



HAL
open science

A MIP-based approach to learn MR-Sort models with single-peaked preferences

Pegdwendé Minoungou, Vincent Mousseau, Wassila Ouerdane, Paolo Scotton

► **To cite this version:**

Pegdwendé Minoungou, Vincent Mousseau, Wassila Ouerdane, Paolo Scotton. A MIP-based approach to learn MR-Sort models with single-peaked preferences. *Annals of Operations Research*, 2022, 10.1007/s10479-022-05007-5 . hal-03812773

HAL Id: hal-03812773

<https://hal.science/hal-03812773>

Submitted on 28 May 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A MIP-based approach to learn MR-Sort models with single-peaked preferences

Pegdwendé Minoungou^{1,2}, Vincent Mousseau¹, Wassila Ouerdane¹, and Paolo Scotton³

¹ MICS, CentraleSupélec, Université Paris-Saclay, Gif-sur-Yvette, France
{pegdwende.minoungou, vincent.mousseau, wassila.ouerdane}@centralesupelec.fr

² IBM, Saclay, France

³ IBM Research – Zurich, Rüschlikon, Switzerland
psc@zurich.ibm.com

Abstract. The Majority Rule Sorting (MR-Sort) method assigns alternatives evaluated on multiple criteria to one of the predefined ordered categories. The Inverse MR-Sort problem (Inv-MR-Sort) consists in computing MR-Sort parameters that match a dataset. Existing learning algorithms for Inv-MR-Sort consider monotone preference on criteria. We extend this problem to the case where the preference on criteria are not necessarily monotone, but possibly single-peaked (or single-valley). We propose a mixed-integer programming based algorithm that learns from the training data the preference on criteria together with the other MR-Sort parameters. Numerical experiments investigate the performance of the algorithm, and we illustrate its use on a real-world case study.

Keywords: Multicriteria Sorting, MR-Sort, Single-Peaked Preferences, Preference Learning

1 Introduction

In this paper, we consider multiple criteria sorting problems in which alternatives evaluated on several criteria are to be assigned to one of the pre-defined ordered categories C^1, C^2, \dots, C^p , C^1 (C^p , respectively) being the worst (best, respectively) category.

Many multiple criteria methods have been proposed in the literature (see e.g. [12],[40]). We are interested in a pairwise comparison based method: the Non-Compensatory Sorting model (NCS, see [4,5]). NCS assigns alternatives to categories based on the way alternatives compare to boundary profiles representing frontiers between consecutive categories and can be viewed as an axiomatic formulation of the Electre Tri method (see [34]). More specifically, we consider a particular case of NCS in which the importance of criteria is additively represented using weights: the Majority Rule Sorting (MR-Sort, see [26]).

In real-world decision problems involving multiple criteria sorting, the implementation of a sorting model requires eliciting the decision-maker’s (DM) preferences and adequately representing her preferences by setting appropriate values for the preference-related parameters. It is usual to elicit the sorting model parameters indirectly from a set of assignment examples, i.e., a set of alternatives with corresponding desired categories. Such preference learning approach has been developed for MR-Sort (Inv-MR-Sort, see, e.g. [26], [37]), and makes it possible to compute MR-Sort parameters that best fit a learning set provided by the DM.

Such a preference learning approach requires considering criteria involving monotone preferences (criteria to be maximized or minimized). This applies in the context of Multiple Criteria Decision Aid (MCDA), in which the decision problem is structured and carefully crafted through an interaction between the DM and an analyst. In contrast, we are interested in this paper in

application in which the evaluations of alternatives on criteria do not necessarily induce monotone preferences. We illustrate hereafter such a situation in the two illustrative examples.

Example 1: Consider a veterinary problem in cattle production. A new cattle disease should be diagnosed based on symptoms: each cattle should be classified as having or not having the disease. New scientific evidence has indicated that the presence of substance A in the animal’s blood can be predictive in addition to usual symptoms. Still, there is no clue how the level of substance A should be considered. Does a high, a low level, or a level between bounds of substance A indicate sick cattle? The veterinarians’ union has gathered a large number of cases and wants to benefit from this data to define a sorting model based on usual symptom criteria and the level of substance A in the animal’s blood. Hence, the sorting model should be inferred from data, even if the way to account for the substance A level is unknown.

Example 2: A computer-products retail company is distributing a new Windows tablet and wants to send targeted marketing emails to clients who might be interested in this new product. To do so, clients are to be sorted into two categories: *potential buyer* and *not interested*. To avoid spamming, only clients in the former category will receive a telephone call. To sort clients, four clients characteristics are considered as criteria, all of them being homogeneous to a currency e.g. € : the turnover over the last year of (i) Windows PC, (ii) Pack Office, (iii) Linux PC, and (iv) Dual boot PC. As the company advertises a new Windows tablet, both first two criteria are to be maximized (the more a client buys Windows PCs and Pack Office, the more she is interested in products with a Windows system), and the third criterion is to be minimized (the more a client buys Linux PCs, the less he/she is interested in products with a Windows system). The marketing manager is convinced that the last criterion should be considered but does not know if it should be maximized or minimized if preferences are single-peaked; a subset of clients has been partitioned into not interested/potential buyers. Based on this dataset, the goal is to simultaneously learn the classifier parameters and the preference direction for the last criterion.

In the previous examples, it is unclear for the DM how to account for some of the data (level of substance A in blood, Dual boot PC turnover) on the classification of alternatives (cattle, client). These examples correspond to single-peaked criteria, i.e. criteria for which preferences are defined according to a “*peak*” corresponding to the best possible value; on such criteria, the preference decreases with the distance to this peak. In other words, the peak corresponds to a target value below which the criterion is to be maximized, and above which the criterion is to be minimized. Such criteria are frequent in the medical domain (getting close to a normal blood sugar level), chemical applications (get close to a neutral PH), ...

In MCDA, there exist works that account the non-monotonicity of preferences in value-based models (see e.g. [9,24,10], and Section 2). However, there does not exist, to the best of our knowledge, such work concerning pairwise comparisons methods. This paper aims to extend the literature on MCDA for non-monotone criteria to outranking methods and, in particular, to MR-Sort. Specifically, we tackle the problem of inferring from a dataset an MR-Sort with possibly non-monotone criteria. The challenge is that this inference problem is already known to be difficult with monotone criteria, see [26].

More specifically, we assume that evaluations on criteria should be either maximized, minimized or corresponds to single-peaked (or single-valley) preferences. We propose a mixed-integer mathematical programming (MIP) approach to learn the MR-Sort parameters and criteria type (gain, cost, single-peaked, or single-valley) from a dataset of assignment examples.

The paper is organized as follows. Section 2 reviews the existing works in the field of MCDA that consider criteria that are not necessarily monotone. The NCS and MR-Sort methods are presented and extended to the case of single-peaked (single-valley) criteria in section 3. In section 4 we specify the Inv-MR-Sort problem in the presence of single-peaked criteria, and a MIP based algorithm is proposed in Section 5. Section 6 presents the performance of the algorithm on a generated dataset and a real-world case study. The last section groups conclusions and further research issues.

2 Related work

In Multiple Criteria Decision Aid (MCDA), preference learning methods require a preference order on criteria. Such preference order on criteria directly results from the fact that alternatives evaluations/scores correspond to performances that are to be maximized (profit criterion) or minimized (cost criterion), which result in monotone preference data. In multicriteria sorting problems, this boils down to a higher evaluation on a profit criterion (on a cost criterion, respectively) favours an assignment to a higher category (to a lower category, respectively).

However, there are numerous situations in which the criteria evaluation is not related to category assignment in a monotone way. Such a situation is indeed considered in the induction of monotone classification rules from data.

Classification methods in the field of machine learning usually account for attributes (features) that are not supposed to be monotone. Some specialized methods have been proposed to consider monotone feature (see [20], [7]), for decision trees [14], or for decision rules [18]. Some of these approaches have been extended to partially monotone data (see [33], [38]). Błaszczyński et al. in [3] present a non-invasive transformation applied to a dominance-based rough set approach to discover monotonicity relationships (positively/negatively global/local monotonicities) between attributes and the decision considering non-ordinal and ordinal classification problems. With their proposed transformation applied on non-monotone data, they can deduce laws with interval conditions on attributes because they are positively monotone in one part of the evaluation space and negatively monotone in the other.

In the context of the multicriteria decision aid, several preference learning/disaggregation approaches consider non-monotone preferences on criteria. To the best of our knowledge, however, almost all these contributions consider a utility-based preference model, in which non-monotone attributes are represented using non-monotone marginal utility functions.

Historically, Despotis and Zopounidis [9] are the first to consider single peaked value functions with an additive piece-wise linear model. The UTA-NM method proposed in [24] allows for non-monotone marginals and prevents over-fitting by introducing a shape penalization. Also, in the context of an additive utility model, Eckhardt and Klieger [13] define a heuristic pre-processing technique to encode the original non-monotone attributes input into a space monotone; in other words, each alternative x originally described by attribute values (x_1, \dots, x_n) is encoded values $f_1(x_1), \dots, f_n(x_n)$, where $f_i(x_i)$ intuitively corresponds to an “average” of DM’s ratings across objects that have value x_i in attribute i . Another contribution proposed by [10] proposes a heuristic approach to learn non-monotone additive value-based sorting model from data.

Liu *et al.* [27] model sorting with a piece-wise linear additive sorting model, using a regularization framework to limit non-monotonicity. Guo *et al.* [19] propose a progressive preference elicitation for multicriteria sorting using a utility model with non-monotone attributes. A framework to rank alternatives with a utility model using slope variation restrictions for marginals is proposed in [16,17]. Based on a mixed-integer program, [21,22] proposes to disaggregate an additive piece-wise linear sorting model with different types of monotone (increasing, decreasing) and non-monotone (single-peaked, single caved) marginal value functions. Recently some contributions aim at inferring non-compensatory sorting models involving non-monotone criteria from data. Sobrie *et al.* [36] consider a medical application in which some attributes are single-peaked, and duplicates these attributes into two criteria (to be maximized and minimized). Moreover, [30] proposed a heuristic to learn an MR-Sort model and criteria preference directions from data. [36] and [30] are forerunners to the present work, but do not investigate in a systematic way how to learn MR-Sort models from non-monotone data, which justifies the present work.

