects should be influenced by the presence of other projects and therefore these kind of rank reversals should be avoided.

In similar situations, it could also be argumented that the multi-criteria decision aid methods which suffers from rank reversal occurrences are vulnerable to ranking manipulations by artificially adding (or removing) some alternatives.

W. De Keyser and P. Peeters [16] were the first to point out that the PROMETHEE methods suffer from rank reversal occurrences.

Before going into the specific details of the rank reversal phenomenon in PROMETHEE, some characteristics of more general methods based on pairwise comparisons will be given.

3.1 Transitivity of preferences and pairwise comparisons

Some important facts have been highlighted by Mareschal et al. [19] concerning methods based on the pairwise comparisons of the alternatives.

These facts will be resumed in the rest of this section. First, the notion of *consistent ranking methods based on pairwise comparison* will be defined. Then, few words will be said about the transitivity of the preferences and finally, it will be shown that consistent ranking methods based on pairwise comparisons with nontransitive preferences matrices suffer from the rank reversal phenomenon [19].

3.1.1 Ranking methods based on pairwise comparisons

Let us again consider a multi-criteria decision problem consisting in the ranking of the alternatives from a finite set A of n alternatives which must maximise k evaluation functions.

If the multi-criteria decision aid method is based on:

1. the computation of an $n \times n$ pairwise preferences matrix *Pref* whose elements $pref_{ij}$ are global preference indices of the alternative a_i over a_j
2. the exploitation of this matrix to obtain a complete preorder

then it is said to be a *ranking method based on pairwise comparisons* (RBPC) [19].

The PROMETHEE methods are examples of such methods where the $n \times n$ preferences matrix Π consist in the $n \times n$ evaluations of $\pi(a_i, a_j)$ for all a_i, a_j in A.

Furthermore, a ranking method based on pairwise comparisons is said to be *consistent* if the ordering obtained with its application on a set of two alternatives is consistent with the 2×2 preferences matrix.
This means that if a consistent ranking method based on pairwise comparisons is applied on a set of two alternatives a_i and a_j, with a_i pairwise preferred over a_j ($pref_{ij} > pref_{ji}$) then a_i should be ranked before a_j [19]. This property seems rather intuitive and necessary for any ranking method based on pairwise comparisons.

By remembering how the outranking flow is computed in PROMETHEE (section 2.2.2), one can easily see that it will be consistent with any 2×2 preferences matrix Π. This means that PROMETHEE is a consistent ranking method based on pairwise comparisons. It should however be noted that the ranking obtained with a larger set of alternatives will not necessarily respect all the pairwise preferences anymore.

As already announced, such methods with nontransitive preferences matrix are susceptible to suffer from the rank reversal phenomenon. One could therefore wonder why these preferences matrix could be nontransitive. Some insights into the transitive nature of the preferences will therefore be given in the next section.

3.1.2 Transitivity of the preferences

In a mono-criterion context, with a criterion $f(.)$ which must be maximized, the preference relation P of the natural dominance relation is trivially transitive:

$$\forall a_i, a_j, a_k \in A : \left\{ \begin{array}{l} a_i P a_j \Rightarrow f(a_i) > f(a_j) \\ a_j P a_k \Rightarrow f(a_j) > f(a_k) \end{array} \right. \Rightarrow f(a_i) > f(a_k) \Rightarrow a_i P a_k \tag{3.1}$$

In a similar way, one could see that the indifference relation is also transitive.

A slightly more elaborated dominance relation can be defined by introducing an indifference threshold q:

$$\forall a_i, a_j \in A : \begin{cases} f(a_i) > f(a_j) + q & \Leftrightarrow \quad a_i P a_j \\ \mid f(a_i) - f(a_j) \mid \leq q & \Leftrightarrow \quad a_i I a_j \\ f(a_i) < f(a_j) - q & \Leftrightarrow \quad a_j P a_i \end{cases} \tag{3.2}$$

This dominance relation defines a *semiorder* [30]. The preference relation P is still transitive but the indifference I relation is not transitive anymore.

$$\begin{aligned} \forall a_i, a_j, a_k \in A : \begin{cases} a_i P a_j \Rightarrow f(a_i) > f(a_j) + q \\ a_j P a_k \Rightarrow f(a_j) > f(a_k) + q \end{cases} \\ \Rightarrow f(a_i) > f(a_k) + q \Rightarrow a_i P a_k \end{aligned} \tag{3.3}$$

The semiorder dominance structure has been introduced by Luce in [17] as a way to represent the human preference model (preference and indifference) without relying on the assumption that the indifference relation is transitive. This hypothesis is justified between others by the fact that a human being cannot discriminate between two evaluations if these are to close one to another.
Luce gives the example of 401 numbered cups of coffee where each cup of coffee i would contain $\left(1 + \frac{i}{100}\right)x$ grams of sugar, with $i = 1 \ldots 401$ and x being the weight in grams of a cube of sugar. When asking to someone to taste any cups i and $i + 1$, he will obviously not taste any difference. On the other hand, when asking to someone to taste the first and the last cup, it is highly probable that he will taste a difference [17].

Tversky goes a step further and shows that when considering a multi-criteria context, even the preference relation should not be considered as being necessarily transitive [28]. Tversky motivate his statement with the following example.

Consider a decision maker D which must hire one of three possible candidates. The candidates are evaluated according to two criteria: *intelligence* and *experience*. The evaluations of the different candidates can be seen in Table 3.1.

Candidates	Intelligence	Experience
x	2ε	6ε
y	3ε	4ε
z	4ε	2ε

Table 3.1: Evaluation table of Tversky's example [28]

The evaluations represent some dimensionless scores according to the corresponding criteria with ε being a constant.

Suppose D values the intelligence more than the experience. He could therefore adopt the following strategy:

- If a candidate c_i is considered by the decision maker D as more intelligent than a candidate c_j, then candidate c_i will be preferred over c_j.
- If two candidates c_i and c_j are both considered by D as being equally intelligent, then the one with the most experience will be preferred.

Now suppose also that D does not find that a difference of score smaller or equal to ε for the intelligence criterion is enough to be significative (this can be due to the fact that the tests are not precise enough, or simply because D does not care about too small differences). The ranking of the candidates, based on the intelligence criterion only, will therefore lead to a semiorder structure (as the ones introduced by Luce).

This has for consequence that the candidate y is considered by D as being equally intelligent to both the two other candidates, but candidates x and z are not considered as equally intelligent (z being considered more intelligent than x). Since y is considered as equally intelligent as x and z, the preference between these pairs will depend on the experience of the candidates.

By defining (I_{int}, P_{int}) and (I_{exp}, P_{exp}) the indifference and preference relations respectively on the intelligence and experience criterion only, the following preference relations are obtained:

$$\begin{cases} yI_{int}z \ \wedge \ yP_{exp}z & \Rightarrow yPz \\ yI_{int}x \ \wedge \ xP_{exp}y & \Rightarrow xPy \\ zP_{int}x & \Rightarrow zPx \end{cases} \tag{3.4}$$

These considerations will lead to the preferences matrix represented in Figure 3.1.

$$P = \begin{array}{c} \\ x \\ y \\ z \end{array} \begin{array}{c} \begin{array}{ccc} x & y & z \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \end{array}$$

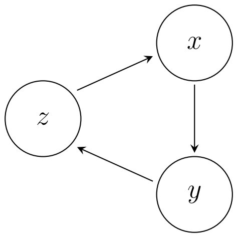


Figure 3.1: Example of non transitive preferences matrix and its graphical representation

Unlike with the pairwise preferences matrices Π of the PROMETHEE methods whose elements are valued between 0 and 1 and represent the strength with which an element is pairwise preferred over another, this matrix is binary and indicates if an alternative is preferred over another, without giving any indication about the strength of the preference.

It can be seen from this example, that the pairwise preferences of a decision maker which seems to use only reasonable assumptions can be nontransitive.

It will now be shown that any ranking method based on pairwise comparisons which is consistent and uses a nontransitive preferences matrix must suffer from the rank reversal phenomenon.

3.1.3 Rank reversal occurrences in consistent RBPC with nontransitive preferences

To demonstrate that rank reversal occurrences are unavoidable in such conditions, we will consider the problem of having to rank the three candidates presented in the previous section.
The considered ranking method will therefore have to aggregate the nontransitive preferences in one global ranking, as shown in Figure 3.2.

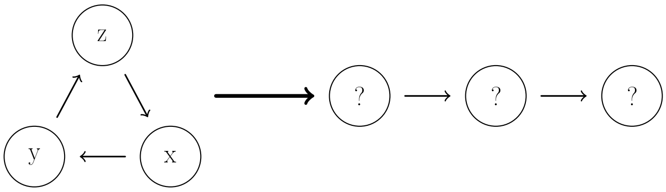


Figure 3.2: Aggregating non transitive preferences in a ranking

Since the method is considered to be consistent by hypothesis, if one alternative is removed, the ranking obtained by applying the method on the remaining alternatives must respect the pairwise preferences of these alternatives (since the method is applied on only two alternatives). These conditions are shown in the Figure 3.3.

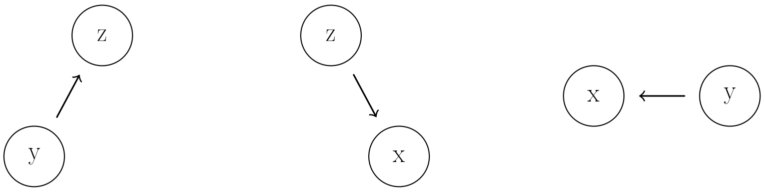


Figure 3.3: Consistent rankings when only two candidates are considered

It should be quite obvious that it is not possible to find any complete ranking that satisfies these three conditions. Therefore, whatever the ranking is, it will lead to some rank reversal occurrence when one of the alternatives is removed.
The reasoning here above has been done for three alternatives but it remains valid for any set of alternatives where the pairwise preferences between these alternatives form a cycle (such as in figure 3.1). We can therefore conclude that the rankings obtained with PROMETHEE on a set of alternatives, from which a subset have nontransitive pairwise preferences, will inevitably suffer from rank reversal occurrences.

However, this is not the only characterization that can be done concerning the possibility of occurrence of rank reversals in the PROMETHEE methods. The next section will address the possibility of rank reversal occurrences between alternatives dominating each other and the impact of the differences of net flows of two alternatives on their possibility to reverse their ranks.

3.2 Some results about the rank reversal phenomenon in the PROMETHEE methods

3.2.1 Respect of the natural dominance relation

An alternative which is at least as good as another alternative on all the criteria and strictly better on at least one criteria is said to dominate this other alternative (see section 1.3.3). There is obviously no reason why an alternative which is dominated by another should be ranked before this other which would be a violation of the natural dominance relation. It is therefore reasonable to hope that any ranking method will respect it.
This also means that there should not be any strict rank reversal between dominated and dominant alternatives (the dominated alternatives could however become indifferent to the dominant ones under some particular circumstances).

It will be shown here under that the PROMETHEE methods respect the dominance relation [26] and do not suffer from these kind of rank reversals.

It will be shown that if alternative a_i is dominating alternative a_j, then its netflow can not be lower that the one of a_j:

$$\left\{ \begin{array}{ll} \forall c = 1, \ldots, k \ : & f_c(a_i) \geq f_c(a_j) \\ \exists h \in \{1, \ldots, k\} : & f_c(a_i) > f_c(a_j) \end{array} \right. \Rightarrow \phi(a_i) \geq \phi(a_j) \tag{3.5}$$

Since a_i is dominating a_j, and the preference functions P_c are nondecreasing, we can see that $\pi(a_i, b) \geq \pi(a_j, b)$ for any b in A:

$$\pi(a_i, b) = \sum_{c=1}^{k} w_c \cdot P_c(a_i, b) \geq \sum_{c=1}^{k} w_c \cdot P_c(a_j, b) = \pi(a_j, b) \tag{3.6}$$

A similar reasoning can be applied to see that $\pi(b, a_i) \leq \pi(b, a_j)$.

Therefore we can conclude that:

$$\begin{aligned} \phi(a_i) &= \phi^+(a_i) - \phi^-(a_i) \\ &= \frac{1}{n-1} \sum_{b \in A} \pi(a_i, b) - \frac{1}{n-1} \sum_{b \in A} \pi(b, a_i) \\ &\geq \frac{1}{n-1} \sum_{b \in A} \pi(a_j, b) - \frac{1}{n-1} \sum_{b \in A} \pi(b, a_j) \\ &\geq \phi(a_j) \end{aligned} \tag{3.7}$$

Showing that the dominance relation is respected.

3.2.2 Rank reversal and net flow differences

Having alternatives dominating each other is not the only situation under which rank reversal cannot occur in the PROMETHEE methods. Indeed, rank reversal can occur only between alternatives whose net flows difference is lower than a given limit.

Here under, an initial bound on the net flow difference between two alternatives will be shown under which rank reversal cannot occur when one alternative is removed in PROMETHEE II [26].

