Omega 136 (2025) 103325

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

Inverse optimization on the evaluations of alternatives in the PROMETHEE II ranking method☆

Alexandre Flachs*, Yves De Smet

Université libre de Bruxelles, Ecole polytechnique de Bruxelles, Computer and Decision Engineering department - SMG research unit, Avenue Franklin D. Roosevelt 50, 1050, Bruxelles, Belgium

ARTICLE INFO

Keywords:
Multi-criteria decision aid
PROMETHEE II
Inverse optimization
Ranking

ABSTRACT

In multi-criteria decision aid, one usually distinguishes three main families of methods: aggregating, outranking and interactive approaches. Among the outranking methods, PROMETHEE II allows decision makers to obtain a complete ranking of the alternatives regarding multiple – usually conflicting – criteria. Despite its popularity and the various theoretical contributions that have been made to it, few work has been focused on the influence of the alternatives' evaluations on the final ranking.

In this contribution, this issue is addressed by proposing a solution to the following question: given a ranking of the alternatives, what are the minimal modifications required on the evaluations of a given alternative in order to improve its position? A novel exact algorithmic procedure is proposed to answer this question even when multiple criteria are involved. In such cases the solution takes the form of a Pareto optimal front. A proof of the correctness of the algorithm and a complexity analysis are provided. Finally, experimental tests of the method on a real life dataset is discussed along with some indications on how it could be presented to decision makers.

1. Introduction

Inverse optimization is the process of, given a solution to an optimization problem, finding the objective function and/or the constraints that would have led to this solution. In that sense it is dual to the classical optimization problem [1]. In this work, an inverse optimization method to the classical problem of ranking a set of alternatives based on multiple criteria is studied. More formally, given two alternatives a_i, a_j, the question of the minimal deviation of a_i's evaluations to be ranked better than a_j is studied.

For example, let us consider a company that wants to recruit new employees. Each candidate could be evaluated based on multiple criteria such as their experience, education, their salary expectation, etc. Multi-Criteria Decision Making (MCDM) methods can be used to rank them based on these criteria and give insights on which one is the best fit for the company. In the context of inverse optimization, the dual question is asked: given a ranking of the candidates, how could one of them change his/her profile (for example his/her expected salary) to reach a better position in the ranking? Or even on a broader perspective: what combination of criteria could he/she try to change to reach this goal ?

In this work, it is important to make a clear distinction between the person, the group, the company, etc… in charge of computing the ranking, and the actors who represent the alternatives. Of course, the first ones are interested in quantifying the robustness of the ranking with respect to preference parameters. For the latter however, these parameters are imposed and their only way to improve their position in the ranking is through their evaluations. The authors argue that this is an important question to study in MCDM. Alternatives can be considered actors in various types of decision problems. The above example is one among many, any call for tender, recruitment problem, or any decision problem where decision makers (DM) are to be compared by an external actor can be subject to such study. Despite the growing importance of group and large scale decision making, the DM is supposed to be a single person in the context of this study and the focus is on the inverse optimization problem mentioned above.

The rest of this article is structured as follows: a brief review of the literature is presented in Section 2. Then, a reminder about PROMETHEE II is given in Section 3. It is followed by Section 4 in which the study of the problem of optimizing the evaluation of one alternative regarding one criterion is developed. The generalization to multiple criteria is

☆ Area: Decision Analysis and Preference Modeling. This manuscript was processed by Associate Editor S. Corrente.

* Corresponding author.
E-mail addresses: alexandre.flachs@ulb.be (A. Flachs), yves.de.smet@ulb.be (Y. De Smet).

https://doi.org/10.1016/j.omega.2025.103325
Received 15 July 2024; Accepted 18 March 2025
Available online 31 March 2025

presented in Section 5. Finally, experimental tests of the proposed method is discussed in Section 6.

2. Literature review

Since the 1950s, operations research has known increasing interest in the field of decision modeling. As most strategic decision problems involve multiple conflicting criteria, this interest naturally extended to multiple-criteria decision making (MCDM) to assist decision makers (DM) in such situations [2–4]. A plethora of MCDM methods have since been proposed to solve different type of multiple-criteria problems, each with its own advantages and drawbacks. Several taxonomies of these methods have been proposed to classify them, see for example [5, 6]. Vincke proposed to classify them into three main categories based on how they model the DM's preferences: aggregating, interactive and outranking methods [7].

Aggregating methods assign some marginal utility value to each alternative regarding each criterion, and then combine these by using a utility function which depends on the method, leading to a global utility value for each alternative. They include among many others: Multi-Attribute Utility Theory (MAUT) [8], the influential Analytical Hierarchy Process (AHP) [9], or the weighted sum method [10] and advanced extensions such as Choquet integrals [11].

Interactive methods usually iterate between a computation phase based on some knowledge on the decision problem, and a dialogue (interaction) phase with the DM. Though some exist for discrete decision problems (*i.e.* problems where the alternative set is discrete) [12], their popularity seem to have decreased in the recent years compared to other methods. They have however been particularly developed and used for continuous problems, often referred to as multi-objective optimization problems. Nimbus [13] and Nautilus [14] for example are well known methods in that field.

Outranking methods differ in various aspects but all follow one important rule: the concordance–discordance principle. It recommends that an alternative *outranks* another if there are sufficient arguments in favor of the first and not enough against it [15], and that the DM should be able to specify these arguments in some way. Some well known families of outranking methods are Electre [15] and Promethee [7].

Promethee I and II were first proposed in 1982 by J.P. Brans as procedures to define a ranking over a set of alternatives. They have been widely used in practice and have been the subject of numerous theoretical studies and methodological extensions. Behzadian et al. [16] reference many applications of Promethee in various practical fields such as finance, environmental management, health care, etc... Among the reasons for this popularity, one could cite the GAIA visualization tool [17], relying on Principal Components Analysis to plot in two dimensions the unicriterion net flow scores (resulting from the computations of the methods), alongside with other information. Moreover, the development of user-friendly software such as PromCalc [18], Decision Lab 2000 [19], Visual Promethee [20], or more recently D-Sight [21] helps DM to use the methods without having to implement them themselves. Some parts of the Promethee II procedure have been axiomatically characterized by Bouyssou first [22], then Marchant [23, 24], and more recently by Dejaegere [25], each providing different insight regarding the method's structure; and several contributions were interested in the problem of rank reversal [see for example 26–29]. Among other extensions, Promethee has been adapted to multi-criteria sorting [30] and clustering [31,32], to the Stochastic Multiobjective Acceptability Analysis (SMAA) framework [33]. More recently the Promethee γ method [34] has been proposed to reduce inconsistencies related to the incomparability relations produced when using Promethee I.

Inverse optimization on MCDM ranking methods is not a new field of research. The most studied problems are related to sensitivity analysis. Their goal is to quantify the robustness of the ranking with respect to some parameters. As most parameters and sometimes performance evaluations involve some subjectivity at some point, sensitivity is considered to be a crucial step in the decision making process [35].

Sensitivity analysis and inverse optimization are directly related. The first one studies how much some parameter can change while preserving some properties of the ranking, the latter studies the minimal deviation on these parameters to modify the ranking. It is expected that a method that can solve a sensitivity analysis problem can be adapted to solve related inverse optimization problems without too many modifications. This is for example the case for the method presented in this work. For this reason, sensitivity analysis ideas and techniques are introduced first, and followed by inverse optimization contributions.

Sensitivity or robustness analysis can take many forms. It is most often used to study how much imprecision, uncertainty or subjectivity in the data can affect the results of a decision aid method. Given a decision problem and an adapted method, one of the most straightforward ways to perform sensitivity analysis is to study the maximal change allowed on one or multiple parameters while preserving the results obtained. In particular for Promethee methods, Mareschal [36] discussed the computation of stability intervals on a given weight to preserve a ranking obtained with Promethee II, while keeping the ratios of other weights unchanged. De Smet and Doan [37] later solved the more generic problem of computing stability intervals on all the weights used in Promethee II. Both these contributions rely on linear programming. More recently, Liu & Liu [28] studied stability intervals on intra-criteria parameters (used in Promethee II to model preferences for each criterion). As the relation between these parameters and the final ranking is more complex, they developed a dedicated algorithm.

On another note, Wolters and Mareschal [38] introduced three different type of sensitivity analyses one could perform, respectively:

1. The sensitivity of a ranking to changes in the performance values of all alternatives on some criteria. This problem is studied on a real-life example by iterations and dependence between criteria is taken into account, but the authors do not provide a general solution to this problem.
2. The "influence" of a change in the performance value of an alternative on the ranking. Similar to the previous problem, the authors do not provide a general solution to this problem.
3. The minimal modification of the weights required to allow a given alternative to be ranked first. In this case, the modification is minimal in the sense of the L_1 norm and the optimal solution can be computed by the mean of a linear program in the case of the Promethee II method.

These questions are relevant to many MCDM methods.

In all these contributions, DM or analysts are expected to perform the sensitivity analysis *after* the MCDM analysis has been made. Some contribution chose another path and propose either to integrate the sensitivity analysis in the MCDM method itself, *e.g.* Stochastic Multiobjective Acceptability Analysis (SMAA) [39]; or to mitigate the effect of the uncertainty on the results, *e.g.* Robust Ordinal Regression (ROR) [40]. It combines a disaggregation–aggregation approach (explained below) with indirect preference information from the DM, which is considered by psychologists clearer and less cumbersome to assess by the DM than direct information.

SMAA is a family of methods extensions developed to handle decision problems in which uncertainty is so important that it has to be taken into account from the beginning of the decision making process. It can be thought of as an adaptation of Monte Carlo simulation to MCDM methods, where parameters are drawn from a distribution representing the uncertainty. In the case of ranking problems, after a large number of simulations, the method derives one rank acceptability index b_i^r for each alternative a_i at each rank r. It translate how probable it is that an alternative a_i obtains rank r in the final ranking from the available information and the explored space. The method has in particular been applied to Promethee methods [33]. It is primarily used as an exploration tool to help DM understand if the uncertainty has much influence on the final ranking, and if it has, where it comes from.

Similar to SMAA, ROR is a method that aims at mitigating the effect of uncertainty on the final ranking [40]. It follows a disaggregation–aggregation approach. In the disaggregation phase, the DM is asked to provide some preference information on a reference subset of the alternatives. *Ordinal regression* is then used to derive valid parameters from this information. The aggregation phase then uses these parameters in a well known method to derive the final ranking. Multiple valid instance of the parameters set can be derived from the same preference information. The specificity of *robust* ordinal regression is to consider all of them in the final ranking. To summarize this information into recommendations for the DM, ROR computes *necessary* and *possible* preference relations. In the case of ranking problems, a necessary relation is a weak preference between any two alternative holding for all the instances of the parameters set, while the possible relations are the ones that are valid for at least one of them. These results can for example be used to compute the highest and lowest possible ranks of each alternative that would be compatible with the provided preference information. These two ranks can themselves be used to compute the stability intervals on the weights used in the method. For example, the choice problem consists of selecting a subset of the best alternatives. Kadziński et al. [41], who adapted ROR to the Promethee II method (under the name Promethee^{GKS}), highlighted that the highest and lowest ranks of an alternative could be used as bounds for the selection of this subset. For example, the subset could be composed of the alternatives that are in the top k for all the instances of the parameters set and are ranked first in at least one.

Most of the aforementioned contributions focus on the sensitivity of the ranking to changes in the preference parameters. Up to recently, very few contributions have studied the sensitivity of the ranking to changes in the performance evaluations of the alternatives. Kadziński et al. [42] proposed a framework for solving various problems of this kind. The authors called this framework “Post Factum Analysis”. It considers variations on both preference parameters and alternatives' performances with two dual questions in mind regarding performances: what is the minimal improvement required to change the ranking, and what is the maximal degradation allowed to preserve it. The first question relates directly to the inverse optimization problem studied in this paper.

Several contributions were focused on the problem of finding the optimal change of a single evaluation of the alternatives on a given criteria. In the context of the TOPSIS method Dutta et al. [43] investigated the impact of a change of one alternative's performance. More specifically, they studied the minimal change required to achieve a given rank, and determine the range of such possible ranks. The authors provided a closed form solution to their problem and an algorithm to identify when this solution is achievable for a given problem. They completed their analysis by providing a method to identify which criteria can be changed to reach the desired goal and, given a criterion to change, which alternatives can be outranked. The problems studied in their work are very similar to the ones presented in this paper, but the solutions relies heavily on the specificity of TOPSIS and cannot be adapted to Promethee methods. To the best of the authors knowledge the similar problem were multiple evaluations are allowed to change has not been studied in any MCDM ranking method.

Let us note that a similar problem has already been formulated in the context of multiple-criteria sorting: the Inverse Multi-Criteria Sorting Problem (IMCSP) [44]. Sorting can be informally described as the task of assigning items evaluated regarding multiple criteria to predefined ordered classes. The question is not “which alternative is better?” but “what class correspond the best to each alternative?”. In this context, the focus of the IMCSP is the minimal change required on an alternative's evaluations for it to be assigned to a different class. To the best of the authors knowledge, the IMCSP has not been studied for Promethee-based sorting methods. Solutions to the IMCSP developed for other methods usually take the form of a linear program.

Other inverse optimization methods have been studied in the literature. In the field of machine learning (ML) for example, prediction or classification models often rely on the optimization of a loss function which is so complex that it is considered as a black box. This behavior is the root of many critics towards ML models, as they are often considered too difficult to interpret [45]. Explainability of a model can be studied using counterfactual explanations: what would have been the optimal input to reach a different output? A detailed review of this field can be found in [46]. This problem differs from the one studied in this work as ML datasets used are usually much larger than the ones used in MCDM, which makes the problem more complex. Moreover, counterfactual explanations are either model agnostic – and thus approximate a solution – or model specific, in which case they are not adapted to most MCDM methods, such as Promethee II.