In this paper, we extend the literature in the following way: we consider non-monotone preferences in the context of an outranking-based sorting model, whereas the literature mainly focuses on additive value-based preference models. We propose a learning-based formulation in which the MR-Sort sorting model and the (possibly non-monotone) structure of preferences on criteria are simultaneously inferred from a set of assignment examples.

3 NCS, MR-Sort, and single-peaked preferences

3.1 NCS: Non-compensatory Sorting

Non-compensatory Sorting (NCS) [4,5] is an MCDA sorting model originating from the ELECTRE TRI method [15]. NCS can be intuitively formulated as follows: an alternative is assigned to a category if (i) it is better than the lower limit of the category on a sufficiently strong subset of criteria, and (ii) this is not the case when comparing the alternative to the upper limit of the category.

Consider the simplest case involving 2 categories Good (\mathcal{G}) and Bad (\mathcal{B}) with the following notations. We denote X_i the finite set of possible values on criterion i , $i \in \mathcal{N} = \{1, \dots, n\}$; we suppose w.l.o.g. that $X_i = [\min_i, \max_i] \subset \mathbb{R}$. Hence, $X = \prod_{i \in \mathcal{N}} X_i$ represents the set of alternatives to be sorted. We denote $\mathcal{A}_i \subseteq X_i$ the set of approved values on criterion $i \in \mathcal{N}$. Approved values on criterion i ($x_i \in \mathcal{A}_i$) correspond to values contributing to the assignment of an alternative to category \mathcal{G} . In order to assign alternative a to category \mathcal{G} , a should have approved values on a subset of criteria which is “sufficiently strong”. The set $\mathcal{F} \subseteq 2^{\mathcal{N}}$ contains the “sufficiently strong” subsets of criteria; it is a subset of $2^{\mathcal{N}}$ up-closed by inclusion. In this perspective, the NCS assignment rule can be expressed as follows:

$$x \in \mathcal{G} \quad \text{iff} \quad \{i \in \mathcal{N} : x_i \in \mathcal{A}_i\} \in \mathcal{F}, \quad \forall x \in X \quad (1)$$

With more than two categories, we consider an ordered set of p categories $C^p \triangleright \dots \triangleright C^h \triangleright \dots \triangleright C^1$, where \triangleright denotes the order on categories. Sets of approved values $\mathcal{A}_i^h \subseteq X_i$ on criterion i ($i \in \mathcal{N}$) are defined with respect to a category h ($h = 2..p$), and should be defined as embedded sets such that $\mathcal{A}_i^2 \supseteq \dots \supseteq \mathcal{A}_i^p$. Analogously, sets of sufficiently strong criteria coalitions are relative to a category h , and are embedded as follows: $\mathcal{F}^2 \supseteq \dots \supseteq \mathcal{F}^p$. The assignment rule is defined bellow, for all $x \in X$, where $\mathcal{A}_i^1 = X_i$, $\mathcal{A}_i^{p+1} = \emptyset$, $\mathcal{F}^1 = \mathcal{P}(\mathcal{N})$, and $\mathcal{F}^{p+1} = \emptyset$.

$$x \in C^h \quad \text{iff} \quad \{i \in \mathcal{N} : x_i \in \mathcal{A}_i^h\} \in \mathcal{F}^h \text{ and } \{i \in \mathcal{N} : x_i \in \mathcal{A}_i^{h+1}\} \notin \mathcal{F}^{h+1} \quad (2)$$

A particular case of NCS corresponds to the MR-Sort rule [26]. When the families of sufficient coalitions are all equal $\mathcal{F}^2 = \dots = \mathcal{F}^p = \mathcal{F}$ and defined using additive weights attached to criteria, and a threshold: $\mathcal{F} = \{F \subseteq \mathcal{N} : \sum_{i \in F} w_i \geq \lambda\}$, with $w_i \geq 0$, $\sum_i w_i = 1$, and $\lambda \in [0, 1]$. Moreover, as the finite set of possible values on criterion i , $X_i = [\min_i, \max_i] \subset \mathbb{R}$, the order on \mathbb{R} induces a complete pre-order \succsim_i on X_i . Hence, the sets of approved values on criterion i , $\mathcal{A}_i^h \subseteq X_i$ ($i \in \mathcal{N}, h = 2 \dots p$) are defined by \succsim_i and $b_i^h \in X_i$ the minimal approved value in X_i at level h : $\mathcal{A}_i^h = \{x_i \in X_i : x_i \succsim_i b_i^h\}$. In this way, $b^h = (b_1^h, \dots, b_n^h)$ is interpreted as the frontier between categories C^{h-1} and C^h ; $b^1 = (\min_1, \dots, \min_n)$ and $b^{p+1} = (\max_1, \dots, \max_n)$ are the lower frontier of C^1 and the upper frontier of C^p . Therefore, the MR-Sort rule can be expressed as:

$$x \in C^h \quad \text{iff} \quad \sum_{i: x_i \geq b_i^h} w_i \geq \lambda \text{ and } \sum_{i: x_i \geq b_i^{h+1}} w_i < \lambda \quad (3)$$

It should be emphasized that in the above definition of the MR-Sort rule, the approved sets \mathcal{A}_i^h can be defined using $b^h \in X$, which are interpreted as frontiers between consecutive categories, only if preferences \succsim_i on criterion i are supposed to be monotone. A criterion can be either defined as a *gain* or a *cost* criterion:

Definition 1. A criterion $i \in \mathcal{N}$ is:

- a *gain criterion*: when $x_i \geq x'_i \Rightarrow x_i \succsim_i x'_i$
- a *cost criterion*: when $x_i \leq x'_i \Rightarrow x_i \succsim_i x'_i$

Indeed, in case of a gain criterion, we have $x_i \in \mathcal{A}_i^h$ and $x'_i \geq x_i \Rightarrow x'_i \in \mathcal{A}_i^h$, and $x_i \notin \mathcal{A}_i^h$ and $x_i > x'_i \Rightarrow x'_i \notin \mathcal{A}_i^h$. Therefore \mathcal{A}_i^h is specified by $b_i^h \in X_i$: $\mathcal{A}_i^h = \{x_i \in X_i : x_i \geq b_i^h\}$. In case of a cost criterion, we have $x_i \in \mathcal{A}_i^h$ and $x'_i \leq x_i \Rightarrow x'_i \in \mathcal{A}_i^h$, and $x_i \notin \mathcal{A}_i^h$ and $x_i < x'_i \Rightarrow x'_i \notin \mathcal{A}_i^h$. Therefore \mathcal{A}_i^h is specified by $b_i \in X_i$: $\mathcal{A}_i^h = \{x_i \in X_i : x_i \leq b_i^h\}$. We study hereafter the MR-Sort rule in the case of single-peaked preferences [2].

3.2 Single-peaked and single-valley preferences

In this paper, we consider preferences that are not necessarily monotone on all criteria.

Definition 2. Preferences \succsim_i on criterion i are:

- *single-peaked preferences with respect to \geq* iff there exist $p_i \in X_i$ such that: $x_i \leq y_i \leq p_i \Rightarrow p_i \succsim_i y_i \succsim_i x_i$, and $p_i \leq x_i \leq y_i \Rightarrow p_i \succsim_i x_i \succsim_i y_i$
- *single-valley preferences with respect to \geq* iff there exist $p_i \in X_i$ such that: $x_i \leq y_i \leq p_i \Rightarrow p_i \succsim_i x_i \succsim_i y_i$, and $p_i \leq x_i \leq y_i \Rightarrow p_i \succsim_i y_i \succsim_i x_i$

In an MCDA perspective, single-peaked preferences (single-valley, respectively) can be interpreted as a gain criterion to be maximized (a cost criterion to be minimized, respectively) below the peak p_i , and as a cost criterion to be minimized (a gain criterion to be maximized, respectively) above the peak p_i . Note also that single-peaked and single-valley preferences embrace the case of gain and cost criteria: a gain criterion corresponds to single-peaked preferences

when $p_i = \max_i$ or single-valley preferences with $p_i = \min_i$, and a cost criterion corresponds to single-peaked preferences when $p_i = \min_i$ or single-valley preferences with $p_i = \max_i$.

