



Poznan University of Technology  
Faculty of Computing and Telecommunications

Doctoral dissertation

Multiple Criteria Decision Analysis methods  
inspired by other sub-disciplines of artificial  
intelligence

**Krzysztof Martyn**

Supervisor: Miłosz Kadziński, Ph.D., Habil.

Poznań, 2023



# Abstract

Systems that assist humans in processing and analyzing information are becoming increasingly popular. Over the years, various areas of artificial intelligence have developed many tools and techniques for this purpose. In this context, Multiple Criteria Decision Analysis (MCDA) provides decision support tools that are highly interpretable, ensuring their recommendations are believable and trustworthy. Nevertheless, to deal with new, more complex problems, incorporating techniques and inspiration from different fields may be essential. This dissertation presents MCDA methods that combine ideas from various AI sub-disciplines. Firstly, there was developed a framework for preference learning algorithms. It infers the parameters of MCDA-inspired models through interpretable artificial neural networks. They are suitable for handling vast, inconsistent preference information. Moreover, incorporating ideas from machine learning, two approaches were employed for modeling non-monotonic marginal value functions within a preference disaggregation framework. One method allows controlling the complexity and interpretability of the inferred model by minimizing the number of changes in monotonicity. The other elucidates the non-monotonic shape as a combination of non-decreasing and non-increasing components. Furthermore, following the multi-label classification problem, an additive value function model was proposed for the newly formulated problem of multiple interrelated decision sorting. Then, novel exploitation methods of preference and outranking relations were developed. They analyze the strength and weaknesses of alternatives using algorithms inspired by website scoring. The scores can be enhanced by the Decision Makers' holistic judgments in the form of subsets of options considered comprehensively strong or weak. The practical usefulness of the proposed methods was demonstrated in real-world problems such as risk management in nanomanufacturing processes and the evaluation of special economic zones or technological parks.





# List of publications

The dissertation consists of the introductory section and the following five original publications:

- [P1] M. Kadziński, K. Martyn, M. Cinelli, R. Słowiński, 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, DOI: 10.1016/j.ijar.2019.11.007.

Number of citations<sup>1</sup>:

- according to Web of Science: 27
- according to Google Scholar: 31

- [P2] M. Kadziński, K. Martyn, M. Cinelli, R. Słowiński, 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*, 218:106879, 2021, DOI: 10.1016/j.knosys.2021.106879.

Number of citations<sup>1</sup>:

- according to Web of Science: 9
- according to Google Scholar: 11

- [P3] K. Martyn and M. Kadziński. Deep preference learning for multiple criteria decision analysis. *European Journal of Operational Research*, 305(2):781–805, 2023, DOI: 10.1016/j.ejor.2022.06.053.

Number of citations<sup>1</sup>:

- according to Web of Science: 5
- according to Google Scholar: 6

---

<sup>1</sup>as on June 1, 2023

- [P4] K. Martyn, M. Martyn, and M. Kadziński. PrefRank: a family of multiple criteria decision analysis methods for exploiting a valued preference relation. *Expert Systems With Applications*, 2023. Submitted.
- [P5] K. Martyn, M. Martyn, and M. Kadziński. ScoreBin: scoring alternatives based on a crisp outranking relation with an application to the assessment of Polish technological parks. *Information Sciences*, 2023. Submitted.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Multiple Criteria Decision Analysis</b>	<b>11</b>
2.1	General scheme of decision aiding process . . . . .	11
2.2	Notation . . . . .	12
2.3	General assumptions . . . . .	13
2.4	Preference information . . . . .	13
2.5	Robustness analysis . . . . .	14
2.6	Reminder on UTADIS method . . . . .	14
2.7	Reminder on PROMETHEE methods . . . . .	15
2.8	Reminder on ELECTRE methods . . . . .	16
2.9	Exploitation methods for valued and crisp relations . . . . .	17
<b>3</b>	<b>Artificial Intelligence</b>	<b>19</b>
3.1	Machine learning . . . . .	19
3.2	Artificial neural networks . . . . .	21
3.3	Preference learning . . . . .	22
3.4	Web mining . . . . .	23
<b>4</b>	<b>Method inspired by deep learning and preference learning</b>	<b>25</b>
4.1	ANN-UTADIS: Preference learning with UTADIS and ANN . . . . .	27
4.2	Illustration of preference models inferred with neural networks . . . . .	29
4.3	Computational experiments . . . . .	29
4.4	Summary . . . . .	32
<b>5</b>	<b>Methods inspired by machine learning</b>	<b>33</b>
5.1	Introduction . . . . .	33
5.2	Non-monotonic criteria . . . . .	35
5.3	Multi-decision sorting problems . . . . .	38
5.4	Results of experiments . . . . .	39
5.5	Comparison of the proposed models . . . . .	45

<b>6</b>	<b>Methods inspired by web mining</b>	<b>47</b>
6.1	PrefRank . . . . .	48
6.1.1	Measures used for comparing the choice or ranking recommendations	50
6.1.2	Experimental comparison between PROMETHEE and PrefRank .	51
6.1.3	Case study concerning evaluation of special economic zones . . . . .	51
6.2	ScoreBin . . . . .	52
6.2.1	Measures used for comparing the choice or ranking recommendations	55
6.2.2	Experimental comparison of results attained by different methods exploiting a crisp outranking relation . . . . .	55
6.2.3	Case study concerning evaluation of technological parks in Poland	56
6.2.4	Robustness analysis . . . . .	57
6.3	Comparison of PrefRank and ScoreBin methods . . . . .	59
<b>7</b>	<b>Summary</b>	<b>61</b>
	<b>Bibliography</b>	<b>67</b>
	<b>Publication reprints</b>	<b>77</b>
	<b>Extended abstract in Polish</b>	<b>217</b>
	<b>Declarations</b>	<b>229</b>

# Chapter 1

## Introduction

Decision problems are situations in which a Decision Maker (DM) needs to decide on a set of alternatives by considering their performances on a set of criteria. Typically, no alternative performs best on all criteria since the latter represents different points of view on the quality of considered options. As a result, there is a set of potentially best solutions, and it depends only on the DM's preferences which of them would be judged the most suitable. It is why the DM needs to provide preference information that reflects his/her value system. The main aim of intelligent decision support systems is to incorporate the elements of such a system and suggest a recommendation that would be consistent with them.

Artificial Intelligence (AI) is a discipline that focuses on creating computer systems and programs able to simulate the cognitive capabilities of humans as learning, understanding, pattern recognition, problem-solving, planning, or decision-making. It covers many sub-disciplines, including:

- Machine Learning (ML) dealing with algorithms and techniques enabling learning based on data and making automated predictions. In ML, we can distinguish deep learning with its primary focus on deep neural networks, which can detect and process complex patterns in data and perform sophisticated tasks.
- Multi-Criteria Decision Aiding (MCDA) developing methods and tools which can support the process of decision making while there exist many criteria and alternatives to consider.
- Natural Language Processing (NLP) which focuses on advancing techniques and models for analyzing, understanding, and generating natural language.
- Recommender systems dealing with algorithms and techniques for personalized recommendations based on a vast set of possible actions. They take into account preference analysis, behaviors, and characteristics of users.

- Web mining focussed on extracting, analyzing, and using information from internet resources like websites, forums, or social media, exploring their structure and users' behaviors.

Many of these sub-disciplines are tightly connected, and different methods can be assigned to a few. This dissertation focuses on methods and problems in the intersection of MCDA with other sub-disciplines of AI.

MCDA methods are inspired by real-world decision-making. They provide recommendations systematically, consistently, and objectively, taking into account only these features of alternatives that are crucial for the DM due to employing his/her preferences. The preference information given by DM may be provided directly, e.g., defining the importance of criteria or model parameters, or indirectly as exemplary decisions made on a subset of reference alternatives or part of the final solution. Additionally, it may refer to the nature of criteria, interactions between them, and the direction of preference of performances on criteria.

We can distinguish three main types of decision problems [37], [7]:

- choice focused on choosing a subset of the most preferred options;
- ranking where alternatives are put in order from the best to the worst;
- sorting (ordinal classification) aiming to assign alternatives to predefined classes that reflect the DM's preference level.

Different sub-disciplines of AI solve similar problems. For instance, web mining considers a ranking of websites based on the connections between them. In recommender systems, one needs to create a ranking of items and suggest a few that are most relevant for the user. Moreover, in ML, one considers both ranking problems as regression or object assignment to predefined classes or the subset of the most relevant labels for the considered item. Multi-label classification problem refers to the issue of making multiple decisions about whether a considered label is relevant or not [106]. Regarding these different sub-disciplines of AI, preference information is usually given indirectly, the reference set is considered as a training set and a way of searching for models' parameters is called supervised learning.

MCDA and machine learning are two sub-disciplines of artificial intelligence that support people in decision-making. Both provide tools and techniques for analyzing different alternatives and recommend to DM solutions relevant to the considered decision problem. However, these two fields' main goals, assumptions, methods, and possibilities differ.

Decision-aiding is based on the dialogue between the DM and an analyst. The former provides information representing his/her value system and preferences. On these

premises, the method builds an analytical model and elaborates recommendations. The latter can be accepted by DM or rejected, leading to reconsidering preference information and possibly their change. In turn, the analyst's task is to properly select a method of acquiring preference information and processing data to ensure that the chosen model reflects the DM's mindset and the characteristics of a tackled problem in the best way.

Over the years, several different preference models were proposed to aggregate alternatives' performances on various criteria. Three main families of methods can be identified:

- methods based on pairwise comparisons in the form of preference or outranking relations between alternatives [97];
- scoring methods assessing options in a holistic way using value-utility and distance-based procedure [10];
- approaches based on sets of "if ... then ..." decision rules [36].

In this doctoral dissertation, we consider only the first two of them. The exemplary methods using these models are more extensively discussed in Chapter 2.

When it comes to machine learning, it provides techniques for pattern recognition in data and making correct predictions even for deeply complex problems. These models process only the data while concluding, and there is typically no need to provide any additional information regarding the problem or criteria. ML is widely used and can be applied to nearly any learning task, e.g., in medical diagnosis [56], recommender systems [86] data analysis, control systems [104], speech and pattern recognition.

Both MCDA and ML may be employed to solve decision problems. However, significant differences between them can be highlighted [12], [102], [20]. Firstly, MCDA entirely focuses on the user, his/her knowledge, and preferences. Solving a problem depends on the DM's preferences. Through their exploitation, methods can reveal his/her priorities. Conversely, ML is mainly directed to models, with the main interest in data mining, data analysis, and pattern recognition. The fundamental goal of ML is to solve problems by optimizing one specific feature, e.g., minimizing a loss function. The diverse objectives are manifested by including different models, the size of the problems considered, techniques employed, and the role of users.

Preference models which are used in MCDA are inspired by actual approaches to decision-making by humans. The desire to incorporate such aspects leads to the ease of interpretability and explainability of these methods. It is a particularly essential attribute since intelligent systems are implemented in various applications (fields). Decision-making problems appear frequently in security, medicine, or natural environment preservation, where improper decisions may cause significant damage. The possibility of its

usage mainly depends on the fact that their predictions and recommendations are credible and trustworthy.

Interpretability of the model states for the ease of understanding the model by humans and cause of a decision made by the model [79], [80]. It may also signify the degree of confidence in how people can predict the behavior of algorithms [82]. The feasibility of the model's interpretability enables DM to verify if the model takes into account proper aspects of the problem and if it is coherent with the information and constraints that are given by DM [7]. The method should explain the specific decision by giving the influence of each score that characterizes the alternative and the existing dependencies between criteria or options. Explanations help improve the transparency and interpretability of the models. Explanations help users, stakeholders, and decision-makers understand the factors and features that influenced the model's decision by providing insights into how a model arrived at a particular prediction. It helps DM better understand his/her preferences and decisions [12]. Moreover, it delivers justifications and knowledge that enable DM to take an active role in the decision process. Explanations enhance trust and confidence in artificial intelligence models. When users can understand and validate the reasoning behind a decision, they are more likely to trust and accept the model's outputs. Explanations also allow users to identify potential biases, errors, or limitations in the model, promoting more informed decision-making.

In the context of decision support systems, the crucial attribute is simulatability, i.e., the ease of correct prediction of algorithm output by DM [82]. It characterizes MCDA methods as they try to reflect the DM's reasoning.

It is in opposition to machine learning methods which focus on extracting essential but, at the same time, some abstract patterns and dependencies in data. This is possible thanks to approximations of complex, highly non-linear transformations that enable solving intricate problems. However, it causes that most methods are black-box-type methods whose interpretability is highly restrained [102]. In this case, both decision explanation and interpretation may be the general approximation of the inner logic and influence of the specific criteria [80].

One of the critical features of the model is its efficiency in reproducing preference information. If the model has low quality of preference information reconstruction, the correctness of the conclusions obtained from the analysis and interpretation of the model will be unlikely [82]. Due to this fact, the acquired solution should respect all constraints given by DM and consider complete preference information. In case of any inconsistencies detection, there should be delivered knowledge which part from the preference information was not reconstructed.

Gathering preference information in a direct dialogue with DM implied that the traditional MCDA methods were designed to learn from small data sets [12]. This information reflects the real value system of DM, thus typically being cohesive. On the contrary, ML



methods were always adapted to cope with huge training data sets with noise and inconsistencies [20]. Data is a set of historical decisions made over some time or aggregate decisions made by many DMs. In order to process them, some advanced statistical and optimization techniques are used, which help to search a parameters space effectively to find the best-fitted model. Due to that, these models can scale efficiently with the increase of alternatives number [102].

Over the years, both MCDA and ML focused on independently developing methods that address the above problems. Nonetheless, recently there has been a great rise in collected data and decisions which need to be analyzed automatically and then interpreted and justified. It led to development research derived from both disciplines, called preference learning [29]. Such methods enable simple scaling with increasing preference information while simultaneously providing the possibility of model interpretation. As part of this discipline, some interpretable ML methods were adapted to decision problems [7], including Rank-SVM [46] or decision trees with monotonicity constraints. Moreover, some MCDA methods were adapted to cope with a great amount of data. The main assumptions of preference learning are described in Chapter 3.3.

Currently, for big data, using precise algorithms is associated with a long time of calculations, which is impractical. It causes the necessity of using heuristics or various techniques to enable the solution's scalability with increasing training data. Accordingly, some proposed methods include heuristics using evolutionary algorithms [19], linear programming models combined with simulated annealing [84], or dedicated metaheuristic [96] to estimate parameters models values.

In the last years, much effort has been made to optimize and reduce the calculation time of AI models. Great attention was assigned to the artificial neural network (ANN) models. Usually, to get high accuracy, they require an enormous amount of training data. However, since they run many similar operations, they can be easily parallelized and implemented on dedicated processing units such as GPU and TPU, which helps to reduce the calculation time significantly. Some techniques, like distributed learning, enable solving problems that do not fit one device. Additionally, many frameworks and optimization techniques appeared that facilitate the effective learning process of models for complex and inconsistent problems [15].

Another aspect is that decision-making is often connected not only with scoring in terms of the quality of each option but also may concern scoring in the context of how favorably a considered alternative compares to others. Usually, DM does not consider a precise utility function and rate alternatives compared to others with the usage of some heuristics. They assume the iterative consideration pairs of options and choose the better one which goes to the next round. This choice may be executed by the selection of an alternative that is better on most of the criteria [2], or rejection based on the first cue that discriminates them starting from the most important criterion [34]. One proposed

a plethora of MCDA methods in the spirit of pairwise comparisons and exploration of the relations in the considered pair of alternatives [6], [23], [71], [52]. The examples of these methods are families of ELECTRE [23] and PROMETHEE [6]. The first one examines if there is strong enough evidence for the assertion “ $a_i$  is at least as good or more preferred to  $a_k$ ” and if there are not vital enough premises against it. The latter focuses on the degree of preference for one option over the other. The considered relation may vary depending on the specific method and can refer to the preference of one alternative over the other, outranking, or indifference. The result from comparison may be a crisp (binary) value, which says only about the existence of the relation, or valued (fuzzy) with information about the degree of relation intensity.

Nevertheless, disclosing the relation between all pairs of alternatives is usually not a decision problem itself but only a step in the decision process. Therefore, the information given as a pairwise comparison matrix or a preference graph must be exploited to gather ranking or recommendation of the most preferred options. These techniques may focus on sorting alternatives in terms of being more preferred and choosing the option that is preferred most often [5]. The other variant of ranking construction is to consider the strengths and weaknesses of all options [6] or iteratively with the usage of descending and ascending distillation procedures [90]. The recommendation of best alternatives may take an option on the top of the ranking or a set of alternatives that are in the kernel of an outranking graph [91].

The ranking creation and choice from available options basing on the dependencies between options also appear in the context of web mining. It can be characterized as extracting valuable and meaningful information from the World Wide Web. It involves analyzing large volumes of web data such as web pages, hyperlinks, and user behavior to discover patterns, trends, and insights that can be used for various purposes. One is page ranking used by search engines and recommender systems. The goal is to provide users with the most relevant and helpful information by distinguishing valuable and trustworthy sites from spam and low-quality pages. For this purpose, a quality score is used by the users and moderators, and the graph of connections between websites [39]. Also, more valuable pages include links to other valuable ones and rarely to suspicious websites [85]. It can also be considered as portals, mainly providers of content and pages that aggregate from many sources and are gates (intermediary), hubs from which it is possible to go to many other websites [58].

With general development, new challenges and decision problems appear more complex and require tools and methods for their resolution. The advantages and possibilities of different sub-disciplines of AI were an inspiration for this doctoral dissertation. It considers different research fields, such as ML, deep learning, and web mining. Among the accomplished works, three papers have been published, and two others have been submitted for publication. The applicability and usefulness of each proposed method were

demonstrated on real-world problems (use cases). These works refer to three research areas:

### 1. MCDA methods inspired by deep learning

Traditional MCDA methods require directly defining the model's parameters or describing a small set of example reference alternatives holistically evaluated by the DM. In this situation, preference disaggregation is run using mathematical programming. In case of any inconsistencies in preference information, the main aim is to reflect the user's preferences as well as possible. However, if the data set of reference options is vast and involves many inconsistencies, such an approach may struggle with reproducing real DM's value system [70]. Additionally, methods based on mathematical programming experience scalability issues with an increase in the size of the reference set.

The second issue is that in some methods, even if the DM provides some indirect preference information concerning the evaluation of alternatives, (s)he still needs to define the shape of a function that transforms alternatives scores to partial preference or outranking degrees.

In the last years, neural network models were learned on training data sets that were bigger and bigger to learn more detailed and representative features, leading to better generalization and more accurate prediction. As a result, many sophisticated optimization techniques were proposed based on gradient learning methods, effectively enabling network weights adjustments to training data. Additionally, in order to shorten the processing time, they can use parallel and distributed computational techniques which enable scaling both vertically and horizontally.

In this doctoral dissertation, a general schema of implementation of MCDA-inspired approaches with the usage of neural networks is proposed. These models solve sorting problems by preference learning from large reference data sets. Moreover, in the case of methods that use additive aggregation functions and methods which aggregate pairwise comparisons, the proposed solutions let to model any monotonic shape of these functions. The essential feature is that the proposed architectures of neural networks enable straightforward interpretation of the model and explanation of a recommended decision. The accuracy and usability of these models were presented via experiments on the ten benchmark data sets used in preference learning.

### 2. MCDA methods inspired by machine learning

Machine learning methods can explore and approximate very complex transformations of input data. Thus, traditional MCDA methods focus on monotonic transformations and experience challenges if the direction of preference is unknown. Hitherto approaches to cope with non-monotonic conversions of criteria values produce complicated solutions; hence their interpretation is intricate.

It was a motivation for this doctoral dissertation to propose methods of criteria modeling for which the preference characteristic is limited or unknown a priori:

- an approach which minimizes the complexity of the function by minimizing the number of monotonicity directions changes for non-monotonic criteria with defined A-shaped, V-shaped, or any other shape,
- an approach which models non-monotonic criteria as a compound of two components, non-decreasing and non-increasing, which enables easier interpretability of the model.

The second inspiration from machine learning is multi-label classification problems. The essence of this issue is that many decisions must be made simultaneously by assigning to the object each label or not. In the case of decision-making, it is common that there is a situation when there are many interdependent decisions to be made or states which decision is more or less adequate to the considered scenario. Consequently, within this doctoral dissertation, a new type of decision problem is defined as a sorting problem with many interdependent decisions. We proposed a suitable method for these challenges and tested it on the risk management problem during nanomaterials production.

### 3. MCDA methods inspired by web mining

In web mining methods, the ranking of the quality of websites mainly depends on the links in the WWW graph. Additionally, the analysis may be extended by defining whether the website is credible, significantly influencing the final position in the ranking.

Methods based on pairwise comparisons provide information on the relation type between alternatives. In order to get the recommendation of the most preferred options or their ranking, it is necessary to aggregate this information. Current methods for exploiting such a relation treat all options equally important. An option that outranks an extremely weak alternative is as important as one that outranks an alternative noticeably strong. On the other hand, these methods do not let control of the final results by DM. In most methods, to get other solutions, the change needs to be introduced at the earlier stage to get a different relation

graph.

The above observations motivated the creation of two new families of methods for exploiting preference matrices: PrefRank for weighted preference relation and ScoreBin for crisp outranking relation. These methods consider dependencies in the graph, weaknesses and strengths of alternatives, and whether the option is easy to outrank. The proposed approaches were tested experimentally in terms of their similarity to other exploitation algorithms. Their application is described in real-world use cases as the ranking of special economic zones in Poland for PrefRank and technological parks in Poland for ScoreBin.

The remainder of this doctoral dissertation is organized in the following way. Chapter 2 consists of the principles of decision aiding process and preference modeling. In particular, it includes the main assumptions and description of selected MCDA methods considered in this doctoral dissertation. Chapter 3 focuses on chosen techniques and aspects from other sub-disciplines in AI, especially machine learning, deep learning, preference learning, and web mining. Chapters 4–6 describe in detail the research areas considered in this doctoral dissertation by introducing MCDA methods inspired by other sub-disciplines of AI. Chapter 7 summarizes the research and proposed methods with the possible directions of future work and scientific research.



## Chapter 2

# Multiple Criteria Decision Analysis

This chapter describes the general characteristics and paradigms of MCDA with chosen methods considered within this doctoral dissertation.

### 2.1 General scheme of decision aiding process

We start with listing the major steps in decision aiding process. The first step of each decision process involves defining a problem and specifying the desired form of the outcome. At this stage, it is required to define which options are considered by DM, their essential features, and also their nature and characteristics. Problems can also be divided by the number of decisions to make. We can distinguish problems with a single decision and several interdependent decisions.

The next step is choosing a preference model, which states the way alternatives' performances on criteria are processed and aggregated. In this doctoral dissertation, two models are considered. The first calculates the comprehensive quality of each alternative using a value or utility function. This model, along with an exemplary UTADIS method, is extensively presented in Chapter 2.6. The other examined model focuses on pairwise comparisons and determines preference or outranking relation for each pair. Preference relation assumes that if alternative  $a_i$  is preferred over  $a_k$ , it means that  $a_i$  is better than  $a_k$ . Outranking relation states that if option  $a_i$  outranks  $a_k$ , then  $a_i$  is at least as good as  $a_k$ . PROMETHEE method, which employs preference relation, is described in Chapter 2.7, whereas the ELECTRE method that incorporates outranking is discussed in Chapter 2.8.

With a choice of preference model, DM is asked to provide preference information. It may concern direct values of model parameters. However, there is also the possibility of indirect preference information, e.g., exemplary class assignments or holistic

score of alternative which is preferred or not. A thorough description of different types of preference information is given in Chapter 2.4.

Based on the preference information, the methods create the exact instance of the preference model that needs to reflect the preferences as precisely as possible. In the case of indirect information, it needs to be firstly disaggregated to the specific model parameters.

In the case of a method that relies on additive value functions, the obtained model is used to assign a comprehensive quality score to all available options. However, for a method that operates on pairwise comparisons, the model offers insights into the relationships between all the alternatives. It also requires additional exploitation of this outcome to solve the initial problem. The procedures for exploiting valued preference relations are described in Chapter 2.7, whereas those for dealing with crisp outranking relations are in Chapter 2.8.

Nonetheless, there can be an infinite number of all possible parameter values compatible with preference information, and selecting one representative model might be challenging. It calls for robustness analysis, which provides information about a broad spectrum of all possible solutions given the uncertainty related to the selection of the compatible preference model instance. The detailed description is in Chapter 2.5.

All the solutions, representative models, and results of robustness analysis are offered to the DM's judgment. (S)he needs to check if the results are satisfying and acceptable. If so, the process of decision-aiding is completed. Otherwise, it is possible to choose another representative model, or the gathered solution may prompt DM to rethink preference information by its specification or change. Then the model is constructed once again, and the process is continued until DM approves the solution.

## 2.2 Notation

In this doctoral dissertation, the following notation is used:

- $A = \{a_1, a_2, \dots, a_i, \dots, a_n\}$  – a finite set of  $n$  alternatives;
- $A^R = \{a_1^*, a_2^*, \dots\} \subseteq A$  – a finite set of reference alternatives, which the DM accepts to critically judge in a holistic way;
- $G = \{g_1, g_2, \dots, g_j, \dots, g_m\}$  – a finite set of  $m$  evaluation criteria,  $g_j : A \rightarrow \mathbb{R}$  for all  $j \in J = \{1, \dots, m\}$ ;
- $X_j = \{x_j \in \mathbb{R} : g_j(a_i) = x_j, a_i \in A\}$  – a set of all different performances on  $g_j$ ,  $j \in J$ ;



- $x_j^1, x_j^2, \dots, x_j^{n_j(A)}$  – increasingly ordered values of  $X_j$ ,  $x_j^k < x_j^{k+1}$ ,  $k = 1, 2, \dots, n_j(A) - 1$ , where  $n_j(A) = |X_j|$  and  $n_j(A) \leq n$ ;
- $C_1, C_2, \dots, C_p$  -  $p$  pre-defined, preference ordered classes, where  $C_{h+1}$  is preferred to  $C_h$ ,  $h = 1, \dots, p - 1$  ( $H = \{1, \dots, p\}$ ).