Suppose that alternative x is removed from the set of alternatives A. The resulting set will be noted A'. The net flow of an alternative a_i in the new set is given by:

$$\begin{aligned} \phi'(a_i) &= \frac{1}{n-2} \sum_{b \in A'} (\pi(a_i, b) - \pi(b, a_i)) \\ &= \frac{1}{n-2} \sum_{b \in A} (\pi(a_i, b) - \pi(b, a_i)) - \frac{1}{n-2} (\pi(a_i, x) - \pi(x, a_i)) \\ &= \frac{n-1}{n-2} \phi(a_i) - \frac{1}{n-2} (\pi(a_i, x) - \pi(x, a_i)) \end{aligned} \tag{3.8}$$

Using the same reasoning for an alternative a_j, we can express the net flow difference between the two alternatives in the set A' according to their net flow difference in A:

$$\begin{aligned}\phi'(a_i) - \phi'(a_j) =& \frac{n-1}{n-2}(\phi(a_i) - \phi(a_j)) \\ &- \frac{1}{n-2}(\pi(a_i, x) - \pi(x, a_i) - \pi(a_j, x) + \pi(x, a_j))\end{aligned} \tag{3.9}$$

If we suppose that a_i was initially ranked before a_j ($\phi(a_i) - \phi(a_j) > 0$), there will be no rank reversal ($\phi'(a_i) - \phi'(a_j) > 0$) only if:

$$(n-1)(\phi(a_i) - \phi(a_j) > (\pi(a_i, x) - \pi(x, a_i) - \pi(a_j, x) + \pi(x, a_j)) \tag{3.10}$$

By remembering that $\pi(a_\alpha, a_\beta)$ can only take values between zero and one (see equation 2.14 in section 2.2.1) we can conclude that there cannot be any rank reversal between a_i and a_j when an alternative is removed if their net flow difference is greater than:

$$\begin{aligned}(n-1)(\phi(a_i) - \phi(a_j) &> (1 - 0 - 0 + 1) \\ \phi(a_i) - \phi(a_j) &> \frac{2}{n-1}\end{aligned} \tag{3.11}$$

This bound, which is only valid when one alternative is removed, can however be generalized to the cases where N alternatives are removed:

$$\phi(a_i) - \phi(a_j) > \frac{2N}{n-1} \tag{3.12}$$

However, this first bound is quite weak and not really often applicable. A reinforcement of this bound has been given in [29] where it is shown that the removal of one alternative cannot cause any rank reversal between alternatives which's net flow difference is greater than:

$$\phi(a_i) - \phi(a_j) > \frac{2}{n-1} \cdot W_{ij} \tag{3.13}$$

where:

$$W_{ij} = \sum_{c=1, f_c(a_i) \geq f_c(a_j)}^{k} w_c \tag{3.14}$$

represents the total weight of the coalition of criteria where a_i is as least as good as a_j [29].

3.3 Comments about the rank reversal phenomenon and conclusion

As it has been seen, the rank reversal problem is far from being a trivial one. There seems to be situations where the rank reversal phenomenon could be seen as reasonable (with nontransitive preferences) but there are also cases where it should be avoided (such as in the example of the sport stadium).

It should also be noted that this phenomenon is not restricted to the area of multi-criteria decision aid. Indeed, it had already been highlighted earlier in social choice theory, and more particularly in Kenneth Arrow's famous impossibility theorem [3] and its condition of *Independence of irrelevant alternatives.*

The problematic of rank reversal in the multi-criteria decision aid domain has already been addressed and is still addressed by numerous authors (some of them have already been mentioned in this chapter) and it is important to point out that the issue of the legitimacy of this phenomenon has not been settled yet.

The rest of this work is not aimed at answering this question. It rather starts from the ascertainment that rank reversals occurs in the PROMETHEE methods, and proposes an investigation of two approaches aimed at reducing or suppressing them.

The first approach is called ROBUST PROMETHEE and will be described in the next chapter. It is based on the repetition of PROMETHEE II a great number of times on small subsets of alternatives chosen at random.

The second approach is called REFERENCED PROMETHEE. It is based on the application, for each of the considered alternatives, of PROMETHEE II on a set constituted of this alternative and of predefined reference profiles. This method, by construction, does not suffer from rank reversals.

These two methods unfortunately do not provide solutions without any cost. The distinctive features of these methods, as well as the additional information needed from the decision maker for each of them will be detailed in their corresponding chapters.

Chapter 4

The ROBUST PROMETHEE method

ROBUST PROMETHEE is a multi-criteria decision aid method based on PROMETHEE II. It has originally been proposed by De Smet in [25].

As indicated by its name, it is aimed to be more robust than PROMETHEE II (with respect to rank reversal occurrences). Such as for the notion of rank reversals, the notion of robustness is not univocally defined (see [24], [13] and [4]). In this work, the following definition of a robust solution will be used:

> *"a robust solution shall behave well in (slightly) different conditions, meaning that it is as much as possible immune to small changes in the conditions it was designed for."* [4]

More specifically, it is hoped that ROBUST PROMETHEE will produce rankings more robust with respect to the sets of alternatives on which it is applied. This means that if the set of alternatives is slightly changed (in particular if one alternative is added or removed), then the ranking should remain globally the same (excepted for the added alternative). If this method is effectively robust in this regard, then it should less suffer from the rank reversal phenomenon. This last assertion will be studied in this chapter.

The idea used to make ROBUST PROMETHEE robust consists in repeating PROMETHEE II, a certain amount of times, on different samplings of the alternatives. These samplings (or subsets) consist of a fixed quantity of alternatives taken uniformly at random from A (without replacements). The final ranking will then be built using all the sub-rankings obtained with the different samplings.

ROBUST PROMETHEE is a method based on pairwise comparisons (see section 3.1.1). The rankings obtained during the iterations are used to compute a pairwise probability matrix P which has the role of the $n \times n$ matrix of pairwise preferences:

- The elements p_{ij} are computed as the probability that the alternative a_i would be ranked before the alternative a_j in one of the random subsets.
- Once the iterations are completed, the alternatives would be ranked according to their net flow scores computed on the pairwise probability matrix.

These elements p_{ij} are therefore indicators that are used to asses the probability that alternative a_i should be ranked before a_j by the decision maker, independently of the subsets used. Therefore, just as the net flow score computed on the Π matrix, is the centered score that minimizes the sum of the square deviations for the pairwise preferences of the alternatives (see section 2.3.2), ROBUST PROMETHEE's net flow score will be a score that minimizes the sum of the square deviations for the pairwise probabilities of being ranked before the other alternatives.

This chapter is organized as follow. First, a more detailed description of the algorithm of this method will be given, then, the first empirical results will be presented, and finally, some of its mathematical properties will be studied.

4.1 The ROBUST PROMETHEE algorithm

ROBUST PROMETHEE is based on the repetition of PROMETHEE II on different samplings of the alternatives. We will denote with R this quantity of repetitions and m the size of the samplings (or subsets). The value of m is supposed to be constant.

Mathematically speaking, the method can easily be defined with the help of the two following families of functions:

$$A_{mr}(a_i) = \begin{cases} 1 & \text{if } a_i \text{ is in the } m \text{ selected alternatives at iteration } r \\ 0 & \text{else} \end{cases} \tag{4.1}$$

$$Beats_r(a_i, a_j) = \begin{cases} 1 & \text{if } a_i \text{ is ranked before } a_j \text{ at the iteration } r \\ \frac{1}{2} & \text{if } a_i \text{ and } a_j \text{ have the same ranking} \\ 0 & \text{else} \end{cases} \tag{4.2}$$

The first ones, $A_{mr}(a_i)$, simply serve as indicators of whether a given alternative is selected at the rth iteration of the procedure.
The other ones, $Beats_r(a_i, a_j)$, indicate if a specific alternative is ranked before another specific alternative. In other words, these functions indicate whether the alternative a_i has beaten alternative a_j. If one of these two alternatives was not selected at the rth iteration, then the function will be equal to zero.

With the help of these two family of functions, the elements of the probability matrix can be assessed as follows:

$$p_{ij} = \frac{\sum\limits_{r=1}^{R} Beats_r(i,j)}{\sum\limits_{r=1}^{R} A_{mr}(i) \cdot A_{mr}(j)} \tag{4.3}$$

Finally, the net flow is computed on the preference matrix. This net flow will be denoted further as robust flow (ϕ_{rob}) to prevent any ambiguities.

$$\phi_{rob}(a_i) = \frac{1}{n-1} \sum_{j=1}^{n} (p_{ij} - p_{ji}) \tag{4.4}$$

However, this method could in practice be implemented in a computationally more efficient manner. A proposition of a more efficient algorithm can be found in appendix A.

In the following section, the first empirical results will be analysed.

4.2 First empirical results

Some first results of ROBUST PROMETHEE have been given in [25] where it has been applied on a simplified version of the HDI dataset [2]. This simplified version of the dataset consists in a subset of the twenty best ranked countries which are evaluated using only two criteria.

These first results, obtained with R equal to 10.000 and m equal to 5, were quite encouraging and lead to two observations:

- the ranking obtained with ROBUST PROMETHEE is quite close with the one obtained with PROMETHEE II (Kendall rank correlation coefficient between the two rankings is equal to 0.968).
- the use of ROBUST PROMETHEE did not lead to any rank reversal (instead of twenty when using the classical PROMETHEE II method).

To compute the number of rank reversal occurrences caused by a method, the following procedure is applied. First, an initial ranking $\mathscr{R}$ is built using the set A of n alternatives. Then, one alternative a_x is removed from this set (and from the initial ranking) and a new ranking $\mathscr{R}'$ is computed. The minimal number of pairwise inversions that are needed to obtain the original ranking ($\mathscr{R} - a_x$) from the new one ($\mathscr{R}'$) is the number of rank reversal caused by the removal of a_x. This operation is repeated for each a_x in A. The total number of rank reversals obtained with the method is equal to the sum of the rank reversals obtained for all the iterations of this procedure.

In the rest of this section, some further empirical investigation of the ROBUST PROMETHEE method will be carried out on the already mentioned HDI data set, as well as on subsets of the SHANGHAI ranking [1] and of the EPI data set [14].

Some technical details such, as the values of parameters of the PROMETHEE II method ($w_c, f_c(.), p, q$), will not be detailed in this work but can be found on the *GitHub* repository hosting the code of all the tests performed during this master thesis[10]. However, it should be clear that rankings obtained with ROBUST PROMETHEE will only be compared to rankings obtained with PROMETHEE II using the same sets of parameters. For reasons of ease, all tests are performed with preference functions of type 3 (see section 2.1).

4.2.1 Empirical results with the HDI data set

The numbers of rank reversals caused by the ROBUST PROMETHEE method on the HDI data set for different values of R and m are represented in Table 4.1.

$R\backslash m$	3	5	6	7	8	10	15
500	2.2	3.85	0.3	1.55	13.9	28.9	13
1000	0.55	0.8	0.05	0.15	11.95	24.15	10.9
5000	0	0.05	0	0	4.95	18.35	10.9
10000	0	0	0	0	3.55	18	8.1

Table 4.1: Quantity of rank reversals with a subset of 20 alternatives from the HDI data set (average of 20 repetitions).

As it has already been stated, the method does not yield any rank reversal for appropriate values of R and m. This is a significant improvement since PROMETHEE II yields 20 rank reversals when applied with this data set. However this number of rank reversals is highly dependent of R and m.

First, it can be seen that, up to a certain point, the number of rank reversals decreases as the value of R increases. This is not surprising and can be intuitively understood. When performing R iterations, it is as if an sample of size R of all the possible subsets of alternatives A_m was evaluated. The smaller the sample size is, the higher the variances of it's properties are, and the less the elements p_{ij} of our pairwise probability matrix are significant and stable. Therefore, the value of R should be chosen large enough.

The effects of the parameter m, on the other hand, are not so trivial. It can be seen that while R is relatively small, the value of m should not be taken neither too small, neither too large, its optimal value being 6 for this data set. On the other hand, if R is large enough, the values of m smaller or equal to 7 all lead to a ranking with no rank reversal occurrence.

4.2.2 Empirical results with the SHANGHAI data set

The numbers of rank reversals caused by ROBUST PROMETHEE on a random subset of 20 alternatives from the SHANGHAI data set for different values of R and m are represented in table 4.2. This set of alternatives consists in six criteria to maximise.

$R\backslash m$	4	6	8	9	12	15	18
1000	53.70	21.95	14.40	12.25	14.25	14.70	12.30
5000	46.70	18.50	11.45	10.75	12.3	14.45	12.0
10000	41.20	19.05	11.40	10.25	12.15	14.15	12.0
20000	40.25	19.00	11.50	10.30	12.10	14.25	12.0

Table 4.2: Quantity of rank reversals with a subset of 20 alternatives from the SHANGHAI data set (average of 20 repetitions).

With this set of alternatives, it is not possible anymore to completely avoid rank reversals.