When considering MR-Sort with single-peaked criteria, approved sets can not be represented using frontiers between consecutive categories. However, approved sets should be compatible with preferences, i.e. such that:

$$\begin{cases} x_i \in \mathcal{A}_i^h \text{ and } x'_i \succ_i x_i \Rightarrow x'_i \in \mathcal{A}_i^h \\ x_i \notin \mathcal{A}_i^h \text{ and } x_i \succ_i x'_i \Rightarrow x'_i \notin \mathcal{A}_i^h \end{cases} \quad (4)$$

In case of a single-peaked criterion with peak p_i , we have:

$$\begin{cases} x_i \in \mathcal{A}_i^h \text{ and } p_i \leq x'_i \leq x_i \Rightarrow x'_i \in \mathcal{A}_i^h \\ x_i \in \mathcal{A}_i^h \text{ and } x_i \leq x'_i \leq p_i \Rightarrow x'_i \in \mathcal{A}_i^h \\ x_i \notin \mathcal{A}_i^h \text{ and } p_i \leq x_i \leq x'_i \Rightarrow x'_i \notin \mathcal{A}_i^h \\ x_i \notin \mathcal{A}_i^h \text{ and } x'_i \leq x_i \leq p_i \Rightarrow x'_i \notin \mathcal{A}_i^h \end{cases} \quad (5)$$

Therefore it appears that with a single-peaked criterion with peak p_i , the approved sets \mathcal{A}_i^h can be specified by two thresholds $\bar{b}_i^h, \underline{b}_i^h \in X_i$ with $\underline{b}_i^h < p_i < \bar{b}_i^h$ defining an interval of approved values: $\mathcal{A}_i^h = [\underline{b}_i^h, \bar{b}_i^h]$. Analogously, for a single-valley criterion with peak p_i , the approved sets \mathcal{A}_i^h can be specified using $\bar{b}_i^h, \underline{b}_i^h \in X_i$ (such that $\underline{b}_i^h < p_i < \bar{b}_i^h$) as $\mathcal{A}_i^h = X_i \setminus]\underline{b}_i^h, \bar{b}_i^h[$

Given a single-peaked criterion i for which approved set is defined by the interval $\mathcal{A}_i^h = [\underline{b}_i^h, \bar{b}_i^h]$. Consider the function $\phi_i : X_i \rightarrow X_i$ defined by $\phi_i(x_i) = |x_i - \frac{\bar{b}_i^h + \underline{b}_i^h}{2}|$, i.e., the absolute value of $x_i - \frac{\bar{b}_i^h + \underline{b}_i^h}{2}$. Then, the approved set can be conveniently rewritten as : $\mathcal{A}_i^h = \{x_i \in X_i : \phi_i(x_i) \leq \frac{\bar{b}_i^h - \underline{b}_i^h}{2}\}$. In other words, when defining approved sets, a single-peaked criterion can be re-encoded into a cost criterion, evaluating alternatives as the distance to the middle of the interval $[\underline{b}_i^h, \bar{b}_i^h]$, and a frontier corresponding to half the width of this interval.

Analogously, given a single-valley criterion i for which approved set are defined by the interval $\mathcal{A}_i^h = X_i \setminus]\underline{b}_i^h, \bar{b}_i^h[$. Using the same function ϕ_i , approved set can be conveniently rewritten as : $\mathcal{A}_i^h = \{x_i : \phi(x_i) \geq \frac{\bar{b}_i^h - \underline{b}_i^h}{2}\}$. Hence, when defining approved sets, a single-valley criterion can be re-encoded into a gain criterion, evaluating alternatives as the distance to the middle of the interval $[\underline{b}_i^h, \bar{b}_i^h]$, and a frontier corresponding to half the width of this interval.

4 Inv-MR-Sort: Learning an MR-Sort model from assignment examples

MR-Sort preference parameters, e.g. weights, majority level, and limit profiles, can be either initialized by the “end-user”, i.e. the decision-maker, or learned through a set of assignment examples called a learning set. We are focusing on the learning approach. The aim is to find the MR-Sort parameters that “best” fit the learning set.

We consider as input a learning set, denoted L , composed of assignment examples. Here, an *assignment example* refers to an alternative $a \in A^* \subset X$, and a desired category $c(a) \in \{1, \dots, p\}$. In our context, the determination of MR-sort parameters values relies on the resolution of a mathematical program based on assignment examples: the *Inv-MR-Sort* problem

takes as input a learning set L and computes weights $(w_i, i \in \mathcal{N})$, majority level (λ) , and limit profiles $(b_h, h = 2..p)$ that best restore L , i.e. that maximizes the number of correct assignments.

This learning approach – also referred to as preference disaggregation – has been previously considered in the literature. In particular, [31],[39] learned the ELECTRE TRI parameters using mathematical programming formulation (non-linear programming for the former, mixed-integer programming for the latter). In contrast, [11] propose an evolutionary approach to do so. Later, a more amenable model, the MR-Sort – which derives from the ELECTRE TRI method and requires fewer parameters than ELECTRE TRI – was introduced by Leroy et al. in [26]. They proposed a MIP implementation for solving the *Inv-MR-Sort* problem. In contrast, Sobrie et al. [35] tackled it with a metaheuristic, and Belahcene et al. [1] with a Boolean satisfiability (SAT) formulation. Other authors proposed approaches to infer MR-Sort incorporating veto phenomenon [28], and imprecise/missing evaluations [29], and [32] presented an interactive elicitation for the learning of MR-Sort parameters with given profiles values. Recently [23] proposes an enriched preference modelling framework that accounts for a different type of input. Lastly, [30] proposed an extension of Sobrie’s algorithm for solving the *Inv-MR-Sort* problem with latent preference directions, i.e. considering criteria whose preference direction, in terms of gain/cost, is not known beforehand.

In this paper, we aim to extend the resolution of the *Inv-MR-Sort* problem to the case where each criterion can be either a cost criterion, a gain criterion, a single-peaked criterion, or a single-valley criterion.

5 Exact resolution of *Inv-MR-Sort* with single-peaked criteria

In this section, we present a Mixed Integer Programming (MIP) formulation to solve the *Inv-MR-Sort* problem when each criterion can either be a cost, gain, single-peaked, or single-valley criterion. More precisely, the resolution will take as input a learning set containing assignment examples and computes:

- the nature of each criterion (either cost, gain, single-peaked, or single-valley criterion),
- the weights attached to criteria w_i , and an associated majority level λ ,
- the frontier between category C^h and C^{h+1} , i.e. – as defined in Section 3 – the value b_i^h such that if criterion i is a cost or a gain criterion, and the interval $[b_i^h, \bar{b}_i^h]$ if criterion i is a single-peaked or single-valley criterion.

For the sake of simplicity, we describe the mathematical formulation in the case of two categories; the extension to more than two categories is discussed in Section 5.6.

Let us consider a learning set L , provided by the Decision Maker, containing assignment examples corresponding to a set of reference alternatives $A^* = A^{*1} \cup A^{*2}$ partitioned into 2 subsets $A^{*1} = \{a^j \in A^* : a^j \in C^1\}$ and $A^{*2} = \{a^j \in A^* : a^j \in C^2\}$. We denote by J^* , J^{*1} , and J^{*2} the indices j of alternatives contained in A^* , A^{*1} , and A^{*2} , respectively.

In the MIP formulation proposed in this section, we represent single-peaked or single-valley criteria only. As discussed in Section 3.2, this is not restrictive because the cost and gain criteria are particular cases of single-peaked (or single-valley) criteria, with a peak corresponding to the endpoints of the evaluation scale.

5.1 Variables and constraints related to approved sets and profiles

Suppose that criterion i is single-peaked and that the set of approved values is defined by $\mathcal{A}_i = [\underline{b}_i, \bar{b}_i]$. Let us denote $b_i^\perp = \frac{\bar{b}_i + \underline{b}_i}{2}$ the middle of the interval of approved values. Consider an alternative $a^j \in A^*$ in the learning set; its evaluation on criterion i is approved (i.e., $a_i^j \in \mathcal{A}_i$) if $a_i^j \in [\underline{b}_i, \bar{b}_i]$. The condition $|a_i^j - b_i^\perp| \leq \frac{\bar{b}_i - \underline{b}_i}{2}$ guaranties that $a_i^j \in [\underline{b}_i, \bar{b}_i]$. This allows to rewrite the set \mathcal{A}_i as $\mathcal{A}_i = \{x_i \in X_i : |x_i - b_i^\perp| \leq \frac{\bar{b}_i - \underline{b}_i}{2}\}$.

To test whether $a_i^j \in \mathcal{A}_i$, we define $\alpha_i^j = a_i^j - b_i^\perp$ such that $a_i^j \in \mathcal{A}_i \Leftrightarrow |\alpha_i^j| \leq \frac{\bar{b}_i - \underline{b}_i}{2}$. In other words, we re-encode criterion i as a cost criterion representing the distance to b_i^\perp , and accepted values correspond to α_i^j which are lower or equal to $\frac{\bar{b}_i - \underline{b}_i}{2}$ (i.e., the half the interval $[\underline{b}_i, \bar{b}_i]$).

In the following, we denote $b_i = \frac{\bar{b}_i - \underline{b}_i}{2}$. Hence, in our formulation, the sets \mathcal{A}_i are defined using two variables: b_i^\perp representing the middle of the interval $[\underline{b}_i, \bar{b}_i]$, and b_i representing half of the interval $[\underline{b}_i, \bar{b}_i]$ allowing to define \mathcal{A}_i using b_i and b_i^\perp as $\mathcal{A}_i = \{x_i \in X_i : |x_i - b_i^\perp| \leq b_i\}$.

In order to linearize the expression $|\alpha_i^j| = |a_i^j - b_i^\perp|$ in the MIP formulation, we consider two positive variables α_i^{j+} , α_i^{j-} (defined such that $|\alpha_i^j|$ is equal to $\alpha_i^{j+} + \alpha_i^{j-}$) and binary variables β_i^j verifying constraints (6a)-(6c), where M is an arbitrary large positive value. Constraints (6b) and (6c) ensure that at least one variable among α_i^{j+} and α_i^{j-} is null.