## 2.3 General assumptions

In this doctoral dissertation, we make the following assumptions regarding considered problems and decision-makers.

Firstly, we assume that DM may not have any established preference model or a strict algorithm for scoring alternatives. Moreover, the decisions made by DM reflects his/her current preferences. It indicates that the preference information given by DM may not be consistent [53]. In case of finding any inconsistencies and pointing that to DM, (s)he can change his/her mind. Therefore, the main task of decision aiding is not to discover the existing preference model in DM's mind but to propose one that follows his/her preferences the best [12]. The DM knows which characteristics of the problem are crucial and which do not influence the decision that is made. It means that all criteria are complete and without any redundancies. Moreover, many methods assume the interdependence of criteria. The DM may define the direction of preference, which means that his/her preferences are monotonic on a considered criterion, or (s)he may be unsure of the nature of the criterion, and it needs to be discovered by model. Usually, two main types of monotonic criteria are considered: gain (non-decreasing) and cost type (non-increasing). In addition, DM can specify a criterion as strictly increasing or decreasing. In some real-world decision scenarios, the assumptions on monotonicity are too simplified a representation of DM's preferences, and they can be non-monotonic [88]. Moreover, scoring each alternative by DM is demanding and time-consuming, making it impossible to rate all options. Thus models can be used to comprehensively evaluate the options outside the reference set.

## 2.4 Preference information

As MCDA methods require the participation of DM in the decision process, the phase of gathering preference information is one of the most crucial in decision-aiding. The DM has various possibilities to express his/her preferences. Some may be easier for him/her, and (s)he can be more confident, but some may cause some difficulties and be more mentally demanding. Asking for the same preference information in different ways may lead to different outcomes, which may cause inconsistencies and solutions with lower quality [53].

Some methods assume providing direct preference information as precise model parameter values concerning, for instance, the importance of criteria, dependencies between them, or pairwise comparison thresholds. Parameters may have a clear interpretation [17], but it still requires from DM the knowledge of how the method works and the influence of these parameters on the final solution. Additionally, when DM does not have a well-defined preference model in mind, it might be for him/her cognitively demanding [10].

Conversely, the DM may provide preferences in an indirect way as local or holistic judgments. Such information may be established as an exemplary assignment of alternative to class, declaring the relation between pair of options or criteria, or ranking of subsets of alternatives. Such preferences can be more natural and less demanding for DM. Including historical decisions made by the DM's, it becomes easier to gain [10].

The disaggregation paradigm allows us to conclude the model's parameter values based on holistic preferences [44]. It focuses on finding an instance of the model which recreates examples of decisions provided by DM. Such models are called consistent or compatible. Usually, to perform preference disaggregation, the methods incorporate mathematical programming [108], evolution algorithm [32], [19], or simulated annealing [84].

## 2.5 Robustness analysis

Frequently, gathered models are only a single instance in an infinite set of all models compatible with preference information. It may cause that different sets of parameters to produce various decisions, considering alternatives from outside the reference set. The results and analysis, which come from different instances, may vary significantly. In order to assess the stability of the model, it is essential to perform a robustness analysis. It verifies a wide spectrum of feasible scenarios and informs DM about potential consequences of preference information given by himself/herself [37] [47]. In the case of mathematical programming models, it is possible to run a precise analysis that can check all possible and necessary relations between alternatives or assignments to classes. Possible relation (assignment) means that at least one compatible model confirms this decision, whereas necessary relation (assignment) requires all possible instances to recommend the same decision. Moreover, stochastic analysis can also be conducted using Monte Carlo simulation, providing information about the distribution of decisions in the space of all coherent models [100].

## 2.6 Reminder on UTADIS method

UTADIS is a preference disaggregation approach adjusted for sorting problems. It utilizes an additive value function to evaluate each alternative by summing the marginal values

on all criteria to the comprehensive value [54] [94]:

$$U(a_i) = \sum_{j=1}^m u_j(g_j(a_i)) = \sum_{j=1}^m u_j(a_i) \in [0, 1], \quad (2.1)$$

where  $u_j$  is a marginal value function. In the original approach, all functions are monotonic, where the direction of monotonicity depends directly on the preference provided by DM. There are distinguished two types of criteria: gain and cost type, which means that the function needs to be non-decreasing and non-increasing, respectively. These functions are defined by the set of characteristic points with their marginal values, which must fulfill the constraints of monotonicity. Marginal values for the remaining values on the criterion are linear interpolations of the two closest points. The DM can determine the characteristic points or encompass all possible values on the considered criterion [37].

The comprehensive value is normalized to the interval [0-1] where one is assigned to the ideal alternative  $a^+$ , i.e., with the most preferred evaluations on all criteria. It implies that the sum of marginal values for the best scores must equal 1. The comprehensive value equals 0 for anti-ideal alternative  $a^-$ , i.e., an option with the worst performances on all criteria. It leads to the necessity of each value of the marginal value function to be equal to 0 for the least preferred evaluation.

A value-driven threshold-based sorting procedure is employed to obtain assignments to classes for all alternatives. It incorporates thresholds  $t_h$  defined on the scale of comprehensive values where  $t_{h-1}$  states for the lower and  $t_h$  for the upper boundary of class  $C_h$ . The value of the worst threshold equals 0, and each subsequent one is greater than the previous, where the upper boundary of the best class is above 1. Preference information in this method appears as exemplary assignments to classes which are translated to constraints  $t_{h-1} \leq U(a_i^*) < t_h$  if the alternative is assigned to class  $C_h$  by DM.

This method is modeled as a mathematical programming problem where the goal is to minimize the number of incorrectly classified alternatives.

## 2.7 Reminder on PROMETHEE methods

The PROMETHEE method aggregates the results of pairwise comparisons of each alternative against all remaining ones into a comprehensive measure of desirability [6]. They calculate marginal preference degrees  $\pi_j(a_i, a_k)$  constructed on differences in alternatives evaluation on the specific criteria. DM selects the shape of the marginal preference function from a set of predefined shapes, with the most typical one illustrated in Figure 2.1. These functions can reflect the uncertainty of DM, which is why they require defining two thresholds: indifference  $q_j$  and preference  $p_j$ . The indifference threshold states the maximal difference in scores, which is negligible for DM. In contrast, the preference threshold means the minimal difference for which DM is sure that one score is better than the other. All functions are non-decreasing and normalized to the interval [0,1].

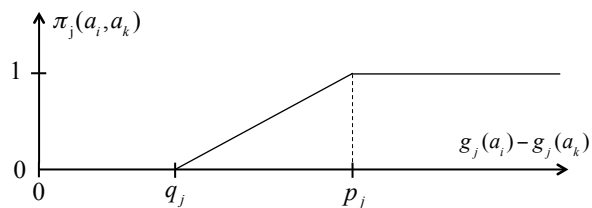


Figure 2.1: Typical marginal preference function used in PROMETHEE.

The marginal preference degrees are aggregated using a weighted sum into a comprehensive preference degree  $\pi(a_i, a_k)$ . These preference degrees are finally aggregated with the usage of the Net Flow Score (NFS) procedure to positive  $S^+(a_i)$  and negative  $S^-(a_i)$  flows [97]. Positive flow states for arguments for the strength of the considered option  $a_i$  and is calculated as an average of preference degree  $a_i$  over other alternatives. Besides, negative flow reflects the weakness of alternative  $a_i$  and means how other options are preferred over  $a_i$  on average.

These two flows can be employed to create a partial ranking by checking the relation between each pair of alternatives according to PROMETHEE I procedure. Indifference appears if both flows are identical. Then, preference shows if one option is better on one flow and not worse on the other. Finally, incomparability occurs if, accordingly to one flow, the alternative is better than another, whereas, on the second flow, the relation is reversed. The appearance of incomparability means that the model has not strong enough evidence to indicate one option is better than the other, and alternatives are significantly different.

With the usage of the PROMETHEE II method, it is possible to create a complete ranking by calculating total flow  $S(a_i) = S^+(a_i) - S^-(a_i)$ . In this case, option  $a_i$  is preferred over  $a_k$  if  $S(a_i) > S(a_k)$ . Alternatives are indifferent if their total flows are equal.

## 2.8 Reminder on ELECTRE methods

ELECTRE methods utilize outranking relation as a preference model by pairwise comparisons to state the existence of this relation [27] [89]. There can be distinguished as a set of thresholds reflecting the uncertainty in options evaluation as indifference threshold  $q_j$  and preference threshold  $p_j$ , whose interpretation is similar to PROMETHEE methods. Moreover, there appears also veto threshold  $v_j$ , which expresses the minimal performance difference, which is so critical that it has the power to invalidate the outranking.

Outranking relation is established via the concordance and discordance tests. Concordance test checks if there exists criteria coalition strong enough to support outranking relation. The concordance index is calculated as a weighted sum of partial concordance indices determined for each criterion. In turn, the discordance test verifies the strength of arguments against the outranking. Moreover, this relationship can be considered both

valued and crisp. The comprehensive concordance and discordances serve as the basis for computing the outranking credibility. Then, to transform it into a crisp one, it must be compared against the credibility threshold (cutting level)  $\lambda$ . For obtaining solutions for ranking or choice problems, it is required to exploit the gathered relation.

## 2.9 Exploitation methods for valued and crisp relations

Once an outranking or preference relation is constructed, it can be exploited in the function of choice or ranking problem to provide an adequate recommendation. In what follows, three state-of-the-art methods serving this purpose are discussed.

Valued relations may be exploited using, e.g., NFS [97] or distillation procedures [90]. Distillation computes the quality of each alternative and iteratively adds them to the constructed order until considering the entire set. In the downward (upward) distillation, the ranking is constructed in a top-down (bottom-up) fashion, retaining alternatives with the greatest (least) quality first. Finally, the two rankings are intersected to obtain a final ranking, which is a partial preorder.

In the case of binary relation, to gather a ranking of options, it is possible to use variants of NFS [97] or Qualification Distillation (QD) [90] adjusted to crisp relations. The other option in the procedure from ELECTRE-Score [25] assigns a score value to each alternative based on pairwise comparisons with reference alternatives.

Considering choice problems, there are several available methods [2]. Firstly, there can be recommended alternatives from the top of the ranking. Secondly, according to social choice theory, one can employ plurality or anti-plurality rule, i.e., choose alternatives that most often outrank other options or those that least often are outranked by others [5]. Thirdly, in the ELECTRE I method, one proposed exploiting an outranking graph to search for its kernel  $K \subseteq A$  as the most preferred subset of alternatives. Kernel  $K$  comprises alternatives not outranked by any other alternative in  $K$ . In contrast, the alternatives outside  $K$  must be outranked by at least one alternative in  $K$ . If there are cycles in the graph, they must be eliminated. In this dissertation, each cycle is aggregated into an auxiliary node that inherits all incoming and outgoing arcs of the alternatives contained in the cycle.



## Chapter 3

# Artificial Intelligence

In this chapter, we describe the main assumptions and concepts of machine learning methods (Section 3.1), artificial neural networks (Section 3.2), and web mining (Section 3.4) in the context of solving decision problems within the scope of this doctoral dissertation.

### 3.1 Machine learning

Machine learning is a sub-discipline of artificial intelligence that can learn and enhance performance through data analysis. The learning process involves examining vast amounts of data to identify previously unrecognized patterns, dependencies, and rules that were not explicitly programmed. As a result, the program can predict new data, make decisions, and solve problems. During the learning process, algorithms can adapt their parameters based on the information included in the training data. The greater the amount of data and the better its representativeness of the problem, the better results can be achieved.

We can distinguish numerous paradigms of learning, among which are:

- Supervised learning is the technique where an algorithm is learned on examples from a training set, and each example is treated as a known correct answer. The method analyzes these examples and tries to find general patterns and rules, which enable predicting answers for new data.
- Unsupervised learning focuses on data analysis for which no clear answers exist. The aim is to find hidden patterns, structures, or groups in data, which help to understand data or find anomalies.
- Reinforcement learning is a paradigm in which a method learns by interacting with the environment and getting the response as a reward or penalty. These

algorithms try to find an optimal strategy of actions to maximize rewards and minimize penalties.

- Active learning is a technique in which an algorithm actively engages with the user by asking questions about scoring a specific item instead of relying on a pre-existing training data set.

This doctoral dissertation focuses explicitly on supervised learning, where evaluating the performance of a trained model is an essential step. This is done using a separate test set independent of the training data to assess the model's ability to make accurate predictions and decisions based on previously unseen data. The ultimate goal of machine learning is to develop models that can generalize information beyond the scope of their training data sets.

Due to the wide range of optimization approaches and their complexity, these techniques require a set of hyperparameters that govern the learning process. A subset of data specifically reserved for validation purposes can be utilized to determine the optimal values for hyperparameters. This validation data helps evaluate the quality of various model variants and chooses the best hyperparameter set.

Machine learning can be applied to a wide range of diverse issues, for instance:

- classification, which means assigning objects to predefined classes or categories based on their features,
- multi-label classification, which enables to assign of one object to multiple classes (labels) simultaneously,
- label ranking where apart from assigning a set of labels, there is also created their ranking where position states for adequacy for this object,
- regression which is a prediction of numerical value for the option,
- clustering, i.e., discovering some clusters in data based on the similarities between objects and assigning them to these groups.

An unquestionable advantage of ML is the possibility of processing large data sets and extracting knowledge from them. Numerous methods are easily scalable, enabling the generate results in a short amount of time. However, it is also connected with the fact that some algorithms require significant training data to achieve high accuracy. If there is not enough high-quality data, models can be less accurate and more prone to overfitting. It means they can be too fitted to training data with a small ability for generalization to new ones. To prevent overfitting, regularization techniques are employed. It focuses on adding extra constraints or sanctions to the learning process to enlarge the generalization ability and avoid over-complicating the algorithm.



ML can discover complicated patterns and dependencies in data, which may be hard to identify with traditional data analysis methods. They can model highly non-linear transformations and dependencies between data. It facilitates the discovery of knowledge and a better understanding of problems. However, many methods do not support a precise definition of data characteristics and relations there. Hence, there is a risk of learning incorrect conclusions based on accidental correlation in data, leading to erroneous predictions and inferences. Moreover, many more complex algorithms are intricate and challenging to interpret and get a comprehensible and precise prediction explanation. It impedes the detection of erroneous dependencies learned by the model and disrupts the possibility of verifying the correctness of the model. All issues mentioned above may lead to reduced trust and social acceptance of these methods, hence their limited usage.

## 3.2 Artificial neural networks

Neural networks are machine learning models that may be applied to almost every task focusing on predicting highly complex output based on well-defined input data. Their knowledge representation model is a structure consisting of many neurons grouped in layers. Many architectures define different layers and connection types, which are adjusted to various problems. The most common are Feed Forward Networks, Convolutional Neural Networks, and Recurrent Neural Networks.

To capture non-linear dependencies, the output of neurons undergoes a non-linear transformation through an activation function. The best-known activation functions are:

- Sigmoid and hyperbolic tangent, which map input data to intervals  $[0,1]$  and  $[-1,1]$ , respectively, using the logistic function,
- Rectified Linear Unit (ReLU), which outputs 0 for negative input values and preserves positive values,
- LeakyReLU is derived from ReLU but introduces a small scaling factor for negative values.

Neural networks are trained using gradient-based techniques, such as Stochastic Gradient Descent. In this approach, the successive optimization steps aim to minimize the loss function, gradually adjusting the model's parameters to approximate the desired output values. The training data is divided into smaller groups known as batches to expedite the training process and enhance learning stability. Within one step of optimization, the model generates predictions for all objects included in a single batch. This process is repeated for all batches until the whole training data set is processed, called an epoch. When batch includes the whole data set, it is called Batch Gradient Descent, and whereas the approach when the training data set divides into many batches, it is called Mini Batch

Gradient Descent [92]. Many optimization techniques exist, including additional regularization or adaptation of the speed of learning of each parameter or momentum, which are supposed to make the learning process faster and reduce the influence of overfitting. The exemplary techniques are Adam[57] and AdamW [72].

In deep learning, other approaches are employed to address overfitting and enhance model robustness against noise in the data. Techniques such as dropout and data augmentation are utilized for this purpose [107], [93]. The former is a regularization technique that randomly turns off some neurons during training. It causes the reduction of dependencies between neurons and forces neural networks to use a wide range of input data. The latter focuses on creating artificial input data by different transformations on training data, enlarging the data’s diversity.

Most computations in neural networks involve matrix transformations, particularly matrix multiplications. Specialized hardware such as GPUs and TPUs excel in performing fast matrix calculations, thereby reducing the training and prediction time of neural networks. Furthermore, distributed learning across multiple computing machines is feasible, allowing for efficient scaling of neural networks. This scalability enables training larger models on extensive datasets, improving model efficiency and accuracy.

### 3.3 Preference learning

Preference learning is a research area at the intersection of ML and MCDA, which focuses on model reasoning based on user preferences. In this context, preferences may be considered a set of constraints or requirements to solve decision problems that can be violated somewhat, contrary to the MCDA approach [29].

Many machine learning methods only optimize a strictly defined goal. Usually, in many real-world examples, DM cannot characterize an ideal solution and possess limited knowledge about possible solutions. It leads to the situation when the expected goal is impossible to achieve, or the result is unsatisfactory and can be improved [4]. Preference learning provides numerous solutions, including methods of preference acquisition and prediction based on empirical data. They are used in multiple applications like marketing, recommender systems, computer games, e-commerce, or web browsers [30]. Preference information may come from different sources and be presented, for instance, as direct feedback from users in the form of like or dislike, the score on the ordinal scale as stars, or indirectly as clicks in links to usage or ordering a product.

Preference learning advances the concept of supervised learning to learning from a training set with a known preference. It may also include more general types of information like relative preference or information about preferences considering specific object features. Methods, which are proposed in preference learning, focus mainly on ranking problems, especially:

- Object ranking – involves ordering items based on preference using pairwise comparisons (a ranking problem in MCDA).
- Instance ranking – in this case, the goal is to assign an alternative to one from a set of predefined classes ordered by preference. It can be seen as a sorting problem in MCDA.
- Label ranking – an order of labels that best fits each object is created. This ranking type focuses on determining the most suitable labels for each object.

Models proposed in preference learning are ease in model interpretability and the possibility to provide additional preference information, e.g., regarding criteria monotonicity. Preference models can be divided into four main groups. The first group focuses on models that are based on utility or value functions, which capture the overall preference of an object or alternative. The second group involves learning the preference relation between pairs of objects, allowing for direct comparison and ranking. The third group utilizes local preference aggregation techniques, which consider the preferences of neighboring objects or alternatives. Finally, the fourth group encompasses model-based preference learning, where sophisticated models are constructed based on particular assumptions regarding the preferences relations [30].

Among this area, some interpretable machine learning methods were adapted to decision problems [7], for instance, Rank SVM [46], or decision trees with monotonicity constraints. Moreover, several MCDA methods were altered to address problems with a great amount of data. These are Choquet integral [22] to handle interactions between criteria, model additive value functions [68], or a method that generates a monotone rule ensemble based on Dominance-based Rough Set Approach [14].

### 3.4 Web mining

Web mining, a sub-discipline of AI, involves extracting, gathering, discovering, and analysis of information from various internet sources [8]. It is essential in the context of information retrieval on the internet. Web browsers create personalized rankings of adequate websites based on a query and its context. They score sites in terms of accuracy and quality of content, reliability of links, the popularity of the website, and its reputation. It is helpful for users to find the information they need and be sure that the presented pages are credible, valuable, and connected with their interests.

In web mining, we can distinguish three major areas:

- web content mining focuses on extracting and analyzing relevant data from websites, focusing on their content.

- web structure mining directs to the analysis of connections between web pages represented as a directed graph where websites correspond to vertices and hyperlinks to arcs,
- web usage mining involves analyzing user behavior and interactions on the internet, specifically their interactions with websites.

In this doctoral dissertation, we focus only on web structure mining.

Usually, websites are not independent documents defined only by their content. They also include connections to other websites as hyperlinks that refer to their subject. The analysis of websites graph enables one to discover subject groups, identify essential web pages, analyze social networks, and detect fake or spam pages.

In this context, the most popular approach is PageRank, which states the importance and popularity of web pages [85]. It enables scoring and ranking pages based on the quality of links that lead to them. Over the years, the PageRank algorithm was an essential technique for scoring web pages in the Google browser. The main idea is that websites with many incoming links from other pages with high rankings are indications that the site is valuable and vital. This concept was employed in numerous fields, for instance, scoring robot swarms based on individuals [11], the analysis of social networks [64], or identifying genes connected with some diseases [81]. Different variants of PageRank are used in network security by spam identification, detrimental pages [62], or botnets [28]. In particular, one can use the TrustRank algorithm [39], which enables assigning websites to a set of trusted pages and then propagating information through the graph of website connections.

The other algorithm of network analysis is HITS which focuses on identifying two types of websites: hubs and authorities [58]. The algorithm assigns two scores for each page, exhibiting a mutually reinforcing relationship. The authority score quantifies the value of the page's content. A good authority must be linked by many good hubs, being regarded as a meaningful source for a particular topic. The hub value captures the value of each page's links to other sites. A good hub points to many good authorities.

The other algorithm similar to HITS is Salsa [66], which divides pages to hub and authorities but differs in their identification. This method investigates second-degree neighborhoods. Pages are considered good hubs if they link to websites with links from good hubs. Then, pages are good authorities if they are pointed by websites linking to other good authorities. A random walk on a bipartite graph calculates the quality of hubs and authorities. One part corresponds to hubs, and the other to authorities. Each page can belong to both groups. Connections in the graph correspond to links between pages.

## Chapter 4

# Method inspired by deep learning and preference learning

Over the last few years, the amount of gathered and processed data increased dramatically. In particular, some data sets include historical records for decision problems with a large volume of available options and decisions made for them in the past [45]. Analyzing them in an understandable way and verifying the correctness of the conclusions is crucial for companies that gather the data. Over the years, MCDA methods have delivered tools to analyze and support the decision-making process, which is straightforward in interpretation and delivers credible explanations of their recommendations.

Historically, decision problems involved a relatively small number of reference alternatives. It comes from the fact that each piece of preference information from the DM is created by his/her holistic analysis of the available options [21]. Due to that, traditional MCDA methods were designed to learn from a small amount of preference information. Values of model parameters are often established by disaggregation of preference information via mathematical programming. The gathered model tries to recreate the DM's way of thinking by the best reflection of his/her decisions. Unfortunately, in the case of large data sets consisting of highly inconsistent preferences in data, models have difficulties with effective extraction of preference [70].

One of the most significant advantages of machine learning is the possibility of scalability, which enables to analyze large data sets. In particular, deep learning copes well with discovering model parameters for complicated problems, which include massive and highly noisy training data. A more precise description of neural networks is available in Chapter 3.2. These features of ANN were the reason to use them in the context of decision aiding, which learns from historical decisions or patterns as exemplary options. Specifically, [73] described the Adaptive Feedforward ANN approach to rank the set of discrete alternatives using a highly nonlinear value function. Further, [42] used ELECTRE-based single-layer perceptron to solve multi-criteria classification problems,

whereas [38] proposed NN-MCDA method consisting of two parts: linear additive value model and compound nonlinear fully connected deep neural network. The disadvantage of solutions described in [42] and [38] is the difficulty in interpreting the entire model or its parts. Additionally, they do not allow the definition of strictly monotonic criteria.

On the other hand, the methods proposed within the scope of preference learning solve decision problems that are greater in size. They include methods from MCDA adapted to cope with a large amount of reference data and ML methods modified to learn monotonic preferences. The detailed description of preference learning is in Chapter 3.3.

The problem of coping with an enormous amount of reference data in MCDA methods and possibilities offered by deep learning was the motivation to propose several methods inspired by both disciplines in this doctoral dissertation. In [74], we proposed architectures of neural networks which apply preference learning inspired by highly interpretable MCDA methods. In particular, we considered methods based on Ordered Weighted Average (OWA) operator [105], Choquet integral [1], TOPSIS [43], UTADIS, PROMETHEE, and ELECTRE. The latter three are described in Chapter 2. We propose a model of preference learning which accustoms deep learning techniques to discover parameter values of the model from the large set of reference data. These methods address sorting or instance ranking problems, and to accomplish this, each of them assigns a comprehensive score  $Sc(a_i)$  to  $a_j \in A$ . This score is then employed in a value-driven threshold-based sorting procedure (Chapter 2.6). It is worth adding that values of thresholds which separate classes, are model parameters adjusted during training. The objective function is the minimization of average regret for reference alternatives which states the distance from thresholds delimiting the desired class in case an alternative is misclassified or to zero.

The networks used to derive parameters for the OWA-, Choquet-, and distance-based models are shallow and contain from one to two linear layers. However, the ANNs proposed for UTADIS, PROMETHEE, and ELECTRE can be categorized as deep learning models [15] due to their multiple hidden layers and their ability to process various levels of data (such as criteria, alternatives, pairs of alternatives, and assignments). The network for ANN-ELECTRE involves the greatest number of layers and units among all introduced methods, with one input layer, five hidden layers, and one output layer. Hidden layers are necessary to capture the complexity of the value- and outranking-based MCDA methods. All components and units of the proposed architectures are suitably adjusted so that the proposed models retain constraints on the monotonicity of the criteria. However, the raw weight values obtained from multiple layers, some of which apply nonlinear transformations to the data, are not easily interpretable for users. Therefore, we ensure that users are presented with the final models of ANN-UTADIS, ANN-PROMETHEE, and ANN-ELECTRE. These models provide a summary of the comprehensive contribution of individual criteria. They consider the transformations applied by various layers, the activation functions used for nonlinear processing, and the normalization of scores

to a more easily interpretable range of alternatives' scores.