PROMETHEE II leads to 14 rank reversals. The effectiveness in reducing this rank reversal quantity of ROBUST PROMETHEE thus highly depends on the value of m used and is limited. It can also be seen that the quantity of rank reversal is not monotonically decreasing with the increase of R.

The remaining rank reversals obtained using the optimal value of m (9) and a great enough value of R (5000) are further analysed in Table 4.3.

The two first columns contain the indices of the alternatives whose rank are reversed during the procedure. The two next ones contain the number of times these rank reversal happened with the ROBUST PROMETHEE and PROMETHEE II methods respectively.

a_i	a_j	Mean ROB	Mean PII	$\Delta\phi$	$\Delta\phi_{rob}$
10	7	2.10	2	0.00770	0.01627
18	14	0.60	0	0.00962	0.05815
19	2	3.00	3	0.00623	0.03515
10	9	2.00	2	0.00722	0.03765
9	7	3.05	7	0.00048	0.02137

Table 4.3: Analyse of remaining rank reversals on a subset of 20 alternatives from the SHANGHAI data set, m=9, R=5000, 20 repetitions

It can be seen that one “new kind” of rank reversal is introduced with the use of ROBUST PROMETHEE as the alternatives a_{18} and a_{14} can have their rank reversed (which is not the case with PROMETHEE II). However, this kind of rank reversal does not happen often (only slightly more than once every two executions).

The last columns contain the net flow and robust flow differences between the two concerned alternatives. It can be observed that the flow differences between the alternatives which suffer from rank reversal are relatively small.

4.2.3 Empirical results with the EPI data set

The numbers of rank reversals caused by ROBUST PROMETHEE on a subset of the 20 first alternatives from the EPI data set for different values of R and m are represented in table 4.4. This data set consists in nine criteria to maximise.

$R\backslash m$	3	4	7	9	12	14	16	18
500	49.1	63.3	31.8	22.8	10.8	9	8	9.0
1000	37.5	49.9	27.2	19.4	10.5	8.7	7.8	9.0
5000	19.4	38.3	25.5	20.2	11.3	9	8.7	9.0
8000	18.4	34.0	25.8	19.0	11.2	9	7.8	9.0

Table 4.4: Quantity of rank reversals with a subset of 20 alternatives from the EPI data set (average of 10 repetitions).

The use of PROMETHEE II on the same set of alternatives leads to 11 rank reversals.

The observations that can be done from these results are very similar to the ones done with the SHANGHAI data set. First, the decrease in the quantity of rank reversal is again relatively small. Second, the effectiveness of ROBUST PROMETHEE on this data set strongly depends on the value of m.

On the other hand, the best results are obtained with values of m which are high enough, which is quite different of what had been seen with the SHANGHAI data set and is exactly the opposite of what had been seen with the HDI data set.

Here under, the remaining rank reversals obtained using the optimal value of m (16) and a great enough value of R (5000) have been further analysed:

a_i	a_j	Mean ROB	Mean PII	$\Delta\phi$	$\Delta\phi_{rob}$
19	18	2.0	2	0.00332	0.09299
18	15	1.0	1	0.00399	0.13705
10	6	1.0	0	0.00493	0.09773
18	4	1.0	1	0.00311	0.10381
19	15	4.1	6	0.00067	0.04406
15	14	1.0	1	0.00422	0.09175

Table 4.5: Analyse of rank reversals on a subset of 20 alternatives from the EPI data set, m=16, R=5000, 10 repetitions

Again, the same remarks as with the SHANGHAI data set can be made:

- nearly all rank reversals caused by ROBUST PROMETHEE are also caused by the PROMETHEE II method,
- rank reversal occurs only between alternatives having close flows.

This last observation is important as it leads to the conjecture that ROBUST PROMETHEE will perform better (suffer from fewer rank reversal occurrences) when the alternatives of the problem are sufficiently discriminated.

After having empirically studied the results of the application this method on various data sets of different natures, we will now focus on studying it and more precisely, some of its properties related to the randomness of the selection of the subsets.

4.3 Mathematical properties of ROBUST PROMETHEE

So far, it has been observed empirically that ROBUST PROMETHEE was producing a ranking quite similar to the PROMETHEE II ranking, that it is more efficient when the alternatives are sufficiently discriminated and when the number of iterations is high enough, and that the optimal size of the subsets is not trivial to determine as it vary according to the set of alternatives on which the method is applied.

In the first part of this section, some lower bound on the number of iterations that are needed to obtain a meaningful probability matrix P (and therefore a meaningful ranking) will be given. Then it will be shown that, due to the random selection of the subsets and to the presence of rank reversal in PROMETHEE II, ROBUST PROMETHEE does not always respect the dominance relation.

4.3.1 Pairwise comparisons probability – lower bound on the number of iterations

As already stated, the ROBUST PROMETHEE ranking is obtained by computing the net flow score of each alternative on the probability matrix P. An element of P, p_{ij}, is computed using equation 4.3 and should represent the probability that an alternative a_i should be ranked before an alternative a_j.

However, in practice, p_{ij} makes sense only if alternative a_i and alternative a_j have been compared pairwise at least once during the ROBUST PROMETHEE procedure, i.e. when:

$$\sum_{r=1}^{R} A_{mr}(i) \cdot A_{mr}(j) \neq 0 \tag{4.5}$$

Two upper bounds on the probability that there exist at least two alternatives which have not been compared pairwise during the ROBUST PROMETHEE method will be given here under. These upper bounds on the probability will depend only on the number of alternatives of the problem, the size of the subsets, and the number of iterations of the method. These bounds will allow us to deduce a lower bound on the values of R and m given a desired upper bound probability that two alternatives have not been compared pairwise, and given a set of alternatives (with fixed size n).

The number of distinct couples that can be made from the n alternatives of the decision problem is given by:

$$N := C_n^2 \tag{4.6}$$

and, at each iteration, we compare m random alternatives pairwise which means that:

$$M := C_m^2 \tag{4.7}$$

distinct couples are evaluated.

To find the desired bound on the probability that there exists a couple of alternatives which has not been evaluated, we will use some known results of the *coupon collector's problem* [20].

The coupon collector's problem can be described as follow:

> *At each iteration i, Di balls are picked at random from an urn containing k red balls. These balls are painted in white before being put back in the urn. After how many draws will all the balls be painted in white ?*

It is obvious that this problem can be applied to the one of comparing all alternatives pairwise in the ROBUST PROMETHEE method. Instead of having to observe k red balls, there are N couples of alternatives that should be compared and instead of observing D_i balls at iteration i, M couples are compared. However, since we are interested in finding an upper bound of the probability, the assumption will be made that the couples are drawn one at the time which will highly simplify the reasoning. This assumption is sound since when the couples are drawn M at a time, we have the guarantee that these M couples are distinct one from the other which is not the case otherwise. The upper bound on the probability and the lower bound on the values of R and m that will be found will therefore be higher than the real ones.

We will define:

$D :=$ the number of random couples needed to observe all the couples at least once.

The probability of finding a new couple after that $(i-1)$ couples have already been found is given by:

$$p_i = \frac{N-(i-1)}{N} \tag{4.8}$$

The expectation and the variance of D are given by the following relations (see appendix B):

$$E[D] = N \cdot H_N \qquad Var[D] \leq \frac{\pi^2 N^2}{6} \tag{4.9}$$

with H_N being the Nth harmonic number

Chebyshev Inequality

Knowing the variance and the expectation of the random variable D, a probabilistic bound can be found using Chebyshev's inequality:

$$P[|D - N \cdot H_N| \geq c\frac{\pi N}{\sqrt{6}}] \leq \frac{1}{c^2} \tag{4.10}$$

By setting $p = \frac{1}{c^2}$ the desired upper bound probability:

$$P[|D - N \cdot H_N| \geq \frac{\pi N}{\sqrt{6p}}] \leq p \tag{4.11}$$

Since we are only interested to find a bound on the probability that all the couples have not yet been evaluated, the absolute value in the previous equation will be simplified as follow:

$$P[D \geq N \cdot H_N + \frac{\pi N}{\sqrt{6p}}] \leq p \tag{4.12}$$

Once again, this simplification has for consequence that the bound that will be found on the number of iterations and the size of the subsets is higher than the real one.

Since this bound should be valid after that $R \cdot M$ random couples have been evaluated, we find the following restriction:

$$\begin{aligned} R \cdot M \geq N \cdot H_N + \frac{\pi N}{\sqrt{6p}} \\ \Rightarrow \quad P[D \geq R \cdot M] \leq p \end{aligned} \tag{4.13}$$

Finally, by replacing M and N by their definitions (4.6 and 4.7) we find the desired bound:

$$\begin{aligned} R\frac{m^2 - m}{2} \geq \frac{n^2 - n}{2} \cdot H_{\frac{n^2-n}{2}} + \frac{\pi \frac{n^2-n}{2}}{\sqrt{6p}} \\ R(m^2 - m) \geq (n^2 - n)H_{\frac{n^2-n}{2}} + \frac{\pi(n^2 - n)}{\sqrt{6p}} \end{aligned} \tag{4.14}$$

Union bound argument

Another bound can easily be found simply by making the union bound of the probabilities that one couple has not be evaluated after the evaluation of c random couples. The probability $P(i, c)$ of not having found a specific couple (called here the ith couple) after c random couples have been drawn is lower than:

$$\begin{aligned} P(i,c) &= (1-\frac{1}{N})^c \\ &\leq e^{\frac{-c}{N}} \end{aligned} \tag{4.15}$$

By making the union of the probabilities for the N couples:

$$P[D > c] \leq N \cdot e^{-\frac{c}{N}} \tag{4.16}$$

This inequality can be justified by the fact that $P[D > c]$ can be seen as the probability that at least one couple has not yet been evaluated after the evaluation of c couples, and by remembering Boole's inequality:

$$P\left(\bigcup_i B_i\right) \leq \sum_i P(B_i) \tag{4.17}$$

with the B_i forming a set of events.

Setting $p = N \cdot e^{-\frac{c}{N}}$ the desired upper bound probability, the above bound becomes:

$$P[D \geq N \cdot \ln(\frac{N}{p})] \leq p \tag{4.18}$$

Such as for the bound found using Chebyshev's inequality, this bound should be valid after that $R \cdot M$ random couples have been evaluated:

$$\begin{aligned} R \cdot M &\geq N \cdot \ln(\frac{N}{p}) \\ \Rightarrow \quad & P[D \geq R \cdot M] \leq p \end{aligned} \tag{4.19}$$

This restriction can easily be translated in terms of parameters of the ROBUST PROMETHEE method (R, m and n):

$$R \cdot (m^2 - m) \geq (n^2 - n)\ln(\frac{n^2-n}{2p}) \tag{4.20}$$

More details about the two reasonings here above can be found in [20].

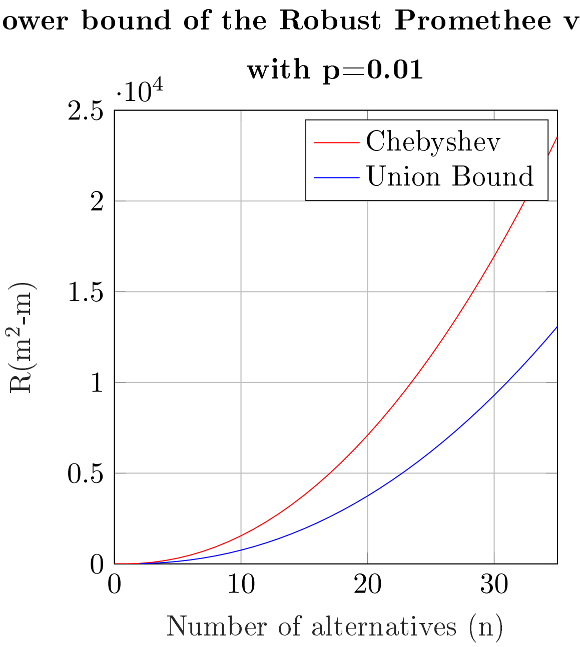


Figure 4.1: lower bound for R and m

Comparison of the bounds

Since the harmonic number has a logarithmic asymptotic behaviour (but is greater than a logarithm) we can deduce that the bound found using the union bound argument will be tighter than the one found using Chebyshev's inequality. This statement is confirmed by the Figure 4.1 illustrating the bound on R and m in function of the number of alternatives.

It can be seen that for sets of 20 alternatives, $R(m^2 - m)$ should be higher than about 3750. This does not correspond to the values that have been found during the empirical investigation of the previous section where the optimal values of R had to be more or less equal to 5000 (which would lead $R(m^2 - m)$ to be equal to $1e5$ when m is equal to 5).

This should not be surprising as the bound that have been found only guarantees that the probability that at least one couple of alternatives has not been evaluated during the procedure is under 1%, in other words, that the probability that all couples of alternatives have not been compared at least once is under 1%. Of course, each couple of alternatives should be evaluated more than once in order for the matrix P to correctly represent the probabilities of the alternatives to be ranked before others.