$$\alpha_i^j = a_i^j - b_i^\perp = \alpha_i^{j+} - \alpha_i^{j-} \quad (6a)$$

$$0 \leq \alpha_i^{j+} \leq \beta_i^j M \quad (6b)$$

$$0 \leq \alpha_i^{j-} \leq (1 - \beta_i^j) M \quad (6c)$$

Let $\delta_{ij} \in \{0, 1\}$, $i \in \mathcal{N}$, $j \in J^*$, be binary variables expressing the membership of evaluation a_i^j in the approved set \mathcal{A}_i ($\delta_{ij} = 1 \Leftrightarrow a_i^j \in \mathcal{A}_i$). In order to specify constraints defining δ_{ij} , we need to distinguish the case where criterion i is a single-peaked or a single-valley criterion. In the first case, the single-peaked criterion is transformed into a cost criterion and the following constraints hold :

$$\delta_{ij} = 1 \iff |\alpha_i^j| \leq b_i \implies M(\delta_{ij} - 1) \leq b_i - (\alpha_i^{j+} + \alpha_i^{j-}) \quad (7a)$$

$$\delta_{ij} = 0 \iff |\alpha_i^j| > b_i \implies b_i - (\alpha_i^{j+} + \alpha_i^{j-}) < M \delta_{ij} \quad (7b)$$

$$\delta_{ij} \in \{0, 1\} \quad (7c)$$

In the second case, the single-valley criterion is transformed conversely into a gain criterion as follows :

$$\delta_{ij} = 1 \iff |\alpha_i^j| \geq b_i \implies M(\delta_{ij} - 1) \leq (\alpha_i^{j+} + \alpha_i^{j-}) - b_i \quad (8a)$$

$$\delta_{ij} = 0 \iff |\alpha_i^j| < b_i \implies (\alpha_i^{j+} + \alpha_i^{j-}) - b_i < M \delta_{ij} \quad (8b)$$

$$\delta_{ij} \in \{0, 1\} \quad (7c)$$

In order to jointly consider both cases (7a)-(7b) and (8a)-(8b) in the MIP, we introduce a binary variable σ_i , $i \in \mathcal{N}$ which indicates whether criterion i is a single-peaked ($\sigma_i = 1$) or single-valley criterion ($\sigma_i = 0$). When $\sigma_i = 1$, the constraints (9c) and (9d) concerning the single-peaked criteria hold while the constraints (9a) and (9b) for single-valley criteria are relaxed, and conversely when $\sigma_i = 0$.

$$-M\sigma_i + M(\delta_{ij} - 1) \leq \alpha_i^{j+} + \alpha_i^{j-} - b_i \quad (9a)$$

$$\alpha_i^{j+} + \alpha_i^{j-} - b_i < M\delta_{ij} + M\sigma_i \quad (9b)$$

$$M(\sigma_i - 1) + M(\delta_{ij} - 1) \leq b_i - \alpha_i^{j+} - \alpha_i^{j-} \quad (9c)$$

$$b_i - \alpha_i^{j+} - \alpha_i^{j-} < M\delta_{ij} + M(1 - \sigma_i) \quad (9d)$$

$$\delta_{ij} \in \{0, 1\} \quad (7c)$$

$$\sigma_i \in \{0, 1\} \quad (9e)$$

Lastly, in order to restrain the bounds of the single-peaked/single-valley interval within $[\min_i, \max_i]$, we add the 2 following constraints :

$$b_i^\perp - b_i \geq \min_i \quad (10a)$$

$$b_i^\perp + b_i \leq \max_i \quad (10b)$$

5.2 Variables and constraints related to weights

As in [26], we define the continuous variables c_{ij} , $i \in \mathcal{N}$, $j \in J^*$ such that $\delta_{ij} = 0 \Leftrightarrow c_{ij} = 0$ and $\delta_{ij} = 1 \Leftrightarrow c_{ij} = w_i$, where $w_i \geq 0$ represent the weight of criterion i with the normalization constraint: $\sum_{\forall i \in \mathcal{N}} w_i = 1$. To ensure the correct definition of c_{ij} , we impose:

$$c_{ij} \leq \delta_{ij} \quad (11a)$$

$$\delta_{ij} - 1 + w_i \leq c_{ij} \quad (11b)$$

$$c_{ij} \leq w_i \quad (11c)$$

$$0 \leq c_{ij} \quad (11d)$$

5.3 Variables and constraints related to the assignment examples

So as to check whether assignment examples are correctly restored by the MR-Sort rule, we define binary variables $\gamma_j \in \{0, 1\}$, $j \in J^*$ equal to 1 when the alternative a^j is correctly assigned, 0 otherwise. The constraints below guarantees the correct definition of γ_j (where $\lambda \in [0.5, 1]$ represents the MR-Sort majority threshold).

$$\sum_{i \in \mathcal{N}} c_{ij} \geq \lambda + M(\gamma_j - 1), \forall j \in J^{*2} \quad (12a)$$

$$\sum_{i \in \mathcal{N}} c_{ij} < \lambda - M(\gamma_j - 1), \forall j \in J^{*1} \quad (12b)$$

5.4 Objective function

The objective for the *Inv-MR-Sort* problem is to identify the MR-Sort model which best matches the learning set. Therefore, in order to maximize the number of correctly restored assignment examples, the objective function can be formulated as: $Max \sum_{j \in J^*} \gamma_j$

Finally, the MIP formulation for the *Inv-MR-Sort* problem with single-peaked and single valley criteria is given bellow (where M an arbitrary large positive value, and ε an arbitrary small positive value). Table 1 synthesizes the variables involved in this mathematical program.

Variable	Domain	Number of variables	Definition
α_i^{j+}	\mathbb{R}^+	$n \times A^* $	first component of the absolute value $ a_i^j - b_i^\perp $
α_i^{j-}	\mathbb{R}^+	$n \times A^* $	second component of the absolute value of $ a_i^j - b_i^\perp $
β_i^j	$\{0, 1\}$	$n \times A^* $	binary variable indicating the sign of $a_i^j - b_i^\perp$
σ_i	$\{0, 1\}$	n	$\sigma_i = 1$ if criterion i is single-peaked, $\sigma_i = 0$ if i is single-valley
γ_j	$\{0, 1\}$	$ A^* $	$\gamma_j = 1$ if alternative a^j is correctly assigned by the model, $\gamma_j = 0$ if not
δ_{ij}	$\{0, 1\}$	$n \times A^* $	$\delta_{ij} = 1$ if $a_i^j \in \mathcal{A}_i$, $\delta_{ij} = 0$ if $a_i^j \notin \mathcal{A}_i$
c_{ij}	$[0, 1]$	$n \times A^* $	$c_{ij} = 1$ if $a_i^j \in \mathcal{A}_i$ (i.e, if $\delta_{ij} = 1$), $c_{ij} = 0$ if $a_i^j \notin \mathcal{A}_i$ (i.e, if $\delta_{ij} = 0$)
b_i^\perp	\mathbb{R}	n	middle of the interval $[b_i, \bar{b}_i]$
b_i	\mathbb{R}	n	value of half the width of the interval $[b_i, \bar{b}_i]$ on criterion i
w_i	$[0, 1]$	n	weight of criterion i
λ	$[0, 1]$	1	majority threshold

Table 1: Description of decision variables

$$\max \sum_{j \in J^*} \gamma_j \quad (13a)$$

$$\sum_{i \in \mathcal{N}} c_{ij} \geq \lambda + M(\gamma_j - 1) \quad \forall j \in J^{*2} \quad (12a)$$

$$\sum_{i \in \mathcal{N}} c_{ij} + \varepsilon \leq \lambda - M(\gamma_j - 1) \quad \forall j \in J^{*1} \quad (12b)$$

$$\sum_{i \in \mathcal{N}} w_i = 1 \quad (13b)$$

$$c_{ij} \leq \delta_{ij} \quad \forall j \in J^*, \forall i \in \mathcal{N} \quad (11a)$$

$$c_{ij} \geq \delta_{ij} - 1 + w_i \quad \forall j \in J^*, \forall i \in \mathcal{N} \quad (11b)$$

$$c_{ij} \leq w_i \quad \forall j \in J^*, \forall i \in \mathcal{N} \quad (11c)$$

$$b_i^\perp - a_i^j = \alpha_i^{j+} - \alpha_i^{j-} \quad \forall j \in J^*, \forall i \in \mathcal{N} \quad (6a)$$

$$\alpha_i^{j+} \leq \beta_i^j M \quad \forall j \in J^*, \forall i \in \mathcal{N} \quad (6b)$$

$$\alpha_i^{j-} \leq (1 - \beta_i^j) M \quad \forall j \in J^*, \forall i \in \mathcal{N} \quad (6c)$$

$$-M \cdot \sigma_i + M(\delta_{ij} - 1) \leq \alpha_i^{j+} + \alpha_i^{j-} - b_i \quad \forall j \in J^*, \forall i \in \mathcal{N} \quad (9a)$$

$$\alpha_i^{j+} + \alpha_i^{j-} - b_i + \varepsilon \leq M \cdot \delta_{ij} + M \cdot \sigma_i \quad \forall j \in J^*, \forall i \in \mathcal{N} \quad (9b)$$

$$M \cdot (\sigma_i - 1) + M(\delta_{ij} - 1) \leq b_i - \alpha_i^{j+} - \alpha_i^{j-} \quad \forall j \in J^*, \forall i \in \mathcal{N} \quad (9c)$$

$$b_i - \alpha_i^{j+} - \alpha_i^{j-} + \varepsilon \leq M \cdot \delta_{ij} + M \cdot (1 - \sigma_i) \quad \forall j \in J^*, \forall i \in \mathcal{N} \quad (9d)$$

$$b_i^\perp - b_i \geq \min_i \quad \forall i \in \mathcal{N} \quad (10a)$$

$$b_i^\perp + b_i \leq \max_i \quad \forall i \in \mathcal{N} \quad (10b)$$

$$c_{ij} \in [0, 1], \delta_{ij} \in \{0, 1\} \quad \forall j \in J^*, \forall i \in \mathcal{N} \quad (13c)$$

$$\alpha_i^{j+}, \alpha_i^{j-} \in \mathbb{R}^+ \quad \forall j \in J^*, \forall i \in \mathcal{N} \quad (13d)$$

$$\beta_i^j \in [0, 1] \quad \forall j \in J^*, \forall i \in \mathcal{N} \quad (13e)$$

$$b_i \in \mathbb{R}, w_i \in [0, 1], b_i^\perp \in \mathbb{R}, \sigma_i \in \{0, 1\} \quad \forall i \in \mathcal{N} \quad (13f)$$

$$\gamma_j \in \{0, 1\} \quad \forall j \in J^* \quad (13g)$$

$$\lambda \in [0.5, 1] \quad (13h)$$

5.5 Interpretation of the optimal solution

Once the optimal solution to the above mathematical program is found, it is necessary to derive, from the optimal solution, the corresponding MR-Sort model, i.e:

- the nature of each criterion (either cost, gain, single-peaked, or single-valley criterion),
- the weights attached to criteria w_i , and associated majority level λ ,
- the frontier between category C^1 and C^2 , i.e., the value b_i if criterion i is a cost or a gain criterion, and the interval $[b_i, \bar{b}_i]$ if criterion i is a single-peaked or single-valley criterion.