## 4.1 ANN-UTADIS: Preference learning with UTADIS and ANN

In this section, the general scheme of the proposed methods is described in reference to ANN-UTADIS. A detailed description of all remaining methods is available in [74]. ANN-UTADIS extends the UTADIS method, which is a preference disaggregation method that quantifies a comprehensive quality of each alternative using an additive value function:

$$U(a_i) = \sum_{j=1}^m w_j u_j(g_j(a_i)), \quad (4.1)$$

where  $u_j \in [0, 1]$  is a marginal value function and  $w_j$  is a weight associated with criterion  $g_j$ .

To represent marginal value values, it is necessary to define a neural network able to model any monotonic function. Only a non-decreasing function is considered as the transformation to non-increasing is conducted by negating the function. One of the basic features which are on the theoretical foundations of ANN is that the neural network  $u(x)$  with one hidden layer and sigmoidal activation function  $\sigma$  can approximate any continuous  $N$ -dimensional function with accuracy depending on the number of neurons or components  $L$  in the hidden layer [13]:

$$u(x) = \sum_{k=1}^L \alpha_k \sigma(y_k^T \mathbf{x} + \theta_k), \quad (4.2)$$

where  $\alpha_k, \theta_k \in \mathbb{R}$  and  $y_k \in \mathbb{R}^N$  are weights of this network and  $\mathbf{x} \in \mathbb{R}^N$  is an input vector. By limiting parameters  $\alpha_k \in \mathbb{R}_{\geq 0}$  and  $y_k \in \mathbb{R}_{\geq 0}^N$ , function  $u(x)$  is monotonic. The most often used sigmoidal functions are sigmoid and hyperbolic tangent. However, both of them have the problem of gradient vanishing, which causes troubles while learning some particular neurons or even totally stops them for specific input values. To mitigate these difficulties, a function *LeakyHardSigmoid* is introduced:

$$\text{LeakyHardSigmoid}(x) = \begin{cases} \delta x, & \text{if } x < 0, \\ x, & \text{if } 0 \leq x \leq 1, \\ \delta(x - 1) + 1, & \text{if } x > 1, \end{cases} \quad (4.3)$$

where  $\delta$  is a slope factor, a very small value in the range  $[0, 1)$ . This function is not sigmoidal and cannot approximate level segments of non-decreasing functions. However, gradually decreasing the slope to zero during training makes it possible to make the *LeakyHardSigmoid* function equivalent to *HardSigmoid*. Neural network  $u(x)$  defined

in Eq. 4.2 for one-dimensional vector  $\mathbf{x}$  and  $\mathbf{y}$  with non-negative weights  $\alpha_k$  and  $y_k$  and activation function  $\sigma$  *LeakyHardSigmoid* is further described as *Monotone Block*. The number of components  $L$  limits the maximal number of the function  $u(x)$  breakpoints; however, it can be lower if  $\alpha_k = 0$ .

The architecture of the network used in the ANN-UTADIS method is shown in Figure 4.1. Initially, input data needs to be scaled to interval  $[0,1]$  by using, e.g., min-max scaling and cost-type criteria need to be transformed to their gain-type counterparts. Each performance of the alternative is processed by a non-decreasing function inside *Monotone Block*. Subsequently, the individual marginal values for each criterion are combined into a comprehensive value using a linear layer, as shown in Eq. 4.1. This layer's weights ( $w_j$ ) are enforced to be positive values to maintain the predetermined preference directions. Since the output from *Monotone Block* is not normalized to the interval  $[0,1]$ , we perform a min-max scaling of comprehensive scores:

$$Sc_{ANN-UTADIS}(a_i) = \frac{U(a_i) - U(a^-)}{U(a^+) - U(a^-)}. \quad (4.4)$$

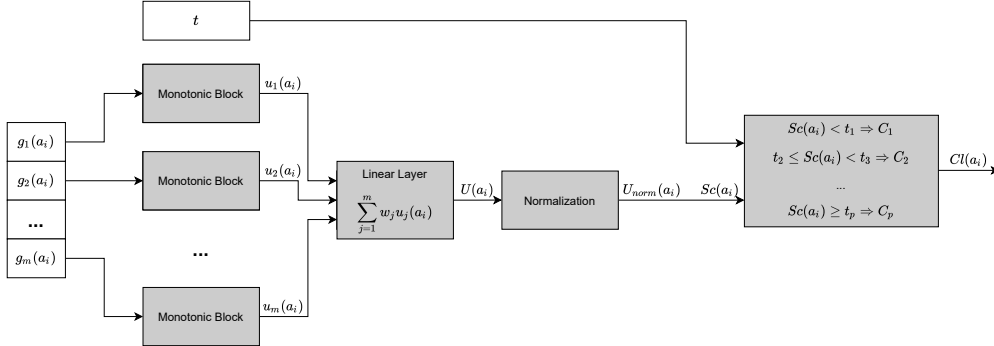


Figure 4.1: The neural network architecture employed by the ANN-UTADIS method.

The neural network employed in ANN-UTADIS optimizes various parameters: weights  $w_j$ , class thresholds  $t$ , and parameters used in each *Monotone Block*. To make parallel computations easier, hyperparameter  $L$  should be the same for all criteria, enabling operations on tensors instead of scalars. To summarize, the model of ANN-UTADIS consists of one input layer, three hidden, and one output layer.

The idea of utilizing a monotone block to model any monotonic function was also utilized in two other proposed methods, i.e., ANN-PROMETHEE and ANN-ELECTRE. This transformation was used to discover the shape of the marginal preference function for PROMETHEE and the shape of marginal concordance and discordance functions for ELECTRE. It removes the requirement of defining a specific shape of these functions from the DM and allows for a better fit to data. Additionally, the ANN-ELECTRE network can directly learn the values of the preference threshold, allowing the functions that comprise this model to be interpreted.



Batch Gradient Descent is employed to accelerate the training process and remove dependence on the order of considered alternatives. If it is impossible, using Mini Batch Gradient Descent is recommended. Although then the order of processing options may influence the results, it has an insignificant effect when the size of the batch is large enough.

The data augmentation technique is utilized to enhance the model’s resistance to noise, improve its robustness, enhance its ability to generalize, and reduce overfitting. This is achieved by adding Gaussian noise to the training data, which is diverse in each epoch.

## 4.2 Illustration of preference models inferred with neural networks

To present the model and its interpretation, let us consider a two-class classification problem Employee Rejection / Acceptance (ERA) [40]. This problem focuses on a student survey that investigates the willingness to hire an employee based on four candidate features, such as experience and verbal skills. All the criteria in the survey are in the form of gain type and have been pre-processed to be within the range of 0 to 1. The models were obtained by training the ANN-UTADIS on 80% randomly chosen reference alternatives. The model consists of 4 marginal value functions shown in Figure 4.2. The interpretation of these plots is identical to the original UTADIS method. Among all criteria, the highest influence on comprehensive value has criterion  $g_3$  and the lowest – criterion  $g_2$ . It is also easily noticeable that the change in values between 0 and 0.1 for criterion  $g_2$  has a minor impact on comprehensive value. In contrast, even a slight change in criterion  $g_1$  between values 0.8 and 1.0 causes a big difference in comprehensive value. Additionally, we can see that the influence of criterion  $g_3$  is almost linear.

## 4.3 Computational experiments

To evaluate the effectiveness of the ANN-inspired methods discussed in this doctoral dissertation, extensive experiments were conducted on nine benchmark data sets sourced from the UCI repository (<http://archive.ics.uci.edu/ml/>) and the WEKA software [40]. These data sets consist of over hundred to 1700 alternatives evaluated on 4 to 8 criteria, adjusted to the problem of binary sorting. Moreover, these data sets include from 14 thousand to nearly 3 million pairwise comparisons, which are directly reconstructed by methods like ANN-PROMETHEE and ANN-ELECTRE.

To investigate the level of inconsistency in each data set, we performed the following steps. First, we computed the number of pairs of alternatives for which the holistic scores were coherent with the dominance principle or for which it was violated. Moreover, we checked if indifferent options were classified the same. Finally, we analyzed how

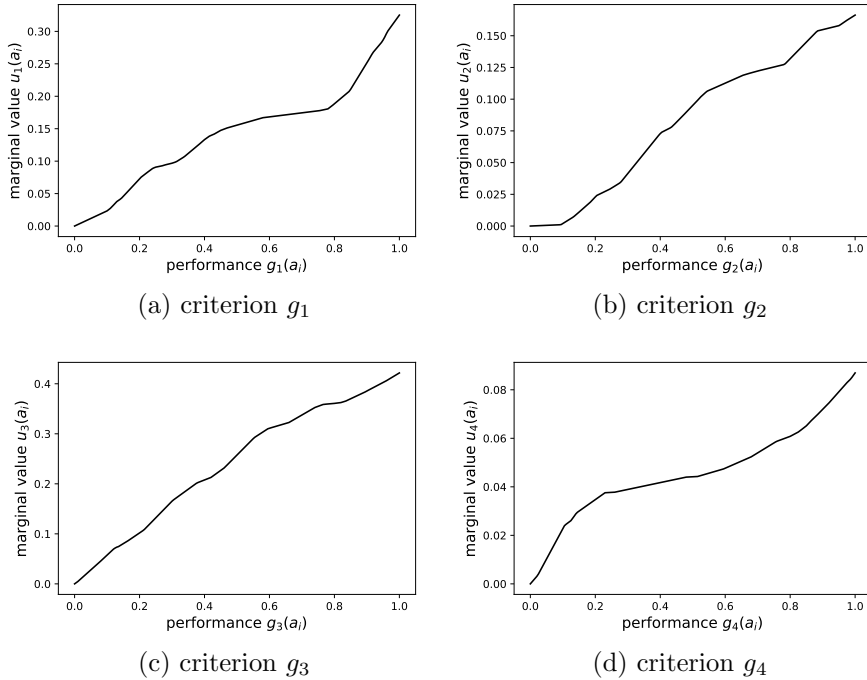


Figure 4.2: Marginal value functions scaled by criteria weights constructed by ANN-UTADIS for the ERA dataset.

many criteria in each problem disrupt the preference monotonicity rule by comparing mean scores in each class. This study showed that all considered data sets consist of inconsistent preference information. However, the number of inconsistencies in different problems varied. It made the recreation of preference information for some problems challenging, whereas it was not demanding for others.

As evaluation measures, we employed three measures: the standard misclassification rate (0/1 loss), F1, and AUC (Area Under the Curve), which reflects a ranking error, i.e., the average amount of changes in ranking from comprehensive values to make results identical. Moreover, we tested three scenarios of solving a stated problem in each data set which is supposed to quantify how well the models cope with knowledge generalization. We focused on different training and test set proportions from the whole data set. The first scenario assumed that the training set was vastly larger than the test (80% to 20%); in the second one, the sizes of both sets were the same (50% vs. 50%), and in the last scenario, the training set was significantly smaller than the test set (20% to 80%). To ensure a more robust analysis, each experiment was repeated 100 times, and the results were averaged over all iterations.

To determine the optimal values of hyperparameters, grid search tests were conducted, evaluating the classification quality across different parameter values. We considered the

following parameters: learning rate  $lr$ , components number in monotonic block  $L$ , and standard deviation  $\xi$  in Gaussian noise used for data augmentation. Based on those experiments, ranges of values for these parameters were established, which allowed for achieving the highest scores for each data set. Results for AUC for the ERA data set and model ANN-UTADIS are presented in Figure 4.3. Most of the gathered solutions were similar, but the best scores are for  $lr = 0.05$ ,  $L = 20$ , and  $\xi = 0.05$ .

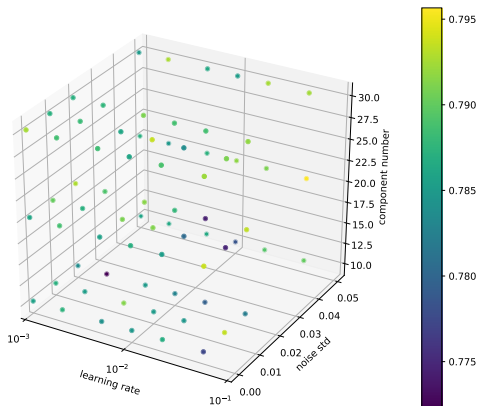


Figure 4.3: The AUC value attained for the ANN-UTADIS and different hyperparameter values for the ERA dataset.

All acquired solutions were confronted with results from other state-of-the-art preference learning methods. For comparison, we selected models of linear regression [41], Choquistic regression, kernel logistic regression with the polynomial kernel (KLR-ply) and Gaussian kernel (KLR-rbf) [99], the MORE algorithm [14], LMT [65], MR-Sort with Mixed-Integer Program (MIP) [67] and dedicated metaheuristic (META) [96], UTADIS with predefined characteristic points [108] and with with the characteristic points corresponding to all unique performances (UTADIS-G).

The best scores were achieved for models ANN-UTADIS, ANN-Ch-Uncons. (Choquet integral model without constraints on the range of weights values), CR, KLR-rbf, and ANN-PROMETHEE for AUC measure without ANN-PROMETHEE for 0/1 loss and F1. ANN-PROMETHEE and ANN-ELECTRE achieved higher accuracy on AUC than 0/1 loss because these methods correctly recreated most relations from pairwise comparisons. However, they made errors during the classification. Considering different types of problems, the difficulty of the challenge came from its inconsistency and was also reflected in the efficiency of accomplished results by all models. While comparing methods based on a similar preference model, e.g., ANN-UTADIS was statistically significantly better (Wilcoxon test) on the test set than methods utilizing mathematical programming, i.e., UTADIS and UTADIS-G. There were several reasons for this. Firstly, the objective of minimizing the sum of regrets, as done by UTADIS and UTADIS-G, was not aligned with the perspective captured by AUC. Additionally, implementing a Monotonic

Block in ANN-UTADIS allowed for more flexible inference of marginal value functions, enabling the better fitting of characteristic points in the input data. Finally, using data augmentation in ANN-UTADIS helped prevent overfitting issues that may arise with UTADIS-G.

## 4.4 Summary

This part of the thesis introduced interpretable and explainable models of preference learning. They allow learning from the reference set, with the known preference, and predict preference for other options. The described learning algorithms enable finding parameters of the monotonic sorting model with the optimization techniques proposed for deep learning. As a result, we eliminate the need for arbitrarily determining the values of meta-parameters, such as the shapes of preference functions or the characteristic points of marginal value functions. Instead, the method can employ more general per-criterion (value, preference, concordances, or discordance) functions, allowing greater freedom in adjusting to the data. Moreover, the proposed approaches can learn from highly inconsistent data, which are too large to be processed with traditional approaches effectively. We presented the possibilities the elaborated approaches offer on examples of learning from over a thousand alternatives and coping with problems consisting of millions of pairwise comparisons. Additionally, the proposed methods have accuracy comparable with other preference learning methods. The predictive performance is particularly favorable for ANN-UTADIS, ANN-UTADIS, ANN-Ch-Uncons. and ANN-PROMETHEE models.

## Chapter 5

# Methods inspired by machine learning

### 5.1 Introduction

The common feature of machine learning methods is the possibility of recreating very complex transformations of input data to achieve the highest accuracy. These models are supposed to discover complex patterns in data and employ them for new options predictions.

In real-world problems, there can exist criteria for which the direction of preference is equivocal. It is often when a range of most desired values exists, and all below or above are less favored. For instance, in some medical data, there is an interval of acceptable values, which states the proper parameter of a physiological feature of the patient. In contrast, the scores from outside this range may be evidence of some abnormalities or illness. On the other hand, there can exist criteria that are fully non-monotonic.

The usage of highly complex transformations to recreate data may be connected with the problem of overfitting. It means that the model has a low ability for generalization and, at the same time, a high adjustment to single training observations. As a result, many regularization techniques exist to constrain the model's complexity. On the one hand, they try to force the model to employ only information and transformations which matter. Moreover, they try to direct the model to more straightforward transformations. The more complex the model, the more challenging its interpretation.

The existence of non-monotonic criteria in many real-world problems motivated the works of many researchers. First, some methods use a set of predefined shapes of marginal value functions [38], [88], [16]. In particular, [88] defines many monotonic criteria, including level type, exponential, stepwise, and non-monotonic. Secondly, more general algorithms aim to handle non-monotonicity in a broader sense without focusing on specific shapes and limiting the complexity of these functions [33], [18]. The last group

aims to minimize the complexity of non-monotonic functions. In this context, [59] introduced penalization for non-monotonicity changes using MILP models. On the other hand, [32] and [70] restrict the variability of the slope shape of the value function to ensure the most interpretable sorting model. While this approach allows for modeling any shape of the function, it also allows for an unlimited number of changes in the direction of monotonicity.

In decision problems, the complexity of the model influences both the overfitting and the ease of interpretability. The primary goal is to create a model which recreates DM's preferences. The existing methods do not directly define the number of changes in monotonicity and do not explain non-monotonic functions.

The above observations motivated the development of two approaches to modeling non-monotonic criteria. In [48], the complexity of the model is defined as the number of monotonicity changes. A DM who wants the most interpretable model prefers solutions with fewer monotonicity changes in criteria. In turn, [49] introduces non-monotonic functions, which are decomposed into two monotonic components: non-decreasing and non-increasing. Despite direct constraints on the complexity of function, a straightforward interpretation of non-monotonic criteria is enabled by such a solution.

The common problem considered in machine learning is multi-label classification. It assumes an object is assigned to one class for each label or decision. There exist many ways to cope with such problems like binary relevance [9], label powerset [101], or probabilistic classifier chains [9]. Binary relevance focuses on transforming the problem into multiple classification problems and considering them separately. In this case, we lose information about dependencies between decisions. The label powerset transforms a problem into a classification problem of all possible subsets of labels. This technique requires a large amount of data to represent each class appropriately. It might be highly challenging for data with a rare combination of labels. Probabilistic classifier chains create chains of classifiers so that each classifier predicts a label based on previous classifiers. The result of running such a classifier depends on the order of considered decisions and requires multiple considerations of the same problem. The limitations of the existing approaches inspired the creation of a specialized method for multi-decision sorting problems proposed in [49].

This section briefly discusses two works, [48] and [49], inspired by ML. Both incorporate a threshold-based value-driven sorting procedure to the additive value function [35], [109]. The remainder of UTADIS methods is shown in Chapter 2.6. The usefulness of this research was presented on real-world problems concerning risk management related to handling nanomaterials in different conditions.

## 5.2 Non-monotonic criteria

### Minimization of the number of monotonicity changes

This section describes how to model per-criteria preference functions based on partial information about DM's preferences. In particular, this information may define the type of criterion as gain or cost, level-monotonic [88], or one for which the direction of preference a priori cannot be defined. Moreover, DMs may establish non-monotonic A- and V-shaped criteria or the criterion with unknown monotonicity constraints that can take any shape. The plots of exemplary marginal functions are presented in Figure 5.1. Mixed-Integer Linear Programming (MILP) is used to adjust the non-monotonic character of the marginal value functions to the available assignment examples. The complexity of these functions is controlled by minimizing monotonicity changes for value functions across all criteria. Each type of preference direction is defined as a set of constraints on marginal values, assuring the proper shape of function and normalization of criteria. The model is normalized when an anti-ideal alternative has a comprehensive value equal to 0, whereas an ideal one has a score equal to 1. It means that the sum of marginal values assigned to the most preferred performances needs to be 1. These performances are unknown a priori as their position depends on the shape of marginal value functions. It is the reason for establishing each criterion's most and least preferred value.

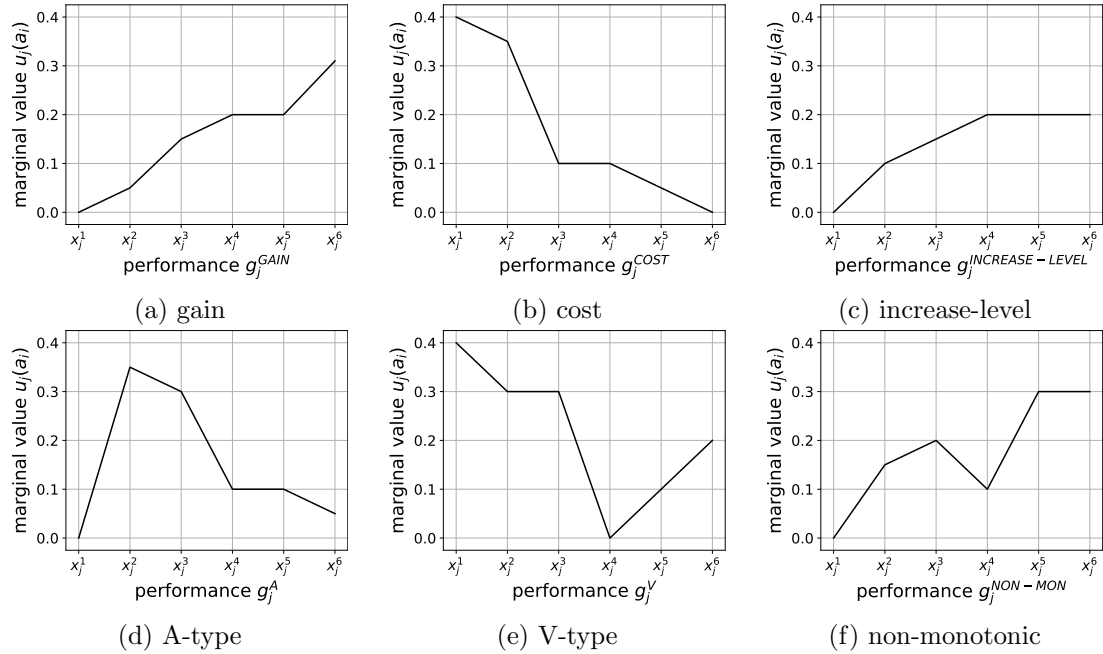


Figure 5.1: Example marginal value functions representing different types of requirements with respect to their monotonicity.

The shape of the gain-type criterion is guaranteed by forcing the non-negative value of the difference between partial values for consecutive performances  $u_j(x_j^k) \leq u_j(x_j^{k+1})$ .

Then, for the cost-type criterion, the value assigned to each performance cannot be greater than for the next one  $u_j(x_j^k) \geq u_j(x_j^{k+1})$ .

Suppose the DM states a monotonic criterion but does not specify the preference direction. In that case, it is modeled as a combination of two criteria: non-decreasing and non-increasing. Moreover, a binary variable is introduced to indicate which components are considered while setting the value of the other components to zero.

A-type criterion means that the most preferred value is somewhere in between the extreme performances. Hence it involves a single change in monotonicity from non-decreasing to non-increasing. The shape of this function is modeled with the use of binary variables  $v_{j,k}^{opt}$  connected with the pairs of consecutive scores on criterion  $x_j^{k-1}$  and  $x_j^k$ . The binary variable  $v_{j,k}^{opt}$  is equal to one for the first pair for which there is a non-increasing value. This enables non-decreasing shapes until score  $x_j^{k-1}$  and non-increasing after that. Due to the characteristic of this function, the performance  $x_j^{k-1}$  is the most preferred value, which will be used in the normalization. On the other hand, the least preferred performances can be only extreme values. An additional binary variable is introduced to enforce that the marginal value for at least one of them equals zero. Thus either the highest or the lowest performance is the least preferred. We model the V-type criterion similarly by requiring that the least preferred performance is somewhere inside the performance range.

Level-monotonic criteria types are the ones which, until or from some a priori unknown performance, remain constant. There are four distinguished variants of criteria: increase-level, level-increase, decrease-level, and level-decrease. They are modeled by joint constraints connected with gain or cost type criteria, with these for A-type and V-type. For instance, increase-level criteria can be enforced by compiling the requirements for A- and gain-type functions. In this case, binary variables  $v_{j,k}^{opt}$  indicate the performance that initiates the level of constant preference. The other considered variants are as follows:

- level-increase as a combination of gain- and V-type criteria,
- decrease-level as a compilation of V- and cost-type criteria,
- level-decrease combines constraints from cost- and A-type criteria.

Finally, let us consider the criterion for which no a priori information regarding monotonicity can have any shape. In general, avoiding defining any constraints for such functions would be possible. However, since the goal is to control the complexity of the inferred marginal value functions, the two sets of binary variables  $v_{j,mon-dir}^{k,k-1}$  and  $v_{j,change-mon}^{k,k-2}$  is introduced, which captures the number of changes in monotonicity between the neighboring performance sub-intervals. The first set defines if there is a non-decreasing or non-increasing part between values  $x_j^k$  and  $x_j^{k-1}$ . Then, the second



one stores information about the change in monotonicity. The value of  $v_{j,change-mon}^{k,k-2}$  is set to one if there appeared a change in monotonicity direction in the point  $x_j^{k-1}$  using binary variables  $v_{j,mon-dir}^{k,k-1}$  and  $v_{j,mon-dir}^{k-1,k-2}$ . As for the non-monotonic criterion, any score can be the least or most preferred, so two more sets of binary variables are used.

The above-discussed constraints modeling the shape of the marginal value function are combined with the constraints representing the DM's preference information. They form the Mixed Integer Programming problem that allows minimizing the number of monotonicity changes in the following way:

$$Minimize : NM = \sum_{j \in G_A \cup G_V} \sum_{p=2}^{n_j(A)} v_{j,p}^{opt} + \sum_{j \in G_{NON-MON}} \sum_{k=3}^{n_j(A)} v_{j,change-mon}^{k,k-2}, \text{ s.t. } E^{AR}.$$

The computational complexity of Mixed Integer Programming problems mainly depends on the number of employed variables. The number of continuous and binary variables required to model proposed types of criteria is shown in Table 5.1. Additionally, other variables essential to model preference information must be included. Their number may differ for various problems. For sorting problems, we need to incorporate  $p$  continuous variables, which are limiting profiles between classes and  $|A^R|$  variables connected with the comprehensive values of reference alternatives.

Table 5.1: The number of continuous and binary variables in the MIP problem depends on the type of criterion.