Finally, the reader should be aware that many of the existing methods in the literature that consider the problem of improving the ranking of alternatives focus on the implementation of improvement in complex environment, rather than mathematical requirements as it is the case in the proposed paper. One can for example consider the use of multi-criteria benchmarking [47] or long-term improvement planning [48]. Such methods are deemed interesting when an alternative plans to improve its performance over a long period time, usually by analyzing how and where higher ranked alternatives perform better than it does. This contribution's aim is to propose a new method to improve the rank of alternatives in short term regarding time (if possible), and to clarify the mathematical requirements of the problem to allow DM to interpret and use them in a complex environment.

In the context of the Promethee II methods, most contributions focused on the sensitivity of the ranking to changes in the preference parameters. First on weights by exploiting the linear structure of the problem, then on intra-criteria parameters by developing dedicated algorithms. To the best of the authors' knowledge, no formal work has been done on the question of the requirements to improve the ranking of alternatives by changing multiple performances of an alternative.

3. A reminder about Promethee II

Let $A = \{a_1, \dots, a_n\}$ be a set of n alternatives and $F = \{f_k : A \to \mathbb{R} | k = 1, \dots, q\}$ be a set of q criteria to be maximized (without loss of generality). Promethee II is a method to rank the alternatives of A based on these criteria. One can write

$$\begin{aligned} &a_i \succ a_j \Leftrightarrow a_i \text{ is preferred to } a_j \\ &a_i \sim a_j \Leftrightarrow a_i \text{ is indifferent to } a_j \\ &a_i \succsim a_j \Leftrightarrow a_i \text{ is preferred or indifferent to } a_j \end{aligned} \tag{1}$$

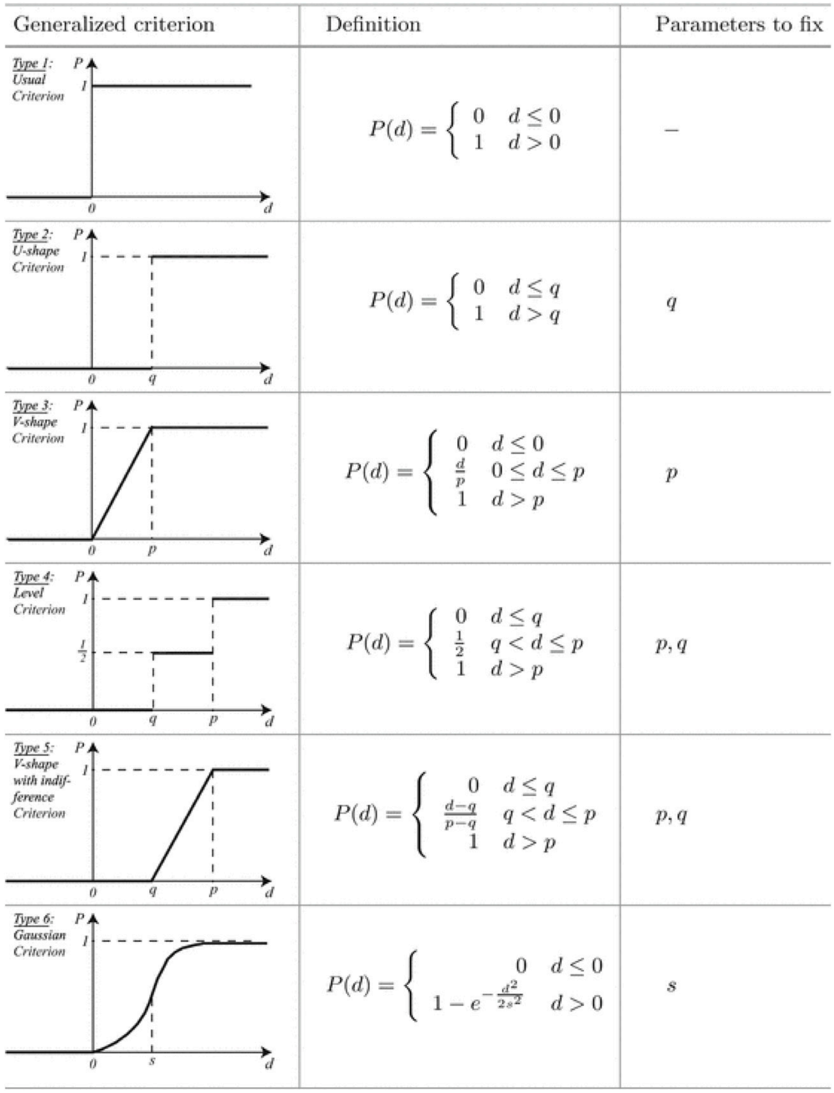
Promethee II addresses this problem based on pairwise comparisons between alternatives. The different steps of this method are summarized in this section. We refer the interested reader to [17] for a more complete description.

First, for each criterion independently, it requires to compute the difference of the evaluations f_k of two alternatives a_i and a_j:

$$d_{ij}^k = f_k(a_i) - f_k(a_j) \tag{2}$$

It then removes scaling effect from the criterion by normalizing this difference using a function called preference function $\mathcal{F}_k$. These are non-decreasing functions $\mathcal{F}_k : \mathbb{R} \to [0,1]$ such that $\mathcal{F}_k(x) = 0 \ \ \forall x \leq 0$ and $\lim_{x\to\infty} \mathcal{F}_k(x) = 1$. The index k is there to emphasize the fact that each original criterion can be associated to a different preference function. The pair between a criterion and a corresponding preference function is called a generalized criterion. In what follows, both names are used without distinction.

In theory, any preference function can be used as long as it respects the above conditions, but one usually considers a set of six generalized



Generalized criterion	Definition	Parameters to fix
Type 1: Usual Criterion	$P(d)=\begin{cases}0 & d\le 0\\ 1 & d>0\end{cases}$	–
Type 2: U-shape Criterion	$P(d)=\begin{cases}0 & d\le q\\ 1 & d>q\end{cases}$	q
Type 3: V-shape Criterion	$P(d)=\begin{cases}0 & d\le 0\\ \frac{d}{p} & 0\le d\le p\\ 1 & d>p\end{cases}$	p
Type 4: Level Criterion	$P(d)=\begin{cases}0 & d\le q\\ \frac{1}{2} & q<d\le p\\ 1 & d>p\end{cases}$	p, q
Type 5: V-shape with indifference Criterion	$P(d)=\begin{cases}0 & d\le q\\ \frac{d-q}{p-q} & q<d\le p\\ 1 & d>p\end{cases}$	p, q
Type 6: Gaussian Criterion	$P(d)=\begin{cases}0 & d\le 0\\ 1-e^{-\frac{d^2}{2s^2}} & d>0\end{cases}$	s

Fig. 1. Graphs of the six preference functions proposed for PROMETHEE methods along with their analytical definitions.
Source: Taken from [17].

criteria that are commonly used. Among them, the V-shape criterion (or linear criterion without indifference zone) requires one parameter called the preference threshold $p_k \in \mathbb{R}_0^+$ and is defined as follows.

$$\mathcal{F}_k^{(V)}(d_{ij}^k) = \begin{cases} 0 & \text{if } d_{ij}^k \le 0 \\ \frac{d_{ij}^k}{p_k} & \text{if } 0 < d_{ij}^k < p_k \\ 1 & \text{if } p_k \le d_{ij}^k \end{cases} \tag{3}$$

For the sake of clarity, one often writes $\pi_{ij}^k = \mathcal{F}_k^{(V)}\left(d_{ij}^k\right)$. Other preference functions are summarized in Fig. 1.

The normalized differences computed for all pairs of alternatives and over all criteria are then summed together to obtain the global preference between the two alternatives over all criteria.

$$\pi_{ij} = \sum_{k=1}^{q} w_k \cdot \pi_{ij}^k \tag{4}$$

The weights w_k are positive normalized parameters assumed to be given by the decision maker. The resulting value π_{ij} is thus between 0 and 1.

The *positive, negative* and *net flow scores* are obtained by summing the preferences of an alternative a_i over all the other alternatives as defined in Eqs. (5)–(7).

$$\phi_i^+ = \frac{1}{n-1}\sum_{j=1}^{n} \pi_{ij} \tag{5}$$

$$\phi_i^- = \frac{1}{n-1}\sum_{j=1}^{n} \pi_{ji} \tag{6}$$

$$\phi_i = \phi_i^+ - \phi_i^- = \frac{1}{n-1}\sum_{j=1}^{n} \left(\pi_{ij} - \pi_{ji}\right) \tag{7}$$

They represent respectively the average preference of a_i over all other alternatives, the average preference of all other alternatives over a_i, and the balance between the two.

PROMETHEE II uses the net flow score as an outranking function. The preference relation defined is a comparison between the net flows of two alternatives.

$$a_i \succ a_j \Leftrightarrow \phi_i > \phi_j \Leftrightarrow R_{ij} > 0 \tag{8}$$

$$a_i \sim a_j \Leftrightarrow \phi_i = \phi_j \Leftrightarrow R_{ij} = 0 \tag{9}$$

$$a_i \succsim a_j \Leftrightarrow \phi_i \ge \phi_j \Leftrightarrow R_{ij} \ge 0 \tag{10}$$

Where we defined $R_{ij} = \phi_i - \phi_j$, the net flow scores difference.

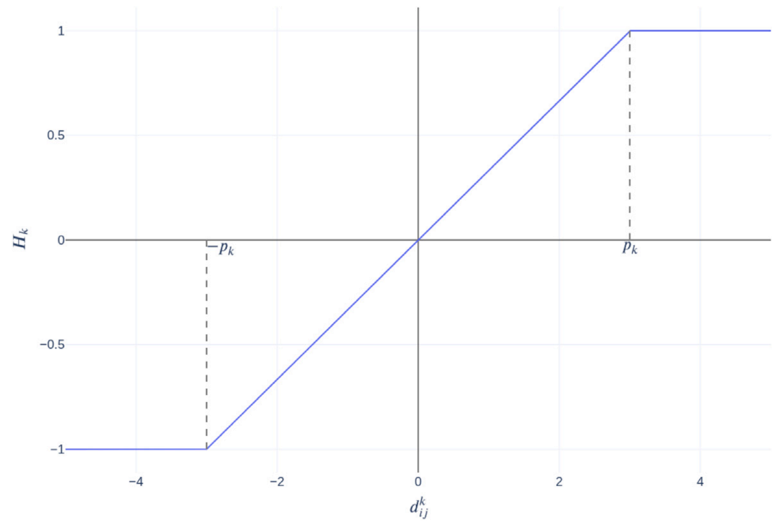


Fig. 2. Symmetrical V-Shape preference function with $p_k = 3$.

Let us note that the net flow score can be developed as in Eq. (11).

$$\phi_i = \frac{1}{n-1}\sum_{j=1}^{n}\left(\pi_{ij} - \pi_{ji}\right) = \frac{1}{n-1}\sum_{j=1}^{n}\sum_{k=1}^{q} w_k(\pi_{ij}^k - \pi_{ji}^k) = \sum_{k=1}^{q} w_k \cdot \phi_i^k \quad (11)$$

Where ϕ_i^k is the unicriterion net flow.

$$\phi_i^k = \frac{1}{n-1}\sum_{j=1}^{n}(\pi_{ij}^k - \pi_{ji}^k) = \frac{1}{n-1}\sum_{j=1}^{n} H_k(d_{ij}^k) \quad (12)$$

$H_k(d_{ij}^k) = \mathcal{F}_k(d_{ij}^k) - \mathcal{F}_k(d_{ji}^k)$ is the symmetrical preference function. An example in the case of the V-shape function is given in Fig. 2.

Let us note that H_k is an odd function and that it is monotone non-decreasing in d_{ij}^k, due to the monotony restrictions on $\mathcal{F}_k$. $H_k(d_{ij}^k)$ is thus a non-decreasing function of $f_k(a_i)$ and a non-increasing function of $f_k(a_j)$.

The focus of this work is the problem of minimizing the change required on the evaluations of one alternative on (possibly) multiple criteria to improve its position in a ranking. However, the nature of the relation between an evaluation $f_k(a_i)$ and the net flow score of a_i is mostly characterized by the generalized criterion $\mathcal{F}_k$, which is in general not linear, nor smooth or convex. In contrast to the problem of optimizing weights studied in aforementioned works [36,37] which exploits linearity, this non-convex behavior tends to indicate the absence of any general exact optimization method. Nonetheless, an exact algorithmic approach to solve the described problem is presented below for a family of preference functions.

4. Optimizing the evaluation of one alternative regarding one criterion

The first focus of this work is the problem below.

Problem 1 (*Minimal Change of One Evaluation*). Let us consider a ranking obtained with PROMETHEE II such that $a_i \succ a_1$ for some $i, 1 < i \leq n$. Let δ_1^1 be such that the evaluations of a_1 regarding the first criterion is changed from $f_1(a_1)$ to $f_1(a_1) + \delta_1^1$. What is the minimal value of δ_1^1 required to change the relative position between a_1 and another alternative a_i in the PROMETHEE II ranking?

The main idea behind the proposed approach relies on the fact that if preference functions are piecewise linear, they induce a piecewise linearity of the net flow score difference between two alternatives. Problem 1 is thus studied in the case of a V-Shape preference function as it is complex enough to show how to deal with difficulties. The results can easily be adapted for usual, U-shape (also called quasi-criterion), level, or V-shape with indifference regions preference functions. Those can be found in Appendix A in which every traditional preference function is covered.

The hypothesis that the evolving alternative is a_1 on the evaluation $f_1(a_1)$ is done without loss of generality. As the relative ranking between a_1 and a_i is given by the sign of the net flow difference, R_{1i} is first studied to show that it is piecewise linear in the evaluation of a_1. The explicit form of R_{1i} and the parts in which it is linear are then computed such that the equation $R_{1i} \geq 0$ can be solved in each of them, leading to a solution to Problem 1.