Criteria weights w_i , and associated majority level λ are directly obtained from the corresponding variables in the optimal solution. The preference directions and criteria limit profiles are deduced as follows:

- Case $\sigma_i = 1$ (criterion i is represented as a single-peaked criterion in the optimal solution):
 - if $b_i^\perp - b_i \leq \min_{j \in J^*} \{a_i^j\}$, then criterion i is a cost criterion, and the maximal approved value on criterion i is $b_i^\perp + b_i$, i.e. $\mathcal{A}_i =]-\infty, b_i^\perp + b_i]$, see Fig. 1 case 3,
 - if $b_i^\perp + b_i \geq \max_{j \in J^*} \{a_i^j\}$, then criterion i is a gain criterion, and the minimal approved value on criterion i is $b_i^\perp - b_i$, i.e. $\mathcal{A}_i = [b_i^\perp - b_i, \infty[$, see Fig. 1 case 2,
 - otherwise, i is a single-peaked criterion, and $\mathcal{A}_i = [b_i^\perp - b_i, b_i^\perp + b_i]$, see Fig. 1 case 1
- Case $\sigma_i = 0$ (criterion i is represented as a single-valley criterion in the optimal solution):
 - if $b_i^\perp - b_i < \min_{j \in J^*} \{a_i^j\}$, then criterion i is a gain criterion, and the minimal approved value on criterion i is $b_i^\perp + b_i$, i.e. $\mathcal{A}_i = [b_i^\perp + b_i, \infty[$, see Fig. 2 case 3,
 - if $b_i^\perp + b_i > \max_{j \in J^*} \{a_i^j\}$, then criterion i is a cost criterion, and the maximal approved value on criterion i is $b_i^\perp - b_i$, i.e. $\mathcal{A}_i =]-\infty, b_i^\perp - b_i]$, see Fig. 2 case 2,
 - otherwise, i is a single-valley criterion, and $\mathcal{A}_i =]-\infty, b_i^\perp - b_i] \cup [b_i^\perp + b_i, \infty[$, see Fig. 2 case 1.

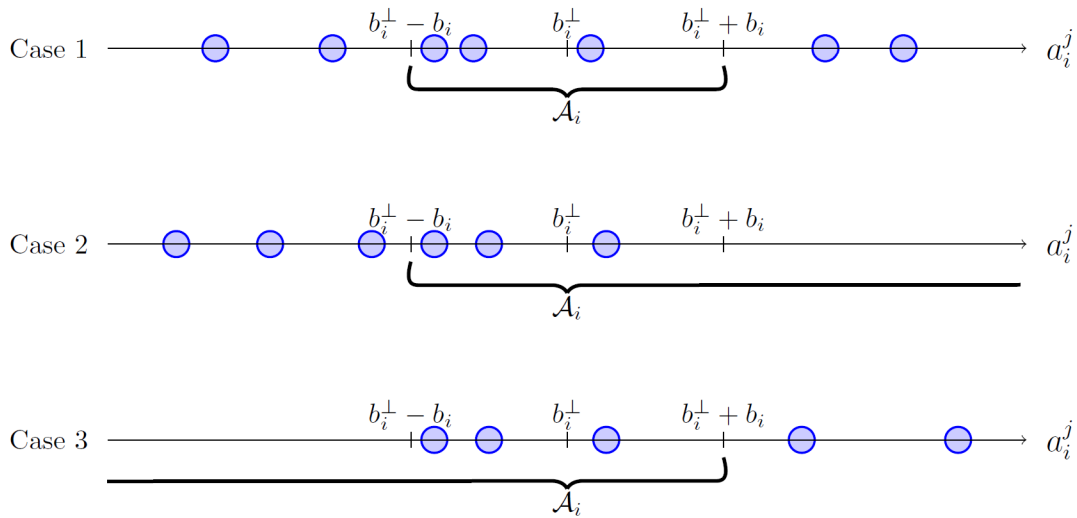


Fig. 1: Three cases for single-peaked criteria

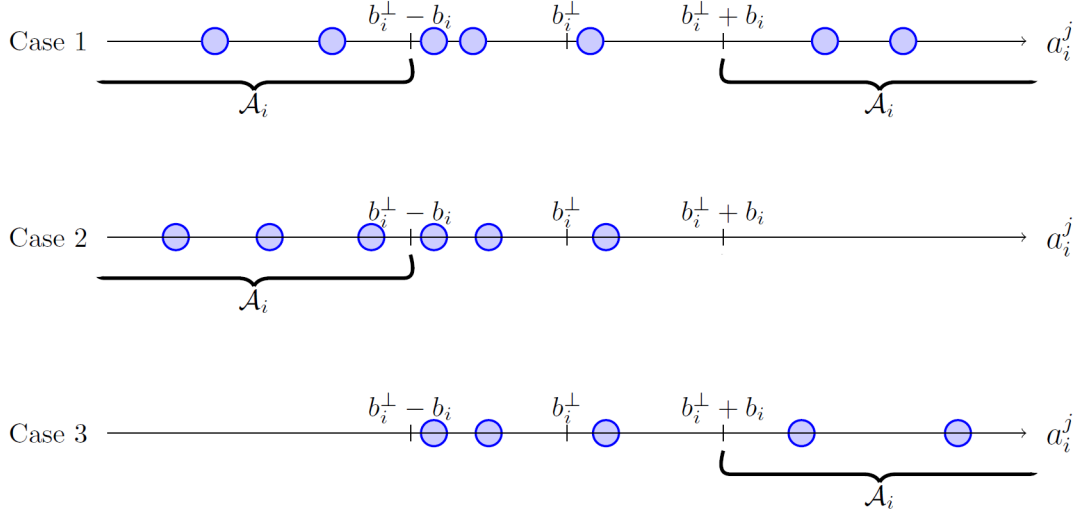


Fig. 2: Three cases for single-valley criteria

5.6 Extension to more than 2 categories

Our framework can be extended to more than two categories, at the cost of adding supplementary variables and constraints to the mathematical program. So as to extend to p categories ($p > 2$), sets of approved values $\mathcal{A}_i^h \subseteq X_i$ on criterion i ($i \in \mathcal{N}$) should be defined with respect to a category level h ($h = 2, \dots, p$), and should be embedded such that $\mathcal{A}_i^p \subseteq \mathcal{A}_i^{p-1} \subseteq \dots \subseteq \mathcal{A}_i^2$.

In the MIP formulation, the variables δ_{ij} , c_{ij} , α_i^{j+} , α_i^{j-} , β_i^j , b_i , and b_i^\perp should be indexed with a category level $h = 2..p$, and become δ_{ij}^h , c_{ij}^h , α_i^{jh+} , α_i^{jh-} , β_i^{jh} , b_i^h , and $b_i^{h\perp}$, respectively. Constraints (12a) and (12b) relative to the assignment examples should be replaced by the following ones:

- $\sum_{i \in \mathcal{N}} c_{ij}^{p-1} \geq \lambda + M(\gamma_j - 1)$, $\forall a_j \in C^p$
- $\sum_{i \in \mathcal{N}} c_{ij}^1 + \varepsilon \leq \lambda - M(\gamma_j - 1)$, $\forall a_j \in C^1$
- $\sum_{i \in \mathcal{N}} c_{ij}^{h-1} \geq \lambda + M(\gamma_j - 1)$, $\forall a_j \in C^h \subset [C^2, C^{p-1}]$
- $\sum_{i \in \mathcal{N}} c_{ij}^h + \varepsilon \leq \lambda - M(\gamma_j - 1)$, $\forall a_j \in C^h \subset [C^2, C^{p-1}]$

Lastly, constraints on b_i^h , and $b_i^{h\perp}$ should be imposed so as to guaranty that the approved sets are embedded such that $\mathcal{A}_i^{p-1} \subseteq \mathcal{A}_i^{p-2} \subseteq \dots \subseteq \mathcal{A}_i^1$, i.e., $[b_i^{p-1\perp} - b_i^{p-1}, b_i^{p-1\perp} + b_i^{p-1}] \subseteq [b_i^{p-2\perp} - b_i^{p-2}, b_i^{p-2\perp} + b_i^{p-2}] \subseteq \dots \subseteq [b_i^{1\perp} - b_i^1, b_i^{1\perp} + b_i^1]$.

6 Experiments, results and discussion

In this section, we report numerical experiments to empirically study how the proposed algorithm behaves in terms of computing time, ability to generalize, and ability to restore an MR-Sort model with the correct preference direction (gain, cost, single-peaked, or single-valley). The experimental study involves artificially generated datasets and ex-post analysis of a real-world case study.

6.1 Tests on generated datasets

In this section we are focusing on synthetic data.

6.1.1 Experimental design

Assuming a generated MR-Sort model \mathcal{M}^0 representing perfectly the Decision Maker preferences, we first randomly generate n -tuples of values considered as alternatives (each tuple corresponding to n criteria evaluations). Then we simulate the assignments of these alternatives following the model \mathcal{M}^0 and obtain, therefore, assignment examples which constitute the learning set L , used as input to our MIP algorithm. Alternatives are generated in such a manner to obtain a balanced dataset (i.e. equal number of assignments in each category). The Inv-MR-Sort problem is then solved using the proposed algorithm and, as a result, generating a learned model noted \mathcal{M}' .

Generation of instances and model parameters. We consider a learning set of 200 assignment examples. A vector of performance values of alternatives is drawn in an independent and identically distributed manner, such that the performance values are contained in the unit interval discretized by tenths. We then randomly generate profiles values (either b_i , or \underline{b}_i and \bar{b}_i) for each criterion; also these values are chosen with the unit interval discretized by tenths. In order to draw uniformly distributed weights vectors (see [6]), we uniformly generate $|\mathcal{N}| - 1$ random values in $[0, 1]$ sorted in ascending order. We then prepend 0 and append 1 to this set of values obtaining a sorted set of $|\mathcal{N}| + 1$ values. Finally, we compute the difference between each successive pair of values resulting in a set of $|\mathcal{N}|$ weights such that their sum is equal to 1. We randomly draw λ in $[0,1]$.