Criterion type	# continuous variables	# binary variable
Gain, cost	$n_j(A)$	0
Monotonic non-defined type	$3n_j(A) + 1$	1
A-type, V-type	$n_j(A) + 1$	$n_j(A)$
level-type	$n_j(A)$	$n_j(A) - 1$
non-monotonic with a controlled number of monotonicity changes	$n_j(A) + 1$	$4n_j(A) - 3$
non-monotonic as a composition of cost and gain type	$3n_j(A) + 1$	0

## Non-monotonicity as the composition of gain-type and cost-type components

The need for easy interpretability of non-monotonic criteria has driven the introduction of various approaches to model these criteria. Specifically, the marginal value functions  $u_j(x_j^k)$  of criteria that may exhibit non-monotonic behavior are represented as a sum of marginal value functions one of a gain-type  $u_{j,g}(x_j^k)$  and the other of a cost-type  $u_{j,c}(x_j^k)$ . In the case of discovering by model the direction of preference, one of the components is set to zero, and the criterion becomes monotonic. Otherwise, both components have positive values, and the criterion can have any non-monotonic shape. However, there can occur that the marginal value is greater than zero for each characteristic point.

It is undesirable due to two reasons. Firstly, the comprehensive value of the anti-ideal alternative is greater than zero in this case. It means that the range of comprehensive value is reduced, which makes the space of possible solutions smaller. Secondly, the marginal value function, which has the least preferred value greater than zero, is harder to interpret. To prevent such a scenario, the marginal value function should be normalized so that at least one value is zero, which can be obtained using additional bias value. It was defined for each criterion and is subtracted from marginal values and adds the constraint that each marginal value needs to be greater or equal to zero.

The number of continuous variables required for modeling this type of criterion is equal to  $3n_j(A) + 1$ . When comparing this non-monotonic criterion to the one described in the previous section, the main difference is that the non-monotonic criterion, being a composition of gain- and cost-type criteria, does not require binary variables. However, there is no possibility to control the complexity of the shape of the function.

### 5.3 Multi-decision sorting problems

Quite often, multiple decisions must be made for the same alternatives. The problem of multi-decision sorting focuses on simultaneously assigning alternatives to one of the predefined classes for many decisions. These classes express the level of quality or risk using a predefined scale consistent across all decision attributes.

It is necessary to employ separate models for each decision to capture the varying relevance of specific evaluations and criteria across different decisions. However, inter-decisional constraints have been introduced to address the interdependencies between decisions. These constraints reflect the relations between comprehensive values associated with the same alternative across multiple value functions used to classify that alternative based on different decision attributes.

It is important to emphasize that the number and interpretation of classes remain the same across all decision attributes. In this way, the classes specified by the DM determine the order of labels associated with each reference alternative. When the class  $C_{DM}^{D_s}(a^*)$  assigned to reference alternative  $a^*$  in decision  $D_s$  is more preferred than the class  $C_{DM}^{D_t}(a^*)$  in another decision  $D_t$ , it indicates that the corresponding label  $D_s$  is more suitable or fitting for that particular alternative. As a result, the comprehensive value of  $a^*$  associated with  $D_s$  should be greater than its corresponding value associated with  $D_t$ :

$$\left. \begin{array}{l} \text{for all } a^* \in A^R : \\ \text{if } C_{DM}^{D_s}(a^*) > C_{DM}^{D_t}(a^*) : \\ U^{D_s}(a^*) \geq U^{D_t}(a^*) + \varepsilon. \end{array} \right\} E^R(\text{inter} - \mathcal{D}) \quad (5.1)$$

## Inconsistent preference information

This part of the dissertation also focussed on dealing with inconsistencies in preference information regarding the holistic assignment of reference alternatives. The main goal of optimization is the minimization of several alternatives which are incompatible with the model. To achieve this, we used binary variables associated with each alternative. The variable is set to one if an alternative needs to be removed from the reference set. Then constraints the alternative took part in are always satisfied.

## 5.4 Results of experiments

### Use case description

The practical usefulness of the methods proposed in [48] and [49] is demonstrated in a case study concerning exposure management related to handling nanomaterials in different conditions [83]. Nanomaterials are particles between 1 and 100 nanometers in size whose physicochemical properties differ significantly from those of larger-sized materials composed of the same atoms. Due to these specific properties, nanomaterials are used in many fields, such as construction, electronics, environmental management, and healthcare. The production, processing, and use of nanomaterials can involve exposure to health and life risks. These effects are an object of concern. They are still being studied, and the safety standards are mainly based on analogous chemical manufacturing processes. Different precautions can be used to minimize the corresponding risks depending on the specific exposure situation. These precautions can be considered decision attributes, within which predefined classes have been defined to represent different levels of risk. In [48], we considered the necessity of using respirators by workers as personal protective equipment when handling nanomaterials. In turn, in [49], we additionally examined fume hood, fume hood with HEPA filter as engineering controls, and HEPA vacuum cleaner, corresponding to the work practices.

The set of alternatives used in experiments is composed of exposure scenarios for existing and future nanomaterials and manufacturing processes [83]. Each alternative was evaluated on ten criteria that define the features of nanomaterials like *particle size*, their *toxicity*, *airborne capacity*, *detection limit*, and parameters associated with the manufacturing process such as *exposure limit*, *quantity*, *number of employees*, *engineering controls*, and *multiple exposures*. The alternatives were scored on the five-point scale, which states the requirement of each precaution:  $C_1$  (required; the least preferred class),  $C_2$  (might be required),  $C_3$  (optional),  $C_4$  (might be optional), and  $C_5$  (not required; the most preferred class). Preference information shows the assignment of reference alternatives to one of the predefined classes for each decision attribute.

In what follows, we present the results of experiments conducted using methods pro-

posed in [48] and [49], including interpretation of gathered models and explanation of chosen decisions.

## Results for minimization of the number of monotonicity changes

In [48], we used 30 reference alternatives for the model construction and 21 non-reference options. We considered the following assumptions regarding criteria types:

- *number of employees* and *engineering controls* are non-monotonic,
- *particle size* is increase-level,
- *detection limit* is gain-type,
- all remaining criteria are cost-type.

The constructed model involves two changes in monotonicity from non-increasing to non-decreasing for criterion *engineering controls* and from non-decreasing to non-increasing for *number of employees*. The plots of marginal value functions for these criteria are presented in Figure 5.2 for the representative model. What is worth noticing is the shape of these functions. In particular, *engineering controls* is V-shaped with the least preferred value Open-NP. Moreover, *number of employees* is A-shaped with the most preferred value on 11-50 employees. The criterion *particle size* is increase-level, involving constant preference for the range 2-10nm.

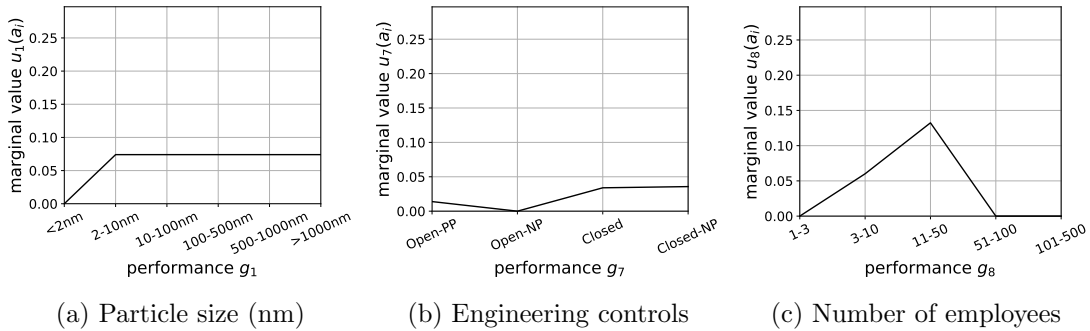


Figure 5.2: Marginal value functions for the exposure management of nanomaterials in the context of using a respirator.

The influence of each criterion on the recommended class can be estimated by considering the maximum contribution of each criterion to the comprehensive value (see Table 5.2). The most significant impact on the comprehensive value can be attributed to *detection limit*, *duration of exposure* and *airborne capacity*, whereas the lowest impact is associated with *quantity*, *multiple exposures*, and *toxicity*. Analyzing the differences in marginal values between specific performances on criteria allows for identifying transitions that can significantly improve the requirement level of using a respirator. An example of such a significant difference is the change in team size from 50-100 employees to

11-50 employees, which involves a change of 0.1324. On the contrary, the increase in the team size does not influence the decision that is made.

Table 5.2: The maximal shares of the individual criteria in the comprehensive values (in %).

Criterion	Maximal share
Particle size (nm) ( $g_1$ )	7.40%
Toxicity ( $g_2$ )	4.97%
Airborne capacity ( $g_3$ )	14.55%
Detection limit ( $g_4$ )	26.83%
Exposure limit (f/cc) ( $g_5$ )	6.62%
Quantity (Kg) ( $g_6$ )	2.09%
Engineering controls ( $g_7$ )	3.57%
Number of employees ( $g_8$ )	13.24%
Duration of exposure (h) ( $g_9$ )	17.77%
Multiple exposure (number) ( $g_{10}$ )	2.96%

The explanation of the decision is based on describing which features have an impact on the final decision. It lets the DM understand the whole decision process better. The impact of the individual criteria on the comprehensive values and the relations between the latter ones for different decision attributes are demonstrated in Figure 5.3. Each option has a share of each criterion's marginal function on the final score. We also provide limiting profiles for classes. It is easily seen that, e.g., alternative  $a_{18}$  was assigned to the most preferred class  $C_5$  mainly because of the highly preferred score on *detection limit* ( $g_4$ ) and *duration of exposure* ( $g_9$ ).

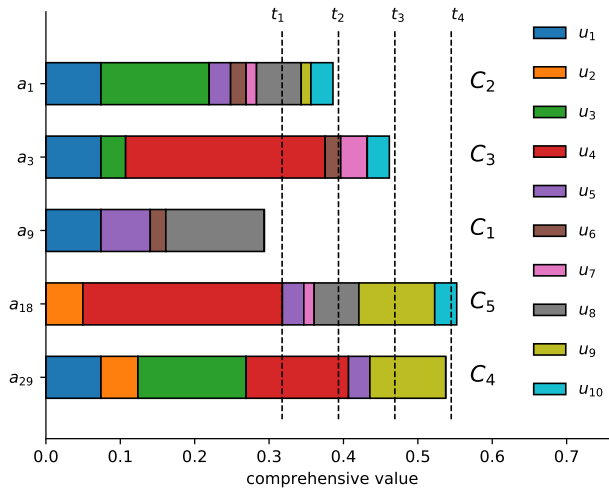


Figure 5.3: Marginal and comprehensive values as well as class assignments for the five example reference exposure scenarios.

## Robustness analysis for minimization of the number of monotonicity changes

The constructed marginal value functions are ones from the infinite number of functions compatible with the preference information from the DM and the minimal number of monotonicity changes. An analysis of possible assignments  $C_P(a)$  of non-reference alternatives was performed to verify the stability of sorting recommendation. It investigates if there is at least one compatible instance for which alternative  $a$  is assigned to class  $C_h$ . It is computed by transforming an original optimization problem with additional constraints of assigning alternative  $a$  to the considered class  $C_h$ .

An additional constraint is set to establish the number of monotonicity changes equal to those previously found. If such a set of equations is feasible, then  $a$  can possibly be assigned to  $C_h$ . Otherwise,  $a$  cannot be assigned to  $C_h$  with any compatible model instance. For the proposed model involving non-monotonic criteria, when alternative  $a$  is possibly assigned to class  $C_h$  and  $C_k$  where  $h \geq k+1$ , then it needs to be assigned to all classes in-between those mentioned above. This is called the “no jump property” [37].

Given a set of all possible binary vectors  $\mathcal{V}$ , two situations can occur:

1.  $\mathbf{v}_1 = \mathbf{v}_2$  for all  $\mathbf{v}_1, \mathbf{v}_2 \in \mathcal{V}$  which means that all criteria have changes in monotonicity, the most and least preferred values are in the same characteristic points. In this situation, the possible assignment  $C_P(a)$  of alternative  $a$  is an interval without jumps.
2.  $\mathbf{v}_1 \neq \mathbf{v}_2$  for at least one  $\mathbf{v}_1, \mathbf{v}_2 \in \mathcal{V}$  states for the situation where criteria have different shapes or changes of monotonicity are in different characteristic points. Then, the possible assignment  $C_P(a)$  of alternative  $a$  is a union of intervals where nothing can be said about the presence or absence of jumps.

As a result of a conducted robustness analysis, three alternatives were necessarily classified into only one class, and the rest of the options were possibly classified into intervals of 2 or 3 classes without jumps.

## Results for a multi-decision problem

In [49], we conducted experiments using 40 reference alternatives and 5 non-reference ones. In contrast to the problem solved in [48], apart from *number of employees* and *engineering controls*, also *particle size* is treated as a non-monotonic criterion. The main goal of optimization is to minimize the number of alternatives whose desired assignments to classes is not recreated. As a secondary criterion, we minimize a sum of biases for non-monotonic criteria. The discovered most optimal model recreates assignments for 37 reference alternatives.

The marginal value functions for the *particle size* and four decision attributes are presented in Figures 5.4. The shapes of these functions are non-monotonic, but they can be easily decomposed to gain- and cost-type components and bias values 0.10, 0.03, 0.07, 0.04 for *respirator*, *fume hood*, *fume hood with HEPA filter*, *HEPA vacuum cleaner* correspondingly. Besides *fume hood with HEPA filter*, the most preferred is the biggest size of particles  $> 1000\text{nm}$ . For decisions *respirator*, *fume hood with HEPA filter*, *HEPA vacuum cleaner*, the shape of marginal functions has a W shape. It is caused by the increase in a gain component between sizes 2-10nm, 10-100nm, and 100-500nm and the decrease in cost component between 10-100nm, 100-500nm, and 500-1000nm. It leads to high peaks on sizes 0-100nm or 100-500nm. These plots show the complementary influence of *fume hood* and *fume hood with HEPA filter*, as the first one has the least preferred values for 10-500nm, whereas the second has the most.

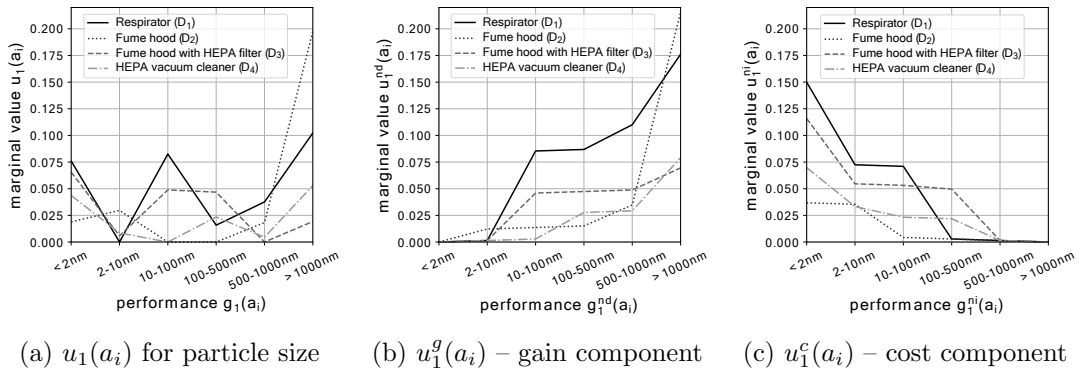


Figure 5.4: Marginal value functions for particle size for four decision attributes.

This complementary influence can also be seen while considering the maximal share of each criterion in the comprehensive value (Table 5.3). The model should reflect not only the alternative assignments to classes but also the ranking of the requirements of each precaution for each alternative. It is the reason why marginal functions focus on different aspects of scenarios. Criteria with the most significant influence for *fume hood* simultaneously have the lowest for *fume hood with HEPA filter* and the other way round. The exception is *airborne capacity* criterion has a high impact on all decisions, whereas *quantity* has marginal influence on all but decision  $D_3$ .

Class assignments for decision *respirator* with the explanation as marginal and comprehensive values for five reference alternatives are presented in Figure 5.5(a). They show one intra-decision relation, which means that options scored by DM as safer have a higher comprehensive value than scenarios connected with classes with greater levels of the requirement of precautions. Considering alternative  $a_2$ , it was assigned to class  $C_2$  as its comprehensive values  $U^{D_1}(a_2)$  appeared between thresholds  $b_1$  and  $b_2$  which limit classes  $C_1$  from  $C_2$  and  $C_2$  from  $C_3$  respectively. The reason for that was small values on a few criteria: *detection limit* ( $u_4^{D_1}(a_2) = 0$ ), *quantity* ( $u_6^{D_1}(a_2) = 0.0014$ ), *duration*

Table 5.3: The maximal shares of the individual criteria in the comprehensive values (in %) for four decision attributes.

Criterion	Respirator ( $D_1$ )	Fume hood ( $D_2$ )	Fume hood with HEPA filter ( $D_3$ )	HEPA vacuum cleaner ( $D_4$ )
Particle size ( $g_1$ )	10.22%	20.6%	7.32%	6.18%
Toxicity ( $g_2$ )	3.58%	0.79%	12.04%	9.97%
Airborne capacity ( $g_3$ )	17.5%	18.15%	21.82%	16.44%
Detection limit ( $g_4$ )	15.27%	0.44%	8.87%	12.58%
Exposure limit ( $g_5$ )	12.24%	11.91%	9.5%	20.57%
Quantity ( $g_6$ )	3.56%	0.59%	12.88%	6.09%
Engineering controls ( $g_7$ )	9.4%	16.26%	5.73%	7.7%
Number of employees ( $g_8$ )	7.93%	14.65%	4.22%	10.22%
Duration of exposure ( $g_9$ )	11.66%	14.21%	4.29%	2.58%
Multiple exposure ( $g_{10}$ )	8.58%	2.35%	13.28%	7.63%

of exposure ( $u_9^{D_1}(a_2) = 0.0048$ ), and multiple exposures ( $u_{10}^{D_1}(a_2) = 0.0065$ ). The main influence on decision was assigned to airborne capacity ( $u_3^{D_1}(a_2) = 0.1750$ ) and number of employees ( $u_8^{D_1}(a_2) = 0.0793$ ).

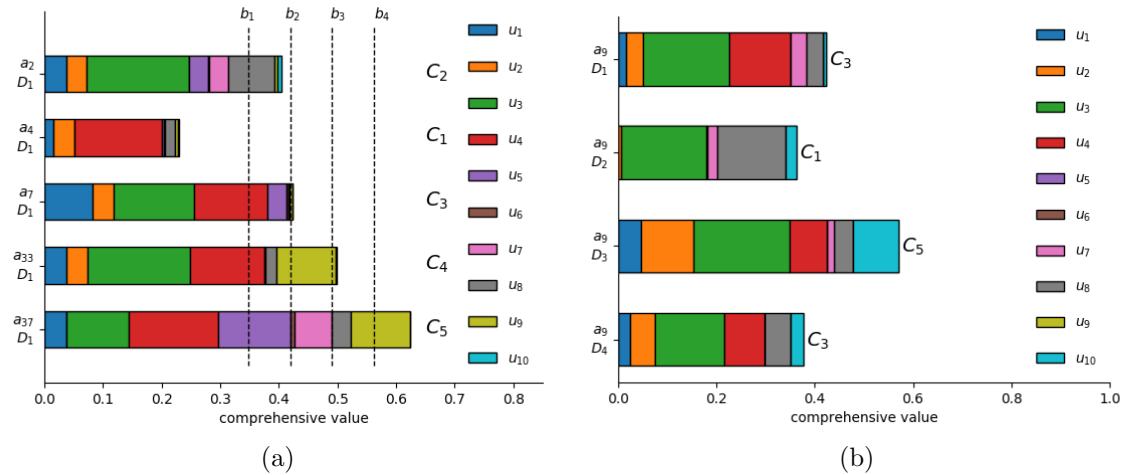


Figure 5.5: Marginal and comprehensive values and class assignments demonstrating (a) intra-decision relations for five reference exposure scenarios in terms of respirator ( $D_1$ ); (b) inter-decision relations for the alternative  $a_9$  in terms of four decision attributes (Respirator –  $D_1$ , Fume hood –  $D_2$ , Fume hood with HEPA filter –  $D_3$ , and HEPA vacuum cleaner –  $D_4$ ).

Figure 5.5(b) illustrates the inter-decision relations and the influence of individual criteria on the comprehensive values for the example scenario  $a_9$ . Depending on the decision, the same option can be scored differently, leading to various classes. Alternative  $a_9$  was assigned by the DM to the best class for decision *fume hood with HEPA filter* ( $D_3$ ), *respirator* ( $D_1$ ) and *HEPA vacuum cleaner* ( $D_4$ ) was optional and *fume hood* ( $D_2$ ) was required. It meant that the most suitable decision not to use precaution for this option was  $D_3$  and the least  $D_2$ . This ranking of precautions was reflected in comprehensive values for particular decisions  $U^{D_3} > U^{D_1}, U^{D_4} > U^{D_2}$ . The comprehensive value of  $a_9$  for *fume hood with HEPA filter* ( $D_3$ ) came mainly from highly preferred scores on criteria with



great maximal shares, i.e., *airborne capacity*, *toxicity*, *multiple exposures*, and *detection limit*. On the other hand, the relatively small comprehensive value for *fume hood* came from the fact that it had less preferred performances on criteria *particle size*, *exposure limit*, *duration of exposure*, and *engineering controls* which for this decision had a high maximal share. It caused that for this decision, option  $a_9$  had the lowest comprehensive value.

## 5.5 Comparison of the proposed models

The proposed works focus on solving two potentially similar but completely different problems, i.e., sorting for a single decision [48] or multiple interdependent decisions [49].

In both papers, we presented new approaches to modeling non-monotonic criteria as mathematical programming problems. In [48], ten criteria were defined, of which three were non-monotonic. The main aim of the proposed method is to guarantee the model's interpretability which is accomplished by favoring lower complexity. This feature of the model is controlled in two ways. There can be direct constraints on the shape of the function or minimization of the number of changes in monotonicity directions. It is obtained using binary variables, which restrain the possible shapes.

On the other hand, in [49], non-monotonic criteria are implemented as the compound of two components: non-decreasing and non-increasing. It allows us to receive a function whose interpretation focuses on these two components. There is no need to use binary variables which control the number of monotonicity changes. However, this results in the possibility of obtaining complex functions.

### Comparison of experiments results

The usefulness of the proposed methods in both papers was tested on the real-world problem of exposure management of engineered nanomaterials. Although a different set of alternatives was used and various descriptions of *particle size* criterion, as well as other formulations of the problem, there appeared some similarities between obtained models. Considering decision attribute *respirator*, it was easily noticeable that in both cases, maximal shares of the representative model were similar, i.e., *airborne capacity*, *detection limit* and *duration of exposure* had high values, whereas *toxicity*, *quantity* and *multiple exposures* had low. Moreover, in the case of plots of marginal functions, both models pointed to Open-NP as the least preferred for *engineering controls* and for criterion *number of employees* teams with 101-500 employees. In [49], non-monotonic criteria were more complex, e.g., for *number of employees*, there were three changes of monotonicity directions. In turn, in [48], when this number was minimized, there was only one change for *number of employees*.

The differences in obtained models came from different objective functions and additional constraints between decisions. In [49], besides assigning alternatives to classes, the model had to recreate a ranking of decision attributes that state which precautions were least required for the considered scenario. Due to this, the model considered different aspects connected with various precautions. For all decisions, the essential criterion was *airborne capacity*.

The presented plots of marginal value functions are only a single instance from an infinite number of models compatible with preference information. Robustness analysis was conducted in [48] to analyze the necessary and possible assignments for non-reference alternatives.

## Chapter 6

# Methods inspired by web mining

This chapter presents methods of exploitation of valued preference relation, called PrefRank [75], and crisp outranking relation, called ScoreBin [76]. Methods based on discovering preference relations are one of the most popular in MCDA. These relations can be divided into valued and crisp. Their interpretation may differ, depending on the context [97]. In the case of valued relation, it can signify the degree extent to which users recognize one option as better than the other [69], percentage share of compatible instances of the model which confirm preference [51]. Moreover, it can signify the strength or credibility level of the statement " $a_i$  is at least as good  $a_k$ " or " $a_i$  is more preferred to  $a_k$ " [6], [23], [31]. Then, binary relation may indicate the existence of outranking relation [26] or strict preference [3].

The set of all relations can be described as a directed graph with alternatives as vertices and the discovered relations as arcs. In real cases, it is relatively rare that an outranking or preference relation points to one option as the best or allows ordering alternatives unambiguously. Hence these relations need to use additional exploitation techniques to get recommendations of the most preferred options or their ranking. These techniques are described in Chapter 2.9.

However, these methods can be criticized. When it comes to approaches for creating ranking, NFS calculates entering and leaving flows by simple weighted aggregation of in- and out-coming arcs for each alternative. Nevertheless, this method does not consider the preference graph's structure. Specifically, when an alternative is preferred to a relatively good option to the same degree as a relatively worse solution, both instances contribute equally to the comprehensive strength of the alternative. Similarly, if the alternative is considered worse than highly favorable or weak options, these instances are equally significant in terms of the alternative's overall weakness. As a result, all pairwise comparisons are given equal discriminative power, which is solely determined by the preference index values.

Furthermore, the distillation procedures do not provide explicit and comprehensive

scores or numerical values for alternatives. This limitation becomes problematic when a cardinal ranking is desired as the method’s output. Additionally, the distillation process does not consider the difficulty or ease of outranking other alternatives, which can be relevant in decision-making scenarios. Moreover, using ELECTRE-Score necessitates the specification of supplementary reference profiles and preference information, enabling the assignment of precise scores to them. This additional requirement increases the cognitive demand of the process.

Regarding choice problems, methods based on social choice theory do not consider relations between alternatives. There might be recommended options that are outranked by many others. The method of graph kernel from ELECTRE I may recommend relatively weak alternatives. It means that many other alternatives can outrank the option in kernel unless they are in graph kernel. Additionally, the user has no control over the number of recommended options. Finally, none of the described techniques allows additional indirect preference information, influencing the final ranking. The only possibility is to pass additional information or modification of method parameters during relation creation.

Therefore, this doctoral dissertation proposes two families of methods PrefRank and ScoreBin. These methods are inspired by web structure mining, which creates a ranking based on hyperlinks (Section 3.4).