For the sake of clarity, the notation $\delta = \delta_1^1$ is used below. From now on, the evaluation of a_1 regarding the first criterion is noted $f_1(a_1)+\delta$. The net flow score with $\delta = 0$ is noted $\widetilde{\phi}$ and similarly, any other function evaluated with $\delta = 0$ is noted with a tilde on top of it. Let us write $R_{1i}(\delta) = \phi_1 - \phi_i$. It follows from Eq. (11) that the influence of δ on $R_{1i}(\delta)$ is only on the first term of the sum, hence:

$$\begin{aligned}\phi_1 - \widetilde{\phi}_1 &= \frac{1}{n-1}\sum_{k=1}^{q} w_k \sum_{j=1}^{n}\left[H_k(d_{1j}^k) - \widetilde{H}_k(d_{1j}^k)\right] \\ &= \frac{w_1}{n-1}\sum_{j=1}^{n}\left[H_1(d_{1j}^1) - \widetilde{H}_{1j}(d_{1j}^1)\right] \\ &= \frac{w_1}{n-1}\sum_{j=1}^{n} H_1(d_{1i}^1) - w_1\widetilde{\phi}_1^1 \end{aligned} \quad (13)$$

Since every criterion except the first remain unchanged. Similarly,

$$\begin{aligned}\phi_i - \widetilde{\phi}_i &= \frac{1}{n-1}\sum_{k=1}^{q} w_k \sum_{j=1}^{n}\left[H_k(d_{ij}^k) - \widetilde{H}_k(d_{ij}^k)\right] \\ &= \frac{w_1}{n-1}\left[H_1(d_{i1}^1) - \widetilde{H}_1(d_{i1}^1)\right] \end{aligned} \quad (14)$$

Since $H_k(d_{ij}^k) = \widetilde{H}_k(d_{ij}^k)$ for $k \neq 1$ or $j \neq 1$. Combining the two previous equations, one gets:

$$\begin{aligned} R_{1i}(\delta) - R_{1i}(0) &= \phi_1 - \widetilde{\phi}_1 - \left(\phi_i - \widetilde{\phi}_i\right) \\ &= \frac{w_1}{n-1}\left[\sum_{j=1}^{n} H_1(d_{1j}^1) + \widetilde{H}_1(d_{i1}^1) - H_1(d_{i1}^1)\right] - w_1 \cdot \widetilde{\phi}_1^1 \end{aligned} \quad (15)$$

Finally, the net flow score difference between a_1 and a_i is:

$$R_{1i}(\delta) = \frac{w_1}{n-1}\left[b_{1i}^1 + H_1(d_{1i}^1) + \sum_{j=1}^{n} H_1(d_{1j}^1)\right] \quad (16)$$

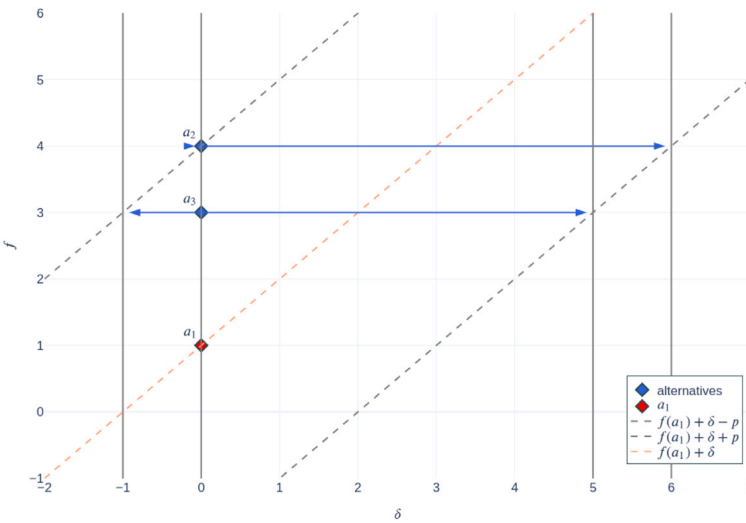


Fig. 3. Evolution with respect to δ of the A^δ-partition given by Eq. (19) of a set of alternatives A. The alternatives of A are represented on the y-axis. The dashed gray lines represent the lower and higher bounds of A_m^δ. For a given value of δ, every alternative with an evaluation below the lowest line belongs to A_l^δ, and every alternative with an evaluation above the highest line belongs to A_r^δ. The rest of the alternatives belong to A_m^δ. The blue arrows represent the range of values of δ for which the alternatives belong to A_m^δ.

with b_{1i}^1 a constant given by:

$$b_{1i}^1 = \frac{n-1}{w_1}\left[R_{1i}(0) - w_1 \cdot \widetilde{\phi}_1^1 + \widetilde{H}_1(d_{i1}^1)\right] \tag{17}$$

According to the monotony of $H_1(d_{ij}^1)$ noted in Section 3, one can deduct from Eq. (16) that the net flow score difference R_{1i} has similar properties and in particular is non decreasing in δ.

To further develop the expression of R_{1i}, one needs to add hypotheses regarding the type of preference function used. Considering that the first criterion is modeled using a V-shape preference function, $H_1(d_{1j}^1)$ has the below form:

$$H_1(d_{1j}^1) = \begin{cases} 0 & \text{if } a_j = a_1 \\ -1 & \text{else if } f_1(a_1) + \delta - f_1(a_j) < -p_1 \\ \frac{f_1(a_1)+\delta-f_1(a_j)}{p_1} & \text{else if } -p_1 \le f_1(a_1) + \delta - f_1(a_j) \le p_1 \\ 1 & \text{else, } p_1 < f_1(a_1) + \delta - f_1(a_j) \end{cases} \tag{18}$$

This expression implicitly defines a partition of the set of alternatives $A \setminus \{a_1\}$ into three parts. Given $f_1(a_1)$ and δ, let $A_l^\delta, A_m^\delta, A_r^\delta, \{a_1\}$ be the following disjoint subsets of A:

$$\begin{aligned} A_r^\delta &= \{a_j \in A | f_1(a_1) + \delta + p_1 < f_1(a_j)\} \\ A_m^\delta &= \{a_j \in A \setminus \{a_1\} | f_1(a_1) + \delta - p_1 \le f_1(a_j) \le f_1(a_1) + \delta + p_1\} \\ A_l^\delta &= \{a_j \in A | f_1(a_j) < f_1(a_1) + \delta - p_1\} \end{aligned} \tag{19}$$

It is obvious that $A = A_l^\delta \cup A_m^\delta \cup A_r^\delta \cup \{a_1\}$. Partitions of A defined as above are later called A^δ-partitions.

Let us note that different values of δ can lead to the same A^δ-partitions. This observation is at the core of the proposed approach to solve Problem 1 and is illustrated in Fig. 3.

Using the A^δ-partition, $H_1(d_{1j}^1)$ can be rewritten as:

$$H_1(d_{1j}^1) = \begin{cases} -1 & \text{if } a_j \in A_r^\delta \\ \frac{f_1(a_1)+\delta-f_1(a_j)}{p_1} & \text{if } a_j \in A_m^\delta \\ 1 & \text{if } a_j \in A_l^\delta \\ 0 & \text{if } a_j = a_1 \end{cases} \tag{20}$$

for any $a_j \in A$. The sum in the net flow score difference (Eq. (16)) can be computed as follows:

$$\sum_{j=1}^{n} H_1(d_{1j}^1) = \sum_{a_j \in A_l^\delta} H_1(d_{1j}^1) + \sum_{a_j \in A_m^\delta} H_1(d_{1j}^1) + \sum_{a_j \in A_r^\delta} H_1(d_{1j}^1) \tag{21}$$

$$= |A_l^\delta| + \sum_{a_j \in A_m^\delta} \left(\frac{f_1(a_1) + \delta - f_1(a_j)}{p_1}\right) - |A_r^\delta| \tag{22}$$

$$= \frac{|A_m^\delta|}{p_1}\delta + |A_l^\delta| - |A_r^\delta| + \frac{f_1(a_1)|A_m^\delta|}{p_1} - \frac{1}{p_1}\sum_{a_j \in A_m^\delta} f_1(a_j) \tag{23}$$

Let $I \subset \mathbb{R}$ be a region such that the A^δ-partition defined in Eq. (19) is the same for any $\delta \in I$. Thus inside I, the net flow score difference is a linear function of δ and can be written as the canonical form of Eq. (24).

$$R_{1i}(\delta) = \frac{w_1}{n-1}\left(b_{1i}^1 + \alpha_I + \beta_I \cdot \delta\right), \quad \delta \in I \tag{24}$$

where α_I and β_I are constant constants (within each interval I) given in Eq. (25). The index I is used to emphasize that they differ from one interval I to another and is omitted later on.

$$\begin{aligned} \alpha &= |A_l^\delta| - |A_r^\delta| + \frac{f_1(a_1)|A_m^\delta|}{p_1} - \frac{1}{p_1}\sum_{a_j \in A_m^\delta} f_1(a_j) + \begin{cases} -1 & \text{if } a_i \in A_r^\delta \\ \frac{f_1(a_1)-f_1(a_i)}{p_1} & \text{if } a_i \in A_m^\delta \\ 1 & \text{if } a_i \in A_l^\delta \end{cases} \\ \beta &= \frac{|A_m^\delta|}{p_1} + \begin{cases} \frac{1}{p_1} & \text{if } a_i \in A_m^\delta \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{25}$$

Once this interval I has been identified, the optimization problem is linear and, consequently, can easily be solved.

It is clear from Eq. (19) that any $\delta \in \mathbb{R}^+$ leads to the definition of exactly one A^δ-partition and, by construction, any interval I as defined above corresponds to exactly one A^δ-partition. All such intervals are hence pairwise disjoint and their collection forms a partition of $\mathbb{R}^+$. Such partitions are called $\mathbb{R}_A^k$-partitions. For a given ranking problem, an $\mathbb{R}_A^k$-partition can be associated to each criterion, hence the index k. The rest of this section is dedicated to the identification $\mathbb{R}_A^k$-partitions.

An illustration of the correspondence between the A^δ-partitions , the net flow score difference and an $\mathbb{R}_A^k$-partition is given in Fig. 4.

A^δ-partitions and $\mathbb{R}_A^k$-partitions are fundamentally different: a $\mathbb{R}_A^k$-partition divides $\mathbb{R}$ into multiple intervals, each of which is associated to a different A^δ-partition. In each part of the $\mathbb{R}_A^k$-partition, R_{1i} is linear in δ and the corresponding A^δ-partition defines the coefficients of linearity.

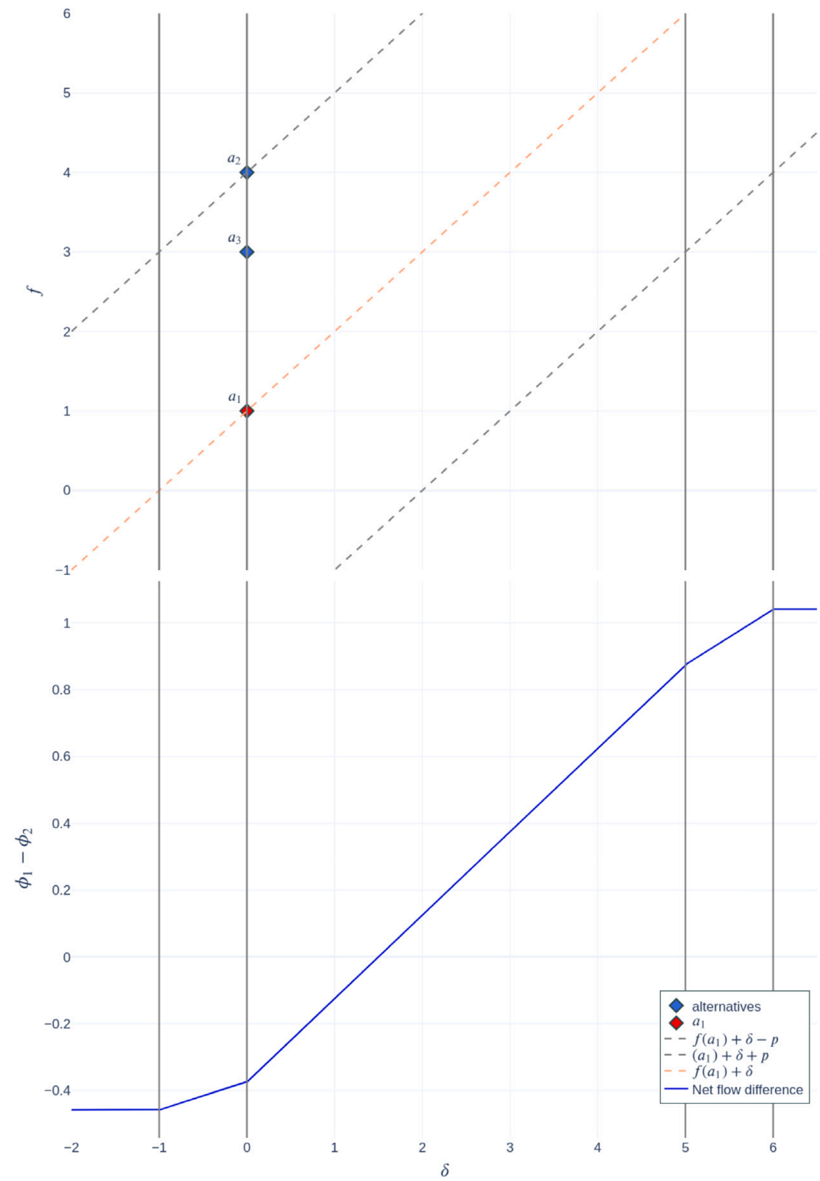


Fig. 4. Correspondence between the regions of $\mathbb{R}$ with the same A^δ-partition and the linearity of the net flow score difference in each of these regions. The collection of such regions is called the $\mathbb{R}^k_A$-partitionof the problem regarding the given criterion. Its bounds are represented by vertical gray lines. The blue lines represent the net flow score difference between two alternatives in each part. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

An algorithmic approach is introduced below to find the $\mathbb{R}^k_A$-partition of a given alternative on a given criterion, from the corresponding A^δ-partitions. This method is based on the two following results. First, Lemma 1 describes a necessary and sufficient condition on the evolution of A^δ-partitions when δ varies. Then, Theorem 1 uses Lemma 1 to quantify how much an evaluation should change to change the A^δ-partition.