The bound found should therefore more be seen as a strict minimum for the methods usability, and certainly not as an optimal value.

4.3.2 Non respect of the dominance relation

As already shown, PROMETHEE II respects the natural dominance relation (see section 3.2.1). Unfortunately, ROBUST PROMETHEE on the other hand is not guaranteed to necessarily respect this relation. The following paragraphs will give an example of an application where an alternative a_w, which dominates slightly another alternative $a_{w'}$, will be ranked after this other alternative.

The evaluations of the alternatives considered for the example are given in Table 4.6.

a	$f_1(.)$	$\dots$	$f_c(.)$	$\dots$	$f_k(.)$
a_1	$f_1(a_1)$	$\dots$	$f_c(a_1)$	$\dots$	$f_k(a_1)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_{t_1}	$f_1(a_{t_1})$	$\dots$	$f_c(a_{t_1})$	$\dots$	$f_k(a_{t_1})$
a_{t_2}	$f_1(a_{t_2})$	$\dots$	$f_c(a_{t_2})$	$\dots$	$f_k(a_{t_2})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_{t_x}	$f_1(a_{t_x})$	$\dots$	$f_c(a_{t_x})$	$\dots$	$f_k(a_{t_x})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_w	$f_1(a_w)$	$\dots$	$f_c(a_w)$	$\dots$	$f_k(a_w)$
$a_{w'}$	$f_1(a_w)$	$\dots$	$< f_c(a_w)$	$\dots$	$f_k(a_w)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
a_n	$f_1(a_n)$	$\dots$	$f_c(a_n)$	$\dots$	$f_k(a_n)$

Table 4.6: Evaluation table of a set of alternatives not necessarily respecting the dominance relation with the ROBUST PROMETHEE method

This set A of alternatives cannot be any set. It must include the following alternatives:

a_w, $a_{w'}$: these two alternatives are our witnesses as the dominated alternative ($a_{w'}$) will be ranked before the dominant one (a_w). However, to justify the assumptions that will be done in the rest of the reasoning, these two alternatives must be considered to be very similar one to the other (their evaluations are equal on all the criteria excepts f_c where a_w is slightly better).

a_t: at least two alternatives written as $a_{t_1} \dots a_{t_x}$ must suffer from the rank reversal phenomenon with the alternatives a_w and $a_{w'}$ when the PROMETHEE II method is applied on sets of m alternatives chosen from A (containing of course the concerned alternative a_t and a_w or $a_{w'}$).

The characterisation of the alternatives a_t has the following consequence. During the application of the ROBUST PROMETHEE procedure, at some iterations the alternatives a_w and $a_{w'}$ could be ranked before some alternative from a_t, and at some other iterations they could be ranked after some alternative from a_t.

Now suppose that at each of the $r = 1 \ldots R$ iterations one of the following cases happen:

1. If both a_w and $a_{w'}$ are selected in the m random alternatives, then no alternative a_t is selected:

$$\big(A_{mr}(a_w) \cdot A_{mr}(a_{w'})\big) \cdot \sum_{t=t_1}^{t_x} A_{mr}(a_t) = 0 \tag{4.21}$$

2. If a_w and one alternative a_t are selected in the m random alternatives, then a_w will be ranked after the alternative from a_t at this iteration.

$$A_{mr}(a_w) \cdot A_{mr}(a_t) = 1 \Rightarrow Beats_r(a_t, a_w) = 1 \qquad \forall t = t_1 \ldots t_x \tag{4.22}$$

   This means that the alternatives a_t will always be ranked before the alternative a_w and that the following elements of the matrix P will have the following values:

$$p_{wt} = 0 \qquad p_{tw} = 1 \qquad \forall t = t_1 \ldots t_x \tag{4.23}$$

3. If $a_{w'}$ and one alternative a_t are selected in the m random alternatives, then $a_{w'}$ will be ranked before the alternative from a_t at this iteration.

$$A_{mr}(a_{w'}) \cdot A_{mr}(a_t) = 1 \Rightarrow Beats_r(a_{w'}, a_t) = 1 \qquad \forall t = t_1 \ldots t_x \tag{4.24}$$

   This means that the alternatives a_t will always be ranked after the alternative $a_{w'}$ and that the following elements of the matrix P will have the following values:

$$p_{w't} = 1 \qquad p_{tw'} = 0 \qquad \forall t = t_1 \ldots t_x \tag{4.25}$$

It will also be assumed, that since a_w and $a_{w'}$ are similar one to the other, they will obtain similar rankings when compared with the other alternatives:

$$p_{w'i} \simeq p_{wi} \qquad p_{iw'} \simeq p_{iw} \qquad \forall i \neq w, w', t_1 \ldots t_x \tag{4.26}$$

Finally, it should also be noted that $p_{ww'}$ is not always strictly equal to 1 but will lie between $[0.5, 1]$ even if $a_{w'}$ is dominated by a_w. This is due to the fact that $Beats_r(a_w, a_{w'})$ can be equal to 0.5 since two alternatives dominating each other could obtain the same ranking (see equations (3.5) and (4.2)).

When computing the difference between the robust flows of a_w and $a_{w'}$ computed on the probability matrix, the following results will be obtained:

$$\begin{aligned}
\phi_{rob}(a_{w'}) - \phi_{rob}(a_w) &= \frac{1}{n-1}\sum_{i=1}^{n}[(p_{w'i} - p_{iw'}) - (p_{wi} - p_{iw})] \\
&\simeq \frac{1}{n-1}\sum_{t=t_1}^{t_x}[(p_{w't} - p_{tw'}) - (p_{w't} - p_{tw'})] \\
&\quad + \frac{1}{n-1}\big((p_{ww'} - p_{w'w}) - (p_{w'w} - p_{ww'})\big) \\
&\geq \frac{1}{n-1}[\sum_{t=t_1}^{t_x} 2 - 2] \\
&> 0
\end{aligned} \tag{4.27}$$

The last inequality is justified by the hypothesis of this example which stated that the set $a_t = a_{t_1} \dots a_{t_x}$ was composed of at least two alternatives.

It can therefore be seen from this example that the dominance relation could be violated by the ROBUST PROMETHEE method. This phenomenon is closely related to the presence of rank reversal in PROMETHEE II. In fact, it is also related to some form of nontransivity. The dominated alternative $a_{w'}$ cannot be ranked before the dominant alternative a_w in one of the subsets, but it can be ranked before some alternatives a_t. These alternatives a_t, on the other hand, can be ranked before a_w in some subsets.

However, even when there exists alternatives a_t which can be ranked before or after some alternatives a_w and $a_{w'}$ depending on the subsets, a_w will always have a probability at least as great of being ranked before a_t than $a_{w'}$. Therefore, since the ROBUST PROMETHEE method must be employed with R relatively high, unbalanced repartition of the samples that could lead to violations of the dominance relation are not likely to happen in practical problems.

4.4 Synthesis of the ROBUST PROMETHEE method and conclusion

After having studied ROBUST PROMETHEE mathematically and using empirical results, we can conclude that it has some advantages and some drawbacks which both should be mitigated.

First, it can be said that this method seems, up to a certain point, to be succeeding in its purpose of reducing the number of rank reversals. The improvement is however not always as drastic as first observed with the HDI data set and it is neither guaranteed that the method will be effective on all data sets.

It has also been shown that the dominance relation was not necessarily respected and some counterintuitive rankings could be obtained. Such violations should however have very low probabilities to occur and have not yet been observed.

The most limiting characteristic of ROBUST PROMETHEE that has been found is probably that the determination of appropriate parameters is not an easy task and vary from one data set to another. Furthermore the variations of the best values of the parameter m are not small variations. It has been seen that this optimal value could be equal to about 30% of the size of the alternative set as it could also be equal to 80%. This could be a serious obstacle for solving practical decision aid problems as the parameters that should be used will not be known in advance.

Finally, it should also be pointed out that the fact that some rank reversal are still occurring should not automatically discredit this method. This chapter has mainly consisted in studying the quantity of rank reversal occurrences but it has also been seen that ROBUST PROMETHEE was producing rankings relatively similar as PROMETHEE II. It should therefore still be studied if the rankings produced are satisfying or not (are they more satisfying than the ones produced by the original PROMETHEE II method ?).

In the following chapter, a new method, REFERENCED PROMETHEE, will be studied. This method handles the rank reversal problem with a different approach. Instead of building a ranking similar to the one produced by PROMETHEE II and trying to reduce the number of rank reversal, by construction, REFERENCED PROMETHEE avoids any rank reversal. It will however be studied if it can produce rankings similar to the ones produced by PROMETHEE II.

Chapter 5

The REFERENCED PROMETHEE method

The second modification of PROMETHEE II that will be studied in this work is called REFERENCED PROMETHEE and has initially been proposed by Doan and De Smet in [11].

The main idea of this method to avoid the rank reversal phenomenon is to use a fixed set of p reference profiles ($\mathscr{R}$).

Iteratively, each of the alternatives a_i from A is added to $\mathscr{R}$[11]:

$$\mathscr{R}_i = \mathscr{R} \cup a_i \tag{5.1}$$

Its score is then computed as its net flow score computed on $\mathscr{R}_i$ [11]:

$$\phi_{ref}(a_i) = \frac{1}{p} \sum_{r \in \mathscr{R}_i} [\pi(a_i, r) - \pi(r, a_i)] \tag{5.2}$$

This net flow will be called the referenced flow.

It is not necessary to apply the full PROMETHEE II method on $\mathscr{R}_i$ since we are not interested in the net flows obtained by the reference profiles, neither by their pairwise preferences $\pi(r, r')$. Since the number of reference profiles should be relatively small compared to the number of alternatives, the computational complexity of REFERENCED PROMETHEE will be smaller than the one of PROMETHEE II.

Finally, the alternatives are ranked according to their referenced flows. In this way, their scores only depend on a fixed set of reference profiles and therefore the relative ordering between two alternatives can not be influenced by other alternatives. This leads to the impossibility for rank reversals to happen but comes with the additional cost of having to find a fixed and appropriated set of reference profiles.

As it will be seen, the choice of the reference profiles set is of crucial importance. This chapter mainly consists in the study of the influence of these sets, and methods to find or build satisfactory ones.

The judgment whether a ranking is appropriate or not given the parameters provided by the decision maker goes beyond the scope of this work. We will therefore focus on finding sets of reference profiles that produce rankings similar to the ones produced by PROMETHEE II, as it can be assumed, due to the wide usage of this method that its rankings are satisfactory enough. To do so, the quality of a set of reference profiles will be assessed using as indicator Kendall's rank correlation coefficient between the PROMETHEE II rankings and the REFERENCED PROMETHEE ones obtained with this set of references.

During the rest of this chapter, as with ROBUST PROMETHEE, rankings obtained with REFERENCED PROMETHEE will only be compared to rankings obtained with PROMETHEE II which have been obtained with the same sets of shared parameters $(w_c, f_c(.), \dots)$.

Before studying how to build or find satisfactory sets of reference profiles, some characterisations of the method and of references sets will be discussed.

5.1 Characteristics of sets of reference profiles

The following properties of reference profiles sets have already been demonstrated in [11]:

- Permutations of the evaluations for a given criterion between different reference profiles do not influence the REFERENCED PROMETHEE ranking. This means that we can arbitrarily impose that the references in a set must pairwise dominate each other.
- REFERENCED PROMETHEE respects the natural dominance relation.

In this section, two other characteristics of the method will be investigated. First, the minimal number of references needed to discriminate all the alternatives will be found. This quantity will then be used to study if it is possible to reproduce the PROMETHEE II rankings with such small amounts of reference profiles.

5.1.1 Number of reference profiles needed to discriminate all the alternatives

A first question that comes to mind when trying to characterize the sets of reference profiles is how many profiles are needed to discriminate all the alternatives of a decision problem.

Indeed, as the next sections focus on trying to find or build references sets, it seems interesting to at least have a rough idea of a lower bound on the quantity of reference profiles that would be needed.

Instinctively, one could suppose that the quantity of references necessary will grow as the size of A grows. It can be seen from Table 5.1 that this is not the case.

	n			
p	**10**	**20**	**40**	**80**
2	13	45	196	840
3	11	47	196	886
5	11	53	173	833
10	16	44	195	829
15	11	34	171	814
25	12	59	165	786

Table 5.1: Quantity of couples of indifferent alternatives (sum of 60 repetitions from the EPI, SHANGHAI, GEQ data sets with random parameters with an indifference threshold of 0.001) for a set of n alternatives evaluated with p reference profiles.

As expected, the number of draws between the alternatives increases with the quantity of alternatives of the decision problem, but it does not (or at least not significantly) decrease with the quantity of reference profiles used.

It can therefore be concluded that few reference profiles are sufficient to discriminate as well as possible all the alternatives of a problem (with an indifference threshold that should however decrease as the number alternatives increases).