## 6.1 PrefRank

This section describes a family of weighted preference relation exploitation methods called *PrefRank* presented in [75].

Similarly to the NFS method, these approaches aggregate preference degrees to strengths  $S^+(a_i)$  and weaknesses  $S^-(a_i)$  of alternative. They can be employed to create a partial ranking using the procedure known from PROMETHEE I or to establish a complete ranking as in the PROMETHEE II method (see Section 2.7). The main difference between NFS, PrefRank, and ScoreBin is the calculation of  $S^+(a_i)$  and  $S^-(a_i)$ . Correspondingly to NFS, strength and weakness in PrefRank are made as a normalized elementary strength  $\phi^+(a_i)$  and weakness  $\phi^-(a_i)$  to sum up to one:

$$S^+(a_i) = \frac{\phi^+(a_i)}{\sum_{k=1}^n \phi^+(a_k)} \text{ and } S^-(a_i) = \frac{\phi^-(a_i)}{\sum_{k=1}^n \phi^-(a_k)}. \quad (6.1)$$

Contrary to NFS,  $\phi^+(a_i)$ ,  $\phi^-(a_i)$  aggregate preference degrees as a weighted sum instead of a simple sum:

$$\phi^+(a_i) = \sum_{k=1}^n \pi(a_i, a_k) \cdot \omega^+(a_k) \text{ and } \phi^-(a_i) = \sum_{k=1}^n \pi(a_k, a_i) \cdot \omega^-(a_k), \quad (6.2)$$

where  $\omega^+(a_k)$  and  $\omega^-(a_k)$  are weights of alternatives whose interpretation differs for each PrefRank variant. Different aspects of the relation between alternatives can be considered by using them. We propose three variants: PrefRank I, II and III, for which inspiration was derived from methods for website ranking: PageRank, HITS, and Salsa.

### **PrefRank I**

When calculating the strength of each alternative in PrefRank I, it is appreciated to be preferred to relatively good alternatives rather than bad ones. That is if an alternative is better than some other option which, in turn, is preferred to a significant degree – over all or the majority of other solutions, some bonus should be implied. Conversely, if being preferred to a relatively poor alternative that, on its own, does not prove its superiority over other solutions, the alternative’s strength should not be significantly increased. On the other hand, when calculating the weakness of each alternative, it is perceived as a more significant disadvantage to be outranked by relatively weak rather than strong alternatives. It means that if an alternative is worse than some other option which, in turn, is strongly outpreferred by many other solutions, this should lead to a significant penalty. However, proving worse than some strong alternatives revealing no or limited deficiencies when other options are compared against it should not add much to the alternative’s weakness. Such effect can be achieved with the following weights:

$$\omega^+(a_k) = S^+(a_k) \text{ and } \omega^-(a_k) = S^-(a_k). \quad (6.3)$$

Considering preference relation as a preference graph, the calculation of strengths in PrefRank I is inspired by the PageRank method, which assumes that a website is good if pointed out by many other good websites. Following this interpretation, strengths can be explained as the probability of finishing in the considered vertex of the preference graph using the random walk algorithm. In this context, the preference degree  $\pi(a_i, a_k)$  equals the probability of moving from  $a_i$  to  $a_k$ . The other interpretation strength in PrefRank I is that they result from an alternative voting system where each option has a voting strength equal to  $\pi(a_i, a_k)$ .

The calculation of values  $S^+(a_i)$  and  $S^-(a_i)$  is made iteratively, assuming all strengths and weaknesses are equal in the first step. The process is finished when the differences between consecutive iterations are negligible.

### **PrefRank II**

The weighting scheme in PrefRank II is inverse to PrefRank I and assumes that a strong alternative should be heavily preferred over weak solutions. The alternative’s strength is computed as the weighted sum of preference degrees with weights interpreted as the weaknesses of solutions it is compared against. On the other hand, a weak alternative is the one vastly outranked by strong alternatives. Hence the alternative’s weakness

is computed as the weighted sum of preference degrees with weights interpreted as the strengths of solutions that are compared with it.

It means that the weights used for strengths and weaknesses calculations are as follows:

$$\omega^+(a_k) = S^-(a_k) \text{ and } \omega^-(a_k) = S^+(a_k). \quad (6.4)$$

PrefRank II is inspired by the HITS method, which distinguishes two roles of websites: hubs and authorities. In this perspective, a page is a good hub if it points to good authorities, and it is a good authority when linked by good hubs. In our adaptation, the alternative's strength is similar to a hub score, and the weakness is similar to an authority score.

### PrefRank III

PrefRank III extends the idea underlying PageRank I taking into account an overall difficulty in being preferred to some alternative estimated by analyzing its relations with all other alternatives. On the one hand, an alternative's great strength derives from being highly preferred to the alternatives outranked by other good solutions. It means that the strength of the option depends proportionally on the strength of the second-degree neighbor alternatives:

$$\omega^+(a_k) = \frac{1}{\sum_{i^*=1}^n \pi(a_{i^*}, a_k)} \sum_{l=1}^n \left[ \frac{\pi(a_l, a_k)}{\sum_{k^*=1}^n \pi(a_l, a_{k^*})} S^+(a_l) \right]. \quad (6.5)$$

In turn, an option's high weakness is implied by being vastly outranked by alternatives that are preferred to other weak solutions:

$$\omega^-(a_k) = \frac{1}{\sum_{i^*=1}^n \pi(a_k, a_{i^*})} \sum_{l=1}^n \left[ \frac{\pi(a_k, a_l)}{\sum_{k^*=1}^n \pi(a_{k^*}, a_l)} S^-(a_l) \right]. \quad (6.6)$$

#### 6.1.1 Measures used for comparing the choice or ranking recommendations

This section presents measures that count similarities between rankings and recommended alternatives.

The Normalized Hit Ratio (NHR) [50] method checks the compatibility of recommendations of best alternatives and is calculated as Jaccard's distance between options at the top of the ranking. Then, Kendall's  $\tau$  coefficient [55] measures the similarity of the relationship between pairs of alternatives. It might be used for the comparison of complete rankings. It assumes that the distance between inverse preference relations  $P$  is two times greater than between preference and indifference  $I$ . Similarly, the Normalized Ranking Distance (NRD) method compares partial rankings [50]. It presumes that the distance between incomparability relation  $R$  and  $I$  is the same as between  $P$  and  $I$ , whereas the distance between  $R$  and  $P$  is 1.5 greater than  $P$  to  $I$ .

### 6.1.2 Experimental comparison between PROMETHEE and PrefRank

We ran experiments to check the similarities between rankings and recommended alternatives between PrefRank and PROMETHEE. They assumed testing of similarities of the results for artificially generated problems. We considered data sets consisting of 4 to 20 (with step 2) alternatives evaluated from 3 to 8 criteria. Then, we determined a valued preference relation using the PROMETHEE method. For each problem size, we generated 100 instances with uniformly distributed performances and criteria weights. The indifference thresholds  $q_j$  were drawn from the interval between 0% and 20% of the performance range on a given criterion, whereas the preference thresholds  $p_j$  were drawn from the interval delimited by  $q_j$  and 50% of the performance range.

For the gathered valued preference relations, we created rankings using PrefRank and PROMETHEE. They were compared with NHR, NRD, and Kendall's  $\tau$ , and the results were averaged among all problem instances. The results showed high similarity in obtained recommendations. The most resembling rankings were obtained for PROMETHEE and PrefRank III. The most contrasting solutions were observed between PrefRank I and PrefRank II. The results of experiments for different problem sizes imply that there was no explicit dependency between the number of alternatives and criteria and the similarity between methods. However, there can be stated that solutions were more alike for a smaller number of criteria and a greater number of alternatives, but there were exceptions to that.

### 6.1.3 Case study concerning evaluation of special economic zones

As part of this dissertation, the PrefRank methods were used to rank special economic zones (SEZs) in Poland. These regions offered favorable investment conditions, enhanced infrastructure, and convenient access to skilled personnel. The goal was to rank 10 SEZs in Poland: Kamienna Góra (KAM), Kostrzyn-Słubice (KOS), Kraków (KRA), Legnica (LEG), Łódź (LOD), Mielec (MIE), Pomorze (POM), Słupsk (SLU), Starachowice (STA), and Tarnobrzeg (TAR). They were characterized based on five criteria: the *total area* each SEZ occupies, *capital expenditures*, the total *number of jobs*, the number of *business permits* and *financial results*. For this problem, criteria weights were determined following the Simos-Roy-Figueira (SRF) method[24]. The comprehensive preference degrees were calculated using the PROMETHEE method. Calculations were made using the *Diviz* platforms [77] in which all PrefRank methods were implemented. The obtained complete rankings were the same for PROMETHEE II and PrefRank III, as well as PrefRank II, and they were as follows  $KOS \succ MIE \succ TAR \succ KRA \succ LOD \succ LEG \succ SLU \succ STA \succ POM \succ KAM$ . The only difference in the complete ranking produced by PrefRank I was that LEG was preferred to LOD.

The resulting partial rankings in the form of a Hasse diagram are shown in Figure 6.1. Only PROMETHEE I and PrefRank III rank all options in the same order. All the techniques agreed on which SEZs were most preferred, i.e., Kostrzyn-Subice, and the least preferred, i.e., Kamienna Gora, while for PrefRank I, the least preferred was also Pomorze.

In turn, in the intermediate part of the ranking, we observed significant differences. Some techniques showed incomparability between pairs of alternatives, while others exhibited preference relationships between them. For example, PrefRank I. KRA was preferred over TAR, whereas for the remaining approaches, they were indifferent.

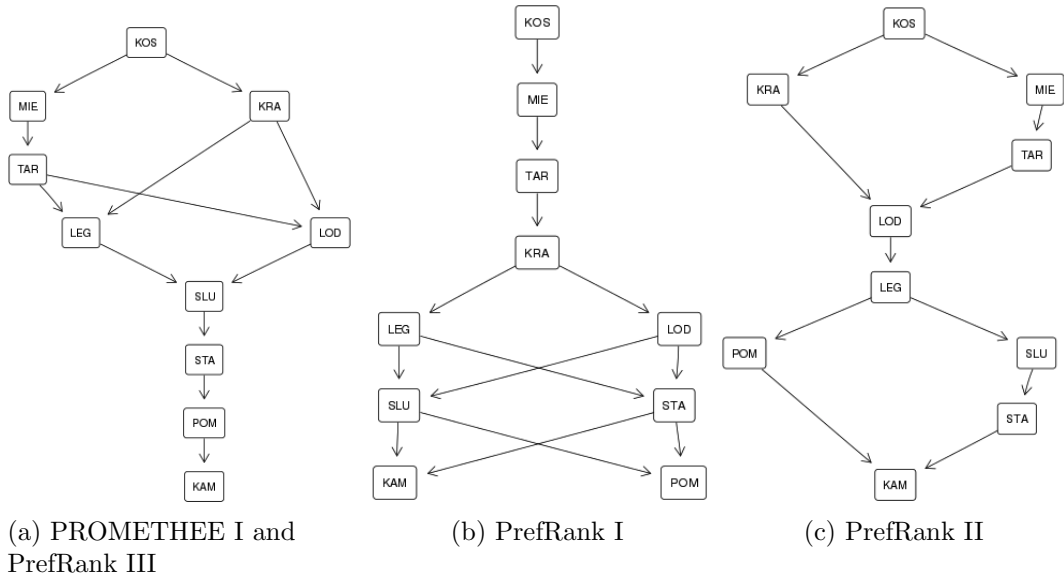


Figure 6.1: Incomplete rankings for the problem of ranking Special Economic Zones in Poland.

## 6.2 ScoreBin

This section introduces the ScoreBin family of procedures that were proposed as part of the work of [76]. In contrast to the PrefRank family, these approaches are designed to exploit crisp outranking relations. These relations are created using the ELECTRE III method (Section 2.8) defined as:

$$\mathbb{1}(a_i, a_k) = \begin{cases} 1 & \text{if } \sigma(a_i, a_k) \geq \lambda \text{ and } i \neq k \\ 0 & \text{else} \end{cases} \quad (6.7)$$

where  $\sigma(a_i, a_k)$  is an outranking credibility value and  $\lambda$  is a cutting level.

As in the PrefRank and NFS approaches, the strength  $S^+(a_i)$  and weakness  $S^-(a_i)$  are determined for each option. However, their calculation and interpretation are different.



In particular, there are two main components included in the strengths (weaknesses), i.e., the part coming from the outranking graph  $G^+(a_i)$  ( $G^-(a_i)$ ) and a bonus (penalty) score  $b_i^+$  ( $b_i^-$ ):

$$S^+(a_i) = b_i^+ + \frac{G^+(a_i)}{\max_{a_k \in A} G^+(a_k)} \text{ and } S^-(a_i) = b_i^- + \frac{G^-(a_i)}{\max_{a_k \in A} G^-(a_k)} \quad (6.8)$$

The component graph is normalized to the range  $[0,1]$  so that the alternative with the highest strength (weakness) in the outranking graph has a score equal to 1. The strength  $S^+(a_i)$  (weakness  $S^-(a_i)$ ) of option  $a_i$  is calculated as the sum of the weights  $\omega^+(a_k)$  ( $\omega^-(a_k)$ ) associated with alternatives  $a_k$  outranked by  $a_i$  (alternatives  $a_k$  that outranked  $a_i$ ):

$$G^+(a_i) = \sum_{k=1}^n \mathbf{1}(a_i, a_k) \omega^+(a_k) \text{ and } G^-(a_i) = \sum_{k=1}^n \mathbf{1}(a_k, a_i) \omega^-(a_k). \quad (6.9)$$

Therefore, if an alternative does not outrank any other, its graph-based strength equals zero. Similarly, if an option is not outranked by any other, its graph-based weakness is also zero. Interpretation of  $\omega^+(a_k)$  and  $\omega^-(a_k)$  is different for each ScoreBin variant. However, irrespective of their definition, alternatives that do not outrank any other option have  $G^+(a_i) = 0$ , while alternatives that are not outranked by any other option have  $G^-(a_i) = 0$ . This means that if  $\omega^+(a_k)$  equals zero, then the outranking of such an option by  $a_i$  would add anything to the  $a_i$  strength. Similarly, being outranked by an option with  $\omega^-(a_k)$  would not increase the alternative's weakness. To ensure the minimal impact of each option  $a_i \in A$  on the strength or weakness of alternatives it is related to, a base bonus  $\alpha^+ \in (0, 1)$  or a penalty  $\alpha^- \in (0, 1)$  is included in its score.

Additionally, this method introduces optional preference information that the DM can provide. It pertains to assigning selected alternatives to the set of strong alternatives  $A_{strong}^* \subseteq A$  or weak alternatives  $A_{weak}^* \subseteq A$ , where one alternative cannot belong to both sets simultaneously. This additional preference information is taken into account as a bonus  $\beta^+ \in \mathbb{R}_{\geq \alpha^+}$  or penalty  $\beta^- \in \mathbb{R}_{\geq \alpha^-}$ . The value of  $\beta^+$  directly affects only the strength of the option, while  $\beta^-$  affects its weakness. Given the bonuses and penalties may serve two purposes, it is possible to consider them under a single variable:

$$b_i^+ = \begin{cases} \beta^+ & \text{if } a_i \in A_{strong}^*, \\ \alpha^+ & \text{else,} \end{cases} \text{ and } b_i^- = \begin{cases} \beta^- & \text{if } a_i \in A_{weak}^*, \\ \alpha^- & \text{else.} \end{cases} \quad (6.10)$$

As the maximum value of the graph component is fixed at 1, the specific values of  $\alpha$  and  $\beta$  can be understood as the ratio of the minimum value that an alternative can receive compared to the graph score obtained by the most preferred option.

The following sections present different variants of the ScoreBin method.

### ScoreBin I

The first variant of ScoreBin increases the strength of ai when it outranks strong alternatives (i.e., with high  $S^+(a_k)$ ) and increases the weakness of ai when it is outranked by weak alternatives (i.e., with high  $S^-(a_k)$ ). Hence, it assumes the following weights:

$$\omega^+(a_k) = S^+(a_k) \text{ and } \omega^-(a_k) = S^-(a_k). \quad (6.11)$$

ScoreBin I draws inspiration from the TrustRank method, which extends the PageRank algorithm with the concept of trust. Websites that have been recognized as trusted receive an additional bonus, which is then propagated to other pages linked to them. The algorithm for determining the strengths and weaknesses is analogous to the method used in PrefRank, and it involves iteratively calculating these scores.

### ScoreBin II

Similarly to PrefRank II, ScoreBin II is inspired by the HITS algorithm. It assumes that the strength is derived from outranking many weak alternatives, and the weakness comes from being outranked by numerous strong alternatives. This requires setting the following weights:

$$\omega^+(a_k) = S^-(a_k) \text{ and } \omega^-(a_k) = S^+(a_k). \quad (6.12)$$

### ScoreBin III

ScoreBin III considers the difficulty and easiness of outranking alternatives. This means that, similarly to PrefRank III and Salsa, an option is considered strong if it outranks an alternative that is also outranked by strong ones. On the other hand, an option is considered weak if it is outranked by an alternative that outranks many weak ones:

$$\omega^+(a_k) = \sum_{l=1}^n \mathbf{1}(a_l, a_k) \cdot S^+(a_l) \text{ and } \omega^-(a_k) = \sum_{l=1}^n \mathbf{1}(a_k, a_l) \cdot S^-(a_l). \quad (6.13)$$

This idea is similar to the concept behind ScoreBin II, where an alternative is judged strong if it outranks weak options, which are outranked by many strong alternatives. The main difference is that ScoreBin II uses weaknesses to calculate strengths and vice versa. This means that the additional preference information provided by the DM, assigning an option to  $A_{strong}^*$ , also indirectly affects the weaknesses of others while assigning alternatives to  $A_{weak}^*$  influences the strengths of others. On the other hand, ScoreBin III uses only the strengths of other alternatives to calculate strengths and only the weaknesses to calculate weaknesses. Therefore, ScoreBin III is similar to ScoreBin I in limiting the impact of positive (negative) information only to alternatives' strengths (weaknesses).

## ScoreBin IV

The fourth variant of ScoreBin combines the first and third counterparts. An alternative is challenging to outrank if few relatively strong options outrank it. On the other hand, an option is easy to outrank if many relatively weak alternatives outrank it. The more challenging an alternative is to outrank, the more favorable it is for the quality of the alternative that does outrank it. The above idea can be implemented using the following weights:

$$\omega^+(a_k) = \frac{1}{\sum_{l=1}^n \mathbb{1}(a_l, a_k) \cdot S^-(a_l)} \text{ and } \omega^-(a_k) = \frac{1}{\sum_{l=1}^n \mathbb{1}(a_k, a_l) \cdot S^+(a_l)}. \quad (6.14)$$

As in ScoreBin I, ScoreBin IV promotes outranking strong options; however, the additional preferential information affects the strengths and weaknesses of other alternatives. For example, if  $a_k \in A_{weak}^*$  outranks  $a_i$ , then  $a_i$  may be considered as less challenging to outrank. Therefore other options that outrank  $a_i$  will have their strength reduced.

### 6.2.1 Measures used for comparing the choice or ranking recommendations

When comparing ranking similarities, the same measures were used for PrefRank, namely NHR, NRD, and Kendall's  $\tau$  coefficient. In addition, because of the comparisons between ScoreBin methods and Electre I, the following metrics were used to compare the set of recommended alternatives and the ranking. The first metric was NHR, which can also be applied to compare the top-rated set from the ranking with the recommended alternatives from the kernel of the graph. The second was the average position (AP) of the set of best options in the complete ranking.

### 6.2.2 Experimental comparison of results attained by different methods exploiting a crisp outranking relation

In this section, the results of experiments examining the similarity between the results obtained from the outranking-based relation exploitation methods, namely ScoreBin I-IV, NFS, QD, and ELECTRE I, will be presented. The experiment involved generating artificial problems consisting of 8 to 20 alternatives (with a step size of 2) and varying evaluation criteria ranging from 3 to 8. The evaluations for these criteria were generated from a uniform distribution within the range of [0-1]. In addition, to check the similarities between different densities of outranking relations, four different values of  $\lambda$  and three different sets of thresholds were tested: low ( $q_j = 0.05, p_j = 0.15, v_j = 0.25$ ), medium ( $q_j = 0.15, p_j = 0.3, v_j = 0.5$ ), and high ( $q_j = 0.25, p_j = 0.45, v_j = 0.75$ ). During these experiments, no additional preferential information was simulated, and the value of the base bonus  $\alpha = 0.1$ .

The experiments revealed that the rankings generated by all the methods were very similar, with the highest similarity observed between methods based on similar concepts, such as ScoreBin I and IV, ScoreBin II and III, and NFS and QD. On the other hand, the most different results were obtained for QD and ScoreBin IV. When considering different problem sizes, the similarity between rankings decreased as the number of alternatives and criteria increased. Compared to ScoreBin I-IV, NFS, QD, and ELECTRE I, the experiments showed relatively low similarity between the graph kernel method and the other methods, with ScoreBin IV generating the most similar recommendations. The average position of the options within the kernel for all methods was around 2.7. When considering all measures, the credibility threshold  $\lambda$  had a negligible impact on the similarity between the rankings. However, for different sets of parameter values for  $q_j$ ,  $p_j$ , and  $v_j$ , it was noticed that the most similar sets of recommended options between ScoreBin I-IV and NFS and QD were associated with low parameter values. In contrast, compared to the graph kernel, the similarity was higher for medium parameter values and lower for high parameter values.

### 6.2.3 Case study concerning evaluation of technological parks in Poland

This section presents the outcomes of a case study evaluating technological parks in Poland [61]. Such parks have created favorable conditions for developing innovative companies, particularly in the advanced technology sector, by providing access to modern infrastructure, scientific knowledge, and financial resources. The objective of our study was to evaluate eleven anonymous technology parks in Poland, which had been assessed based on seven criteria: *sales costs, park buildings' surface, park's localization, total sales, number of services types, overall evaluation of park's management and number of completed projects*.

For this problem, criteria weights were determined using the SRF method, and then an outranking graph was created using the ELECTRE III method (see Figure 6.2 (a)). Next, rankings were created according to ScoreBin I-IV, NFS, and QD methods, and the kernel of the graph was determined, which in this case contained alternatives:  $\{a_8, a_{10}, a_5, a_4, a_7\}$ . We set the value of the base bonus for ScoreBin as  $\alpha = 0.1$  and an enhancement bonus/penalty to  $\beta = 0.8$ .

We considered two scenarios for solving this problem: the first was without and the second with indirect reference information. In the first case, all methods indicated  $a_{10}$  as the best alternative and  $a_9$  as the worst. A Hasse diagram showing the partial ranking for ScoreBin I is shown in Figure 6.2 (b). The resulting rankings for ScoreBin II and III were identical. Then, the greatest similarity can be observed for rankings obtained with ScoreBin IV and NFS. On the other hand, the rankings of ScoreBin I and II (III) differed the most. The average position of the options from the ELECTRE I for all methods was

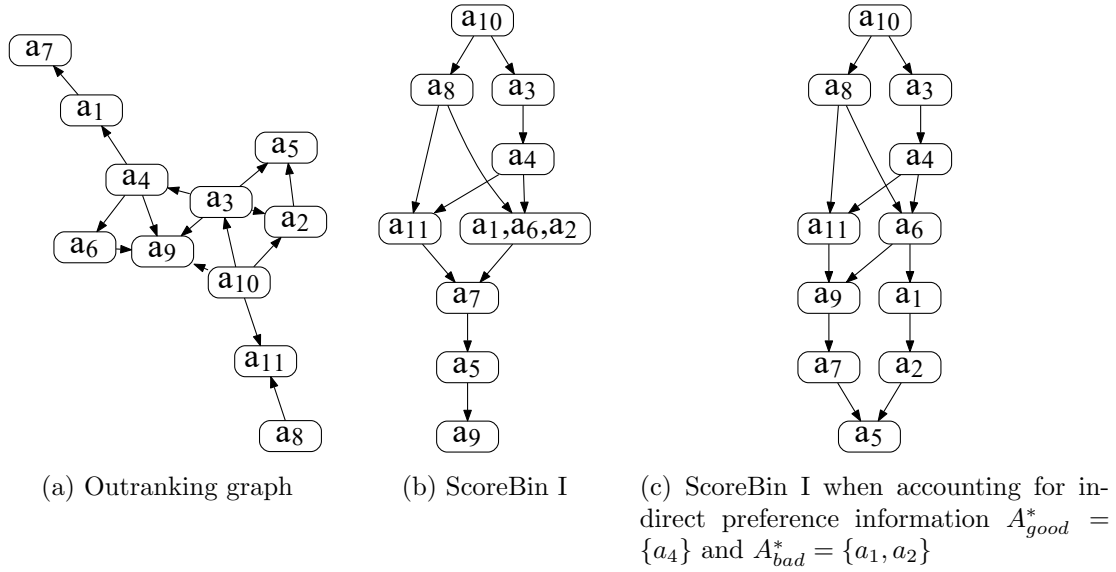


Figure 6.2: Outranking graph (a), incomplete rankings derived with ScoreBin I without additional indirect preference information (b) and with it (c) for the problem of assessing technological parks in Poland.

similar, at about 5.

In contrast, the second scenario contained additional indirect preference information where the DM judged the option  $a_4$  as strong ( $A_{strong}^* = \{a_4\}$ ) while the  $a_1$  and  $a_2$  as weak ( $A_{weak}^* = \{a_1, a_2\}$ ). This preferential information influenced the rankings of all ScoreBin methods. Due to the different ways the preferential information was propagated, the rankings differed much more than in the previous scenario. In the case of partial ranking, all methods still considered alternative  $a_{10}$  as the most preferred, but ScoreBin II-IV methods additionally identify the  $a_4$  option as such. Comparing both scenarios, in the case of ScoreBin I without preferential information (Figure 6.2 (b)), for example, alternative  $a_5$  was preferred over  $a_9$ . However, after adding additional information (Figure 6.2 (c)), the preferred direction for this pair changed, which is related to recognizing alternative  $a_2$  as a weak option.