Lemma 1 (A^δ*-partitions Evolution Follows Shifting Window*)**.** *Let $f_1(a_1)$ be the evaluation of a_1 in a ranking obtained with* Promethee *II, and $\widetilde{A}^\delta_l, \widetilde{A}^\delta_m, \widetilde{A}^\delta_r$ the corresponding A^δ-partition for $\delta = 0$ given by Eq. (19). Let $f_1(a_1) + \delta$, $\delta > 0$ be the increased evaluation, and $A^\delta_l, A^\delta_m, A^\delta_r$ the corresponding A^δ-partition. Then, if it exists, the smallest value δ_{min} of δ for which $A^\delta_m \neq A^{\delta\,\prime}_m$, is such that at least one of the following conditions is satisfied:*

$$\arg\min_{a_j \in \widetilde{A}^\delta_m} f_1(a_j) \in A^\delta_l \tag{26}$$

or

$$\arg\min_{a_j \in \widetilde{A}^\delta_r} f_1(a_j) \in A^\delta_m \tag{27}$$

Moreover, δ_{min} does not exist if $\widetilde{A}^\delta_l = A \setminus \{a_1\}$ or, equivalently, if $\widetilde{A}^\delta_m = \widetilde{A}^\delta_r = \emptyset$.

Lemma 1 can easily be interpreted: when δ is increased, the A^δ-partition changes by moving the alternative(s) of lowest evaluation from A^δ_m (or A^δ_r) to A^δ_l (or A^δ_m).

The proof of this lemma is proposed in two steps: first, two values δ_l and δ_r are computed such that Eq. (26) is satisfied for $\delta = \delta_l$ and Eq. (27) is satisfied for $\delta = \delta_r$. Then, it remains to prove that if $\delta < \min(\delta_l, \delta_r)$ the hypotheses regarding the A^δ-partition are not satisfied. The last part of the lemma is briefly justified.

First, let us note that the definition of A^δ-partitions can be reformulated as follows:

$$A_r^\delta = \{a_j \in A | \delta < f_1(a_j) - f_1(a_1) - p_1\}$$
$$A_m^\delta = \{a_j \in A \setminus \{a_1\} | f_1(a_j) - f_1(a_1) - p_1 \leq \delta \leq f_1(a_j) - f_1(a_1) + p_1\} \tag{28}$$
$$A_l^\delta = \{a_j \in A | f_1(a_j) - f_1(a_1) + p_1 < \delta\}$$

This formulation gives immediate bounds on the extreme values of δ that preserve A-partitions.

Let $a_m = \arg\min_{a_j \in \widetilde{A}_m^\delta} f_1(a_j)$ be the alternative of $\widetilde{A}_m^\delta$ with minimal evaluation and similarly, $a_r = \arg\min_{a_j \in \widetilde{A}_r^\delta} f_1(a_j)$, and let δ_l, δ_r be defined as

$$\delta_l = f_1(a_m) - (f_1(a_1) - p_1)$$
$$\delta_r = f_1(a_r) - (f_1(a_1) + p_1) \tag{29}$$

Let us consider $\delta > \delta_l$ and $A_l^\delta, A_m^\delta, A_r^\delta$ the corresponding A^δ-partition , then $A_m^\delta \neq \widetilde{A}_m^\delta$. This is shown below by simply substituting the definition of δ_l

$$\begin{aligned} \delta > \delta_l &\implies \delta > f_1(a_m) - f_1(a_1) + p_1 \\ &\implies f_1(a_1) + \delta - p_1 > f_1(a_m) \\ &\implies a_m \in A_l^\delta \implies a_m \notin A_m^\delta \end{aligned} \tag{30}$$

As $a_m \in \widetilde{A}_m^\delta$ by hypothesis, $A_m^\delta \neq \widetilde{A}_m^\delta$. Similar results are obtained for a_r if $\delta > \delta_r$. Thus obviously, if $\delta_{\min} = \min(\delta_l, \delta_r)$, then

$$\delta > \delta_{\min} \implies A_m^\delta \neq \widetilde{A}_m^\delta \tag{31}$$

Let us now show that if $\delta < \delta_l$ then $\widetilde{A}_l^\delta$ does not change and if $\delta < \delta_r$ then $\widetilde{A}_r^\delta$ does not change. Thus, if $\delta < \min(\delta_l, \delta_r)$, neither change and $\widetilde{A}_m^\delta$ also remains unchanged. This is done by contradiction. Let us suppose that there is an alternative $a \in \widetilde{A}_m^\delta$ such that for some δ with $0 \leq \delta < \delta_l$ then $a \notin A_m^\delta$. From the monotony of the bounds of A_m^δ, it is only possible if $a \in A_l^\delta$, however:

$$\left.\begin{aligned} a \in \widetilde{A}_m^\delta \\ a \in A_l^\delta \end{aligned}\right\} \implies f_1(a_1) - p_1 \leq f_1(a) < f_1(a_1) + \delta - p_1 \tag{32}$$

And, by hypothesis, $\delta_l > \delta$ which implies $f_1(a_m) > f_1(a_1) + \delta - p_1$ similarly to Eq. (30). Combined with Eq. (32), these imply that $f_1(a) < f_1(a_m)$, which is a contradiction with the definition of $a_{(m)}$ in Eq. (29). This can be adapted to show that if $\delta < \delta_r$ there is no alternative from $\widetilde{A}_r^\delta$ in A_m^δ or A_l^δ, hence $\widetilde{A}_r^\delta$ stays constant. In conclusion, if $\delta < \min(\delta_l, \delta_r)$ both $A_l^\delta = \widetilde{A}_l^\delta$ and $A_r^\delta = \widetilde{A}_r^\delta$, and thus $A_m^\delta = \widetilde{A}_m^\delta$, which ends the second part of the proof.

Regarding the existence of δ_{min}, if $\widetilde{A}_m^\delta = \widetilde{A}_r^\delta = \emptyset$, then neither a_m or a_r exist. It also means that all alternatives (except a_1) are in $\widetilde{A}_l^\delta$, and it is obvious from its definition (Eq. (28)) that increasing δ does not allow any alternative to leave A_l^δ. This concludes the proof of the lemma.

Theorem 1 is an almost direct consequence of Lemma 1 and gives an explicit formula to compute δ_{min} based on the A^δ-partitions.

Theorem 1 (*Smallest Deviations for Change of A^δ-partition*). *Let $f_1(a_1)$ the evaluation of a_1 in a ranking obtained with* Promethee *II using a V-shape criterion and let $A_l^\delta, A_m^\delta, A_r^\delta$ the corresponding A^δ-partition given by Eq. (19). Let δ_{min} be the smallest deviation for which the A^δ-partition changes when $f_1(a_1)$ is increased. Then δ_{min} is given by*

$$\delta_{min} = \min\left(\min_{a_j \in A_m^\delta} f_1(a_j) + p_1, \min_{a_j \in A_r^\delta} f_1(a_j) - p_1\right) - f_1(a_1)$$

Where, by convention, $\min_{a_j \in \emptyset} f_1(a_j) = \infty$.

Theorem 1 is proven by combining Lemma 1 and the definitions of δ_l and δ_r given in Eq. (29).

Algorithm 1 can be used to compute the $\mathbb{R}_A^k$-partition using these two results.

Algorithm 1 Generate $\mathbb{R}_A^k$-partition of one alternative on one criterion

Require: A, f_1, i
$A_l^\delta \leftarrow \emptyset, A_m^\delta \leftarrow \emptyset, R_{part} \leftarrow \emptyset$
$A_r^\delta \leftarrow A \setminus \{a_1\}$ ordered by f_1
$low \leftarrow -\infty$
while $low < \infty$ **do**
  $up \leftarrow \delta_{min}$ according to Theorem 1
  $A_l^\delta, A_m^\delta, A_r^\delta \leftarrow$ Update A^δ-partition using $\delta = up + \varepsilon$ for some small ε
  $R_{part} \leftarrow R_{part} \cup \{[low, up[\}$
  $low \leftarrow up$
end while

A simple example of the execution of Algorithm 1 is given in Appendix B. Once the $\mathbb{R}_A^k$-partition is found, the solution of Problem 1 is straightforward: for each part of the $\mathbb{R}_A^k$-partition , use its associated A^δ-partition to compute the values of α and β given by Eq. (25) and solve the inequality $R_{1i} \geq 0$ using Eq. (24). This is the procedure of Algorithm 2.

Let us note that there is no guarantee that the inequality $R_{1i} \geq 0$ has a solution in any part of the $\mathbb{R}_A^k$-partition , but it can be verified quite easily. As mentioned above, $R_{1i}(\delta)$ is non-decreasing. If $R_{1i}(0) > 0$ the problem is solved immediately. Otherwise, $\lim_{\delta \to \infty} R_{1i}(\delta)$ is easy to compute: every alternative falls in A_l^δ, so $\alpha = n$ and $\beta = 0$. $\beta = 0$ implies that $R_{1i}(\delta)$ is constant after some point δ_0, thus if $\lim_{\delta \to \infty} R_{1i}(\delta) > 0$ the solution to Problem 1 exists and is smaller or equal to δ_0.

In cases with no solution the procedure should yield $\delta = \infty$ by convention. This means that however the evaluation of a_1 is changed, the relative ranking between a_1 and a_i will not change, which is consistent with the fact that the Promethee II method is only partially compensatory: working on only one criterion may not be sufficient to change the ranking.

Since $R_{1i}(\delta)$ is non decreasing, a solution to the inequality $R_{1i}(\delta) \geq 0$ has to be positive. Algorithm 1 provides a way to compute the whole $\mathbb{R}_A^k$-partition of one alternative regarding one criterion but only its positive parts are of interest. Moreover, it does not need to be computed entirely: once a value of δ is found such that the relative ranking of the two alternatives is reversed, the algorithm can stop. Algorithm 2 is a modification of Algorithm 1 that stops as soon as a valid solution is found. If no solution exists, the algorithm stops at the end of the $\mathbb{R}^\delta$-partition and returns ∞.

Algorithm 2 Minimum change on the evaluation of the first alternative regarding the first criterion to reverse its relative ranking regarding another alternative (pseudocode).

Require: A, f_1 as defined in Section 4, i the index of the second alternative.
$A_l, A_m, A_r \leftarrow$ initiate A^δ-partition ▷ See Eq. (19)
$low \leftarrow 0$
while $low < \infty$ **do**
  $up \leftarrow \delta_{min}$ according to Theorem 1
  compute α, β using Eqs. (23) and (25)
  $\delta \leftarrow -\frac{\alpha}{\beta}$
  if $\delta \in [low, up[$ **then** ▷ Valid solution to $R_{1i} \geq 0$
    return δ
  end if
  $A_l, A_m, A_r \leftarrow$ Update A^δ-partition using $\delta = up + \varepsilon$ for some small ε
  $low \leftarrow up$
end while
return ∞

The choice of the data structure to use for the A^δ-partition is obviously an important factor on the efficiency of Algorithm 2. A simple yet efficient solution is to sort the alternatives by their evaluation regarding the criterion of interest and store the result in an array. From now on, $A[i]$ denotes the alternative ranked in position i when they are ordered according to f_1. Then, two indices $up_l, up_m \in [0, n]$ can be used to split this array into three parts corresponding to A_l, A_m and A_r. An element at index i is in A_l if $i < up_l$, in A_m if $up_l \leq i < up_m$ and in A_r otherwise. For example, consider the sorted array of evaluations $[1,2,3,4,5]$ and the indices $up_l = 2$, $up_m = 4$ (by convention, indices start at zero). Then, $A_l = [1,2]$, $A_m = [3,4]$ and $A_r = [5]$. If $up_l = up_m = 4$, then $A_l = [1,2,3,4]$, $A_m = \emptyset$ and $A_r = [5]$. If $up_l = 0$ and $up_m = 5$, then $A_l = \emptyset$, $A_m = [1,2,3,4,5]$ and $A_r = \emptyset$.

With this implementation, updating the A^δ-partition also has a complexity of $O(1)$ since it only involves changing the value of the indices up_l and up_m. Finding $\delta_{\min}$ using Theorem 1 can be done in constant time since the minimum of A_l, A_m, and A_r are given by $A_s[0], A_s[up_l], A_s[up_m]$ respectively. Initializing the A^δ-partition has a complexity of $O(n \log n)$ operations: the alternatives need to be sorted by their evaluation regarding the criterion of interest, then only one loop over the alternatives is needed to compute the indices up_l and up_m.

If the algorithm does not find any solution, it must iterate over all the $\mathbb{R}^k_A$-partition. In the worst case scenario, the $\mathbb{R}^k_A$-partition is composed of $2(n-1)$ parts: if all alternatives start in A^δ_r, then each individually move to A^δ_m, then to A^δ_l. In such circumstances, the *while* loop is executed $O(n)$ times. Each operation in the *while* loop requires $O(1)$ operations when using the implementation of A^δ-partitions described above. Overall, the time complexity of the algorithm is thus $O(n \log n)$.

However, the sorting procedure of the algorithm can be skipped if the alternatives have already been sorted before, which is usually the case. The fastest known way to compute a Promethee II ranking involving any preference function other than a Gaussian was proposed by Calders & Van Assche [49] and requires that the alternatives are sorted by their evaluation for each criterion. If those results are stored, the complexity of Algorithm 2 is reduced to $O(n)$ since the initialization of the A^δ-partition does not require sorting alternatives anymore.

Furthermore, the worst case scenario is only reached if the alternative does not find any finite solution. As discussed above, the existence of such a solution is easy to verify by computing $\lim_{\delta\to\infty} R_{1i}(\delta)$, which also takes constant time.

This theoretical complexity has been empirically verified by running Algorithm 2 on randomly generated datasets of increasing size and computing the time required to reach a solution. Random datasets are not expected to represent any real-world problem, but they are useful to assess the growth of the computation time as a function of only the size of the problem. The growth of average time of computation is not expected to be affected much by this choice. For each given size n (up to ten millions), ten datasets each composed of two criteria and n alternatives were generated with performances following a uniform distribution between zero and ten, and V-Shape preference functions were associated to each criterion with their preference threshold p also drawn at random uniformly between zero and ten. One run of Algorithm 2 was performed on each. Fig. 5 represents the evolution of the average time of computation as a function of the number of alternatives in the problem. The hardware used for the simulations is an Intel Core i9-11900K CPU and 32 GB of RAM. From the figures, it seems that the growth is at most linear, which is consistent with the theoretical complexity of $O(n)$.