In order to assess the ability of the algorithm to restore preference directions, we consider q criteria out of n for which we consider the preference direction as unknown, and we uniformly draw a random preference direction among gain, cost, single-peaked and single-valley. For each single-peaked and single-valley criteria, the peak is uniformly drawn in the $[0,1]$ interval discretized by tenths⁴. Hence, the preference direction of these q criteria are assumed to be unknown. Meanwhile, the remaining $n - q$ criteria are considered as gain criteria.

Performance metrics and tests parameters. To study the performance of the proposed algorithm, we are considering three main metrics.

- *Computing time*: we consider here the time (CPU) necessary to solve the MIP algorithm.
- *Restoration rate of assignment examples*: as our MIP algorithm is an exact method, it is expected that the entire learning set L will be restored by \mathcal{M}' . Therefore we assess the restoration performance on a test set which is run through \mathcal{M}^0 and \mathcal{M}' . This test set comprises randomly generated alternatives not used in the learning set; that is, assignment examples that the algorithm has never seen. This allows to assess the restoration rate (also called classification accuracy in generalization or *CAG*), that is, the ratio between the number of alternatives identically assigned in categories by both \mathcal{M}^0 and \mathcal{M}' , and the number of alternatives.
- *Preference direction restoration rate (PDR)*: considering the set of criteria where the preference direction is unknown, PDR is defined as the ratio between the number of criteria where the preference direction has been correctly restored and the cardinality of this set.

⁴ It should be noted that if the peak is drawn as an extreme value, the single-peaked (or single-valley) criterion actually corresponds to a monotone (gain or cost) criterion.

In order to account for the statistical distribution of all the randomly selected values, we independently select 100 different learning sets, each one associated with a randomly generated \mathcal{M}^0 MR-Sort model. We then ran 100 independent experiments and aggregated the results.

In our experiments, we vary n the number of criteria in $\{4, 5, 6, 7, 8, 9\}$; q the number of criteria with unknown preference directions vary in $\{0, 1, 2, 3, 4\}$, and the number of categories is set to 2. The test set is composed of 10000 randomly generated alternatives.

We executed experiments on a server endowed with an Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz, 80 cores and 384 GB RAM. The tests were performed using the Cplex solver version 20.1.0 [8] on the server using 10 reserved threads and limiting the computing time to a maximum of 1h.

6.1.2 Results

In the following we present the results of the randomly generated tests.

unknown dir. (q)	Number of criteria (n)					
	4	5	6	7	8	9
0	0.34s / 0.65s 100%	0.56s / 1.55s 100%	0.84s / 2.42s 100%	2.38s / 5.18s 100%	2.61s / 8.92s 100%	3.37s / 11.37s 100%
1	1.51s / 3.96s 100%	3.23s / 11.47s 100%	4.53s / 21.14s 100%	7.22s / 35.49s 100%	19.15s / 118.1s / 100%	18.68s / 342.9s 91%
2	6.12s / 38.87s 100%	12.97s / 57.7s 100%	30.1s / 147.6s 94%	54.4s / 430.6s 90%	43.19 / >1h 72%	58.0s / >1h 59%
3	37.48s / 272.7s 95%	76.68s / >1h 89%	72.46s / >1h 80%	76.3s / >1h 54%	59.32s / >1h 47%	25.91s / >1h 31%
4	96.34s / >1h 59%	129.49 / >1h 57%	61.01 / >1h 52%	108.25 / >1h 42%	22.17 / >1h 27%	23.63 / >1h 28%

Table 2: Median CPU time (sec.) of instances solved in 1h / 9th decile of CPU time, and proportion of terminated instances, with 4 to 9 crit. (n), 0 to 4 crit. with unknown pref. dir. (q)

Computing time Table 2 presents the median CPU time of the terminated instances (timeout set to 1 hour). The execution time increases with the number of criteria and the number of criteria with unknown preference direction up to $n = 7$ and $q = 2$. Beyond this limit, the execution time fluctuates, and we observe a relatively large dispersion of CPU times. For instance, when $n = 7$ and $q = 3$, the median value of CPU time is just above 1 minute (76.3 sec.), while the 90th percentile value exceeds 1 hour.

Additionally, Table 2 shows the percentage of instances that terminated within the time limit, set to 1 hour. Unsurprisingly, the number of terminated instances decreases with both the number of criteria and the number of criteria with unknown preference direction. In particular, the rate jumps from 95% with 4 criteria to 31% with 9 criteria in the model when $q = 3$.

Restoration rate of the test set. Regarding the classification accuracy (CAg) of the learned models (involving 4 to 9 criteria in the model and 0 to 4 criteria with unknown preference direction), globally, the performance values are between 0.9 and 0.95, with 0.93 on average. We do not notice a significant trend over both the number of criteria and the number of criteria with unknown preference directions. However, the figures reflect the performance of only terminated instances. Therefore, the CAg rate could degrade when considering executions above the timeout, assuming these are the most difficult instances to learn.

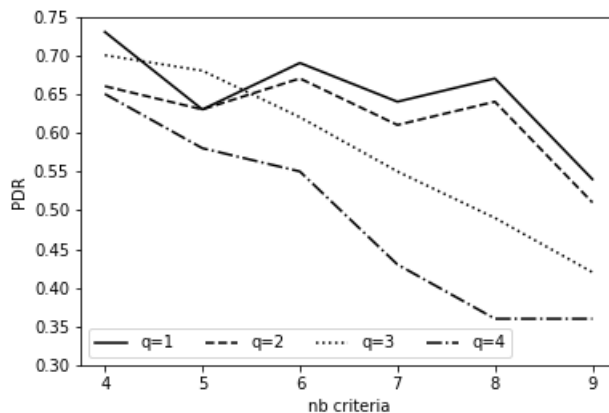


Fig. 3: Preference direction restoration rate (PDR) considering 1 to 4 criteria with unknown preference direction (q) (average performance over terminated instances)

	$w_1 \leq \frac{1}{2n}$	$\frac{1}{2n} < w_1 < \frac{2}{n}$	$w_1 \geq \frac{2}{n}$
PDR	0.44	0.74	0.78

Table 3: PDR (averaged over n) according to the range of weight of criterion c_1

Preference direction restoration rate. Figure 3 illustrates the evolution of the preference direction restoration rate (PDR). Globally, the PDR falls with the increase of the number of criteria in the model. In addition, this indicator degrades moderately with the number of criteria with unknown preference directions with respectively 55% and 35% for $q = 1$ and $q = 4$ considering 9 criteria in the model.

The results illustrated in the Table 3 give more insights of the behaviour of the algorithm regarding PDR. We consider instances involving one criterion with unknown preference direction, $q = 1$ (it corresponds to criterion 1). We analyze how the importance of this criterion (w_1) on the restoration rate of the preference direction. The PDR rate is averaged over the number of criteria in the model ($n \in \{4, \dots, 9\}$) and distributed over three intervals: $[0, \frac{1}{2n}]$, $\frac{1}{2n}, \frac{2}{n}[$, $[\frac{2}{n}, 1]$ which can be interpreted as three levels of importance of w_1 (respectively low level, medium level, and high level). As expected, the average PDR rises with the importance of w_1 ; we have 44% of PDR for a low level of importance whereas 74% and 78% correspond respectively to

a medium and high level of the importance of w_1 . It appears that the MIP has more difficulties in correctly detecting the preference direction of a criterion when this criterion has low importance.

6.1.3 Discussion

The experiments carried out on randomly generated instances give us the following insights.

Although exact methods are typically computationally intensive, the computation time is relatively affordable for medium-sized models (less than 3 minutes for 200 alternatives in the learning set and up to $n = 9$ and $q = 4$ in the model when the timeout is set to 1 hour). Moreover, The computation time could be reduced as our experiments were performed with a limited number of threads set to 10.

The algorithm can restore accurately new assignment examples based on the learned models (0.93 on average up to 9 criteria) and remains relatively efficient with the number of criteria with unknown preference directions. Extended experiments should be done without the limit of time to accurately predict the restoration rate in generalization with the increase of parameters n and q .

Our algorithm restores with difficulty preference directions when the number of criteria grows while keeping the learning set constant. The PDR rate also decreases with the increase of the number of criteria with unknown preference direction in the model with similar learning set sizes (but still greater than the random choice that is 25%). It would be instructive to discover the algorithm’s behaviour in terms of PDR for non-terminated instances for more insights.

Finally, the restoration rate of criteria preference direction correlates with such criteria’s importance in the model. It appears that the preference direction of criteria with importance below $\frac{1}{2n}$ are the most difficult to restore. These results are valid with a learning set of fixed size (200); It would be interesting to investigate experimentally whether larger learning sets would make it possible to accurately learn the direction of preference.

6.2 Tests on a real-world data: the ASA dataset

The ASA⁵ dataset [25] constitutes a list of 898 patients evaluated on 14 medical indicators (see Table 4) enabling to assign patients into 4 ordered categories (ASA1, ASA2, ASA3, ASA4). These categories correspond to 4 different scores that indicate the patient health. Based on the score obtained for a given patient, anesthesiologists decide whether or not to admit such a patient to surgery. The relevance of the dataset for our tests relies on the presence of a criterion with single-peaked preference, which is “Blood glucose level” (i.e. glycemia). For practicality, we restrain the ASA dataset to the 8 most relevant criteria for our experiments. They are in bold in Table 4

To have two categories, we first divide the dataset into two parts: *category 2* representing patients in categories ASA1 and ASA2 (67% of the population) and *category 1* representing those in categories ASA3 and ASA4 (33% of the population).