It will now be investigated if REFERENCED PROMETHEE can reproduce PROMETHEE II rankings and particularly if this can be done with a small quantity of reference profiles.

5.1.2 Possibility of replicating the PROMETHEE II rankings

Before trying to find methods that build references sets that reproduce the PROMETHEE II rankings it should be verified whether such references sets exist or not. To answer this question, a first solution that came to our mind was to use a genetic algorithm [32].

A genetic algorithm is an iterative algorithm inspired by the process of natural selection that took and is still taking place during the evolution of life. These algorithms are usually used to find "good solutions" for nonlinear or hard optimisation problems where it is not possible to find an optimal solution in a reasonable amount of time.

Genetic algorithms work as follows. At each iteration a population of individuals, which is usually called a generation, is considered. Each of them is evaluated using the objective function provided with the problem. The individuals of the population are then combined to form a new generation. This step is called the reproduction. Only those with the best scores are selected for the reproduction and the higher the score they obtain, the more they will reproduce. After the reproduction, some random mutations can also occur (with low probability) which should help the algorithm not to get stuck in a local optimum. There can be different termination conditions for this algorithm. The algorithm can stop after a given number of iterations, after a given amount of elapsed time or when one of the individuals reaches a score judged as sufficient by the user.

A genetic algorithm is particularly well suited for the problem of finding a set of references profiles since:

- The score that must be optimized (the Kendall rank correlation coefficient) is not linear with regard to the reference profiles used.
- Sets of reference profiles are well suited for the reproduction operation. Two new sets can be obtained by shuffling the reference profiles of two existing ones.

A sketch of a general genetic algorithm is illustrated in the Algorithm 1.

This algorithm make uses of some "genetic algorithm specific elements" which had to be defined for our specific research of optimal references sets:

population: collection of possible sets of reference profiles (each set being considered as an individual).

objective function $\mathscr{F}$: Kendall's correlation tau between the ranking obtained by the concerned individual and the ranking obtained with PROMETHEE II.

Algorithm 1 generic genetic algorithm

```
Input: population_size, mutation_probability, $\mathscr{F}$, MAXIT, CEIL
population = initial_popultaion(population_size)
scores = evaluate_population(population, $\mathscr{F}$)
iteration = 0
while max(scores) $<$ CEIL and iteration $<$ MAXIT do
    population_fitness = compute_fitness(scores, population_size)
    parents = chose_parents(population_fitness)
    population = combine(parents)
    population = mutate(population, mutation_probability)
    scores = evaluate_population(population, $\mathscr{F}$)
    iteration = iteration + 1
end while
return population
```

evaluate_population(): evaluate each individual of the population using the objective function.

compute_fitness(): compute the number of times each individual will be parent according to its score.

combine_parents(): randomly form pairs of parents. The reference profiles of a pair of parents are shuffled and separated to form two new sets of reference profiles which will, after mutation, form the individuals of the next generation.

One example of combination of two parents formed by four references can be seen in the figure here under.

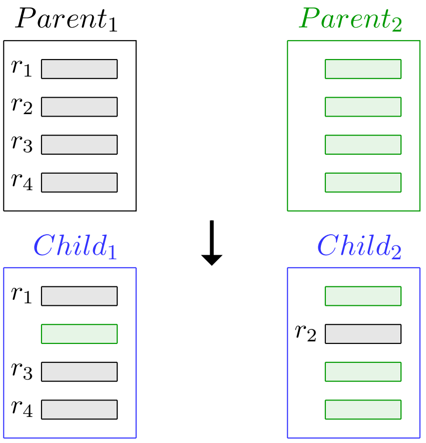


Figure 5.1: Representation of the combination of two sets of reference profiles

mutate(): modify each evaluation, of each reference, in each reference set of the new population with a low probability. The modification of an evaluation consist in multiplying it by a factor taken uniformly at random between 0.75 and 1.25.

This genetic algorithm has been applied on random subsets of alternatives of different sizes. The proportion of subsets for which a set of reference profiles which reproduces the PROMETHEE II ranking exactly has been found is summarised in Table 5.2.

	4 references			5 references		
n	EPI	SHA	GEQ	EPI	SHA	GEQ
20	15/15	15/15	14/15			0/1
25	14/15	13/15	14/15	0/1	2/2	0/1
30	15/15	12/15	12/15		3/3	3/3
40	13/15	8/15	7/15	0/2	3/7	1/8
50	10/15	2/15	2/15	1/5	4/13	5/13

Table 5.2: Proportion of random subsets reproducing the PROMETHEE II ranking found with the genetic algorithm.

First, the algorithm has been applied on 45 different data set of each size searching for sets of 4 reference profiles. Then, the algorithm has been retried for each of the data set for which no perfect set had been found but searching this time for sets of 5 reference profiles.

For each of the different configurations shown in this table, the worst correlation coefficient found from all of the 15 random subsets (or from the remaining subsets when 5 reference profiles where used) is indicated in the Table 5.3.

	4 references			5 references		
n	EPI	SHA	GEQ	EPI	SHA	GEQ
20	1.0	1.0	0.989			0.884
25	0.993	0.94	0.98	0.946	1.0	0.993
30	1.0	0.912	0.96		1.0	1.0
40	0.964	0.828	0.899	0.941	0.948	0.884
50	0.88	0.905	0.841	0.947	0.929	0.875

Table 5.3: Worst τ_K between the PROMETHEE II ranking and REFERENCED PROMETHEE rankings with the best set of reference profiles found with the genetic algorithm.

For most random subsets of size smaller or equal to 30, it has been possible to find sets of reference profiles that reproduce exactly the PROMETHEE II ranking. For the few subsets where this was not the case, the best sets of references found by the genetic algorithm however produced rankings relatively similar to the one obtained with PROMETHEE II (the worst correlation coefficient being 0.912).

For larger subsets of alternatives, there is a strong decrease in the proportion for which a set of reference profiles which perfectly reproduces the PROMETHEE II ranking has been found. However, the sets found by the algorithm again produce quite similar rankings (the worst correlation coefficient being 0.828).

It should also not be forgotten that this genetic algorithm is a heuristic, aimed at providing good solutions for hard optimisation problems. The fact that no perfect set of reference profiles was found in some configurations does not imply that these do not exist.

From these results we can conclude that there seem to always exist small sets of reference profiles producing rankings similar to the PROMETHEE II rankings. It is therefore justified to try to build methods and procedures which should provide such sets of references. This will be done in the next section.

5.2 Constructive methods to build sets of reference profiles

REFERENCED PROMETHEE is based on the comparison of the alternatives with a set of reference profiles. Without providing some tools to find some sets of reference profiles to be used, the method becomes practically unusable. Therefore, two approaches to produce sets of reference profiles will be analysed.

The first approach that will be studied relies on defining a construction strategy that builds sets of reference profiles given the sets of alternatives on which the method is applied. This approach is relatively simple and does not require much additional information from the decision maker.

The second approach, on the other hand, is an interactive approach based on a questioning procedure. The aim of this procedure is to restrain the set of all possible references sets to only ones producing rankings consistent with the decision maker's answers (under the hypothesis that he answers coherently).

5.2.1 Building sets of reference profiles using strategies

When asked to build a set of reference profiles that would be used to apply REFERENCED PROMETHEE on a set of alternatives A, some obvious possibilities could come to mind. For example, a possibility could consist in selecting profiles with evaluations which are equally spread from the worst to the best evaluation of the alternatives for each criterion. Another possibility could consist in using percentiles of the evaluations of the alternatives.

Such ideas will be the basis for what will be called strategies. An infinite number of different strategies could be considered but only four will be compared in this work. These are presented here under:

Strategy I: build p alternatives at random.

Strategy II: build p alternatives as percentiles of the evaluations of A.

Strategy III: build p alternatives equally spaced between the evaluations of A.

Strategy IV: build p alternatives as percentiles between the first and the third quartiles of the evaluations of A.

These strategies have been used to build sets of references for different random subsets from the three following data sets: EPI, SHANGHAI and GEQ (GEQ being a particular subset of the HDI project concerning the gender equality conditions in different countries).

Since it has been seen in the previous section that it was generally possible to reproduce the PROMETHEE II ranking with sets of 4 reference profiles, the simulations in this section, applied on sets of 20 alternatives, have also been performed with strategies producing 4 references.

The correlation tau obtained between the PROMETHEE II ranking and the ones produced by REFERENCED PROMETHEE used with the specified strategies are shown in Table 5.4. The tests have been performed for 9 subsets (S) of alternatives taken uniformly at random.

	EPI			SHA			GEQ		
Strategy	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
I	0.50	0.60	0.44	0.39	0.54	0.61	0.19	0.33	0.37
II	0.75	0.72	0.39	0.56	0.57	0.92	0.63	0.58	0.21
III	0.43	0.45	0.14	0.48	0.42	0.24	0.50	0.23	0.16
IV	0.80	0.46	1.0	0.45	0.79	1.0	0.59	0.57	0.64

Table 5.4: Kendall's correlation coefficients obtained between the PROMETHEE II and REFERENCED PROMETHEE rankings for different strategies. ($n = 20$)

The results seem to show that only strategies II and IV produce rankings more similar to the PROMETHEE II ones than by using sets of random reference profiles (strategy I). There is, on the other hand, no strategy which performs better than any other on all the random subsets, or even which produces satisfying rankings on all the subsets.

In Table 5.5, some statistical estimators of the strategies which have been computed on a larger number of random subsets are shown.

Strategy	τ_{min}	$\mu(\tau)$	$\sigma^2(\tau)$
I	-0.328	0.31	0.045
II	-0.222	0.49	0.051
III	-0.31	0.36	0.043
IV	-0.029	0.69	0.038

Table 5.5: Statistical estimators of Kendall's correlation tau for 3000 randoms subsets of 20 alternatives (from the EPI, SHANGHAI and GEQ data sets)

From these new results it can be confirmed that no strategy is guaranteed to produce similar ranking to PROMETHEE II rankings. Each strategy produced at least one ranking with a negative correlation coefficient for one of the subsets. However, the rankings produced by the fourth strategy have an average coefficient of nearly 0.7.

As the strategies proposed in this section do not produce satisfying enough results, a more elaborated procedure for building sets of reference profiles will be detailed in the next section. This new procedure should provide references sets that produce rankings more similar to the PROMETHEE II ones at the cost of needing more informations from the decision maker.

5.2.2 Building sets of reference profiles using an adaptative questioning procedure

The procedure developed in this section is a simplified adaptive questioning procedure based on the one presented by Eppe and De Smet in [12] with the difference that it will not be aimed at finding a vector of weights w. This new procedure is instead aimed at, given a decision problem (a set A of n alternatives evaluated on k criteria) and given a decision maker (preference functions and a vector of weights w), finding a set $\mathscr{R}$ of p reference profiles which leads to a "satisfactory" REFERENCED PROMETHEE ranking. The resulting questioning procedure would then act as black box to provide such sets to the decision maker.

Since we are interested in this work in finding references sets which reproduce the rankings obtained with the PROMETHEE II method, we will consider an artificial decision maker which provides answers consistent with these rankings.

In the rest of this section, a simplification of the original adaptive questioning procedure presented in [12] will first be described, then the modifications that were needed to apply this method for searching sets of reference profiles will be detailed. Finally, some results of the application of the new method will be discussed.

Adaptive questioning procedure defining a vector of weights

Here under a synthesis of the procedure presented in [12] is given.

The procedure consists in the delimitation, in a k dimensional space, of a domain Ω representing the possible values of the vector of weight w.
Each point α of this domain thus represents a valid weight vector w_α and can be associated to the ranking obtained by applying PROMETHEE II with w_α. Each domaine Ω is therefore associated with a set consisting of all the admissible rankings (that will be called Γ). Even if the number of points in Ω is infinite, there can only be a discrete number of rankings and Γ will therefore be finite and discrete.

The procedure works as follows. At each iteration i, a query is asked to the decision maker. Three types of queries where initially proposed by Eppe and De Smet but only one type will be considered here (since this is the only type of query used in the procedure aimed at finding sets of reference profiles).

This type of query consists in proposing two alternatives of A to the decision maker and asking him which of the two he prefers. His answer will yield a constraint on the domain Ω of possible weight factors, reducing its size. By denoting Ω_i the domain still admissible at the iteration i:

$$\Omega_{i+1} \supseteq \Omega_i \tag{5.3}$$

Suppose for example that the decision maker answers that he prefer alternative a_i over alternative a_j. This means that $\phi(a_i)$ should be greater than $\phi(a_j)$.

With the introduction of the uni-criterion net flow ϕ_c, $\phi(a_i)$ can be written as:

$$\begin{aligned}\phi(a_i) &= \frac{1}{n-1}\sum_{x=1}^{n}\sum_{c=1}^{k} w_c \cdot \big(P_c(a_i,a_x) - P_c(a_x,a_i)\big)\\ &= \sum_{c=1}^{k} w_c \left[\frac{1}{n-1}\sum_{x=1}^{n}\big(P_c(a_i,a_x) - P_c(a_x,a_i)\big)\right] \\ &= \sum_{c=1}^{k} w_c \ \phi_c(a_i)\end{aligned} \tag{5.4}$$

Since all the parameters needed to compute the uni-criterion net flows of any alternative are constant, these will also be constant.