5. Optimizing the evaluations of one alternative regarding multiple criteria

Methods to compute how much to modify a single evaluation of an alternative to influence its position in the ranking were presented in Section 4. A generalization of this approach to multiple evolving criteria is given in this section. Evaluations of one alternative regarding K criteria are now allowed to vary independently. As before, the goal is to find the minimal change in evaluations that would lead to a change in the relative ranking of the alternative and another but multiple values are now to minimize: the deviations regarding each criterion. Such problems fall in the domain of multi-objective optimization. The most general form of a solution set is given by the *Pareto optimal front* [50].

Definition 1 (*Pareto Dominance*). Given a function $F : X \subset \mathbb{R} \to \mathbb{R}^q : x \mapsto (f_1(x), \ldots, f_n(x))$ to maximize (w.l.o.g.), one says that $x \in X$ Pareto dominates $y \in X$ if and only if

$$\forall k \in 1, \ldots, q : f_k(x) \geq f_k(y) \text{ and } \exists j \in 1, \ldots m : f_j(x) > f_j(y)$$

One then writes $x \succ y$ if x Pareto dominates y, $x \nsucc y$ otherwise.

Definition 2 (*Efficient Set and Pareto Front*). the efficient set of X regarding F is a subset $P_X(F)$ that contains all non-dominated alternatives, *i.e.*:

$$P_X(F) = \{x \in X | \forall y \in X : y \nsucc x\}$$

The Pareto optimal front is the image of F restricted to argument values in $P_X(F)$.

Intuitively, the efficient set is the set of candidate solutions for which no other solution is strictly better and the Pareto front are the corresponding values of the objective functions. This definition is valid for optimization problems with constrains by defining $X = dom(F)$ as the set of valid solutions.

Problem 2 (*Minimal Change of Multiple Evaluations*). Let us consider a ranking obtained with Promethee II such that $a_i \succ a_1$ for some $i, 1 < i \leq n$. Let $K \leq q$ and $\vec{\delta} = (\delta_1, \ldots, \delta_K) \in \mathbb{R}^K$ be such that the evaluations of a_1 regarding the K first criteria are changed to

$$\left\{ f_1(a_1) + \delta_1, f_2(a_1) + \delta_2, \ldots, f_K(a_1) + \delta_K \right\}$$

What is the Pareto optimal set or front of $\vec{\delta}$ such that $a_1 \succsim a_i$?

Since the solution to Problem 2 takes the form of an infinite Pareto front, a DM would require further MCDM analysis to select the most appealing one according to her/his preference, most probably from continuous methods of multi-objective optimization. The interested reader can find a review of such methods in [50].

Again, the evaluations of a_1 regarding the K first criteria are considered without loss of generality and the case where the evaluations are computed using V-shape preference functions is considered. The discussion of Appendix A remains valid in this scope.

Besides the increased dimension of the problem, another important difference with the single evaluation case is that this time some evaluations might increase ($\delta_k > 0$) while others decrease ($\delta_k < 0$). Such compensation effects are not relevant in the single evaluation case.

Similarly to the single criterion case, the piecewise linearity of the preference functions implies a piecewise linear behavior of the net flow score difference, as summarized in Theorem 2.

Theorem 2 (*Piecewise Linearity of R_{1i} for Multiple Evaluations*). *Given a ranking obtained with* Promethee II, *for which $\mathcal{F}_k$ $(k = 1, \ldots, K)$ are V-shape preference functions, and the evaluations $f_k(a_1)$ are allowed to vary, each by a value δ_k; the net flow score difference between two alternative a_1 and $a_i, i = 2, \ldots, n$ is piecewise linear in $\vec{\delta} = (\delta_1, \ldots, \delta_K)$.*

Theorem 2 is proven using the notations defined in Section 4 and similar reasoning. This leads to the following expression for $R_{1i} : \mathbb{R}^K \to \mathbb{R}$:

$$(n-1) \cdot R_{1i}(\vec{\delta}) = b_{1i} + \sum_{k=1}^{K} w_k \left(H_k(d^k_{1i}) + \sum_{j=1}^{n} H_k(d^k_{1j}) \right) \tag{33}$$

Where b_{1i} is given by

$$b_{1i} = (n-1) \left(\widetilde{\phi}_1 - \widetilde{\phi}_i - \sum_{k=1}^{K} w_k \cdot \widetilde{\phi}^k_1 \right) - \sum_{k=1}^{K} w_k \cdot \widetilde{H}_k(d^k_{1i}) \tag{34}$$

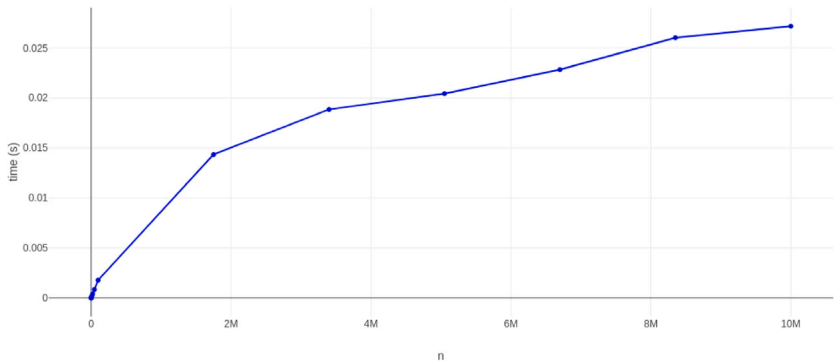


Fig. 5. Evolution of the average time of computation of Algorithm 2 as a function of the number of alternatives in randomly generated problems with V-Shape preference functions. The growth is sublinear, which is consistent with the theoretical complexity of $O(n)$.

This expression again suggests the definition of K A^δ-partitions depending on the evaluations $f_k(a_1)+\delta_k$, using the form given by Eq. (18). For $k = 1,\dots,K$, let us define the following K A^δ-partitions:

$$\begin{aligned} A_r^{\delta_k} &= \{a_j \in A | f_k(a_1) + \delta_k + p_k < f_k(a_j)\} \\ A_m^{\delta_k} &= \{a_j \in A \setminus \{a_1\} | f_k(a_1) + \delta_k - p_k \le f_k(a_j) \le f_k(a_1) + \delta_k + p_k\} \\ A_l^{\delta_k} &= \{a_j \in A | f_k(a_j) < f_k(a_1) + \delta_k - p_k\} \end{aligned} \tag{35}$$

For any $k = 1,\dots,K$, and given one deviation δ_k, $A \setminus \{a_1\}$ is partitioned in three parts $A_l^{\delta_k}, A_m^{\delta_k}, A_r^{\delta_k}$. The above definitions thus give K different partitions of A which are to be used independently in the sums of Eq. (33):

$$\sum_{j=1}^{n} H_k(d_{1j}^k) = \frac{|A_m^{\delta_k}|}{p_k}\delta_k + |A_l^{\delta_k}| - |A_r^{\delta_k}| + \frac{f_k(a_1)|A_m^{\delta_k}|}{p_k} - \frac{1}{p_k}\sum_{a_j \in A_m^{\delta_k}} f_k(a_j) \tag{36}$$

As it was the case for the one criterion case, each term of Eq. (33) is thus piecewise linear in δ_k for some k. Again, for a given set of K A-partitions $\left\{\left\{A_l^{\delta_k}, A_m^{\delta_k}, A_r^{\delta_k}\right\}, k = 1,\dots,K\right\}$, this expression is linear in $\delta_k, k = 1,\dots,K$ on a restricted region I of $\mathbb{R}^K$.

$$(n-1)R_{1i}(\vec{\delta}) = b_{1i} + \sum_{k=1}^{K} w_k \left(\alpha_k + \beta_k \cdot \delta_k\right), \quad \vec{\delta} \in I \subset \mathbb{R}^K \tag{37}$$

Where the expressions of α_k and β_k are similar to Eq. (25) and are given below in Eq. (38).

$$\begin{aligned} \alpha_k &= |A_l^{\delta_k}| - |A_r^{\delta_k}| + \frac{f_k(a_1)}{p_k}|A_m^{\delta_k}| - \frac{1}{p_k}\sum_{a_j \in A_m^{\delta_k}} f_k(a_j) \\ &\quad + \begin{cases} -1 & \text{if } a_i \in A_r^{\delta_k} \\ \frac{f_k(a_1)-f_k(a_i)}{p_k} & \text{if } a_i \in A_m^{\delta_k} \\ 1 & \text{if } a_i \in A_l^{\delta_k} \end{cases} \\ \beta_k &= \frac{|A_m^{\delta_k}|}{p_k} + \begin{cases} \frac{1}{p_k} & \text{if } a_i \in A_m^{\delta_k} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{38}$$

As before, Problem 2 can be solved by finding the regions of $\mathbb{R}^K$ for which all K A^δ-partitions are constant and finding all δ_k in these regions such that $R_{1i} \ge 0$. In what follows, this partition of $\mathbb{R}^K$ is called the $\mathbb{R}_A^K$-partition. As in the single criterion case, there is no guarantee that a solution to $R_{1i} \ge 0$ exists in any region of the $\mathbb{R}_A^K$-partition, even with trade-offs. Again, one could easily compute $\lim_{\vec{\delta}\to(\infty,\dots,\infty)} R_{1i}(\vec{\delta})$ and verify whether it is positive or not to find if a solution exists, the generalization from the situation with one criterion is immediate.

Let us consider the A^δ-partitions $A_l^{\delta_k}, A_m^k, A_r^{\delta_k}$ defined in Eq. (35) for each criterion $k = 1,\dots,K$ for a given set of values δ_k. It is obvious from the definition that if, δ_1 varies, none of $A_l^{\delta_k}, A_m^{\delta_k}, A_r^{\delta_k}$, for $k = 2,\dots,K$ changes. Similarly, if any other δ_k varies, only the kth A-partition can change, thus, the $\mathbb{R}_A^K$-partition is simply the Cartesian product of each $\mathbb{R}_A^k$-partition associated with each criterion. The $\mathbb{R}_A^K$-partition can thus be computed by using Algorithm 1 with each criterion independently, then computing their Cartesian product.

In each part of the $\mathbb{R}_\delta^K$-partition, the net flow difference R_{1i} is linear in each δ_k so the inequality $R_{1i} \ge 0$ defines a half space restricted to the part of linearity, *i.e.* a (possibly infinite) convex polyhedron. The Pareto optimal front in this part is a part of a hyperplane and the complete Pareto front is the union of those parts. Moreover, if every H_k functions is continuous (as it is the case for V-shape) these hyperplanes are given by the equations $R_{1i} = 0$ in each part of the $\mathbb{R}_\delta^K$-partition.

As mentioned in the previous section, with one criterion each $\mathbb{R}_A^k$-partition can be computed with a time complexity in $O(n \log n)$, or $O(n)$ if alternatives have already been sorted, and the computations in each part can be done in constant time. In the multi-criteria case, the Cartesian product of $\mathbb{R}^\delta$-partitions defines at most $(3n)^K$ regions, a naive implementation would verify the existence of a Pareto optimal front in each of these, which again takes constant time. Overall, such a procedure requires $O(K \cdot n)$ operations to compute $\mathbb{R}_A^k$-partitions and $O(n^K)$ operations to solve the problem in all the regions of the Cartesian product. The worst-case time complexity thus increases to $O(n^K)$ when going from one to multiple evolving criteria.

The number K of changing evaluations is expected to be small. The space of admissible changes $\vec{\delta}$ for a given alternative (regarding any number of criteria) in a real decision problem is expected to be small and choosing an efficient value $\vec{\delta}$ is a decision problem in itself. The exponential complexity in K should thus not be a major issue in practice.

Anyhow, the implementation presented above could still be improved regarding its average time-complexity. In the given implementation the $\mathbb{R}_A^K$-partition is entirely computed entirely, but the Pareto optimal front is a union of hyperplanes. By monotonicity of the net flow score difference in each δ_k, if a Pareto optimal front is found in a region, there is no need to verify the existence of a Pareto optimal front in any region corresponding to $\vec{\delta}$ with strictly larger components. Furthermore, once a piece of the Pareto optimal front is found in a part of the $\mathbb{R}_A^K$-partition , it seems reasonable to search for Pareto optimal fronts in the parts adjacent to this one, or at least “close enough” to it in some sense since preference functions are not necessary continuous. These leads are left for further research.

Finally, the reader should now be aware that when using PROMETHEE II to rank alternative, the relative ranking between two alternatives is not independent to third alternatives. This is the main focus of studies on rank reversal occurrences when adding or removing a third alternative mentioned in the introduction [26,27,29]. In this work, evaluations of an alternative are changed in order to improve its ranking relative to only *one* other alternative, but how does it influence the rest of the ranking? If all evaluations of a_1 are improved or unchanged, its relative ranking to any other alternatives can only improve by monotonicity of the preference relation. However, if a compromise is made (some δ_k are positive while other are negative), the presented algorithm does not *a priori* guarantee any behavior of the relative ranking between third alternatives. Without further developments, in order to guarantee that an alternative is ranked first after changing its evaluations the algorithm should be used with respect to every third alternative. This would give $n-1$ Pareto optimal fronts which would need to be merged together. The study of the influence on the whole ranking should be the subject of further research.

6. Experimental results

The presented methods have been tested on the dataset used to compute the Human Development Ranking index (HDR) in 2021 [51]. It is composed of 191 alternatives evaluated on 4 criteria: life expectancy at birth (LE), expected years of schooling (eS), mean years of schooling (μS) and gross national income per capita (GNI). The results are presented below. The methods have been implemented using the Rust programming language [52], and all plots have been generated using the Python programming language [53] with the Plotly library [54]. For the sake of completeness, the methods have also been tested on the Times Higher Education (THE) World University Ranking 2021 dataset [55], using the same procedure. The results are available in Appendix C.