#### 6.2.4 Robustness analysis

ScoreBin methods have two parameters: the minimum alternative contribution  $\alpha$  and strength/weakness enhancement  $\beta$ . The choice of these two values determines the final result obtained. In order to check the stability of the solution, a robustness analysis was performed. It checks the possible solutions in the feasible space of these parameters (A, B). The results of this analysis can be presented in the form of Rank Acceptability Indices (RAIs), i.e., the share of parameters for which alternative  $a_i$  was given  $r$ -th rank. To estimate this value, we used Monte Carlo simulation, which samples the values of the parameters  $\alpha \in [0.005, 1]$  with a step of 0.005 and  $\beta \in (\alpha, 1]$  with a step of 0.005. The

results of the analysis using RAIs for the preference information  $A_{strong}^* = \{a_4\}$  and  $A_{weak}^* = \{a_1, a_2\}$  for the ScoreBin I method and complete ranking are shown in Table 6.1. Meanwhile, Figure 6.3 shows the obtained ranks of the alternative  $a_2$  for different values of  $\alpha$  and  $\beta$ .

From the RAIs table, it can be observed that regardless of the parameter values,  $a_8$  always occupied position 4. On the other hand,  $a_2$  could be positioned anywhere from 6 to 11, with position 6 being achieved for small values of  $\alpha$  and  $\beta$ . In contrast, the worst position occurred for high values of  $\beta$  and low  $\alpha$  when the information about assigning this alternative to weak options had the most significant influence. Analyzing only the strengths,  $a_2$  consistently held rank 4 because it was not assigned to strong options, nor was any variant it outranks included in that set.

Table 6.1: Rank Acceptability Indices (in %) and expected ranks  $ER$  obtained with ScoreBin I for technological parks in Poland in the scenario accounting for additional preference information.

Rank	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$	$a_{11}$
1	-	-	-	3.6	-	-	-	-	-	96.4	-
2	-	-	93.0	4.8	-	-	-	-	-	2.1	-
3	-	-	7.0	91.6	-	-	-	-	-	1.4	-
4	-	-	-	-	-	-	-	100.0	-	-	-
5	0.9	-	-	-	-	83.8	-	-	-	-	16.2
6	30.6	1.3	-	-	-	15.0	2.3	-	-	-	50.0
7	20.5	11.6	-	-	-	1.3	48.3	-	10.9	-	7.5
8	37.6	27.8	-	-	-	-	20.1	-	1.8	-	12.7
9	10.1	50.4	-	-	2.5	-	19.2	-	4.0	-	13.7
10	0.3	7.2	-	-	78.4	-	8.5	-	5.6	-	-
11	-	1.6	-	-	19.1	-	1.5	-	77.7	-	-
$ER$	7.26	8.55	2.07	2.88	10.17	5.18	7.87	4.0	10.37	1.05	6.58

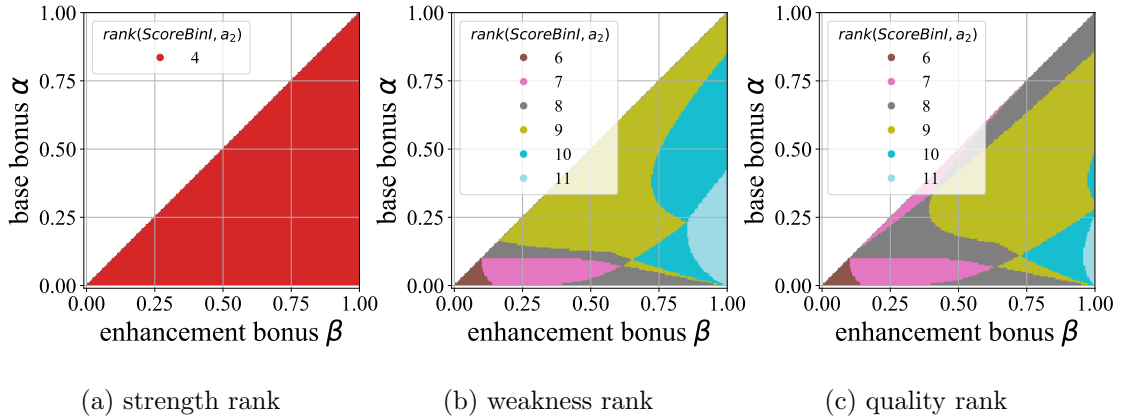


Figure 6.3: Ranks attained by alternative  $a_2$  for uniformly distributed values of parameters  $\alpha$  and  $\beta$  for the problem of assessing technological parks in Poland.

### 6.3 Comparison of PrefRank and ScoreBin methods

This dissertation presents two families of methods for exploiting the weighted preference relation PrefRank and the binary outranking relation ScoreBin. These families were inspired by algorithms derived from web mining and scoring websites. Despite similar inspirations, the two families of methods are significantly different, and ScoreBin methods are an extension of the concepts proposed in PrefRank methods. The main difference is that PrefRank was proposed for a valued relation where we have a numerical score indicating the degree of preference for each pair of alternatives. On the other hand, ScoreBin has been proposed for crisp outranking relations in which we can only determine whether a relation exists. In addition, ScoreBin, compared to PrefRank, extends the ability to control the final ranking by introducing additional indirect preference information. It allows the DM to determine whether an alternative is strong or weak holistically. This information directly affects the option to which it applies and not directly the others through the relation present in the outranking graph. These differences influence the way strengths and weaknesses are determined for the two families. In addition, the ScoreBin family of methods has been extended over the PrefRank methods with a variant based on how challenging the option is to be outranked by other alternatives.





# Chapter 7

## Summary

This chapter summarizes the research and the realization of the dissertation’s aims and outlines perspectives for further study. We refer to the three research areas described in the introduction of this dissertation.

MCDA provides various tools to support DMs in the decision-making process. These methods focus on solving problems such as ranking options in terms of preferences, selecting the best ones, or sorting them into pre-defined, preference-ordered classes. The options considered are evaluated on several potentially conflicting criteria. Under such conditions, an ideal solution typically does not exist. Therefore, eliciting the DM’s preferences is necessary to build the model according to his/her individual value system.

Decision-making problems are becoming increasingly complex due to the increasing number of options to consider, the variety of types of evaluation criteria, and the inconsistencies and uncertainties in the judgments provided by the DM. In addition, the DM may consider many related decisions simultaneously for the same set of options. Moreover, for each decision made, there should be a rationale indicating why that particular decision was made and the impact of individual performances on the decision. At the same time, the decision-making process itself should be easy to interpret by the DM.

This dissertation proposes several MCDA methods addressing these problems that draw inspiration from various AI fields. Five original papers were prepared as part of the research, of which, as of the state (May 31, 2023), three have been published, and two have been submitted for publication. The research was divided into three aims: methods inspired by deep neural networks, machine learning, and web mining.

In the first research area, eight MCDA methods inspired by deep neural networks were proposed. They introduced explainable and interpretable preference learning methods to address the sorting problem. The methods enable the effective reconstruction of DM’s holistic judgments by learning the method’s parameters from large quantities of reference data for which real decisions are known. Artificial Neural Networks (ANNs) are suggested as a computational approach for preference disaggregation. The resulting

model is modified to permit complex monotonic transformations of evaluations while considering constraints on the directions of preferences for individual criteria. The presented techniques allow for a detailed analysis of the model, providing information on the influence of each criterion or criterion group. Additionally, they enable determining which performance difference is critical and which is insignificant. These pieces of information facilitate the straightforward interpretation of the data processing process. The intuitive threshold-based sorting procedure makes it possible to provide reliable explanations for allocating an alternative to a specific class. Specifically, approaches based on scores, distances, and outranking were investigated, incorporating different compensation levels, including interactions between criteria or curvatures of marginal functions.

The main advantage of the proposed methods is the ability to infer all model parameters from the indirect preference information in the form of example class assignments. In addition, more flexible per-criterion (value, preference, concordance, or discordance) functions are used, allowing for more accurate alignment with the input data. As a result, there is no need for the DM to choose the shape of these functions arbitrarily. Secondly, using deep learning techniques enables learning from large and inconsistent preference information, which most traditional methods based on Mixed-Integer Linear Programming would not process within an acceptable time frame. This ability has been demonstrated on various benchmark problems containing over one thousand alternatives or scenarios that involve comparing several million pairs of options. The predictive performance of the proposed methods, in particular ANN-UTADIS, ANN-Ch-Uncons., and ANN-PROMETHEE, is competitive with other state-of-the-art preference learning methods.

In the second research area, two problems were addressed: modeling non-monotonic criteria in additive value function models and dealing with sorting problems with multiple interdependent decisions. Two independent approaches to handling non-monotonic criteria were proposed. The first approach focused on controlling the complexity of these functions by minimizing the number of changes in monotonicity. A wide range of criteria types was explicitly considered, including gain and cost, level-monotonic shapes, monotonic functions without a pre-defined preference direction, A- and V-types where one part is non-increasing, and the other part is non-decreasing, as well as a function without any monotonicity constraints. In the other approach, the non-monotonic criterion combines two components of the non-decreasing and non-increasing types. The resulting marginal function can be of any shape, also providing an understandable justification for this shape.

On the one hand, the outcome of disaggregation of the DM's preferences can be a single representative instance. Such a model provides unequivocal class assignments, along with a justification of the impact of each evaluation on the resulting decision. In addition, it allows for analysis and interpretation of the model by providing information

on what values would have to change so that the classification would be different. On the other hand, an analysis of the robustness of assignments of non-reference alternatives was performed. It determines the possible and necessary classes in a set of compatible instances of the sorting model. An additional contribution is a method for dealing with multi-decision sorting. This approach considers interconnected sorting problems using an additive value function with intra- and inter-decision constraints. The model is built by disaggregating a subset of sample alternatives classified into a single class on each decision attribute.

An analysis was performed using the proposed methods for real sorting problems related to managing exposure to engineered nanomaterials. The first study's analysis concerned predicting precaution levels for needing a respirator. The representative model identified the criteria of detection limits, airborne, and duration of exposure as those with the most significant contribution to the comprehensive value. In contrast, the nanomaterial quantity, exposure frequency, and engineering controls had a minor influence. The second study considers four related precautions that can be used to reduce risk: a respirator, a fume hood with and without a HEPA filter, and a HEPA vacuum cleaner. Airborne capacity, detection limit, and exposure limit were attributed to the highest maximal share in the comprehensive values of alternatives. The marginal value functions obtained for each decision were similar, particularly for the HEPA filter precautions. In contrast, they differed significantly for the fume hood with or without the HEPA, confirming their complementarity.

The last research area was to propose methods for exploiting relations between alternatives that consider dependencies between them. In this dissertation, two families of methods, called PrefRank and ScoreBin, were proposed for analyzing different types of relations. They were inspired by both the Net Flow Score method, which considers option strengths and weaknesses, and the graph analysis algorithms originally proposed within web structure mining. The PrefRank methods are used to analyze valued preference relations, while ScoreBin is used for crisp outranking relations. The individual variants within these two families differ in the weighting scheme implemented when aggregating the results of pairwise comparisons. It allows one to capture different aspects of the option, such as the difficulty and ease of outranking or preferring each alternative and whether it is relatively good or bad. Furthermore, ScoreBin includes optional preference information allowing the DM to specify a subset of strong and weak alternatives.

The proposed methods were compared regarding the similarity of results with other state-of-the-art relations exploitation techniques such as NFS, Qualification Distillation, and ELECTRE I on various simulated decision problems. These experiments showed the greatest similarities between the NFS methods and PrefRank III, ScoreBin II and III, I and IV, and NFS and QD.

Both groups of methods have been tested in real problems. PrefRank was consid-

ered in the problem of ranking Special Economic Zones in Poland. All techniques indicated that Kostrzyn and Ślubice were the most preferred areas for financial growth and job creation. ScoreBin methods were demonstrated on the problem of identifying the best-managed technological parks in Poland, which generated the highest profits and supported the development of both industry and research.

This dissertation proposes decision-support methods that combine ideas from different areas of AI. These studies are significant due to the increasing utilization of intelligent systems for processing and analyzing ever-larger data sets. In these systems, it is often necessary to use techniques that can justify their results, and the analysis process itself is easily interpretable and consistent with the DM's preferences. Moreover, these solutions must deal with the DM's uncertainty and handle complex problems considering non-monotonic preferences over criteria. There are also situations where multiple interdependent decisions must be made for each option. Finally, when considering various scenarios for selection, recommendations should consider the relationships between alternatives and their resulting strengths and weaknesses, as well as how good or weak the alternative is compared.

The experiments and analysis carried out as part of this dissertation provided evidence of the methodological contributions. However, it is essential to acknowledge and address the limitations of these experiments, which will be discussed next.

First, the thesis presents methods for learning preferences from large reference datasets. The experiments presented prove their usefulness in the context of MCDA where traditional methods process modestly-sized datasets [103]. In contrast, other areas of ML process significantly larger datasets. For this reason, it would be valuable to test their accuracy in reconstructing preferential information for such data volumes.

Furthermore, two methods for modeling non-monotonic criteria have been proposed. However, the resulting partial functions have not been compared among themselves or with other state-of-the-art methods. Conducting an analysis comparing criterion complexity, computation time, and overall model quality for these algorithms would allow discovering which models are better suited for different applications. Still, the method's underlying idea and capabilities should be decisive in this aspect. The above discussion can be a potential direction for future research. In addition, the possibilities offered by other areas of AI discussed in this work may also be a start for future study. The following section delves into a discussion of these ideas.

First, as presented in this dissertation, inspiration from different areas of AI can allow decision-making problems to be solved more efficiently. Therefore, it would be beneficial to explore the possibility of utilizing techniques and tools from other areas in the context of decision support. In particular, a technique such as deep transfer learning [98] could transfer information about user preferences between related problems. Additionally, natural techniques that could be utilized in decision-making processes are active learning

methods [87], which would allow for providing DM’s preferences in sequential order, and the model needs to be updated at each step. Federated learning [60] or blockchain [63] techniques could find application in group decision-making methods, where consensus must be reached between different decision-makers.

Another direction for future research is to propose neural preference learning algorithms for other intuitive MCDA approaches. In this context, it would be possible to explore methods that incorporate other types of criteria or interactions between them or that use boundary or characteristic class profiles for sorting.

Third, the ANN-based and proposed methods for handling non-monotonicity could be applied to ranking problems. For this purpose, the preference information would take the form of pairwise comparisons of alternatives, and there would be no need to determine a threshold in the sorting procedure.

Furthermore, when considering an approach that controls the complexity of non-monotonic criteria, extending them to other shapes would be possible. For example, by incorporating polynomial and spline transformations, whose interpretation is essential for real-world problems [95].

Ultimately, computing strengths and weaknesses in PrefRank and ScoreBin methods is carried out similarly. However, it would be possible to simultaneously utilize different combinations of these methods and aggregate their results by averaging them or by examining the spaces of consensus and disagreement [78]. Similarly, in the case of methods inspired by ANN, it is possible to combine multiple architectures into one and aggregate the results into a comprehensive quality measure. The final decision could be determined through either majority voting or weighted voting, where the weights are determined during training.



# Bibliography

- [1] S. Angilella, S. Corrente, S. Greco, and R. Słowiński. Multiple criteria hierarchy process for the choquet integral. In *Evolutionary Multi-Criterion Optimization: 7th International Conference, EMO 2013, Sheffield, UK, March 19-22, 2013. Proceedings 7*, pages 475–489. Springer, 2013.
- [2] D. Bouyssou, T. Marchant, M. Pirlot, A. Tsoukias, and P. Vincke. *Evaluation and decision models with multiple criteria: Stepping stones for the analyst*, volume 86. Springer Science & Business Media, 2006.
- [3] D. Bouyssou and M. Pirlot. An axiomatic approach to tactic. 2006.
- [4] R. Brafman. Preferences, planning, and control. In *Principles of Knowledge Representation and Reasoning*, Proceedings of the International Conference on Knowledge Representation and Reasoning, pages 2–5, United States, Jan. 2008. Institute of Electrical and Electronics Engineers. 11th International Conference on Principles of Knowledge Representation and Reasoning, KR 2008 ; Conference date: 16-09-2008 Through 19-09-2008.
- [5] S. J. Brams and P. C. Fishburn. Approval voting. *American Political Science Review*, 72(3):831–847, 1978.
- [6] J.-P. Brans and Y. De Smet. Promethee methods. In S. Greco, M. Ehrgott, and J. R. Figueira, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 187–219. Springer New York, New York, NY, 2016.
- [7] J.-R. Cano, P. A. Gutiérrez, B. Krawczyk, M. Woźniak, and S. García. Monotonic classification: An overview on algorithms, performance measures and data sets. *Neurocomputing*, 341:168–182, 2019.
- [8] H. Chen and M. Chau. Web mining: Machine learning for web applications. *Annual Review of Information Science and Technology*, 38(1):289–329, 2004.
- [9] W. Cheng, E. Hüllermeier, and K. J. Dembczynski. Bayes optimal multilabel classification via probabilistic classifier chains. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 279–286, 2010.

- [10] M. Cinelli, M. Kadziński, M. Gonzalez, and R. Słowiński. How to support the application of multiple criteria decision analysis? let us start with a comprehensive taxonomy. *Omega*, 96:102261, 2020.
- [11] M. Coppola, J. Guo, E. Gill, and G. C. H. E. de Croon. The PageRank algorithm as a method to optimize swarm behavior through local analysis. *Swarm Intelligence*, 13(3):277–319, 2019.
- [12] S. Corrente, S. Greco, M. Kadziński, and R. Słowiński. Robust ordinal regression in preference learning and ranking. *Machine Learning*, 93:381–422, 2013.
- [13] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.
- [14] K. Dembczyński, R. Słowiński, et al. Learning rule ensembles for ordinal classification with monotonicity constraints. *Fundamenta Informaticae*, 94(2):163–178, 2009.
- [15] L. Deng and D. Yu. Deep learning: methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4):197–387, 2014.
- [16] D. K. Despotis and C. Zopounidis. Building additive utilities in the presence of non-monotonic preferences. *Advances in multicriteria analysis*, pages 101–114, 1995.
- [17] L. C. Dias and V. Mousseau. Eliciting multi-criteria preferences: Electre models. *Elicitation: The science and art of Structuring Judgement*, pages 349–375, 2018.
- [18] M. Doumpos. Learning non-monotonic additive value functions for multicriteria decision making. *OR spectrum*, 34(1):89–106, 2012.
- [19] 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.
- [20] M. Doumpos and C. Zopounidis. Preference disaggregation and statistical learning for multicriteria decision support: A review. *European Journal of Operational Research*, 209(3):203–214, 2011.
- [21] M. Doumpos and C. Zopounidis. Disaggregation approaches for multicriteria classification: an overview. *Preference Disaggregation in Multiple Criteria Decision Analysis: Essays in Honor of Yannis Siskos*, pages 77–94, 2018.
- [22] A. Fallah Tehrani, W. Cheng, K. Dembczyński, and E. Hüllermeier. Learning monotone nonlinear models using the choquet integral. *Machine Learning*, 89:183–211, 2012.



- [23] J. Figueira, S. Greco, B. Roy, and R. Słowiński. An overview of ELECTRE methods and their recent extensions. *Journal of Multi-Criteria Decision Analysis*, 20(1-2):61–85, 2013.
- [24] J. Figueira and B. Roy. Determining the weights of criteria in the ELECTRE type methods with a revised Simos’ procedure. *European journal of operational research*, 139(2):317–326, 2002.
- [25] J. R. Figueira, S. Greco, and B. Roy. Electre-score: A first outranking based method for scoring actions. *European Journal of Operational Research*, 297(3):986–1005, 2022.
- [26] J. R. Figueira, S. Greco, B. Roy, and R. Słowiński. An overview of electre methods and their recent extensions. *Journal of Multi-Criteria Decision Analysis*, 20(1-2):61–85, 2013.
- [27] J. R. Figueira, V. Mousseau, and B. Roy. Electre methods. *Multiple criteria decision analysis: State of the art surveys*, pages 155–185, 2016.
- [28] J. François, S. Wang, R. State, and T. Engel. Bottrack: tracking botnets using netflow and pagerank. In *NETWORKING 2011: 10th International IFIP TC 6 Networking Conference, Valencia, Spain, May 9-13, 2011, Proceedings, Part I 10*, pages 1–14. Springer, 2011.
- [29] J. Fürnkranz and E. Hüllermeier. Preference learning: An introduction. In J. Fürnkranz and E. Hüllermeier, editors, *Preference Learning*, pages 1–17. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [30] J. Fürnkranz, E. Hüllermeier, C. Rudin, R. Slowinski, and S. Sanner. Preference Learning (Dagstuhl Seminar 14101). *Dagstuhl Reports*, 4(3):1–27, 2014.
- [31] J. Geldermann, T. Spengler, and O. Rentz. Fuzzy outranking for environmental assessment. case study: iron and steel making industry. *Fuzzy sets and systems*, 115(1):45–65, 2000.
- [32] 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.
- [33] 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.
- [34] G. Gigerenzer and D. G. Goldstein. Reasoning the fast and frugal way: models of bounded rationality. *Psychological review*, 103(4):650, 1996.

- [35] S. Greco, M. Kadziński, and R. Słowiński. Selection of a representative value function in robust multiple criteria sorting. *Computers & Operations Research*, 38(11):1620–1637, 2011.
- [36] S. Greco, B. Matarazzo, and R. Słowiński. Decision rule approach. *Multiple criteria decision analysis: state of the art surveys*, pages 497–552, 2016.
- [37] S. Greco, V. Mousseau, and R. Słowiński. Multiple criteria sorting with a set of additive value functions. *European Journal of Operational Research*, 207(3):1455–1470, 2010.
- [38] M. Guo, Q. Zhang, X. Liao, F. Y. Chen, and D. D. Zeng. A hybrid machine learning framework for analyzing human decision-making through learning preferences. *Omega*, 101:102263, 2021.
- [39] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 576–587, 2004.
- [40] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
- [41] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant. *Applied logistic regression*. Wiley, New York, 2000.
- [42] Y.-C. Hu. Bankruptcy prediction using electre-based single-layer perceptron. *Neurocomputing*, 72(13-15):3150–3157, 2009.
- [43] C.-L. Hwang and K. Yoon. *Methods for Multiple Attribute Decision Making*, pages 58–191. Springer Berlin Heidelberg, Berlin, Heidelberg, 1981.
- [44] E. Jacquet-Lagrange and Y. Siskos. Preference disaggregation: 20 years of mcda experience. *European Journal of Operational Research*, 130(2):233–245, 2001.
- [45] S. Jeble, S. Kumari, and Y. Patil. Role of big data in decision making. *Operations and Supply Chain Management: An International Journal*, 11(1):36–44, 2017.
- [46] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142, 2002.
- [47] M. Kadziński and K. Ciomek. Integrated framework for preference modeling and robustness analysis for outranking-based multiple criteria sorting with electre and promethee. *Information Sciences*, 352:167–187, 2016.

- [48] M. Kadziński, K. Martyn, M. Cinelli, R. Słowiński, 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.
- [49] M. Kadziński, K. Martyn, M. Cinelli, R. Słowiński, 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*, 218:106879, 2021.
- [50] M. Kadziński and M. Michalski. Scoring procedures for multiple criteria decision aiding with robust and stochastic ordinal regression. *Computers & Operations Research*, 71:54–70, 2016.
- [51] M. Kadziński and T. Tervonen. Robust multi-criteria ranking with additive value models and holistic pair-wise preference statements. *European Journal of Operational Research*, 228(1):169–180, 2013.
- [52] M. Kadziński and T. Tervonen. Stochastic ordinal regression for multiple criteria sorting problems. *Decision Support Systems*, 55(1):55 – 66, 2013.
- [53] D. Kahneman. *Thinking, fast and slow*. Farrar, Straus and Giroux, New York, 2011.
- [54] R. L. Keeney and H. Raiffa. *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge university press, 1993.
- [55] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.
- [56] S. Khan and T. Yairi. A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, 107:241–265, 2018.
- [57] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [58] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
- [59] T. Kliegr. Uta-nm: Explaining stated preferences with additive non-monotonic utility functions. *Preference Learning*, page 56, 2009.
- [60] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016.

- [61] B. Kowalak. Benchmarking of technological parks in Poland (in Polish). Technical report, Polska Agencja Rozwoju Przedsiębiorczości, Warsaw, Poland, 2010.
- [62] V. Krishnan and R. Raj. Web spam detection with anti-trust rank. In *AIRWeb*, volume 6, pages 37–40. Seattle, WA, 2006.
- [63] R. Kumar, A. A. Khan, J. Kumar, N. A. Golilarz, S. Zhang, Y. Ting, C. Zheng, W. Wang, et al. Blockchain-federated-learning and deep learning models for covid-19 detection using ct imaging. *IEEE Sensors Journal*, 21(14):16301–16314, 2021.
- [64] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a Social Network or a News Media? In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pages 591–600, New York, NY, USA, 2010. Association for Computing Machinery.
- [65] N. Landwehr, M. Hall, and E. Frank. Logistic Model Trees. In *European Conference on Machine Learning*, pages 241–252. Springer, 2003.
- [66] R. Lempel and S. Moran. Salsa: the stochastic approach for link-structure analysis. *ACM Transactions on Information Systems (TOIS)*, 19(2):131–160, 2001.
- [67] A. Leroy, V. Mousseau, and M. Pirlot. Learning the parameters of a multiple criteria sorting method. In *International conference on algorithmic decision theory*, pages 219–233. Springer, 2011.
- [68] J. Liu, M. Kadziński, X. Liao, and X. Mao. Data-driven preference learning methods for value-driven multiple criteria sorting with interacting criteria. *INFORMS Journal on Computing*, 33(2):586–606, 2021.
- [69] J. Liu, M. Kadziński, X. Liao, X. Mao, and Y. Wang. A preference learning framework for multiple criteria sorting with diverse additive value models and valued assignment examples. *European Journal of Operational Research*, 286(3):963–985, 2020.
- [70] J. Liu, X. Liao, M. Kadziński, and R. Słowiński. 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.
- [71] J. Liu, X. Liao, W. Zhao, and N. Yang. A classification approach based on the outranking model for multiple criteria ABC analysis. *Omega*, 61:19–34, 2016.
- [72] I. Loshchilov and F. Hutter. Fixing weight decay regularization in adam. 2017.