The answer of the decision maker will therefore result in the following constraint on w [12]:

$$\phi(a_i) > \phi(a_j) \Rightarrow \sum_{c=1}^{k} w_c\big(\phi_c(a_i) - \phi_c(a_j)\big) > 0 \tag{5.5}$$

These are linear constraints that allows to restrict Ω_i as shown in Figure 5.2 taken from [12].

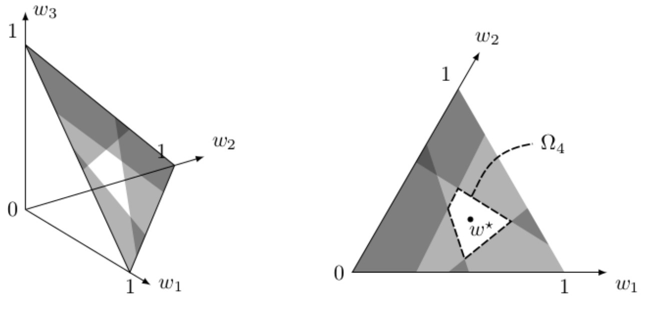


Figure 5.2: Planar representation of the weight domain for a 3-criteria problem with 4 constraints. The darkness of each area is an indicator of the quantity of constraints that are violated inside that area. [12].

The domain of admissible weight factors being reduced at each iteration, the set Γ of the rankings produced associated to this domain will generally also be reduced.

After some number l of iterations, the remaining set Γ_l will contain only rankings which are similar to the ranking which correctly represents the preferences of the decision maker.

This procedure should of course achieve these results by asking the less possible questions. Indeed, if the decision maker has to answer too many queries, then the benefits of using a multi-criteria decision aid method vanish as he still has to be able to provide much of the information about his preferences himself.

Therefore, a particular attention is given to the choice of the queries that will be asked. The query generation scheme used in [12] is based on the idea presented by Iyengar et al. in [15]. It consists in asking queries which split the space of admissible weights the most equally possible in two (maximizing the worst case reduction of the domain). Since the answer of the decision maker can not be anticipated, it is not known which of the two splitted parts of the domain will be eliminated by the new constraint.

The query asked at each iteration i is generated as follows [12]:

1. The center $w^\star$ of the domain of admissible weights factors is computed.
2. A ranking R is established by applying PROMETHEE II with $w^\star$ on A.
3. The alternatives with the best rankings are kept. Each couple of these alternatives will be considered as a query candidate.
4. The cut of Ω_i associated with each query candidate is evaluated and the query splitting this domain the most equally is asked to the decision maker.

Some details of this procedure were voluntarily omitted in this synthesis which is aimed at helping to understand the underlying idea of this procedure. The interested reader can refer to [12] for more details.

Adaptive questioning procedure defining sets of reference profiles

Few differences must be taken into account when trying to find sets of reference profiles instead of vectors of weights. Indeed, in this case, when the decision maker express his preference of an alternative a_i over an alternative a_j, then the referenced flow of a_i will have to be greater than the one of a_j, which means that:

$$\begin{aligned}\frac{1}{p}\sum_{r\in\mathscr{R}}\sum_{c=1}^{k} w_c \cdot \left[P_c(a_i,r) - P_c(r,a_i)\right] \\ > \frac{1}{p}\sum_{r\in\mathscr{R}}\sum_{c=1}^{k} w_c \cdot \big[(P_c(a_j,r) - P_c(r,a_j))\big]\end{aligned} \tag{5.6}$$

Due to the nonlinearity of the preference functions P_c, it is not possible anymore to find linear constraints on the sets of reference profiles $\mathscr{R}$. The domain of admissible points is also not guaranteed to be convex nor connected anymore.

Therefore, the initial domain of possible sets of reference profiles will be represented by a sufficiently large set of points S. These points are selected uniformly at random in the $p \times k$ dimensional space representing all the possible sets of p reference profiles.

At each iteration i, the points which are not consistent with the answer given by the decision maker will be removed from this set. The remaining set, after each query, will be called S_i.

The query generation scheme has also slightly been modified. Query candidates have not been selected using the ranking obtained with the center of admissible points. Indeed, since the constraints are not linear anymore and since the center is not guaranteed to be an admissible point, there does not seem to be any reasons why it should be used. Instead, since the domain considered at each iteration consists of a finite set of points S_i, the ranking $R^\star$ which is obtained with the largest number of points will be used to select the query candidates.

All the couples of pairwise successively ranked alternatives are then selected as query candidates. The query candidate which will lead to the removal of the largest amount of points in S_{i-1} (in the worst case) is proposed to the decision maker.

The procedure is synthesized in Algorithm 2.

Algorithm 2 Adaptive questioning procedure for finding a set of reference profiles

```
Input: $A$, $\mathscr{P}$, $w$, $p$, points_qty, procedure_iterations
PII_ranking = PrometheeII(A, $\mathscr{P}$, $w$)
constraints = $\emptyset$
$S_0$ = initial_points($A$, $p$, points_qty)
for i = 0 ... procedure_iterations do
    $\Gamma_i$, commonest_ranking = analyse($S_i$, PII_ranking, A, $\mathscr{P}$, $w$)
    query = evaluate_candidates($S_i$, commonest_ranking, A, $\mathscr{P}$, $w$)
    constraint$_i$ = ask_DM(query, PII_ranking)
    constraints = constraints $\cup$ constraint$_i$
    $S_{i+1}$ = update_points($S_i$, constraint$_i$)
    if size($S_{i+1}$) $<$ points_qty then
        $S_{i+1}$ = $S_{i+1}$ $\cup$ add_points(constraints, $A$, $p$)
    end if
end for
```

Where the following functions could require some additional information:

initial_points(): build the initial set of points. Each point represents a set of reference profiles composed of p profiles. The evaluations of these initial sets of reference profiles are selected uniformly at random between "0.5×the lowest evaluation in A" and "1.5×the highest evaluation in A".

evaluate_candidates(): computes, for each pair a_i and a_j of successive alternatives in the *commonest_ranking*, the minimal number of points that would be removed from S_i if they formed the next query. This is done by computing for each point whether a_i or a_j would obtain the highest referenced flow. The pair of alternatives which maximizes the minimal number of points that could be removed is returned as the next query.

update_points(): returns a reduced set of points by removing all points from the given set which do not respect the new constraint.

analyse(): analyses all the rankings produced by the remaining points to provide some information about their convergence. This is needed both to be able to analyse the performances of the procedure and to be able to decide after which iteration it should stop.

In the algorithm presented here above, only the different rankings (Γ_i), and the commonest ranking obtained from S_i are computed (for readability purposes). Of course, the other indicators that will be used in the empirical results (such as the correlation coefficients between these rankings and the PROMETHEE II ranking) are also computed with this function.

add_points(): Due to performances issues, it is not possible to start with a set S_0 which contains enough points. Therefore, if at a given iteration, the quantity of points remaining is too low, then some points are added. The points added must of course respect it already imposed constraints.

When several of these constraints have already been imposed, it can become difficult to find new points which satisfy all these constraints at random. To overcome this difficulty, two approaches are used simultaneously. At each iteration, the procedure tries to add some random points and the also tries to add points close to the already existing ones (by multiplying their coordinates by some factors selected uniformly at random between 0.9 and 1.1).

The results of the application of this algorithm on some random sets of alternatives are illustrated in the next section.

Some results of the application of the adaptive questioning procedure

The questioning procedure has been applied on different data sets with contrasting results. The study of its application on three sets of alternatives illustrating this contrast will be detailed here under.

These three sets of alternatives are all composed of 20 or 30 alternatives with weight factors and preference functions taken uniformly at random as usual. The sets of reference profiles considered are always composed of 4 profiles.

The first results are shown in Table 5.6 where the procedure was applied on 20 random alternatives taken from the EPI data set.

i	$\mid S_i \mid$	$\mid \Gamma_i \mid$	τ_{min}	τ_{max}	τ_{μ}	τ_{med}	removed
5	3006	925	-0.200	1.000	0.405	0.410	1441
9	3000	333	-0.189	1.000	0.537	0.610	1387
11	2384	130	-0.126	1.000	0.646	0.715	1584
13	2218	32	0.178	1.000	0.757	0.715	1520
14	2905	22	0.421	1.000	0.916	1.000	611
16	3291	3	0.705	1.000	0.996	1.000	124
17	3167	1	1.000	1.000	1.000	1.000	0

Table 5.6: Adaptive questioning procedure analysis on a subset composed of 20 alternatives taken at random from the EPI data set

The two first columns of the table contain the iteration number and the size of the set of admissible points (at the beginning of this iteration). The next column contains the total number of rankings that are induced by all these points. Then, the four next columns contain some characteristics of these rankings. They respectively contain the minimal, the maximal, the average and the median correlation coefficient between the rankings in Γ_i and the PROMETHEE II ranking. The last column contains the number of points that are removed due to the addition of the new constraint.

This data set is the one where the smallest number of iterations was needed in order to obtain a set of points S containing only points reproducing the PROMETHEE II ranking. Still, 16 questions had to be asked to arrive at this situation. This quantity could seem to be pretty high since A is composed of only 20 alternatives.

More details about the evolution of the admissible correlation coefficients can be seen in the boxplot representation in Figure 5.3.

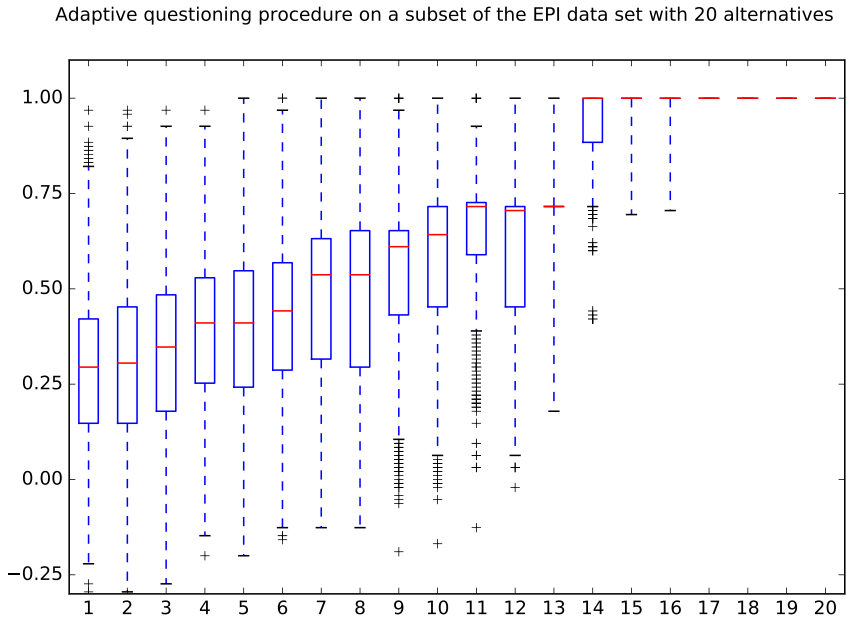


Figure 5.3: Kendall's correlation coefficient between the PROMETHEE II ranking and the admissible rankings during an application of the adaptive questioning procedure on a set of 20 alternatives taken at random from the EPI data set

It can be seen that until the 13th iteration of the procedure the worst correlation coefficient was still about 0.18. Until this iteration, the median, average and minimal correlation coefficient observe a slow but continuous growth. Then, between the 13th and 14th iteration, a sudden growth occurs.

The second set of alternatives on which the results of the adaptive questioning procedure are illustrated is composed of 20 other alternatives taken again at random from the EPI data set.