The purpose of this test is only to illustrate the methods and to show that they work as expected. To do so, the net flow score difference between two alternatives a_1, a_i is computed using the definitions of the PROMETHEE II method with many different values of δ. These values are used to interpolate the net flow score difference as a function of δ, plotted, and compared with the solutions obtained by the methods presented in the previous section. As explained in Section 5, decision makers are expected to restrict the Pareto front of solutions in the end. For visualization purpose, only two criteria are considered at a time. The test is composed of the following steps:

1. Compute the ranking using the PROMETHEE II method with given parameters.
2. Select two alternatives a_i, a_j such that $a_i \succ a_j$, and two criteria k_1, k_2.
3. Solve Problem 1 with these alternatives for both criteria independently. This gives two values δ_1, δ_2.
4. Split the interval $[0, \delta_k + 1], k = 1, 2$ into 1000 points. For each point x, set $\delta_k = x$, and compute the net flow score difference using the definitions given in the PROMETHEE II method with these settings.
5. Plot the net flow score difference as a function of δ_k for $k = 1, 2$ and compare with the value computed in step 3.

The test of the computation of the Pareto optimal front in the case of two criteria followed the same approach. The net flow score difference has been computed for all points in the Cartesian product of two discretized intervals $[0, \delta_1 + 1] \times [0, \delta_2 + 1]$. The results were then plotted as a heatmap over which the Pareto front computed using the method presented in Section 5 was drawn.

The evaluations of alternatives from the HDR are publicly available, the authors also provide weights for each criterion. Those weights were not computed especially for the PROMETHEE II method but are used anyway since the only focus of this section is the testing of the methods presented in this work. The tests presented in this section consider only the V-shape preference function. For simplicity's sake, the preference thresholds are set to the median of absolute values of differences between evaluations of all pairs of alternatives for each criterion. Similar tests were performed with the usual and the V-shape with indifference preference functions, they are available in Appendix A. Using these settings, the head of the ranking computed by PROMETHEE II is Iceland $\succeq$ Norway $\succeq$ Switzerland … The question considered for the test is "how much should Norway improve its life expectancy at birth and/or its mean years of schooling to surpass Iceland in the ranking?".

Table 1
Parameters and the first four alternatives of the HDR. All criteria are normalized using a V-shape preference function.

	LE	eS	μS	GNI
w	1/3	1/6	1/6	1/3
p	71.694	13.405	9.307	12 306.341
Iceland	82.7	19.2	13.8	55 782
Norway	83.2	18.2	13.0	64 660
Switzerland	84.0	16.5	13.9	66 933
Hong Kong	85.5	17.3	12.2	62 607

The evaluations of the first four alternatives and the parameters used for each criterion are available in Table 1.

The method described in Section 4 computes that Norway should improve its life expectancy at birth by $\delta_1 \approx 0.0079$ years, or its mean years of schooling by $\delta_2 \approx 0.00404$ years to surpass Iceland. These results are consistent with the figures obtained by interpolation available in Fig. 6, one can see that for any $\delta \geq \delta_k$, $k = 1, 2$, $R(\text{Norway}, \text{Iceland}) \geq 0$.

However, it might not be acceptable for the decision makers to change any of these values this much without any counterbalance. In such cases, they might be interested in the Pareto optimal front of solutions when the two criteria are considered. The Pareto front computed using the method described in Section 5 is consistent with the heatmap given in Fig. 7.

Computations were performed on a desktop computer with an Intel Core i9-11900K CPU and 32 GB of RAM and took around 3 ms for the two single-criterion and the two-criteria combined. The same procedure for the THE dataset (1526 alternatives) took around 30 ms (details are available in Appendix C).

The range of values spanned by the Pareto optimal front on Fig. 7 is probably much larger than needed by a DM: improving life expectancy of 10 years seems quite unrealistic for example. This example however illustrates how the method can generate complete fronts that can then be a ground of discussion for the DM, who should restrict the range of actions as needed in a second time.

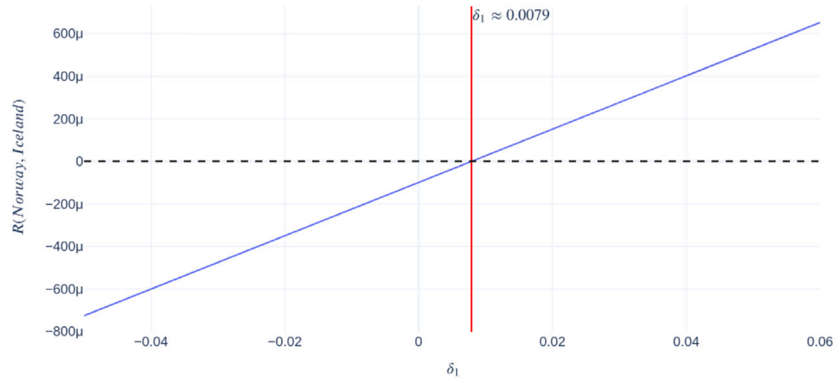
It also tends to show that the PROMETHEE II method is only partially compensatory, it might not be possible to improve its ranking by improving only one criterion. According to Fig. 7, if δ_1 is fixed to -4 years, no value of δ_2, as high as it may be, can make Norway surpass Iceland. And vice-versa, if δ_2 is fixed to -3 years, no value of δ_1 can make Norway surpass Iceland.

7. Conclusion and questions for further research

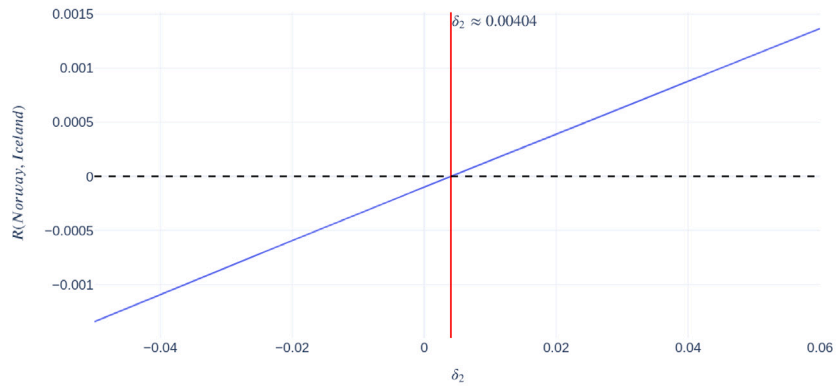
The present study aims at understanding the impact of one alternative's evaluations on the ranking obtained using PROMETHEE II. To do so, an inverse optimization approach to find how much evaluations of the alternative should change to improve its position in the final ranking is proposed. In the case of only one criterion is considered, it was shown that it might not be possible to improve the ranking of an alternative only by changing its evaluation. A computational method was also given to test the existence of a valid solution. For the instances where it is possible, an exact iterative method to find the optimal change in the evaluation was provided and discussed.

When multiple criteria are considered, similar results were obtained regarding both the existence of minimal deviations and the iterative method to find them in the form of a Pareto optimal front. Some of the solutions in this Pareto optimal front correspond to a trade-off: an improvement of the alternative's evaluation regarding one criterion, but a decrease of the evaluation regarding another criterion.

Several questions remain open for further research. For example, the methods presented in Sections 4 and 5 can be seen as tools for



(a) Net flow score difference between Norway and Iceland as a function of the improvement of life expectancy at birth of Norway.



(b) Net flow score difference between Norway and Iceland as a function of the improvement of mean years of schooling of Norway.

Fig. 6. Net flow score difference between Norway and Iceland as a function of δ_1 and δ_2 for (a) life expectancy at birth and (b) mean years of schooling. The red line represents the value computed using the method presented in Section 4. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

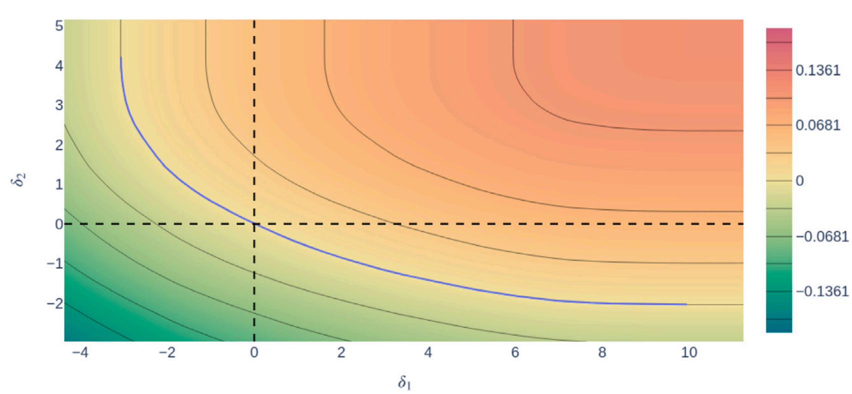


Fig. 7. Heatmap of the net flow score difference between Norway and Iceland as a function of both δ_1 (the increase of life expectancy at birth) and δ_2 (the increase of mean years of schooling), along with the Pareto front of minimal δ_1, δ_2 leading to Norway $\geq$ Iceland computed using the presented methods.

the person in charge of the evaluation of the alternatives to influence their place in the ranking given by PROMETHEE II. It immediately raises questions on the dynamics of the evaluation table. What happens when all alternatives try to improve their ranking by changing their evaluations? This could for example be studied under the prism of game theory, where each alternative would be a player trying to improve its ranking by changing its evaluation(s) within a range of possibilities. What would be the Nash equilibria of such a game?

On another note, this approach could be adapted to study intervals of stability on the evaluations of the alternatives. In this work, the focus was on modifying evaluations to improve the ranking of an alternative, but it could be interesting to study the opposite problem: how much can the evaluations of an alternative decrease without changing its position in the ranking?

Finally, there are some limitations to the methods presented in this work. First, a faster implementation of the multi-criteria case might be possible to reach by exploiting topological properties of the Pareto optimal front and the net flow score difference, in particular its non-decreasing behavior in all criteria and its continuity in the case of continuous preference functions. Second, in the case of multiple criteria changing the evaluations of an alternative by introducing a trade-off between criteria might influence the relative ranking between other alternatives and create rank reversal. It remains to study if improving the ranking of an alternative a to be better than the best alternative b might make another alternative c better than both a and b. Even though no such cases were found in the experiments, it does not seem impossible.

The present study is a first step towards understanding the impact of evaluations on the ranking obtained by PROMETHEE II, further research is needed to understand the practical implications of these results.

CRediT authorship contribution statement

Alexandre Flachs: Writing – original draft, Visualization, Software, Methodology, Conceptualization. **Yves De Smet:** Writing – original draft, Validation, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Other preference functions

As presented in Section 3, the preference function $\mathcal{F}_1(d_{ij}^1)$ can take different forms. The solution to Problem 1 presented above is restricted to V-shape preference functions. This section explains how it can be adapted to any other preference function except the Gaussian (see Fig. 1). Since the Gaussian preference is not linear by part the solution presented above cannot be adapted to it. However, as it is smooth, more general methods could be used to solve Problem 1.

The key idea to adapt the above development is to note that the net flow difference $R(a_1, a_i)$ is piecewise linear because the preference function $\mathcal{F}_1$ also is. In more details, Algorithms 1 and 2 remain true for any piecewise linear preference functions, except the part describing the upper bounds of parts of the $\mathbb{R}$-partition in Algorithm 1 and the values of α, β in Algorithm 2 which are to adapt. These adaptations are presented below. For each piecewise linear preference function, the corresponding A^δ-partitions and the α and β coefficients are given without further justification, as they can be computed following the approach described in Section 4.

Usual preference function

The symmetrical usual preference function is defined as

$$H_k(d_{ij}^k) = \begin{cases} -1 & \text{if } d_{ij}^k < 0 \\ 0 & \text{if } d_{ij}^k = 0 \\ 1 & \text{if } d_{ij}^k > 0 \end{cases} \tag{A.1}$$

The corresponding A^δ-partitions are composed of three parts defined as follows:

$$\begin{aligned} A_l^\delta &= \{a_j \in A | f_k(a_j) - f_k(a_1) - \delta < 0\} \\ A_m^\delta &= \{a_j \in A | f_k(a_j) - f_k(a_1) - \delta = 0\} \\ A_r^\delta &= \{a_j \in A | f_k(a_j) - f_k(a_1) - \delta > 0\} \end{aligned} \tag{A.2}$$

And bounds of the linear parts of $R(a_1, a_i)$ are thus of the form $f_k(a_j) - f_k(a_1), a_j \in A \setminus \{a_1\}$. Finally,

$$\alpha = |A_l^\delta| - |A_r^\delta| + H_k(d_{1j}^k + \delta) \tag{A.3}$$

$$\beta = 0 \tag{A.4}$$

Experimental validation similar to the one presented in Section 6 has been performed on the HDR dataset with the usual preference function. The results are presented in Fig. A.8. The net flow score difference is a step function because the usual preference function is a step function, which is equivalent to the statement $\beta = 0$.

U-shape preference function, quasi-criterion

The symmetrical U-shape preference function is defined as

$$H_k(d_{ij}^k) = \begin{cases} -1 & \text{if } d_{ij}^k < -p_k \\ 0 & \text{if } -p_k \le d_{ij}^k \le p_k \\ 1 & \text{if } d_{ij}^k > p_k \end{cases} \tag{A.5}$$

The corresponding A^δ-partitions are composed of three parts defined as follows:

$$\begin{aligned} A_l^\delta &= \{a_j \in A | f_k(a_j) - f_k(a_1) - \delta < -p_k\} \\ A_m^\delta &= \{a_j \in A | f_k(a_j) - f_k(a_1) - \delta \in [-p_k, p_k]\} \\ A_r^\delta &= \{a_j \in A | f_k(a_j) - f_k(a_1) - \delta > p_k\} \end{aligned} \tag{A.6}$$

It is the same as V-shape preference functions. Bounds of the linear parts of $R(a_1, a_i)$ are thus of the form $f_k(a_j) - f_k(a_1) \pm p_k, a_j \in A \setminus \{a_1\}$. Finally,

$$\alpha = |A_l^\delta| - |A_r^\delta| + H_k(d_{1j}^k + \delta) \tag{A.7}$$

$$\beta = 0 \tag{A.8}$$

similarly to the usual preference function.