- [73] B. Malakooti and Y. Q. Zhou. Feedforward artificial neural networks for solving discrete multiple criteria decision making problems. *Management Science*, 40(11):1542–1561, 1994.
- [74] K. Martyn and M. Kadziński. Deep preference learning for multiple criteria decision analysis. *European Journal of Operational Research*, 305(2):781–805, 2023.
- [75] K. Martyn, M. Martyn, and M. Kadziński. PrefRank: a family of multiple criteria decision analysis methods for exploiting a valued preference relation. *Expert Systems With Applications*, 2023. Submitted.
- [76] K. Martyn, M. Martyn, and M. Kadziński. ScoreBin: scoring alternatives based on a crisp outranking relation with an application to the assessment of Polish technological parks. *Information Sciences*, 2023. Submitted.
- [77] P. Meyer and S. Bigaret. Diviz: A Software for Modeling, Processing and Sharing Algorithmic Workflows in MCDA. *Intelligent Decision Technologies*, 6(4):283–296, 2012.
- [78] G. Miebs and M. Kadziński. Heuristic algorithms for aggregation of incomplete rankings in multiple criteria group decision making. *Information Sciences*, 560:107–136, 2021.
- [79] T. Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.
- [80] C. Molnar. *Interpretable Machine Learning*. 2 edition, 2022.
- [81] J. L. Morrison, R. Breitling, D. J. Higham, and D. R. Gilbert. Generank: using search engine technology for the analysis of microarray experiments. *BMC bioinformatics*, 6:1–14, 2005.
- [82] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.
- [83] S. Naidu. *Towards Sustainable Development of Nanomanufacturing*. PhD thesis, University of Tennessee, Knoxville, 2012.
- [84] A.-L. Olteanu12 and P. Meyer. Inferring the parameters of a majority rule sorting model with vetoes on large datasets. 2014.
- [85] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

- [86] I. Portugal, P. Alencar, and D. Cowan. The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, 97:205–227, 2018.
- [87] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.
- [88] J. Rezaei. Piecewise linear value functions for multi-criteria decision-making. *Expert Systems with Applications*, 98:43–56, 2018.
- [89] B. Roy. The outranking approach and the foundations of electre methods. *Theory and decision*, 31:49–73, 1991.
- [90] B. Roy. *Multicriteria Methodology for Decision Aiding*. Kluwer Academic, Dordrecht, 1996.
- [91] Roy, B. Classement et choix en présence de points de vue multiples. *R.I.R.O.*, 2(8):57–75, 1968.
- [92] S. Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.
- [93] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.
- [94] Y. Siskos, E. Grigoroudis, and N. F. Matsatsinis. Uta methods. *Multiple criteria decision analysis: State of the art surveys*, pages 315–362, 2016.
- [95] O. Sobrie, N. Gillis, V. Mousseau, and M. Pirlot. Uta-poly and uta-splines: additive value functions with polynomial marginals. *European Journal of Operational Research*, 264(2):405–418, 2018.
- [96] O. Sobrie, V. Mousseau, and M. Pirlot. Learning monotone preferences using a majority rule sorting model. *International Transactions in Operational Research*, 26(5):1786–1809, 2019.
- [97] M. Szeląg, S. Greco, and R. Słowiński. Variable consistency dominance-based rough set approach to preference learning in multicriteria ranking. *Information Sciences*, 277:525–552, 2014.
- [98] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu. A survey on deep transfer learning. In *Artificial Neural Networks and Machine Learning–ICANN 2018: 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4–7, 2018, Proceedings, Part III 27*, pages 270–279. Springer, 2018.

- [99] A. F. Tehrani, W. Cheng, K. Dembczyński, and E. Hüllermeier. Learning monotone nonlinear models using the Choquet integral. *Machine Learning*, 89(1):183–211, 2012.
- [100] T. Tervonen and R. Lahdelma. Implementing stochastic multicriteria acceptability analysis. *European Journal of Operational Research*, 178(2):500–513, 2007.
- [101] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.
- [102] W. Waegeman, B. De Baets, and L. Boullart. Kernel-based learning methods for preference aggregation. *4OR*, 7:169–189, 2009.
- [103] J. Wallenius, J. S. Dyer, P. C. Fishburn, R. E. Steuer, S. Zionts, and K. Deb. Multiple criteria decision making, multiattribute utility theory: Recent accomplishments and what lies ahead. *Management science*, 54(7):1336–1349, 2008.
- [104] M. Waltz and K. Fu. A heuristic approach to reinforcement learning control systems. *IEEE Transactions on Automatic Control*, 10(4):390–398, 1965.
- [105] R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on systems, Man, and Cybernetics*, 18(1):183–190, 1988.
- [106] M.-L. Zhang and Z.-H. Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2013.
- [107] S. Zheng, Y. Song, T. Leung, and I. Goodfellow. Improving the robustness of deep neural networks via stability training. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4480–4488, 2016.
- [108] C. Zopounidis and M. Doumpos. PREFDIS: a multicriteria decision support system for sorting decision problems. *Computers & Operations Research*, 27(7-8):779–797, 2000.
- [109] C. Zopounidis and M. Doumpos. Multicriteria classification and sorting methods: A literature review. *European Journal of Operational Research*, 138(2):229–246, 2002.





# Publication reprints



## Publication [P1]

M. Kadziński, K. Martyn, M. Cinelli, R. Słowiński, 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,

DOI: 10.1016/j.ijar.2019.11.007.

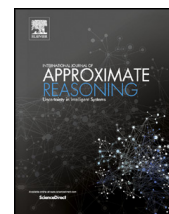
Number of citations<sup>1</sup>:

- according to Web of Science: 27
- according to Google Scholar: 31

---

<sup>1</sup>as on June 1, 2023





# Preference disaggregation for multiple criteria sorting with partial monotonicity constraints: Application to exposure management of nanomaterials



Miłosz Kadziński<sup>a,\*</sup>, Krzysztof Martyn<sup>a</sup>, Marco Cinelli<sup>a,1</sup>, Roman Słowiński<sup>a,b</sup>, Salvatore Corrente<sup>c</sup>, Salvatore Greco<sup>c,d</sup>

<sup>a</sup> Institute of Computing Science, Poznań University of Technology, Poznań, Poland

<sup>b</sup> Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

<sup>c</sup> Department of Economics and Business, University of Catania, Catania, Italy

<sup>d</sup> CORL, Portsmouth Business School, University of Portsmouth, Portsmouth, United Kingdom

## ARTICLE INFO

### Article history:

Received 21 June 2019

Received in revised form 12 September 2019

Accepted 9 November 2019

Available online 15 November 2019

### Keywords:

Multiple criteria sorting  
Incomplete information  
Preference disaggregation  
Non-monotonicity  
Nanomaterials  
Exposure management

## ABSTRACT

We propose a novel approach to multiple criteria sorting incorporating a threshold-based value-driven procedure. The parameters deciding upon the shape of marginal value functions and separating class thresholds are inferred through preference disaggregation from the Decision Maker's incomplete assignment examples and partial requirements on the type of (non-)monotonicity for each marginal value function. These types include standard monotonic shapes, level-monotonic functions, A- and V-types combining increasing and decreasing value trends, and unknown monotonicity constraints. A representative instance of the sorting model compatible with the preference information is constructed by solving a dedicated Mixed-Integer Linear Programming problem. Its complexity is controlled by minimizing the number of changes in monotonicity between all subsequent sub-intervals of marginal value functions. The assignments derived using the constructed representative model are validated against the outcomes of robustness analysis. The proposed method is applied to a real-world problem of exposure management of engineered nanomaterials. We develop a model for predicting precaution level while handling nanomaterials in certain conditions using a respirator. The model captures interrelations between ten accounted evaluation criteria, including both monotonic and non-monotonic criteria, and the recommended class assignment. This makes it suitable for the management of exposure scenarios, which have not been directly judged by the experts.

© 2019 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. Introduction

Multiple Criteria Decision Aiding (MCDA) is one of the fastest developing sub-fields of computer science and operational research [16]. Its importance derives from offering a diversity of approaches for structuring decision problems involving multiple criteria and carrying forward their solution. As the criteria used to represent pertinent viewpoints on the quality

\* Corresponding author at: Institute of Computing Science, Poznań University of Technology, Piotrowo 2, 60-965 Poznań, Poland.

E-mail address: [miłosz.kadzinski@cs.put.poznan.pl](mailto:miłosz.kadzinski@cs.put.poznan.pl) (M. Kadziński).

<sup>1</sup> Current address: Land and Materials Management Division, National Risk Management Research Laboratory, U.S. Environmental Protection Agency (EPA), Cincinnati, Ohio, USA.

of considered alternatives usually do not align to indicate the most preferred alternative, arriving at the problem's solution requires involvement of Decision Maker (DM). (S)he is expected to exchange information with the method in a way ensuring that a recommendation constructed in the course of a decision aiding process is feasible and consistent with his/her value system [36].

The two major components of decision aiding approaches are responsible for querying the DM for suitable inputs and performing the analysis of his/her feedback to produce a recommendation in function of the specific problem to solve [23]. When it comes to the required inputs, their characteristics may be two-fold. On the one hand, they may be imposed by the context of a particular decision problem, hence referring to the characteristics of criteria, type of performances, or specificity of expected results. On the other hand, the inputs may represent the DM's subjective preferences indicating his/her priorities, requirements, and choices that should be respected when deriving the recommendation. Processing such diverse information consists in constructing a preference model of the DM in the context of the considered decision problem, and exploiting this model to produce numerical and other arguments supporting the recommendation.

Most traditional MCDA methods incorporate complete information about the problem and model parameters. Such information takes the form of precise performances of alternatives, well-defined preference directions for all criteria, exact requirements imposed on the provided outcomes, or exact values of preference model parameters [37]. The assumption on availability of such complete information may be questioned on many grounds. When it comes to the model parameters, it may not be possible to obtain their reliable exact estimates from the DM due to a misunderstanding of their meaning, a prohibitively high cognitive effort related to their elicitation, a lack of DM's confidence in providing precise inputs, or an application of some arbitrary transformation of the incomplete judgments to the complete ones (e.g., converting ordinal scales of criteria to cardinal weights). For this reason, the interest in recently developed MCDA approaches has been shifted to acquiring partial preference information at an affordable effort [5,37].

The terms of *incomplete* or *partial information* can be interpreted in two interrelated ways [13,28]. On the one hand, they indicate that the DM's preferences – usually modeled in form of some constraints – can be satisfied by more than one set of parameter values. This implies multiplicity of preference model instances compatible with the DM's statements [37]. On the other hand, incompleteness or partiality of preference information emphasizes that its use may not lead to a univocal recommendation [5]. However, the latter can be made robust by eliciting richer (i.e., more complete) information from the DM [3].

As far as MCDA methods incorporating partial preference information are concerned, the preference disaggregation approaches have been prevailing in the recent years [5,21]. They assume that the DM's preferences have the form of example holistic decisions concerning a subset of reference alternatives. Such judgments may come from historical data, from the DM's better knowledge of some alternatives, or can be implied by a relative easiness of performing a comprehensive evaluation of such alternatives [39].

In this paper, we consider multiple criteria sorting problems oriented toward an assignment of alternatives to pre-defined and preference ordered decision classes [47]. For this purpose, we use a threshold-based value-driven sorting procedure [14, 46]. It incorporates a preference model composed of an additive value function and thresholds separating the classes on a scale of a comprehensive value. The parameter values deciding upon the shape of marginal value functions and separating thresholds are inferred indirectly from the assignment examples, which are composed of reference alternatives and their desired class assignments [14]. The latter ones should be reproduced in the final recommendation, while additionally delimiting the space of admissible values of preference model parameters and influencing the sorting of non-reference alternatives.

The preference disaggregation paradigm has been so far mostly applied in the context of monotone learning data, i.e., criteria with well-defined preference directions [14,26]. These include gain and cost criteria, on which one prefers, respectively, greater or lesser performances. However, the recent trend in MCDA (see, e.g., [11,25,34]) – motivated by numerous real-world applications – consists in accounting for the non-monotonic criteria [1].

The framework proposed in this paper accounts for a wide spectrum of types of monotonic and non-monotonic marginal value functions within a preference disaggregation framework. These types admit specification of partial information concerning the DM's per-criterion preferences implied by the problem's peculiarity. In particular, we consider both gain- and cost-type criteria as well as preference-ordered attributes for which the direction of monotonicity cannot be specified a priori. Furthermore, we account for A- and V-type functions, which combine increasing and decreasing trends in disjoint sub-ranges of the performances scale. We also generalize the latter functions to level-monotonic characteristics, which correspond to the shapes assigning the same marginal value to all performances in a certain performance sub-region, but adhering to monotonicity constraints in the other region [34]. For example, the level-decrease function assigns the same maximal marginal value to a subset of the least performances, while systematically decreasing it from a certain point of the performance scale down to zero being associated with the greatest performance. Finally, we also account for the criteria with unknown monotonicity constraints [25], for which the respective marginal functions are allowed to take any shape.

Similarly to Kliegr [25], we aim at constructing a model whose complexity is controlled by the number of changes in monotonicity between all subsequent sub-intervals of marginal value functions. Minimizing this number, we implement the prudence principle in MCDA, while adjusting the model's complexity to the available incomplete preferences. Hence, the lack of complete information about the monotonicity of particular criteria offers different means for ensuring consistency between the DM's preference information and the model than in traditional MCDA approaches. Indeed, it opposes to

both consistency restoration which eliminates the conflicting DM's statements [30,31] and consistency preservation enforcing compatibility of the new DM's judgments with the previously elicited statements [2,4]. To adjust the non-monotonic character of the marginal value functions to the available assignment examples, we use Mixed-Integer Linear Programming (MILP).

The proposed basic model constructs a single additive value function and a vector of precise class thresholds. However, when using indirect preference information, there may exist multiple instances of the sorting model that would be compatible with it, hence restoring the DM's assignment examples [14,17,22]. In our case, a set of compatible instances of the sorting model is delimited by the minimal number of changes in monotonicity for all marginal value functions. The application of such model instances on the set of non-reference alternatives may lead to different assignments [14,26]. From the viewpoint of robustness analysis, it is thus advisable to examine how the sorting recommendation changes when the complexity of compatible model instances varies within the plausible limits. The results of such an examination take the form of possible assignments, which indicate classes to which a given alternative is assigned by at least one instance of the compatible sorting model. Such assignments can be interpreted as robust conclusions which are supported by the DM's partial preference information.

The proposed method is applied to a real-world problem of exposure management of Engineered Nanomaterials (ENMs). Nowadays, such materials are commonly used in consumer products like cosmetics, clothes and food, which implies that the number of workers exposed to such materials is increasing each year [12,27]. The available approaches proposed for controlling exposure to nanomaterials include the use of personal protective equipment, administrative and work practices control and engineering controls [33]. We develop a model for assessing the suitability of a particular Risk Management Measure (RMM) for exposure management during the manufacturing of ENMs. Specifically, we focus on the use of a respirator while handling nanomaterials in certain conditions. The input preference information concerns a holistic assessment of a subset of exposure scenarios to nanomaterials conducted by a team of experts in view of the recommended level of the selected RMM [32]. In addition, ten descriptors are included in the model development. They include seven monotonic criteria of either gain- or cost-type, a single level-increase criterion, and two non-monotonic variables. The role of constructed model is to capture the interrelations between the evaluation criteria and the recommended level of use of the considered RMM. In this way, the model explains the expert judgments, but it can also be used to assess other exposure scenarios to ENMs. The obtained recommendation is validated against the outcomes of robustness analysis in view of the plurality of sorting model instances compatible with the assignment examples.

The remainder of this paper is organized as follows. In Section 2, we describe the mathematical models underlying the proposed method and review the existing preference disaggregation methods that are able to handle non-monotone data. Section 3 discusses the results of its application to exposure management of engineered nanomaterials. The last section concludes and outlines avenues for future research.

## 2. Construction of threshold-based value-driven sorting model with partially known monotonicity constraints based on the Decision Maker's assignment examples

Let us use the following notation [23]:

- $A = \{a_1, a_2, \dots, a_i, \dots, a_n\}$  – a finite set of  $n$  alternatives;
- $A^R = \{a^*, b^*, \dots\} \subseteq A$  – a finite set of reference alternatives, which the DM accepts to critically judge in a holistic way;
- $G = \{g_1, g_2, \dots, g_j, \dots, g_m\}$  – a finite set of  $m$  evaluation criteria,  $g_j : A \rightarrow \mathbb{R}$  for all  $j \in J = \{1, \dots, m\}$ ;
- $X_j = \{x_j \in \mathbb{R} : g_j(a_i) = x_j, a_i \in A\}$  – a set of all different performances on  $g_j$ ,  $j \in J$ ;
- $x_j^1, x_j^2, \dots, x_j^{n_j(A)}$  – increasingly ordered values of  $X_j$ ,  $x_j^k < x_j^{k+1}$ ,  $k = 1, 2, \dots, n_j(A) - 1$ , where  $n_j(A) = |X_j|$  and  $n_j(A) \leq n$ ;
- $C_1, C_2, \dots, C_p$  –  $p$  pre-defined, preference ordered classes, where  $C_{h+1}$  is preferred to  $C_h$ ,  $h = 1, \dots, p - 1$  ( $H = \{1, \dots, p\}$ ).

### 2.1. Sorting model

To comprehensively assess the quality of alternatives, we use an additive value function defined as follows [24,39]:

$$U(a_i) = \sum_{j=1}^m u_j(g_j(a_i)) = \sum_{j=1}^m u_j(x_j^k) \in [0, 1], \quad (1)$$

where  $u_j$  is a marginal value associated with criterion  $g_j$ ,  $j = 1, \dots, m$ . It is used to evaluate alternatives  $a_i \in A$  from a specific point of view. Observe that in Eq. (1) and in the following with the notation  $u_j(a)$  we mean  $u_j(g_j(a))$ . For all criteria, we use general functions with all unique performances corresponding to the characteristic points [14]. Hence, the shape of  $u_j(a_i)$  is determined by  $u_j(x_j^k)$ ,  $k = 1, 2, \dots, n_j(A)$ .

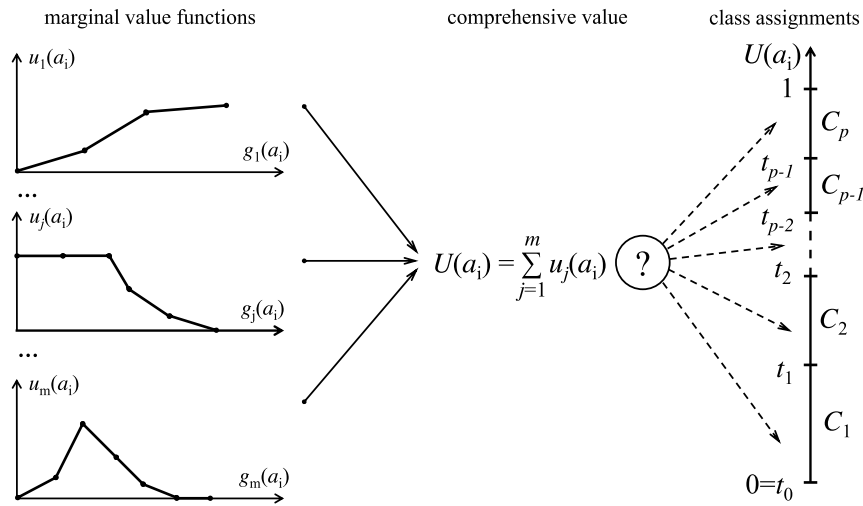


Fig. 1. Value-driven threshold-based sorting procedure.

To classify the alternatives, we use a value-driven threshold-based sorting procedure in which the boundaries between the classes are defined with a vector of thresholds  $t_0, t_1, \dots, t_h, \dots, t_p$ , such that  $t_{h-1}$  and  $t_h$  are, respectively, the lower and upper bounds on a scale of a comprehensive value for class  $C_h$ ,  $h = 1, \dots, p$  [14,46]. Alternative  $a_i \in A$  is assigned to class  $C_h$  in case  $t_{h-1} \leq U(a_i) < t_h$ . Such a procedure is presented graphically in Fig. 1. The set of constraints defining the basic assumptions of the underlying preference model is as follows:

$$\left. \begin{aligned} U(a_i) &= \sum_{j=1}^m u_j(a_i), \text{ for all } a_i \in A, \\ t_h - t_{h-1} &\geq \varepsilon, \quad h = 1, \dots, p, \\ t_0 &= 0, \quad t_p \geq 1 + \varepsilon, \end{aligned} \right\} E^{MODEL} \tag{2}$$

where  $\varepsilon$  is an arbitrarily small positive value.

In the following subsections, we discuss constraints that reconstruct the DM’s preference information and define a set of compatible value functions. We also present the mathematical models for both selection of a single representative sorting model as well as robustness analysis whose results are quantified by means of possible assignments.

### 2.2. Preference information

The parameters of an assumed sorting model are inferred indirectly from the DM’s assignment examples specifying for each reference alternative  $a_i^* \in A^R$  its desired class  $C_{DM(a_i^*)}$  (e.g., alternative  $a_1^*$  should be assigned to class  $C_2$ , whereas alternative  $a_2^*$  should be sorted into class  $C_4$ ) [14,26]. The assignment examples are translated to the following constraints:

$$\left. \begin{aligned} \text{for all } a_i^* \in A^R : \\ U(a_i^*) &\geq t_{DM(a_i^*)-1}, \\ U(a_i^*) + \varepsilon &\leq t_{DM(a_i^*)}. \end{aligned} \right\} E^{ASS-EX} \tag{3}$$

Thus, a comprehensive value of a reference alternative assigned to  $C_{DM}$  should be within the bounds associated with this class.

### 2.3. Compatible sorting model instances

In the proposed approach, we consider a wide spectrum of types of monotonic and non-monotonic marginal value functions within a preference disaggregation framework. These types include standard monotonic shapes, level-monotonic functions, A- and V-types combining increasing and decreasing value trends, and unknown monotonicity constraints.

The existing preference disaggregation methods that are able to handle non-monotone data can be classified into different streams. Firstly, one has proposed to use some specific forms of non-monotonicity or pre-defined shapes of non-monotonic marginal value functions. In this regard, Despotis and Zopounidis [6] and Guo et al. [17] considered the criteria with some mid-point corresponding to the most preferred performance, whereas Rezaei [34] accounted for a rich spectrum of precisely specified shapes including, e.g., A- or V-type functions. Secondly, some more general algorithms have been devised to avoid dealing solely with some specific form of non-monotonicity. In particular, Doumpos [7] used a differential evolution algorithm and Ghaderi et al. [10] introduced a mathematical programming model for constructing non-monotonic



functions, while not directly restraining the model's complexity. The last group of methods aimed at disaggregating holistic judgments while not making any assumptions on the shape of marginal value functions, but controlling their complexity. In this regard, Kliegr [25] penalized the changes of non-monotonicity in the shape of marginal functions using MILP models, whereas Ghaderi et al. [11] and Liu et al. [29] considered minimization of the variation in slope with, respectively, Linear Programming (LP) techniques or a quadratic optimization problem.

In what follows, we discuss constraints that define the shape of marginal value functions depending on the desired types of (non-)monotonicity, and normalize comprehensive values within the  $[0, 1]$  range.

**Shape of marginal value functions.** For each criterion  $g_j$ ,  $j = 1, \dots, m$ , the DM is expected to define the respective requirements on monotonicity of marginal values which are assigned to the respective performances  $x_j^1, x_j^2, \dots, x_j^{n_j(A)}$ . These are implied by the type associated with a given criterion. We consider the following types: gain, cost, monotonic non-defined, A, V, increase-level, decrease-level, level-increase, level-decrease, and non-monotonic. In what follows, we explain their meaning and discuss the respective constraints. Whichever the criterion's type, we require all marginal values to be non-negative:

$$u_j(x_j^k) \geq 0, \quad j = 1, \dots, m, k = 1, \dots, n_j(A). \quad \left. \right\} E^{NON-NEG}$$

The set of constraints involving  $E^{NON-NEG}$  as well as the constraints related to the type of (non-)monotonicity for all criteria will be denoted by  $E^{MON}$ .

- *Gain type* means that the greater  $g_j(a_i)$ , the more preferred alternative  $a_i$  on criterion  $g_j$ , thus implying the non-decreasing trend for the marginal values with the increase in  $g_j(a_i)$  (see Fig. 2a):

$$u_j(x_j^k) \geq u_j(x_j^{k-1}), \quad k = 2, \dots, n_j(A). \quad \left. \right\} E_{GAIN}^{MON}$$

- *Cost type* implies that the greater  $g_j(a_i)$ , the less preferred alternative  $a_i$  on criterion  $g_j$ , thus implying the non-increasing trend for the marginal values with the increase in  $g_j(a_i)$  (see Fig. 2b):

$$u_j(x_j^k) \leq u_j(x_j^{k-1}), \quad k = 2, \dots, n_j(A). \quad \left. \right\} E_{COST}^{MON}$$

- *Monotonic non-defined type* implies that the preference on  $g_j$  adheres to the monotonicity constraints, but whether it is of gain or cost type cannot be specified a priori:

$$\left. \begin{aligned} u_j(x_j^k) &= u_j^\uparrow(x_j^k) + u_j^\downarrow(x_j^k), \quad k = 1, \dots, n_j(A), \\ u_j^\uparrow(x_j^k) &\geq u_j^\uparrow(x_j^{k-1}), \quad k = 2, \dots, n_j(A), \\ u_j^\downarrow(x_j^k) &\leq u_j^\downarrow(x_j^{k-1}), \quad k = 2, \dots, n_j(A), \\ u_j^\uparrow(x_j^k), u_j^\downarrow(x_j^k) &\geq 0, \quad k = 1, \dots, n_j(A), \\ u_j^\uparrow(x_j^{n_j(A)}) &\leq M \cdot (1 - v_{j,cost}^{mon}), \\ u_j^\downarrow(x_j^1) &\leq M \cdot v_{j,cost}^{mon}, \\ v_{j,cost}^{mon} &\in \{0, 1\}, \end{aligned} \right\} E_{NON-DEF}^{MON}$$

where  $M$  is an arbitrarily large positive constant. The marginal value function  $u_j$  is modeled as a sum of values derived from the assumption that  $g_j$  is either of gain ( $u_j^\uparrow$ ) or cost ( $u_j^\downarrow$ ) type. However, only one of them can be activated with the binary variable  $v_{j,cost}^{mon}$ . Specifically, if  $v_{j,cost}^{mon} = 1$ ,  $g_j$  is of cost type. Then,  $u_j^\uparrow(x_j^{n_j(A)}) = 0$  and, thus, all marginal values  $u_j^\uparrow(\cdot)$  are equal to 0. Otherwise,  $u_j^\downarrow(x_j^1) = 0$  and, thus, all marginal values  $u_j^\downarrow(\cdot)$  are equal to 0. This, in turn, implies that  $g_j$  is of gain type.