In the following, we illustrate how to learn the model parameters and the preference type (gain, cost, single-peaked (SP), single-valley) of the criterion “Glycemia” with three different

⁵ ASA stands for “American Society of Anesthesiologists”.

sets of assignment examples chosen in the original set of 898 patients. In this medical application, we suppose that the ‘‘Glycemia’’ criterion type is unknown and expect to ‘‘discover’’ a single-peaked criterion. We report for each experiment the number of distinct performances considered per criterion.

Attribute	Domain (Unit)	Direction
Age	[0 – 105] (year)	min.
Diabetic	{0,1}	min.
Hypertension	{0,1}	min.
Respiratory failure	{0,1}	min.
Heart failure	{0,1}	min.
Heart rate	[55 – 123] (bpm)	SP
Heart rate steadiness	{0,1}	max.
Pacemaker	{0,1}	min.
Atrioventricular block	{0,1}	min.
Left ventricular hypertrophy	{0,1}	min.
Oxygen saturation	[43 – 100](%)	max.
Blood glucose level (glycemia)	[0.5 – 3.8](g/l)	SP
Systolic blood pressure	[9 – 20.5](cmHg)	min.
Diastolic blood pressure	[5 – 13](cmHg)	min.

Table 4: Original criteria in the ASA dataset

First Dataset: we initially consider the whole original dataset with all 898 assignment examples in the learning set as input to our MIP algorithm. We infer the type (gain, cost, single-peaked, single-valley) of criterion Glycemia and the MR-Sort parameters from this first dataset.

The inferred model given in Table 5 is computed in 40h33mn execution time. The obtained model allows restoring $CA = 99.4\%$ of the assignment examples in the learning set. However, in the inferred model, the glycemia criterion is detected as a cost criterion to be minimized (whereas we expect it to be inferred as single-peaked). Note that the inferred value for the limit profile on the glycemia criterion (1.18 g/l) makes it possible to distinguish patients with hyperglycemia from the others but does not distinguish hypoglycemia from normal glycemia (normal glycemia corresponds to [0.9,1.2]). This is due to the distribution of the glycemia values over the patients shown in Figure 4. This distribution shows that all patients with glycemia above 1.2g/l (hyperglycemia) are assigned to Category 1. However, some patients with normal glycemia [0.9,1.2] are also assigned to Category 1, and some patients with glycemia equal to 0.8 g/l or below (hypoglycemia) are assigned to Category 2.

In the following, we check if it is possible to restore the ‘‘correct’’ preference direction (i.e. single-peaked) with a subset of carefully selected patients. To do so, we will remove the patients with normal glycemia ([0.9, 1.2] g/l) assigned to category 1.

Second Dataset : in a second step, we choose to remove the 97 patients of the learning set assigned to Category 1 and whose glycemia values lie within [0.9, 1.2] g/l, i.e., with normal glycemia. Our goal is to foster the algorithm to retrieve a single-peaked preference for the glycemia criterion. The distribution of glycemia values in the new learning set of the remaining 801 patients is provided in Fig. 4.

Using this second learning set, we solve the inference problem with the MIP algorithm. Computation time is 56mn, and the inferred model (see Table 6) restores 99.8% of the learning

Attributes	Instance settings			Model parameters learned		
	#values	Direction	pref. dir.	b_i	w_i	pref. dir.
Age	103 (origin)	min.	known	72.9	0.01	-
Diabetic	2 (origin)	min.	known	0.99	0	-
Hypertension	2 (origin)	min.	known	0	0.01	-
Respiratory F	2 (origin)	min.	known	0.99	0.88	-
Pacemaker	2 (origin)	min.	known	0	0.02	-
Systolic BP	24 (origin)	min.	known	15	0.03	-
Diastolic BP	17 (origin)	min.	known	8.92	0.02	-
Glycemia	82 (origin)	SP	unknown	1.18	0.03	<i>min</i>
				$\lambda =$	0.98	

Table 5: Inferred model with the first dataset (898 assignment examples)

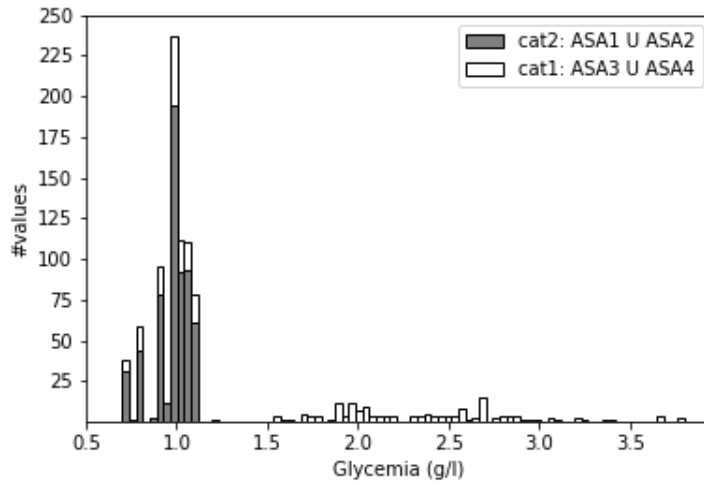


Fig. 4: Distribution of patients glycemia in the first dataset

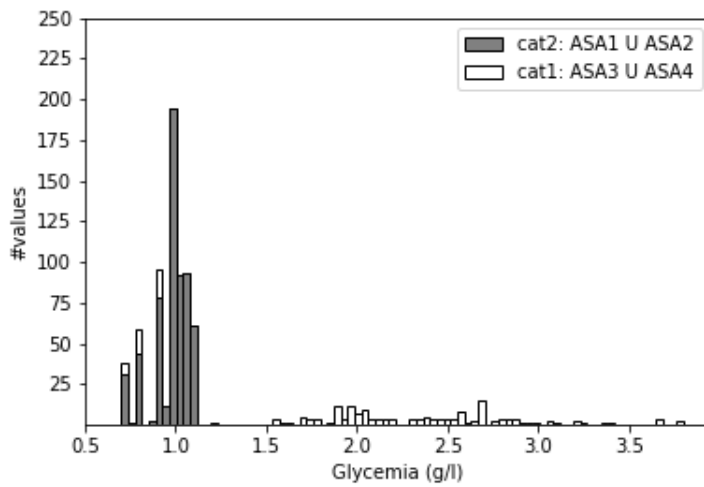


Fig. 5: Distribution of patients glycemia in the second dataset

set. Once again, the restoration rate is high. However, the glycemia criterion is still detected as a cost criterion to be minimized (instead of a single-peaked criterion). The inferred model does not distinguish patients with hypoglycemia from normal glycemia ones.

Attributes	Instance settings			Model parameters learned		
	#values	Direction	pref. dir.	b_i	w_i	pref. dir.
Age	103 (origin)	min.	known	5.9	0	-
Diabetic	2 (origin)	min.	known	0.99	0	-
Hypertension	2 (origin)	min.	known	0	0.01	-
Respiratory F	2 (origin)	min.	known	0	0.01	-
Pacemaker	2 (origin)	min.	known	-0.01	0	-
Systolic BP	23	min.	known	15	0.01	-
Diastolic BP	15	min.	known	8.5	0.01	-
Glycemia	82 (origin)	SP	unknown	1.18	0.96	<i>min</i>
				$\lambda =$	0.99	

Table 6: Inferred model with the second dataset (801 assignment examples)

Third Dataset : Finally, we remove patients in Category 2 for which the glycemia value is lower than 0.9 (hypoglycemia). This new configuration leads to a dataset of 624 patients. In this third dataset, the distribution of glycemia values (see Figure 6) in which hypo and hyperglycemic patients are assigned to Category 1 while patients with normal glycemia are in Category 2.

With this dataset, the MIP algorithm runs in 4mn30s and results are presented in Table 7. The computed model restores all the assignment examples, and glycemia is now detected as a single-peaked criterion. Furthermore, the approved values [0.93, 1.18] can be reasonably interpreted as normal glycemia.

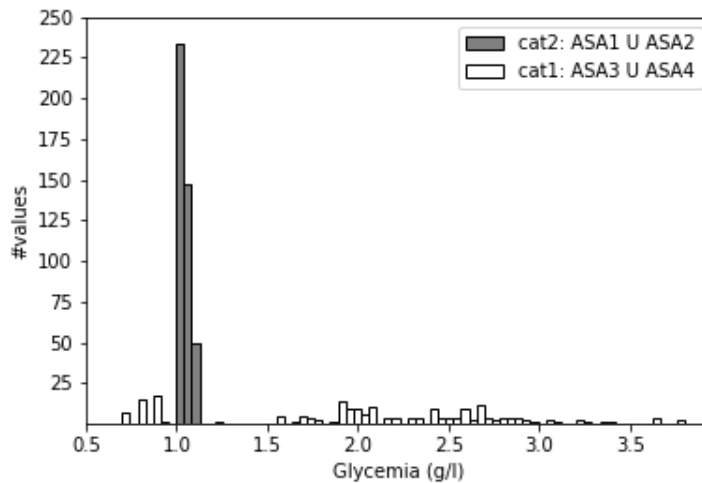


Fig. 6: Patients glycemia in the third dataset

Attributes	Instance settings			Model parameters learned		
	#values	Direction	pref. dir.	b_i	w_i	pref. dir.
Age	103 (origin)	min.	known	3.3	0	-
Diabetic	2 (origin)	min.	known	0	0	-
Hypertension	2 (origin)	min.	known	0	0	-
Respiratory F	2 (origin)	min.	known	0.99	0	-
Pacemaker	2 (origin)	min.	known	0	0	-
Systolic BP	23	min.	known	12.88	0.01	-
Diastolic BP	15	min.	known	9	0.01	-
Glycemia	73 (origin)	SP	unknown	[0.93,1.18]	0.99	SP
				$\lambda =$	1	

Table 7: Inferred model with the third dataset (624 assignment examples)

This illustrative example shows that our model can infer an MR-Sort model and retrieve single-peaked criteria; however, to do so, the learning set should be sufficiently informative. Specifically, when inferring from a dataset a “ground truth” in which a specific criterion i is single-peaked with a set of acceptable values $\mathcal{A}_i = [\underline{b}_i, \overline{b}_i]$, it is necessary that some examples in the learning set are evaluated on criterion i below \underline{b}_i , in $[\underline{b}_i, \overline{b}_i]$, and above \overline{b}_i .