Level preference function

The symmetrical level preference function is defined as

$$H_k(d_{ij}^k) = \begin{cases} -1 & \text{if } d_{ij}^k \le -p_k \\ -\frac{1}{2} & \text{if } -p_k < d_{ij}^k < -q_k \\ 0 & \text{if } -q_k \le d_{ij}^k \le q_k \\ \frac{1}{2} & \text{if } q_k < d_{ij}^k < p_k \\ 1 & \text{if } d_{ij}^k \ge p_k \end{cases} \tag{A.9}$$

The corresponding A^δ-partitions are composed of five parts defined as follows:

$$\begin{aligned} A_1^\delta &= \{a_j \in A | f_k(a_j) - f_k(a_1) - \delta \le -p_k\} \\ A_2^\delta &= \{a_j \in A | f_k(a_j) - f_k(a_1) - \delta \in] - p_k, -q_k[\} \\ A_3^\delta &= \{a_j \in A | f_k(a_j) - f_k(a_1) - \delta \in [-q_k, q_k]\} \\ A_4^\delta &= \{a_j \in A | f_k(a_j) - f_k(a_1) - \delta \in]q_k, p_k[\} \\ A_5^\delta &= \{a_j \in A | f_k(a_j) - f_k(a_1) - \delta \ge p_k\} \end{aligned} \tag{A.10}$$

And bounds of the linear parts of $R(a_1, a_i)$ are thus of the form $f_k(a_j) - f_k(a_1) \pm p_k$ or $f_k(a_j) - f_k(a_1) \pm q_k$ with $a_j \in A \setminus \{a_1\}$. Finally,

$$\alpha = |A_1^\delta| + \frac{|A_2^\delta|}{2} - \frac{|A_4^\delta|}{2} - |A_5^\delta| + H_k(d_{1j}^k + \delta) \tag{A.11}$$

$$\beta = 0 \tag{A.12}$$

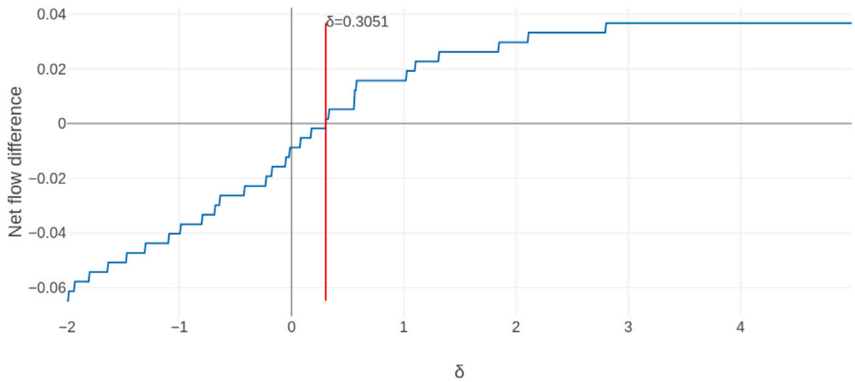


Fig. A.8. Experimental validation of Algorithm 2 adapted to the usual preference function. This represents how the improvement of life expectancy at birth of Iceland influences the difference between its net flow score and the one of Norway.

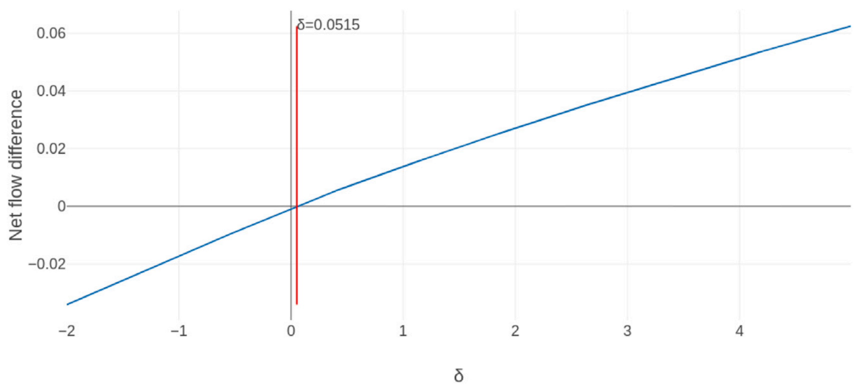


Fig. A.9. Experimental validation of Algorithm 2 adapted to the V-shape with indifference preference function on the HDR dataset. This represents how the improvement of life expectancy at birth of Iceland influences the difference between its net flow score and the one of Norway.

V-shape with indifference preference function

The symmetrical V-shape with indifference preference function is defined as

$$H_k(d_{ij}^k) = \begin{cases} -1 & \text{if } d_{ij}^k \le -p_k \\ \frac{d_{ij}^k + q_k}{p_k - q_k} & \text{if } -p_k < d_{ij}^k < -q_k \\ 0 & \text{if } -q_k \le d_{ij}^k \le q_k \\ \frac{d_{ij}^k - q_k}{p_k - q_k} & \text{if } q_k < d_{ij}^k < p_k \\ 1 & \text{if } d_{ij}^k \ge p_k \end{cases} \tag{A.13}$$

The corresponding A^δ-partitions are composed of the same five parts as the level preference function defined in Eq. (A.10) and the bounds of the $\mathbb{R}$-partition are of the same form. Finally,

$$\alpha = \frac{1}{p_k - q_k}\left[|A_2^\delta|(f_k(a_1) - q_k) + |A_4^\delta|(f_k(a_1) + q_k) - \sum_{a_j \in A_2^\delta \cup A_4^\delta} f_k(a_j) \right] + |A_1^\delta| - |A_5^\delta| + \begin{cases} 1 & \text{if } a_i \in A_1^\delta \\ \frac{f_k(a_1) - f_k(a_j) - q}{p_k - q_k} & \text{if } a_i \in A_2^\delta \\ 0 & \text{if } a_i \in A_3^\delta \\ \frac{f_k(a_1) - f_k(a_j) + q}{p_k - q_k} & \text{if } a_i \in A_4^\delta \\ -1 & \text{if } a_i \in A_5^\delta \end{cases}$$

$$\beta = \frac{|A_2^\delta| + |A_4^\delta|}{p_k - q_k} + \begin{cases} \frac{1}{p_k - q_k} & \text{if } a_i \in A_2^\delta \cup A_4^\delta \\ 0 & \text{otherwise} \end{cases} \tag{A.14}$$

Table B.2
Example evaluation table and parameters of each criterion. Both criteria are to maximize, and to be normalized using V-shape preference functions.

	k_1	k_2
a_1	3	1
a_2	2	4
a_3	2	3
p	2	3
w	0.3	0.7

Experimental validation similar to the one presented in Section 6 has been performed on the HDR dataset with the linear preference function by setting q and p as the first and third quartiles of the absolute difference of alternatives' performances. The results are presented in Fig. A.9.

Appendix B. Simple example of computing $\mathbb{R}^\delta$-partitions

This appendix contains a simple example to illustrate the procedure to compute $\mathbb{R}^\delta$-partitions. The evaluation table is given in Table B.2, it is composed of the alternatives evaluated regarding two criteria to maximize. The corresponding preference functions are both V-shape, and the parameters are given in the table along with the weights. Computing the PROMETHEE II ranking on this dataset gives the ranking $a_2 \succ a_3 \succ a_1$, let us compute the $\mathbb{R}^\delta$-partition of a_1 on the second criterion.

According to Algorithm 1, let us first suppose that $\delta = -\infty$, leading to $A_l^\delta = A_m^\delta = \emptyset$, $A_r^\delta = \{a_3, a_2\}$, where A_r^δ is sorted by increasing evaluation regarding the second criterion. Then, δ_l and δ_r are computed using Eq. (29). Since A_m^δ is empty, δ_l is not defined, and $\delta_l = \infty$. Eq.

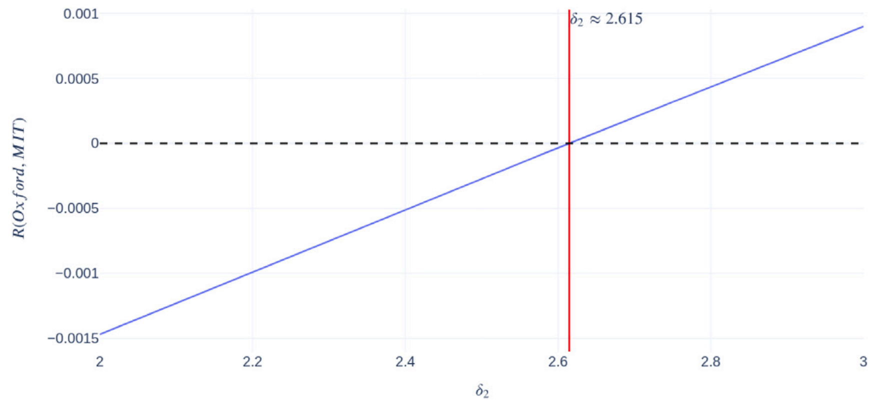


Fig. C.10. Net flow score difference between Oxford University and the Massachusetts Institute of Technology as a function of the increase of citation score of Oxford University.

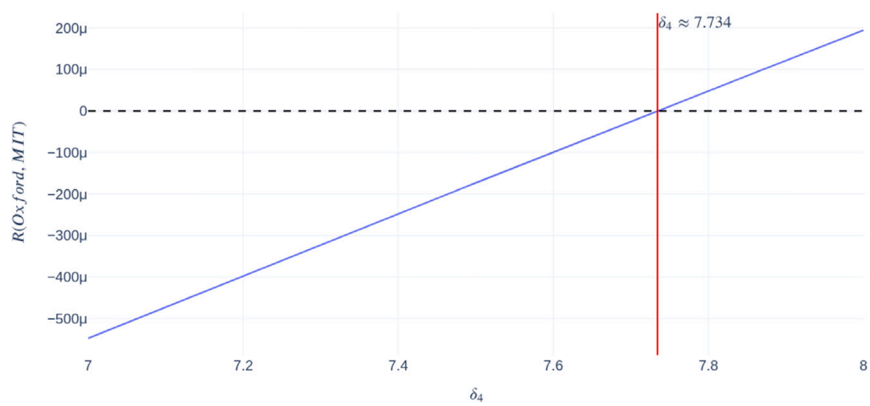


Fig. C.11. Net flow score difference between Oxford University and the Massachusetts Institute of Technology as a function of the increase of international outlook of Oxford University.

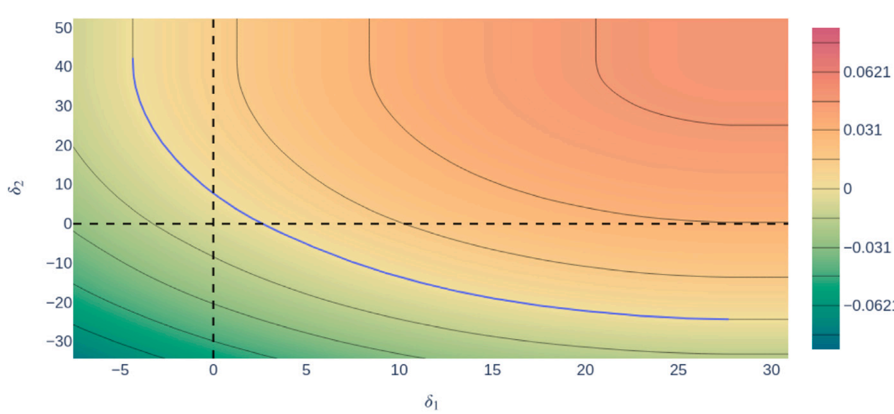


Fig. C.12. Heatmap of the net flow score difference between Oxford University and the Massachusetts Institute of Technology as a function of both δ_1 (the increase of citation score) and δ_2 (the increase of international outlook), along with the Pareto front of minimal δ_1, δ_2 leading to Oxford University ≥ Massachusetts Institute of Technology computed using the presented methods.

(29) gives $\delta_r = 3 - (1 + 3) = -1$. The first part of the $\mathbb{R}^\delta$-partition is thus $]-\infty, -1[$. If $\delta = -1$, the A^δ-partition changes and becomes $A_l^\delta = \emptyset, A_m^\delta = \{a_3\}, A_r^\delta = \{a_2\}$. Using Eq. (29) again, one finds:

$$\delta_l = 3 - (1 - 3) = 5$$

$$\delta_r = 4 - (1 + 3) = 0 \tag{B.1}$$

Since $\delta_r < \delta_l$, $\delta = \delta_r = 0$ and the second part of the $\mathbb{R}^\delta$-partition is $[-1, 0[$. If $\delta = 2$, the A^δ-partition becomes $A_l^\delta = \emptyset, A_m^\delta = \{a_3, a_2\}, A_r^\delta = \emptyset$. Continuing this process, one can find that the $\mathbb{R}^\delta$-partition of a_1 on the second criterion is $rp_2 = \{]-\infty, -1[, [-1, 0[, [0, 5[, [5, 6[, [6, \infty[\}$.

Let us now consider the $\mathbb{R}^2$-partition of a_1. One can verify that the $\mathbb{R}$-partition associated with the first criterion is: $rp_1 = \{]-\infty, -3[, [-3, 1[, [1, \infty[\}$. The $\mathbb{R}^2$-partition associated to a_1 is then the Cartesian product of rp_1 and rp_2. Any pair of intervals from $rp_1 \times rp_2$ defines a region of $\mathbb{R}^2$ for which both the A_k^δ-partitions are constant.

Appendix C. Results of the optimization of the evaluations of the alternatives in the THE ranking

The Times Higher Education World University Ranking 2021 dataset [55] has been used to test the methods presented in this work.

The dataset contains evaluations of 1526 universities on 5 criteria: teaching, research, citations, industry income, and international outlook. The weights are provided by the authors of the dataset: 0.3, 0.3, 0.3, 0.03, and 0.07, respectively. The evaluations are normalized using a V-shape preference function, with parameters p computed as the median of the absolute values of the differences between evaluations of all pairs of alternatives for each criterion: 9.0, 11.3, 27.6, 8.5 and 21.8, respectively.