The results of this application are shown in Table 5.7 and in the boxplot represented in Figure 5.4.

i	$\mid S_i \mid$	$\mid \Gamma_i \mid$	τ_{min}	τ_{max}	τ_{μ}	τ_{med}	points removed
5	3235	851	-0.052	1.000	0.616	0.610	1544
8	3000	272	0.126	1.000	0.791	0.810	1363
12	2248	112	-0.031	1.000	0.878	0.989	376
13	3061	73	0.178	1.000	0.886	1.000	483
16	2438	21	0.452	1.000	0.956	1.000	171
20	2465	1	1.000	1.000	1.000	1.000	

Table 5.7: Adaptive questioning procedure analysis on a subset composed of 20 alternatives taken at random from the EPI data set

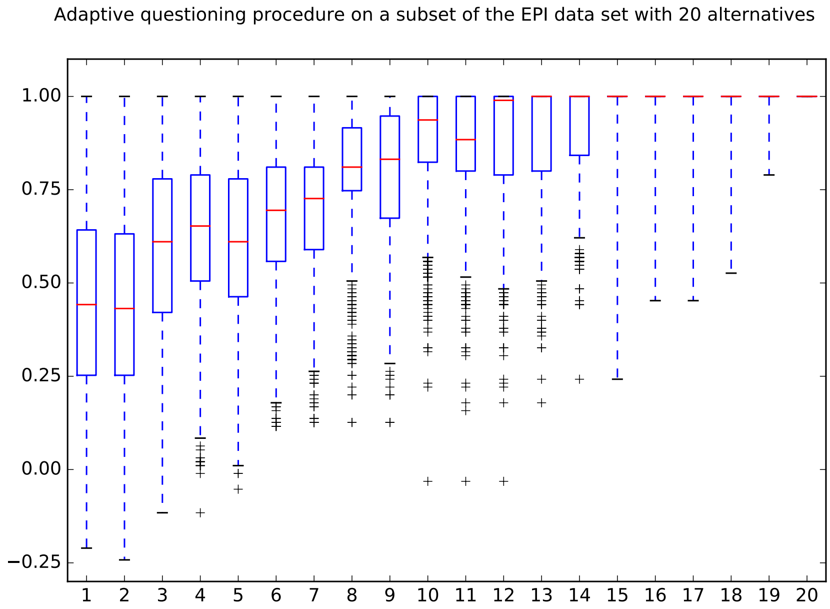


Figure 5.4: Kendall's correlation coefficient between the PROMETHEE II ranking and the admissible rankings during an application of the adaptive questioning procedure on a set of 20 alternatives taken at random from the EPI data set

For this set of alternatives, a larger number of queries had to be asked in order to reduce the set to only points reproducing the PROMETHEE II ranking, and the worst admissible correlation tau stays very low until the 18th query. On the other hand, the median and average correlation coefficient grow faster.

The last set of alternatives on which the procedure is illustrated is a set of 30 alternatives taken at random from the GEQ data set.

Again, the results of this application can be seen in the table and in the boxplot hereunder.

i	$\mid S_i \mid$	$\mid \Gamma_i \mid$	τ_{min}	τ_{max}	τ_{μ}	τ_{med}	points removed
5	3010	2752	-0.195	0.783	0.291	0.296	1451
10	2495	1935	-0.043	0.797	0.357	0.351	1159
15	2019	1146	0.080	0.857	0.415	0.402	947
20	2095	1042	0.075	0.935	0.456	0.420	1056
25	2204	131	0.425	0.848	0.672	0.668	985
30	931	79	0.356	0.935	0.728	0.728	

Table 5.8: Adaptive questioning procedure analysis on a subset composed of 30 alternatives taken at random from the GEQ data set

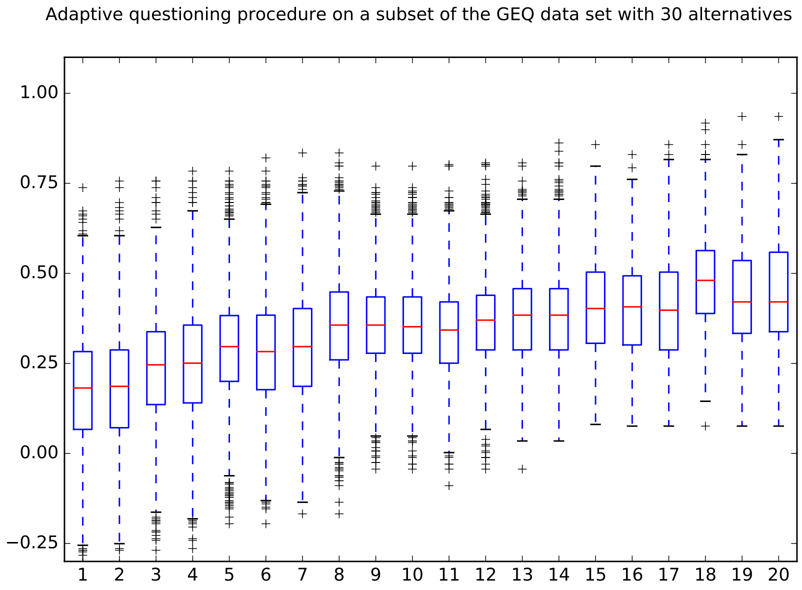


Figure 5.5: Kendall's correlation coefficient between the PROMETHEE II ranking and the admissible rankings during an application of the adaptive questioning procedure on a set of 30 alternatives taken at random from the GEQ data set

This set of alternatives is one of the set of alternatives for which the genetic algorithm presented in section 5.1.2 could not find any set composed of 4 reference profiles reproducing the PROMETHEE II ranking. It is therefore not surprising that the questioning procedure does not succeed at finding one neither.

The correlation coefficients for this application of the procedure do not grow very much. As shown in the Figure 5.5, the median correlation coefficient does nearly not grows during the 20 first iterations. During the 10 next iteration however, the average and median coefficients grow faster but stay under 0.73. It can also be noted that at the beginning of the 30th iteration, the set of admissible points was only composed of 900 points. This is due to the difficulty of the procedure to find new points when some constraints have already been imposed.

The results of the adaptive query procedure applied on these three different data sets are not encouraging. In the best case, about 10 questions were needed to obtain a set of points whose "average" ranking is similar enough to the PROMETHEE II ranking (when the method is applied on a set of 20 alternatives). In the worst case, it can happen that this procedure does not converge to a set of points which produce only rankings similar to the PROMETHEE II method. We can therefore conclude that the procedure is failing in some cases.

However, this procedure has been useful as it produced a lot of points (which, it should not be forgotten, represent sets of reference profiles) which lead to REFERENCED PROMETHEE rankings equal to the PROMETHEE II ones. These sets of reference profiles will briefly be analysed in the next section in order to try to find some possible characterisations of these sets. Such characterisations could be helpful to increase the understanding of the relations between sets of alternatives and the efficiency of strategies used to build set of reference profiles, or eventually to try to imagine new strategies.

5.2.3 Analysis of the satisfactory sets of reference profiles found

Different sets of reference profiles producing a REFERENCED PROMETHEE ranking identical to the PROMETHEE II one were found. These sets of reference profiles have been analysed in order to try to find some similarities which could lead to some indications about how to build such sets of reference profiles.

For instance, the table hereunder shows the expectation of the ratio between the variance of the reference profiles over the variance of the set of alternatives for three different decision problems with alternatives taken from the SHANGHAI data set:

$$E_c = E[\frac{D^2(f_c(r))}{D^2(f_c(a_i))}] \qquad r \in \mathscr{R}, a_i \in A \tag{5.7}$$

A_i	E_1	E_2	E_3	E_4	E_5	E_6
A_1	4.0	4.3	4.7	4.3	2.0	2.5
A_2	5.9	2.9	7.2	7.2	4.7	2.5
A_3	4.1	4.2	2.2	1.7	2.5	3.9

Table 5.9: Expectations of the ratio between the variance of the set of reference profiles and the set of alternatives for sets of reference profiles which reproduce the PROMETHEE II ranking exactly.

We can deduce from this table that the variance of the evaluations of the sets of reference profiles should be higher than the one of the sets of alternatives. However, of how much this variance must be higher is not clear. Indeed, these variances vary significantly from one criterion to another and from one set of alternatives to another. The same observations have been done for sets of alternatives taken from the EPI and GEQ data sets.

Some other estimators have been studied such as, for instance, the expectation of the ratio between the mean of the evaluations of the reference profiles and the mean of the evaluations of the alternatives. These studies did not lead to any interesting property and are therefore not shown.

Different sets of reference profiles reproducing the PROMETHEE II ranking for a same set of alternatives (with the same decision maker) can be represented in a parallel coordinates plot.

A typical parallel coordinate boxplot obtained by plotting two sets of reference profiles is shown in Figure 5.6.

Once again, no particular characteristic has been found during the study of these parallel coordinates plots. The profiles taken from different sets do not seem to have any similarities.

5.3 Conclusions and synthesis of the REFERENCED PROMETHEE method

REFERENCED PROMETHEE does not suffer from the rank reversal phenomenon. However, it produces rankings strongly dependent on the set of reference profiles used.

It has been seen that there generally exist some sets of reference profiles which lead to the same or highly similar rankings as PROMETHEE II, which are considered to be satisfying rankings.

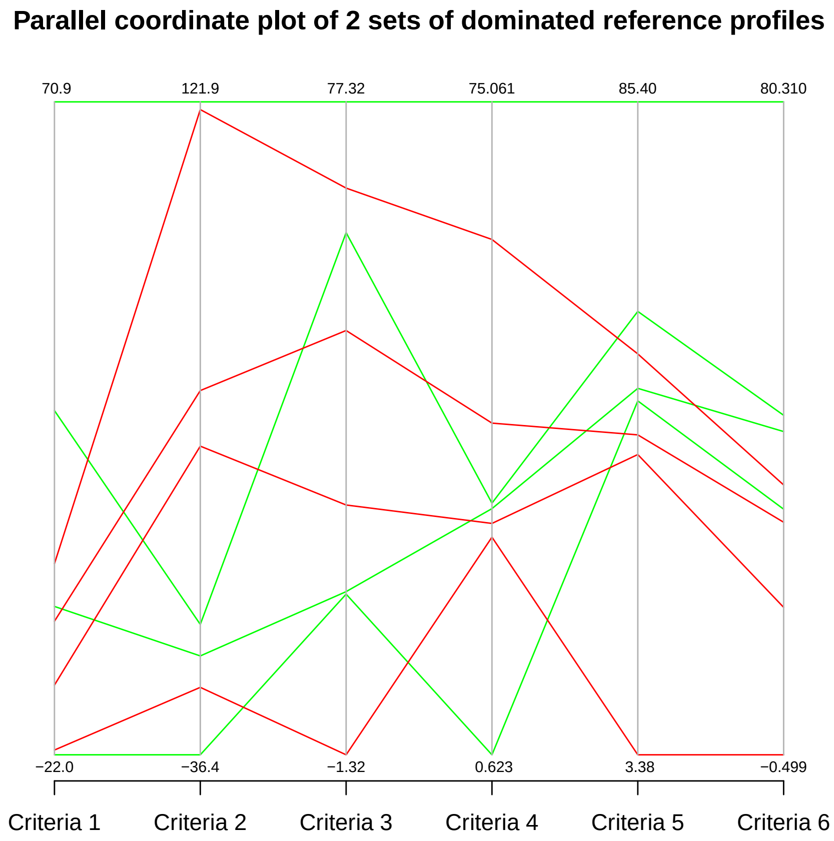


Figure 5.6: Parallel coordinate plot of two sets of reference profiles reproducing the PROMETHEE II ranking for a subset of alternatives taken at random from the SHANGHAI data set.

However, no method was found that was able to build such sets for any decision problems (without using the corresponding PROMETHEE II ranking). The procedures based on the strategies and on the adaptive questioning procedure (sections 5.2.1 and 5.2.2) both produce rankings which similarities with the PROMETHEE II ones strongly depend on the set of alternatives and on the decision maker. These rankings are in some cases not similar enough to the PROMETHEE II ranking to be considered as satisfying.

Chapter 6

Conclusion

As it has been seen in the first chapter, not all decision problems can efficiently be solved with a purely qualitative approach. When the decision problem reality is too complex, a mathematical model should be used. The discipline of helping the decision maker with his decision problem is called operational research or decision aid.

In the large majority of practical decision aid applications, a model based on the optimisation of a single criterion is used. This is called the monocriterion approach. Unfortunately it has some shortcomings. In particular, it has been seen that this approach does not correctly model how the human mind approaches decision problems when their contexts consist in conflicting aspects of different natures. Another approach, the multi-criteria approach is then more adapted.

Some multi-criteria decision aid methods have been analysed in this work. For instance, the PROMETHEE methods have been studied in chapter 2.

When facing multi-criteria decision problems, one could be confronted to the rank reversal phenomenon. At first glance, one could reject methods suffering from this phenomenon as it seems counter-intuitive and irrational. However, it has been seen, in the dedicated chapter (chapter 3), that this phenomenon is more complex and it's legitimacy is far from being trivial.

Nevertheless, two approaches have been proposed to deal with this phenomenon within the framework of the PROMETHEE II method.

The first one, ROBUST PROMETHEE, has been analysed in chapter 4. It works by applying PROMETHEE II on numerous random samplings. Until now, it seems to be reducing the quantity of rank reversals with regard to PROMETHEE II while keeping a similar ranking (which therefore can be considered as satisfactory). However, its efficiency is strongly dependent on the quantity of alternatives considered and on the size of the subsets on which PROMETHEE II is applied. Due to the random nature of this method, it also has the unfortunate characteristic of not guaranteeing that the ranking produced will respect the natural dominance relation. This characteristic should however be mitigated since no violations have until now been observed.