7 Conclusion and future work

This paper proposes a MIP-based method to infer an MR-Sort model from a set of assignment examples when considering possibly non-monotone preferences. More precisely we learn an Mr-Sort model with criteria that can be either of type (i) cost, (ii) gain, (iii) single-peaked or (iv) single-valley criteria. Our inference procedure simultaneously infers from the dataset an MR-Sort model and the type of each criterion.

Our experimental test on simulated data shows that the MIP resolution can handle a dataset involving 200 examples and nine criteria. Experiments suggest that the correct restoration of the criteria type (i)-(iv) requires a dataset of significant size.

Our work opens avenues for further research. First, it would be interesting to test our methodology on real-world case studies to assess further and investigate our proposal’s performance and applicability. Another research direction aims at pushing back computational barriers: our MIP resolution approach faces a combinatorial explosion. The design of an efficient heuristic would be beneficial in this respect.

References

1. K. Belahcene, C. Labreuche, N. Maudet, V. Mousseau, and W. Ouerdane, ‘An efficient SAT formulation for learning multiple criteria non-compensatory sorting rules from examples’, *Computers & Operations Research*, **97**, 58–71, (2018).
2. D. Black, ‘On the rationale of group decision-making’, *Journal of Political Economy*, **56**(1), 23–34, (1948).
3. J. Blaszczynski, S. Greco, and R. Slowinski, ‘Inductive discovery of laws using monotonic rules’, *Engineering Applications of Artificial Intelligence*, **25**(2), 284–294, (2012).
4. D. Bouyssou and T. Marchant, ‘An axiomatic approach to noncompensatory sorting methods in mcdm, i: The case of two categories’, *European Journal of Operational Research*, **178**, 217–245, (February 2007).
5. D. Bouyssou and T. Marchant, ‘An axiomatic approach to noncompensatory sorting methods in MCDM, II: More than two categories’, *European Journal of Operational Research*, **178**(1), 246–276, (2007).

6. J. Butler, J. Jia, and J. Dyer, ‘Simulation techniques for the sensitivity analysis of multi-criteria decision models’, *European Journal of Operational Research*, **103**(3), 531–546, (1997).
7. J.-R. Cano, P.A. Gutierrez, B. Krawczyk, M. Wozniak, and S. Garcia, ‘Monotonic classification: An overview on algorithms, performance measures and data sets’, *Neurocomputing*, **341**, 168–182, (2019).
8. IBM ILOG Cplex, *IBM ILOG CPLEX Optimization Studio CPLEX User’s Manual, Version 12, Release 8*, IBM ILOG, 20.1.0 edn., 2017.
9. D. K. Despotis and C. Zopounidis, *Building Additive Utilities in the Presence of Non-Monotonic Preferences*, 101–114, Springer, 1995.
10. M. Doumpos, ‘Learning non-monotonic additive value functions for multicriteria decision making’, *OR Spectrum*, **34**(1), 89–106, (2012).
11. M. Doumpos, Y. Marinakis, M. Marinaki, and C. Zopounidis, ‘An evolutionary approach to construction of outranking models for multicriteria classification: The case of the ELECTRE TRI method’, *European Journal of Operational Research*, **199**(2), 496–505, (2009).
12. M. Doumpos and C. Zopounidis, *Multicriteria Decision Aid Classification Methods*, Kluwer Academic Publishers, Dordrecht, 2002.
13. A. Eckhardt and T. Kliegr, ‘Preprocessing algorithm for handling non-monotone attributes in the UTA method’, in *Proceedings of the ECAI-12 Workshop on Preference Learning: Problems and Applications in AI (PL-12)*, eds., J. Fürnkranz and E. Hüllermeier, (2012).
14. A. Feelders, ‘Monotone relabeling in ordinal classification’, in *ICDM 2010, The 10th IEEE International Conference on Data Mining*, eds., G.I. Webb, B. Liu, C. Zhang, D. Gunopulos, and X. Wu, pp. 803–808. IEEE Computer Society, (2010).
15. J. Figueira, V. Mousseau, and B. Roy, ‘Electre methods’, in *Multiple criteria decision analysis: State of the art surveys*, 133–153, Springer, (2005).
16. M. Ghaderi, F. Ruiz, and N. Agell, ‘Understanding the impact of brand colour on brand image: A preference disaggregation approach’, *Pattern recognition letters*, **67**, 11–18, (2015).
17. M. Ghaderi, F. Ruiz, and N. Agell, ‘A linear programming approach for learning non-monotonic additive value functions in multiple criteria decision aiding’, *European Journal of Operational Research*, **259**(3), 1073 – 1084, (2017).
18. S. Greco, B. Matarazzo, and R. Slowinski, ‘Rough sets theory for multicriteria decision analysis’, *European Journal of Operational Research*, **129**(1), 1–47, (2001).
19. M. Guo, X. Liao, and J. Liu, ‘A progressive sorting approach for multiple criteria decision aiding in the presence of non-monotonic preferences’, *Expert Systems with Applications*, **123**, 1 – 17, (2019).
20. P.A. Gutiérrez and S. García, ‘Current prospects on ordinal and monotonic classification’, *Prog. Artif. Intell.*, **5**(3), 171–179, (2016).
21. M. Kadzinski, K. Martyn, M. Cinelli, R. Slowinski, S. Corrente, and S. Greco, ‘Preference disaggregation for multiple criteria sorting with partial monotonicity constraints: Application to exposure management of nanomaterials’, *International Journal of Approximate Reasoning*, **117**, 60–80, (2020).
22. M. Kadzinski, K. Martyn, M. Cinelli, R. Slowinski, S. Corrente, and S. Greco, ‘Preference disaggregation method for value-based multi-decision sorting problems with a real-world application in nanotechnology’, *Knowledge-Based Systems*, **218**, (2021).
23. M. Kadzinski and M. Martyn, ‘Enriched preference modeling and robustness analysis for the ELECTRE Tri-B method’, *Annals of Operations Research*, 1–35, (2020).
24. T. Kliegr, ‘UTA-NM : Explaining stated preferences with additive non-monotonic utility functions’, in *Proceedings of ECML PKDD Workshop on Preference Learning*, (2009).
25. M.A. Lazouni, M.A. Chikh, and S. Mahmoudi, ‘A new computer aided diagnosis system for pre-anesthesia consultation’, *Journal of Medical Imaging and Health Informatics*, **3**(4), 471–479, (2013).
26. A. Leroy, V. Mousseau, and M. Pirlot, ‘Learning the parameters of a multiple criteria sorting method’, in *International Conference on Algorithmic Decision Theory*, pp. 219–233. Springer, (2011).
27. J. Liu, X. Liao, M. Kadzinski, and R. Slowinski, ‘Preference disaggregation within the regularization framework for sorting problems with multiple potentially non-monotonic criteria’, *European Journal of Operational Research*, **276**(3), 1071–1089, (2019).
28. P. Meyer and A. Olteanu, ‘Integrating large positive and negative performance differences into multicriteria majority-rule sorting models’, *Computers and Operations Research*, **81**, 216–230, (2017).
29. P. Meyer and A. Olteanu, ‘Handling imprecise and missing evaluations in multi-criteria majority-rule sorting’, *Computers and Operations Research*, **110**, 135–147, (2019).
30. P. Minoungou, V. Mousseau, W. Ouerdane, and P. Scotton, ‘Learning an MR-sort model from data with latent criteria preference direction’, in *DA2PL’2020, from multiple criteria Decision Aid to Preference Learning*, Trento, Italy, (2020).
31. V. Mousseau and R. Slowinski, ‘Inferring an ELECTRE TRI model from assignment examples’, *Journal of global optimization*, **12**(2), 157–174, (1998). Springer.

32. O. Nefla, M. Öztürk, P. Viappiani, and I. Brigui-Chtioui, ‘Interactive elicitation of a majority rule sorting model with maximum margin optimization’, in *ADT 2019, 6th International Conference on Algorithmic Decision Theory*, (2019).
33. S. Pei and Q. Hu, ‘Partially monotonic decision trees’, *Information Sciences*, **424**, 104–117, (2018).
34. B. Roy, ‘The outranking approach and the foundations of Electre methods’, *Theory and Decision*, **31**(1), 49–73, (1991).
35. O. Sobrie, *Learning preferences with multiple-criteria models*, Ph.D. dissertation, Université de Mons (Faculté Polytechnique) and Université Paris-Saclay (CentraleSupélec), 2016.
36. O. Sobrie, M.A. Lazouni, S. Mahmoudi, V. Mousseau, and M. Pirlot, ‘A new decision support model for preanesthetic evaluation’, *Computer Methods and Programs in Biomedicine*, **133**, 183–193, (2016).
37. O. Sobrie, V. Mousseau, and M. Pirlot, ‘Learning monotone preferences using a majority rule sorting model’, *Int. Trans. Oper. Res.*, **26**(5), 1786–1809, (2019).
38. H. Wang, M. Zhou, and K. She, ‘Induction of ordinal classification rules from decision tables with unknown monotonicity’, *European Journal of Operational Research*, **242**(1), 172–181, (2015).
39. J. Zheng, S. Metchebon Takougang, V. Mousseau, and M. Pirlot, ‘Learning criteria weights of an optimistic Electre Tri sorting rule’, *Computers and Operations Research*, **49**, 28 – 40, (2014).
40. C. Zopounidis and M. Doumpos, ‘Multicriteria classification and sorting methods: A literature review’, *European Journal of Operational Research*, **138**(2), 229–246, (2002).