The head of the ranking computed by PROMETHEE II is the Massachusetts Institute of Technology $\succeq$ Oxford University $\succeq$ Stanford University $\succeq$ Harvard University …

The question considered for the test is "how much should Oxford University improve its citation score and/or international outlook to surpass the Massachusetts Institute of Technology in the ranking?".

The results are displayed in Figs. C.10–C.12. As for the HDR dataset, the methods give consistent results with the interpolation of the net flow score difference as a function of δ. The computation took around 30 ms for the two single-criterion and the two-criteria combined on a desktop computer with an Intel Core i9-11900K CPU and 32 GB of RAM.

Data availability

The datasets we used are freely available online and referenced in the manuscript.

References

[1] Chan TCY, Mahmood R, Zhu IY. Inverse optimization: Theory and applications. Oper Res 2023.

[2] Greco S, Ehrgott M, Figueira JR, Ehrgott M, Figueira JR, Greco S. Multiple criteria decision analysis: State of the art surveys, Second edition.. International series in operations research & management science, vol. 233, New York, NY: Springer; 2016.

[3] Greco S, Słowiński R, Wallenius J. Fifty years of multiple criteria decision analysis: From classical methods to robust ordinal regression. European J Oper Res 2024. http://dx.doi.org/10.1016/j.ejor.2024.07.038, URL https://www.sciencedirect.com/science/article/pii/S0377221724005988.

[4] Roy B. Decision-aid and decision-making. European J Oper Res 1990;45(2):324–31. http://dx.doi.org/10.1016/0377-2217(90)90196-I, URL https://www.sciencedirect.com/science/article/pii/037722179090196I, OR for Engineers Expert Systems and Decision-Aid.

[5] Cinelli M, Kadziński M, Gonzalez M, Słowiński R. How to support the application of multiple criteria decision analysis? let us start with a comprehensive taxonomy. Omega 2020;96:102261. http://dx.doi.org/10.1016/j.omega.2020.102261, URL https://www.sciencedirect.com/science/article/pii/S0305048319310710.

[6] Wątróbski J, Jankowski J, Ziemba P, Karczmarczyk A, Zioło M. Generalised framework for multi-criteria method selection. Omega 2019;86:107–24. http://dx.doi.org/10.1016/j.omega.2018.07.004, URL https://www.sciencedirect.com/science/article/pii/S0305048317308563.

[7] Vincke P. Multicriteria decision-aid. Chichester: J. Wiley; 1992.

[8] Dyer JS. Maut — Multiattribute utility theory. In: Multiple criteria decision analysis: state of the art surveys. New York, NY: Springer New York; 2005, p. 265–92. http://dx.doi.org/10.1007/0-387-23081-5_7.

[9] Saaty TL. The analytic hierarchy and analytic network processes for the measurement of intangible criteria and for decision-making. In: Greco S, Ehrgott M, Figueira JR, editors. Multiple criteria decision analysis: state of the art surveys. New York, NY: Springer New York; 2016, p. 363–419. http://dx.doi.org/10.1007/978-1-4939-3094-4_10.

[10] Bouyssou D, Marchant T, Pirlot M, Tsoukiàs A, Vincke P. Aggregation procedures. In: Evaluation and decision models with multiple criteria: stepping stones for the analyst. Boston, MA: Springer US; 2006, p. 169–235. http://dx.doi.org/10.1007/0-387-31099-1_5.

[11] Grabisch M. The application of fuzzy integrals in multicriteria decision making. European J Oper Res 1996;89(3):445–56. http://dx.doi.org/10.1016/0377-2217(95)00176-X.

[12] Vanderpooten D. The interactive approach in MCDA: A technical framework and some basic conceptions. Math Comput Modelling 1989;12(10):1213–20. http://dx.doi.org/10.1016/0895-7177(89)90363-4, URL https://www.sciencedirect.com/science/article/pii/0895717789903634.

[13] Miettinen K, Mäkelä MM. NIMBUS — Interactive method for nondifferentiable multiobjective optimization problems. In: Tamiz M, editor. Multi-objective programming and goal programming: theories and applications. Berlin, Heidelberg: Springer Berlin Heidelberg; 1996, p. 50–7. http://dx.doi.org/10.1007/978-3-642-87561-8_5.

[14] Miettinen K, Eskelinen P, Ruiz F, Luque M. NAUTILUS method: An interactive technique in multiobjective optimization based on the nadir point. European J Oper Res 2010;206(2):426–34. http://dx.doi.org/10.1016/j.ejor.2010.02.041, URL https://www.sciencedirect.com/science/article/pii/S0377221710001785.

[15] Roy B. The outranking approach and the foundations of electre methods. Theory and Decision 1991;31(1):49–73. http://dx.doi.org/10.1007/BF00134132.

[16] Behzadian M, Kazemzadeh R, Albadvi A, Aghdasi M. PROMETHEE: A comprehensive literature review on methodologies and applications. European J Oper Res 2010;200(1):198–215. http://dx.doi.org/10.1016/j.ejor.2009.01.021.

[17] Brans J-P, De Smet Y. PROMETHEE methods. In: Greco S, Ehrgott M, Figueira JR, editors. Multiple criteria decision analysis: state of the art surveys. New York, NY: Springer New York; 2016, p. 187–219. http://dx.doi.org/10.1007/978-1-4939-3094-4_6.

[18] Brans J-P, Mareschal B. The PROMCALC & GAIA decision support system for multicriteria decision aid. Decis Support Syst 1994;12(4):297–310. http://dx.doi.org/10.1016/0167-9236(94)90048-5.

[19] Geldermann J, Zhang K. Software review: "decision lab 2000". J Multi- Criteria Decis Anal 2001;10:317–23. http://dx.doi.org/10.1002/mcda.311.

[20] Mareschal B, De Smet Y. Visual PROMETHEE: Developments of the PROMETHEE & GAIA multicriteria decision aid methods. In: 2009 IEEE international conference on industrial engineering and engineering management. 2009, p. 1646–9. http://dx.doi.org/10.1109/IEEM.2009.5373124.

[21] Hayez Q, De Smet Y, Bonney J. D-sight: A new decision making software to address multi-criteria problems. Int J Decis Support Syst Technol 2012;4:1–23. http://dx.doi.org/10.4018/jdsst.2012100101.

[22] Bouyssou D. Ranking methods based on valued preference relations: A characterization of the net flow method. European J Oper Res 1992;60(1):61–7. http://dx.doi.org/10.1016/0377-2217(92)90333-5.

[23] Marchant T. Valued relations aggregation with the borda method. J Multi-Criteria Decis Anal 1996;5(2):127–32. http://dx.doi.org/10.1002/(SICI)1099-1360(199606)5:2<127::AID-MCDA96>3.0.CO;2-A.

[24] Marchant T. Cardinality and the borda score. European J Oper Res 1998;108(2):464–72. http://dx.doi.org/10.1016/S0377-2217(97)00164-1.

[25] Dejaegere G, Boujelben MA, De Smet Y. An axiomatic characterization of promethee ii's net flow scores based on a combination of direct comparisons and comparisons with third alternatives. J Multi- Criteria Decis Anal 2022;29(5–6):364–80. http://dx.doi.org/10.1002/mcda.1781.

[26] Coquelet B, Dejaegere G, De Smet Y. Analysis of third alternatives' impact on PROMETHEE II ranking. J Multi- Criteria Decis Anal 2024;31(1–2):e1823. http://dx.doi.org/10.1002/mcda.1823.

[27] De Keyser W, Peeters P. A note on the use of PROMETHEE multicriteria methods. European J Oper Res 1996;89(3):457–61. http://dx.doi.org/10.1016/0377-2217(94)00307-6.

[28] Liu X, Liu Y. Sensitivity analysis of the parameters for preference functions and rank reversal analysis in the PROMETHEE II method. Omega 2024;103116. http://dx.doi.org/10.1016/j.omega.2024.103116.

[29] Mareschal B, De Smet Y, Nemery P. Rank reversal in the PROMETHEE II method: Some new results. In: 2008 IEEE international conference on industrial engineering and engineering management. 2008, p. 959–63. http://dx.doi.org/10.1109/IEEM.2008.4738012.

[30] Nemery De Bellevaux P. On the use of multicriteria ranking methods in sorting problems. 2008.

[31] De Smet Y. P2CLUST: An extension of PROMETHEE II for multicriteria ordered clustering. In: 2013 IEEE international conference on industrial engineering and engineering management. 2013, p. 848–51. http://dx.doi.org/10.1109/IEEM.2013.6962531.

[32] De Smet Y, Montano Guzmán L. Towards multicriteria clustering: An extension of the k-means algorithm. European J Oper Res 2004;158(2):390–8. http://dx.doi.org/10.1016/j.ejor.2003.06.012, URL https://www.sciencedirect.com/science/article/pii/S0377221703004910, Methodological Foundations of Multi-Criteria Decision Making.

[33] Corrente S, Figueira JR, Greco S. The SMAA-PROMETHEE method. European J Oper Res 2014;239(2):514–22. http://dx.doi.org/10.1016/j.ejor.2014.05.026, URL https://www.sciencedirect.com/science/article/pii/S0377221714004494.

[34] Dejaegere G, De Smet Y. Promethee γ: A new promethee based method for partial ranking based on valued coalitions of monocriterion net flow scores. J Multi-Criteria Decis Anal 2023;30(3–4):147–60. http://dx.doi.org/10.1002/mcda.1805, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/mcda.1805.

[35] Demir G, Chatterjee P, Pamucar D. Sensitivity analysis in multi-criteria decision making: A state-of-the-art research perspective using bibliometric analysis. Expert Syst Appl 2024;237:121660. http://dx.doi.org/10.1016/j.eswa.2023.121660, URL https://www.sciencedirect.com/science/article/pii/S0957417423021620.

[36] Mareschal B. Weight stability intervals in multicriteria decision aid. European J Oper Res 1988;33(1):54–64. http://dx.doi.org/10.1016/0377-2217(88)90254-8.
[37] Doan NAV, De Smet Y. An alternative weight sensitivity analysis for PROMETHEE II rankings. Omega 2018;80:166–74. http://dx.doi.org/10.1016/j.omega.2017.08.017.
[38] Wolters W, Mareschal B. Novel types of sensitivity analysis for additive MCDM methods. European J Oper Res 1995;81(2):281–90. http://dx.doi.org/10.1016/0377-2217(93)E0343-V, URL https://www.sciencedirect.com/science/article/pii/0377221793E0343V.
[39] Lahdelma R, Hokkanen J, Salminen P. SMAA - stochastic multiobjective acceptability analysis. European J Oper Res 1998;106(1):137–43. http://dx.doi.org/10.1016/S0377-2217(97)00163-X, URL https://www.sciencedirect.com/science/article/pii/S037722179700163X.
[40] Greco S, Słowiński R, Figueira JR, Mousseau V. Robust ordinal regression. In: Ehrgott M, Figueira JR, Greco S, editors. Trends in multiple criteria decision analysis. Boston, MA: Springer US; 2010, p. 241–83. http://dx.doi.org/10.1007/978-1-4419-5904-1_9.
[41] Kadziński M, Greco S, Słowiński R. Extreme ranking analysis in robust ordinal regression. Omega 2012;40(4):488–501. http://dx.doi.org/10.1016/j.omega.2011.09.003, URL https://www.sciencedirect.com/science/article/pii/S030504831100140X.
[42] Kadziński M, Ciomek K, Rychły P, Słowiński R. Post factum analysis for robust multiple criteria ranking and sorting. J Global Optim 2016;65(3):531–62.
[43] Dutta B, Singha T, Goh M, Lamata M-T, Verdegay J-L. Post factum analysis in TOPSIS based decision making method. Expert Syst Appl 2019;138:112806. http://dx.doi.org/10.1016/j.eswa.2019.07.023.
[44] Mousseau V, Özpeynirci Ö, Özpeynirci S. Inverse multiple criteria sorting problem. Ann Oper Res 2018;267(1):379–412. http://dx.doi.org/10.1007/s10479-017-2420-8.
[45] Miller T. Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence 2019;267:1–38. http://dx.doi.org/10.1016/j.artint.2018.07.007, URL https://www.sciencedirect.com/science/article/pii/S0004370218305988.
[46] Guidotti R. Counterfactual explanations and how to find them: literature review and benchmarking. Data Min Knowl Discov 2024;38(5):2770–824. http://dx.doi.org/10.1007/s10618-022-00831-6.
[47] Augusto M, Lisboa J, Yasin M, Figueira JR. Benchmarking in a multiple criteria performance context: An application and a conceptual framework. European J Oper Res 2008;184(1):244–54. http://dx.doi.org/10.1016/j.ejor.2006.10.052, URL https://www.sciencedirect.com/science/article/pii/S0377221706011015.
[48] Hubinont J, De Smet Y. Long-term multi-criteria improvement planning. Decis Support Syst 2021;149:113606. http://dx.doi.org/10.1016/j.dss.2021.113606, URL https://www.sciencedirect.com/science/article/pii/S0167923621001160.
[49] Calders T, Van Assche D. PROMETHEE is not quadratic: An O(qnlog(n)) algorithm. Omega 2018;76:63–9. http://dx.doi.org/10.1016/j.omega.2017.04.003.
[50] Deb K. Multi-objective optimization. In: Burke EK, Kendall G, editors. Search methodologies: introductory tutorials in optimization and decision support techniques. Boston, MA: Springer US; 2014, p. 403–49.
[51] UNDP (United Nations Development Programme). Human development report 2021-22. UNDP (U. N Dev Programme) 2022. URL http://report.hdr.undp.org.
[52] Matsakis ND, Klock II FS. The rust language. In: ACM sIGAda ada letters, vol. 34, no. 3. ACM; 2014, p. 103–4.
[53] Van Rossum G, Drake FL. Python 3 reference manual. Scotts Valley, CA: CreateSpace; 2009.
[54] Plotly Technologies Inc. Collaborative data science. 2015, URL https://plot.ly.
[55] Times Higher Education (THE). World University Rankings. 2021, URL www.timeshighereducation.com/world-university-rankings/2021/world-ranking.