Also due to the random nature of the sampling selection, it could happen that two alternatives would never be part simultaneously of one of the subsets on which PROMETHEE II is applied. This would lead the ROBUST PROMETHEE method to produce some unexpected and undesirable results. An upper bound on the probability that this phenomen happen has been found. This bound depends on the number of repetitions of PROMETHEE II on the random subsets and on the size of these subsets. However, it has been seen that this bound should be used with caution as the method often need each pair of alternatives to be part of the same subset more than once to perform optimally.

A lot of analysis could still be done on this method:

- The rankings produced are similar, but not an exact reproduction of the ones produced by PROMETHEE II. Are these differences relevant? Are they leading to a ranking which is more in adequation with the decision makers preferences?
- All the analysis has been performed by using random subsets of constant size (as proposed by the method). ROBUST PROMETHEE could be adapted and used with subsets of various sizes or even with subsets which are not built at random.

The second method analysed in this work is called REFERENCED PROMETHEE. It is based on the application of PROMETHEE II with a set of fixed reference profiles.

The first characteristic observed for this method is that it is not necessary to use a large number of reference profiles to discriminate large sets of alternatives. For this reason, the majority of the tests have been performed using sets of 4 reference profiles. Then, it has been observed that it was generally possible for small sets of alternatives to reproduce the rankings obtained with PROMETHEE II, by using an appropriate set of profiles. In our case, these appropriate sets of reference profiles have been found using a genetic algorithm.

Knowing this, some procedures have been developed in order to build sets of reference profiles leading to rankings as close as possible to the ones produced by PROMETHEE II. The first one is based on the development of strategies (section 5.2.1) and did not show any encouraging results.

The second one, more elaborated, consists in an iterative questioning procedure. The results of the experimentations of this procedure were not very satisfying neither. Indeed, it is not always yielding sets of reference profiles reproducing the PROMETHEE II ranking, and when it does, it is only after that the decision maker has answered a consequent amount of questions.

As for the first method, different research question are still to be investigated. First of all, the influence of the size of the sets of reference profiles should be analysed. Indeed, only few references being sufficient to discriminate large sets of alternatives, the procedures tested in this chapter were gerenally used with 4 reference profiles. Moreover, the procedures were tested on random sets of alternatives, with preference functions with random ceils and random weight factors. It could be interesting to study the effect of the modifications of individual parameters on their efficiency to find satisfactory sets of reference profiles.

Appendices

Appendix A

ROBUST PROMETHEE algorithm

Here under, an algorithm is proposed to compute the robust flow of a set of alternatives:

Algorithm 3 ROBUST PROMETHEE algorithm

```
Input: $A, \mathscr{P}, w, R, m$
1: $P = n \times n$ matrix of zeros
2: for $r = 1 \ldots R$ do
3:     $A_{mr}$ = random sample of $m$ alternatives from $A$
4:     $Ranking_r$ = PROMETHEE_II($A_{mr}, w, \mathscr{P}$)
5:     $P$ = update($P, Ranking_r$)
6: end for
7: $P$ = normalise($P$)
8: $\Phi$ = netflow($P$)
9: return ranking($\Phi$)
```

The following parts of the algorithm shoud require some additional commnents:

Input $\mathscr{P}$: the vector of preference functions. These preference functions must come with the values of their corresponding parameters. For example, a preference function of type V should be given with its values of p and q (see section 2.1).

line 5, update: returns an updated probability matrix according to ranking given. Here under, one can see an example of how a given probability matrix is updated if the function is given the following ranking : $[a_\alpha \succ a_\gamma \succ a_\beta]$.

$$P = \begin{array}{cc} & \begin{array}{ccccc} & \beta & & \gamma & \end{array} \\ \begin{array}{c} \\ \alpha \\ \\ \gamma \\ \\ \end{array} & \begin{pmatrix} . & . & . & . & . \\ . & x & . & y & . \\ . & . & . & . & . \\ . & z & . & . & . \\ . & . & . & . & . \end{pmatrix} \end{array} \rightarrow \begin{array}{cc} & \begin{array}{ccccc} & \beta & & \gamma & \end{array} \\ \begin{array}{c} \\ \alpha \\ \\ \gamma \\ \\ \end{array} & \begin{pmatrix} . & . & . & . & . \\ . & x+1 & . & y+1 & . \\ . & . & . & . & . \\ . & z+1 & . & . & . \\ . & . & . & . & . \end{pmatrix} \end{array}$$

At this stage, the elements p_{ij} of the matrix contain the total number of times that the alternative a_i has been ranked before the alternative a_j in one of the past iterations.

line 7, normalised: returns a normalised probability matrix. To normalise this matrix, all the elements p_{ij} are divided by the sum of themself and of the conjugate elements of the matrix: p_{ji}. An example is shown here under :

$$P = \begin{array}{cc} & \begin{array}{ccccc} & i & & j & \end{array} \\ \begin{array}{c} \\ i \\ \\ j \\ \\ \end{array} & \begin{pmatrix} . & . & . & . & . \\ . & . & . & y & . \\ . & . & . & . & . \\ . & x & . & . & . \\ . & . & . & . & . \end{pmatrix} \end{array} \rightarrow \begin{array}{cc} & \begin{array}{ccccc} & i & & j & \end{array} \\ \begin{array}{c} \\ i \\ \\ j \\ \\ \end{array} & \begin{pmatrix} . & . & . & . & . \\ . & . & . & \frac{y}{x+y} & . \\ . & . & . & . & . \\ . & \frac{x}{x+y} & . & . & . \\ . & . & . & . & . \end{pmatrix} \end{array}$$

This operation is done (after the R iterations of the PROMETHEE II method) to ensure that all p_{ij} are included between zero and one and represent a probability.

Appendix B

Expectation and variance in the coupon collector's problem

In the context of the coupon collector's problem, the variable d_i is a random variable denoting the number of random balls that must be drawn from the urn in order to obser the ith ball once $(i-1)$ balls have already been observed.

The probability of finding a new ball at the next draw once $(i-1)$ of the N balls have already been found is constant and given by:

$$p_i = \frac{N-(i-1)}{N} \tag{B.1}$$

Consider now an experiment which two only possible outcomes are *succes* with probability p and *failure* with probability $(1-p)$. A random variable Y associated to this experiment which takes the value 1 if the experiment succeeds and 0 if it fails is said to be a bernoulli random variable [20].

Each draw, once $i-1$ balls have already been observed and until ith ball is observed, can be seen as an bernoulli experiment. Therefore, the random variable d_i can also be seen as the distribution of the number of bernoulli experiments of equal probabilities that must be performed until the first experiment succeeds [20].

This defines d_i as a geometric random variable of parameter p_i with the following probability distribution on $n = 1, 2...$ [20]:

$$Pr[d_i = n] = (1-p_i)^{n-1} p_i \tag{B.2}$$

The expectation and the variance of the geometric random variable d_i is given by the following relations [20]:

$$E[d_i] = \frac{1}{p_i} \qquad Var[d_i] = \frac{1 - p_i}{p_i^2} \tag{B.3}$$

Since D is defined as the random variable representing the total number of draws needed, it is equal to the sum of the d_i.

$$D = \sum_i d_i \tag{B.4}$$

Its expectation and its variance can therefore be computed as follow [20]:

$$\begin{aligned} E(D) &= \sum_{i=1}^{N} E(d_i) \\ &= \sum_{i=1}^{N} \frac{1}{p_i} \\ &= \frac{N}{N} + \frac{N}{N-1} + \cdots + \frac{N}{1} \\ &= N \cdot H_N \end{aligned} \tag{B.5}$$

With H_N being the Nth harmonic number

$$\begin{aligned} Var(D) &= \sum_{i=1}^{N} Var(d_i) \\ &= \sum_{i=1}^{N} \frac{1 - p_i}{p_i^2} \\ &\leq \sum_{i=1}^{N} \frac{1}{p_i^2} \\ &= \sum_{i=1}^{N} \left(\frac{N}{N - i + 1} \right)^2 \\ &= N^2 \sum_{i=1}^{N} \left(\frac{1}{i} \right)^2 \\ &\leq \frac{\pi^2 N^2}{6} \end{aligned} \tag{B.6}$$

With $\frac{\pi^2}{6}$ being the result of the Riemann ζ function evaluated in 2.

Bibliography

[1] Academic ranking of world universities. Available: `http://www.shanghairanking.com/ARWU2016.html`, last consulted: 24-05-2017.

[2] Human development reports. Available: `http://hdr.undp.org/`, last consulted: 24-05-2017.

[3] K.J. Arrow. A difficulty in the concept of social welfare. *Journal of political economy*, 58(4):328–346, 1950.

[4] C. Barrico and C.H. Antunes. Robustness analysis in evolutionary multi-objective optimization, 2006.

[5] M. Behzadian, R.B. Kazemzadeh, A. Albadvi, and M. Aghdasi. Promethee: A comprehensive literature review on methodologies and applications. *European Journal of Operational Research*, 200(1):198 – 215, 2010.

[6] V. Belton and T. Gear. On a short-coming of saaty's method of analytic hierarchies. *Omega*, 11(3):228–230, 1983.

[7] J.P. Brans and B. Mareschal. The promcalc & gaia decision support system for multicriteria decision aid. *Decision support systems*, 12(4-5):297–310, 1994.

[8] J.P. Brans and Y. De Smet. *PROMETHEE Methods*, pages 187–219. Springer New York, New York, NY, 2016.

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

[10] G. Dejaegere. Promethee. `https://github.com/gdejaege/Promethee`, last consulted: 31-05-2017, 2017.

[11] N.A.V. Doan and Y. De Smet. On the use of reference profiles to compute alternative promethee ii rankings: A preliminary study. In *Industrial Engineering and Engineering Management (IEEM), 2016 IEEE International Conference on*, pages 326–330. IEEE, 2016.

[12] S. Eppe and Y. De Smet. An adaptive questioning procedure for eliciting promethee ii's weight parameters. *International Journal of Multicriteria Decision Making*, 4(1):1–30, 2014.

[13] R. Hites, Y. De Smet, N. Risse, M. Salazar-Neumann, and P. Vincke. About the applicability of {MCDA} to some robustness problems. *European Journal of Operational Research*, 174(1):322 – 332, 2006.

[14] A. Hsu and al. 2016 environmental performance index. New Haven, CT: Yale University. Available: `www.epi.yale.edu`, last consulted: 24-05-2017.

[15] V.S. Iyengar, J. Lee, and M. Campbell. Evaluating multiple attribute items using queries. In *Proceedings of the 3rd ACM conference on Electronic Commerce*, pages 144–153. ACM, 2001.

[16] W. De Keyser and P. Peeters. A note on the use of promethee multicriteria methods. *European Journal of Operational Research*, 89(3):457–461, 1996.

[17] R.D. Luce. Semiorders and a theory of utility discrimination. *Econometrica*, 24(2):178–191, 04 1956. Date revised - 2013-06-12; Last updated - 2013-09-16; SubjectsTermNotLitGenreText - 3889 6071 1542 11325; 5403 8010 4025; 13221; 13521.

[18] B. Mareschal and J.P. Brans. *Prométhée-Gaia: une méthodologie d'aide à la décision en présence de critères multiples.* Number 2013/9303. Editions Ellipses, December 2002.

[19] B. Mareschal, Y. De Smet, and Ph. Nemery. Rank reversal in the promethee ii method: some new results. In *2008 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 959–963. IEEE, 2008.

[20] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, New York, NY, USA, 2005.

[21] B. Roy. Classement et choix en présence de points de vue multiples. *RAIRO - Operations Research - Recherche Opérationnelle*, 2(V1):57–75, 1968.

[22] B. Roy. *Méthodologie multicritère d'aide à la décision.* Economica, Paris, 1985.

[23] B. Roy. Decision-aid and decision-making. *European Journal of Operational Research*, 45(2):324 – 331, 1990.

[24] B. Roy. Robustness in operational research and decision aiding: A multifaceted issue. *European Journal of Operational Research*, 200(3):629 – 638, 2010.

[25] Y. De Smet. About the computation of robust promethee ii rankings: Empirical evidence. In *Industrial Engineering and Engineering Management (IEEM), 2016 IEEE International Conference on*, pages 1116–1120. IEEE, 2016.

[26] Y. De Smet. *MATH-H-405 Decision Engineering.* 2016.

[27] Y. De Smet and K. Lidouh. *An Introduction to Multicriteria Decision Aid: The PROMETHEE and GAIA Methods*, pages 150–176. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[28] A. Tversky. 19 intransitivity of preferences. *Preference, Belief, and Similarity*, page 433, 1969.

[29] C. Verly and Y. De Smet. Some results about rank reversal instances in the promethee methods. *International Journal of Multicriteria Decision Making 71*, 3(4):325–345, 2013.

[30] Ph. Vincke. *Multicriteria Decision-Aid.* J. Wiley, New York, 1992.

[31] Y.M. Wang and Y. Luo. On rank reversal in decision analysis. *Mathematical and Computer Modelling*, 49(5):1221–1229, 2009.

[32] D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4(2):65–85, 1994.