When modelling marginal value functions for the criteria of gain, cost, or monotonic non-defined types, we required that monotonicity is non-strict. This admits marginal values assigned to a pair of performances  $x_j^{k-1}$  and  $x_j^k$  for  $k = 2, \dots, n_j(A)$ , to be equal. In case the DM would expect the marginal function to be strictly monotonic, the respective weak inequalities should be replaced with their strict counterparts involving  $\varepsilon$ . For example, for gain-type criteria, constraint  $u_j(x_j^k) \geq u_j(x_j^{k-1})$  contained in  $E_{GAIN}^{MON}$  should be replaced with  $u_j(x_j^k) \geq u_j(x_j^{k-1}) + \varepsilon$ .

- *A-type* means that the most preferred performance potentially does not align with any extreme performance, hence admitting at most one change of monotonicity from non-decreasing to non-increasing (see Fig. 2c):

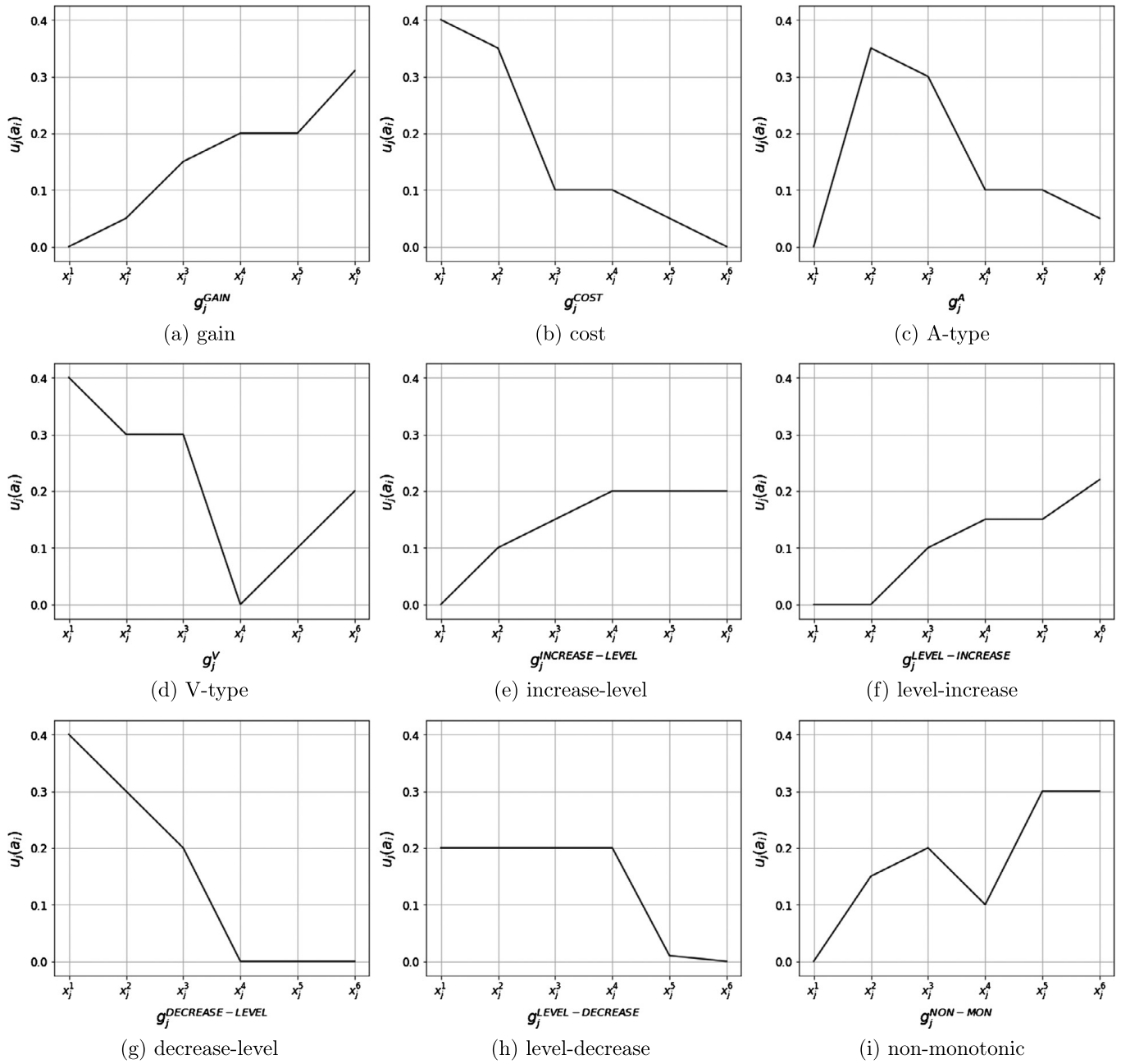


Fig. 2. Example marginal value functions representing different types of requirements with respect to their monotonicity.

$$\left. \begin{aligned}
 &M \cdot \sum_{p=2}^k v_{j,p}^{opt} + u_j(x_j^k) \geq u_j(x_j^{k-1}), \quad k = 2, \dots, n_j(A), \\
 &u_j(x_j^k) \leq u_j(x_j^{k-1}) + M \cdot (1 - \sum_{p=2}^k v_{j,p}^{opt}), \quad k = 2, \dots, n_j(A). \\
 &\sum_{p=2}^{n_j(A)} v_{j,p}^{opt} \leq 1, \\
 &v_{j,p}^{opt} \in \{0, 1\}, \quad p = 2, \dots, n_j(A).
 \end{aligned} \right\} E_A^{MON}$$

Note that  $v_{j,p}^{opt}$  is allowed to be 1 for at most one  $p \in \{2, \dots, n_j(A)\}$ . If  $v_{j,p}^{opt} = 1$ , then the following constraints hold:

$$\left. \begin{aligned}
 &u_j(x_j^k) \geq u_j(x_j^{k-1}), \quad \text{if } p \geq 3, \quad k = 2, \dots, p-1, \\
 &u_j(x_j^k) \leq u_j(x_j^{k-1}), \quad k = p, \dots, n_j(A).
 \end{aligned} \right\}$$

Thus, if  $v_{j,2}^{opt} = 1$ ,  $u_j$  is non-increasing (i.e.,  $g_j$  is of cost type); if  $v_{j,p}^{opt} = 1$ , for  $3 \leq p \leq n_j(A)$ , then  $u_j$  is of pure A-type, whereas  $v_{j,p}^{opt} = 0$  for  $p \in \{2, \dots, n_j(A)\}$  implies that  $u_j$  is non-decreasing (i.e.,  $g_j$  is of gain type).

- *V-type* means that the least preferred performance potentially does not align with any of the extreme performances, hence admitting at most one change of monotonicity from non-increasing to non-decreasing (see Fig. 2d):

$$\left. \begin{aligned} u_j(x_j^k) &\leq u_j(x_j^{k-1}) + M \cdot \sum_{p=2}^k v_{j,p}^{opt}, \quad k = 2, \dots, n_j(A), \\ M \cdot (1 - \sum_{p=2}^k v_{j,p}^{opt}) + u_j(x_j^k) &\geq u_j(x_j^{k-1}), \quad k = 2, \dots, n_j(A), \\ \sum_{p=2}^{n_j(A)} v_{j,p}^{opt} &\leq 1, \\ v_{j,p}^{opt} &\in \{0, 1\}, \quad p = 2, \dots, n_j(A). \end{aligned} \right\} E_V^{MON}$$

The role of binary variable  $v_{j,p}^{opt}$  is analogous to the case of A-type function.

- *Increase-level type* implies that  $u_j$  is non-decreasing up to a certain (though not indicated a priori) performance and then reaches saturation, hence remaining constant from this point up to the greatest performance (see Fig. 2e). This type of function can be enforced by putting together the requirements for A- and gain-type functions, i.e.:

$$E_A^{MON}, E_{GAIN}^{MON} \} E_{INC-LEV}^{MON}$$

- *Level-increase type* implies that  $u_j$  is constant up to a certain performance (thus, assigning zero to the respective marginal values), and non-decreasing in the range between this point and the greatest performance (see Fig. 2f). This type of function can be enforced by putting together the requirements for V- and gain-type functions, i.e.:

$$E_V^{MON}, E_{GAIN}^{MON} \} E_{LEV-INC}^{MON}$$

- *Decrease-level type* implies that  $u_j$  is non-increasing up to a certain performance and then assigns zero to marginal values corresponding to all remaining performances (see Fig. 2g), i.e.:

$$E_V^{MON}, E_{COST}^{MON} \} E_{DEC-LEV}^{MON}$$

- *Level-decrease type* implies that  $u_j$  is constant up to a certain performance (thus, assigning the maximal value to the respective marginal values), and non-increasing in the range between this point and the greatest performance (see Fig. 2h), i.e.:

$$E_A^{MON}, E_{COST}^{MON} \} E_{LEV-DEC}^{MON}$$

- *Non-monotonic type* means that there is no prior information on the monotonicity of criterion  $g_j$  (see Fig. 2i). In general, it would be possible to avoid defining any constraints for such functions, but since we aim at controlling the complexity of the inferred marginal value functions, we will include the following constraint set which captures the number of changes in monotonicity between the neighboring performance sub-intervals:

$$\left. \begin{aligned} M \cdot (1 - v_{j,mon-dir}^{k,k-1}) + u_j(x_j^k) &\geq u_j(x_j^{k-1}), \quad k = 2, \dots, n_j(A), \\ u_j(x_j^k) &\leq u_j(x_j^{k-1}) + M \cdot v_{j,mon-dir}^{k,k-1}, \quad k = 2, \dots, n_j(A), \\ v_{j,mon-dir}^{k,k-1} - v_{j,mon-dir}^{k-1,k-2} + M \cdot v_{j,change-mon}^{k,k-2} &\geq 0, \quad k = 3, \dots, n_j(A), \\ v_{j,mon-dir}^{k,k-1} - v_{j,mon-dir}^{k-1,k-2} - M \cdot v_{j,change-mon}^{k,k-2} &\leq 0, \quad k = 3, \dots, n_j(A), \\ v_{j,mon-dir}^{k,k-1} &\in \{0, 1\}, \quad k = 2, \dots, n_j(A), \\ v_{j,change-mon}^{k,k-2} &\in \{0, 1\}, \quad k = 3, \dots, n_j(A). \end{aligned} \right\} E_{NON-MON}^{MON}$$

If  $u_j(x_j^k) \geq u_j(x_j^{k-1})$ , then  $v_{j,mon-dir}^{k,k-1} = 1$  and  $u_j$  is non-decreasing between characteristic points  $x_j^{k-1}$  and  $x_j^k$ . If  $u_j(x_j^k) \leq u_j(x_j^{k-1})$ , then  $v_{j,mon-dir}^{k,k-1} = 0$  and  $u_j$  is non-increasing between characteristic points  $x_j^{k-1}$  and  $x_j^k$ . If there is a change in the monotonicity direction of  $u_j$  between three characteristic points  $x_j^{k-2}$ ,  $x_j^{k-1}$ , and  $x_j^k$  (i.e., either  $v_{j,mon-dir}^{k,k-1} = 1$  and  $v_{j,mon-dir}^{k-1,k-2} = 0$ , or  $v_{j,mon-dir}^{k,k-1} = 0$  and  $v_{j,mon-dir}^{k-1,k-2} = 1$ ), then  $v_{j,change-mon}^{k,k-2} = 1$ . Otherwise (i.e., either  $v_{j,mon-dir}^{k,k-1} = 1$  and  $v_{j,mon-dir}^{k-1,k-2} = 1$ , or  $v_{j,mon-dir}^{k,k-1} = 0$  and  $v_{j,mon-dir}^{k-1,k-2} = 0$ ),  $v_{j,change-mon}^{k,k-2} = 0$  and there is no change in the monotonicity direction of  $u_j$  between  $x_j^{k-2}$  and  $x_j^k$ . Thus, the sum of  $v_{j,change-mon}^{k,k-2} \in \{0, 1\}$ , for  $k = 3, \dots, n_j(A)$ , represents the number of changes in the monotonicity of  $u_j$ .

**Normalization.** For the sake of interpretability, an additive value function is normalized to the  $[0, 1]$  interval. This is attained by means of two types of constraints. On the one hand, the marginal values of the least preferred performances on all

criteria need to be zero. In this way, a comprehensive value of an anti-ideal alternative is also equal to zero. On the other hand, the marginal values assigned to the most preferred performances on all criteria need to sum up to one, i.e.:

$$\sum_{j=1}^m u_j^{best} = 1, \left. \vphantom{\sum_{j=1}^m} \right\} E^{NORM-1}$$

where  $u_j^{best}$  is the greatest marginal value for criterion  $g_j$ ,  $j = 1, \dots, m$ . The set of constraints involving  $E^{NORM-1}$  as well as dedicated normalization constraints related to the type of (non-)monotonicity for all criteria will be denoted by  $E^{NORM}$ . The respective constraints which allow to identify the least and the most preferred performances which are assigned, respectively, zero and a maximal marginal value are discussed individually for each criterion type:

- For gain, increase-level, and level-increase criteria, the least performance is assigned a marginal value of zero, i.e.:

$$u_j(x_j^1) = 0, \left. \vphantom{u_j(x_j^1)} \right\} E_{GAIN}^{NORM-0}$$

whereas the greatest performance is the most preferred one, i.e.:

$$u_j(x_j^{n_j(A)}) = u_j^{best}, \left. \vphantom{u_j(x_j^{n_j(A)})} \right\} E_{GAIN}^{NORM-1}$$

- For cost, decrease-level, and level-decrease criteria, the greatest performance is assigned a marginal value of zero, i.e.:

$$u_j(x_j^{n_j(A)}) = 0, \left. \vphantom{u_j(x_j^{n_j(A)})} \right\} E_{COST}^{NORM-0}$$

whereas the least performance is the most preferred one, i.e.:

$$u_j(x_j^1) = u_j^{best}, \left. \vphantom{u_j(x_j^1)} \right\} E_{COST}^{NORM-1}$$

- For monotonic criteria with non-defined type of monotonicity, the less preferred performance is either the least (if  $v_{j,cost}^{mon} = 0$ ) or the greatest one (if  $v_{j,cost}^{mon} = 1$ ), i.e.:

$$\left. \begin{aligned} u_j^\downarrow(x_j^{n_j(A)}) &\leq M \cdot (1 - v_{j,cost}^{mon}), \\ u_j^\uparrow(x_j^1) &\leq M \cdot v_{j,cost}^{mon}, \end{aligned} \right\} E_{NON-DEF}^{NORM-0}$$

whereas the most preferred performance is either the greatest (i.e.,  $v_{j,cost}^{mon} = 0$ ) or the least one (if  $v_{j,cost}^{mon} = 1$ ), i.e.:

$$\left. \begin{aligned} u_j^{best} - u_j^\downarrow(x_j^1) &\geq -M \cdot (1 - v_{j,cost}^{mon}), \\ u_j^{best} - u_j^\uparrow(x_j^1) &\leq M \cdot (1 - v_{j,cost}^{mon}), \\ u_j^{best} - u_j^\uparrow(x_j^{n_j(A)}) &\geq -M \cdot v_{j,cost}^{mon}, \\ u_j^{best} - u_j^\downarrow(x_j^{n_j(A)}) &\leq M \cdot v_{j,cost}^{mon}. \end{aligned} \right\} E_{NON-DEF}^{NORM-1}$$

- For A-type criteria, the least preferred performance is either  $x_j^1$  (if  $v_{j,norm-0} = 0$ ) or  $x_j^{n_j(A)}$  (if  $v_{j,norm-0} = 1$ ), i.e.:

$$\left. \begin{aligned} u_j(x_j^1) &\leq u_j(x_j^{n_j(A)}) + M \cdot v_{j,norm-0}, \\ M \cdot (1 - v_{j,norm-0}) + u_j(x_j^1) + \varepsilon &\geq u_j(x_j^{n_j(A)}), \\ u_j(x_j^1) &\leq M \cdot v_{j,norm-0}, \\ u_j(x_j^{n_j(A)}) &\leq M \cdot (1 - v_{j,norm-0}), \\ v_{j,norm-0} &\in \{0, 1\}, \end{aligned} \right\} E_A^{NORM-0}$$

whereas the most preferred performance is either  $x_j^{k-1}$  (if  $v_{j,k}^{opt} = 1$  for some  $k = 2, \dots, n_j(A)$ ) or  $x_j^{n_j(A)}$  (if  $v_{j,k}^{opt} = 0$  for all  $k = 2, \dots, n_j(A)$ ), i.e.:

$$\left. \begin{aligned} u_j^{best} - u_j(x_j^{k-1}) &\geq -M \cdot (1 - v_{j,k}^{opt}), \quad k = 2, \dots, n_j(A), \\ u_j^{best} - u_j(x_j^{k-1}) &\leq M \cdot (1 - v_{j,k}^{opt}), \quad k = 2, \dots, n_j(A), \\ u_j^{best} - u_j(x_j^{n_j(A)}) &\geq -\sum_{k=2}^{n_j(A)} v_{j,k}^{opt}, \\ u_j^{best} - u_j(x_j^{n_j(A)}) &\leq \sum_{k=2}^{n_j(A)} v_{j,k}^{opt}. \end{aligned} \right\} E_A^{NORM-1}$$

- For V-type criteria, the less preferred performance is either  $x_j^{k-1}$  (if  $v_{j,k}^{opt} = 1$  for some  $k = 2, \dots, n_j(A)$ ) or  $x_j^{n_j(A)}$  (if  $v_{j,k}^{opt} = 0$  for all  $k = 2, \dots, n_j(A)$ ):

$$\left. \begin{aligned} u_j(x_j^{k-1}) &\leq 1 - v_{j,k}^{opt}, \quad k = 2, \dots, n_j(A), \\ u_j(x_j^{n_j(A)}) &\leq \sum_{k=2}^{n_j(A)} v_{j,k}^{opt}, \end{aligned} \right\} E_V^{NORM-0}$$

whereas the most preferred performance is either  $x_j^1$  (if  $v_{j,norm-1} = 1$ ) or  $x_j^{n_j(A)}$  (if  $v_{j,norm-0} = 0$ ), i.e.:

$$\left. \begin{aligned} u_j(x_j^1) &\leq u_j(x_j^{n_j(A)}) + M \cdot v_{j,norm-1}, \\ M \cdot (1 - v_{j,norm-1}) + u_j(x_j^1) &\geq u_j(x_j^{n_j(A)}), \\ u_j^{best} - u_j(x_j^{n_j(A)}) &\leq M \cdot v_{j,norm-1}, \\ u_j^{best} - u_j(x_j^{n_j(A)}) &\geq -M \cdot v_{j,norm-1}, \\ u_j^{best} - u_j(x_j^1) &\leq -M \cdot (1 - v_{j,norm-1}), \\ u_j^{best} - u_j(x_j^1) &\geq M \cdot (1 - v_{j,norm-1}), \\ v_{j,norm-1} &\in \{0, 1\}. \end{aligned} \right\} E_V^{NORM-0}$$

- For non-monotonic criteria  $u_j(x_j^k)$  needs to be equal to 0 for at least one characteristic point  $x_j^k$ ,  $k = 1, \dots, n_j(A)$ , such that  $v_{j,norm-0}^k = 1$ :

$$\left. \begin{aligned} u_j(x_j^k) - M \cdot (1 - v_{j,norm-0}^k) &\leq 0, \quad k = 1, \dots, n_j(A), \\ \sum_{k=1}^{n_j(A)} v_{j,norm-0}^k &\geq 1, \quad k = 1, \dots, n_j(A), \\ v_{j,norm-0}^k &\in \{0, 1\}, \quad k = 1, \dots, n_j(A). \end{aligned} \right\} E_{NON-MON}^{NORM-0}$$

Similarly, the maximal marginal value needs to be assigned to at least one characteristic point  $x_j^k$ ,  $k = 1, \dots, n_j(A)$ , such that  $v_{j,norm-1}^k = 1$ :

$$\left. \begin{aligned} &\text{for } k = 1, \dots, n_j(A) : \\ u_j(x_j^k) &\geq u_j(x_j^i) - M \cdot (1 - v_{j,norm-1}^k), \quad i = 1, \dots, k-1, k+1, \dots, n_j(A), \\ u_j^{best} - u_j(x_j^k) &\leq M \cdot v_{j,norm-1}^k, \\ u_j^{best} - u_j(x_j^k) &\geq -M \cdot v_{j,norm-1}^k, \\ \sum_{k=1}^{n_j(A)} v_{j,norm-1}^k &\geq 1, \\ v_{j,norm-1}^k &\in \{0, 1\}, \quad k = 1, \dots, n_j(A). \end{aligned} \right\} E_{NON-MON}^{NORM-1}$$

Overall, a set of sorting model instances (i.e., additive value functions and class thresholds) compatible with the DM's assignment examples and requirements on the (non-)monotonicity of particular criteria can be defined as follows:

$$E^{AR} = E^{MODEL} \cup E^{ASS-EX} \cup E^{MON} \cup E^{NORM}.$$

## 2.4. Sorting recommendation

In this section, we discuss two complementary ways of exploiting a set of compatible sorting model instances. Arbitrary selection of a single representative instance leads to precise assignments for all alternatives, whereas robustness analysis reveals all possible sorting recommendations that follow the DM's preference information and the use of an assumed preference model.

### 2.4.1. Selection of a single representative sorting model

To select a representative sorting model, we minimize the number of changes in monotonicity for all marginal value functions  $u_j$ ,  $j = 1, \dots, m$ , by solving the following optimization problem:

$$\text{Minimize : } NM = \sum_{j \in G_A \cup G_V} \sum_{p=2}^{n_j(A)} v_{j,p}^{opt} + \sum_{j \in G_{NON-MON}} \sum_{k=3}^{n_j(A)} v_{j,change-mon}^{k,k-2}, \quad \text{s.t. } E^{AR},$$

where  $G_A$  and  $G_V$  are subsets of, respectively, A- and V-type criteria admitting at most one change in monotonicity (their number is represented by  $\sum_{p=2}^{n_j(A)} v_{j,p}^{opt}$ ), and  $G_{NON-MON}$  is a subset of criteria for which no monotonicity requirements have been specified (in this case, the number of changes in monotonicity is captured by  $\sum_{k=3}^{n_j(A)} v_{j,change-mon}^{k,k-2}$ ). Let us denote the minimal number of such changes by  $NM^*$ .

Note that the above objective function is applicable only when at least one criterion is of A-, V- or non-monotonic type. Otherwise, there are no changes in monotonicity for any marginal value function and hence  $NM$  is equal to zero. Then, a standard approach to derive a representative sorting model consists in treating  $\varepsilon$  contained in  $E^{AR}$  as a variable and solving the following problem:

$$\text{Minimize : } \varepsilon, \text{ s.t. } E^{AR}.$$

#### 2.4.2. Robustness analysis

Solving the problems presented in Section 2.4.1 leads to a selection of some arbitrary marginal value functions and class thresholds compatible with the DM's partial preference information. Its analysis is beneficial in terms of providing precise recommendation along with information on the importance of particular criteria, trade-offs between criteria, or distribution of class thresholds [15]. However, in view of the incompleteness of DM's preferences, there exist multiple compatible instances of the sorting model whose recommendation for the non-reference alternatives may be different. To verify the stability of sorting recommendation, we refer to the concept of *possible assignment*, which indicates a set of classes to which a given alternative can be assigned by at least one compatible instance of the sorting model [14,22]. The validity of such an assignment for alternative  $a \in A$  and class  $C_h, h = 1, \dots, p$ , can be verified by considering the following set of constraints, which exploits a set of models with the minimal number of changes in monotonicity for all marginal value functions:

$$\left. \begin{aligned} E^{AR}, \\ NM^* = \sum_{j \in G_A \cup G_V} \sum_{p=2}^{n_j(A)} v_{j,p}^{opt} + \sum_{j \in G_{NON-MON}} \sum_{k=3}^{n_j(A)} v_{j,change-mon}^{k,k-2}, \\ U(a) \geq t_{h-1}, U(a) + \varepsilon \leq t_h. \end{aligned} \right\} E(a \rightarrow^P C_h)$$

If  $E(a \rightarrow^P C_h)$  is feasible and  $\varepsilon^* = \max \varepsilon, \text{ s.t. } E(a \rightarrow^P C_h)$  is greater than 0,  $a$  can be possibly assigned to  $C_h$ . In case  $E(a \rightarrow^P C_h)$  is infeasible or  $\varepsilon^* \leq 0$ ,  $a$  cannot be assigned to  $C_h$  with any compatible instance of the sorting model. The set of all classes to which  $a$  can be possibly assigned is denoted by  $C_P(a)$ . In case  $C_P(a)$  is a singleton,  $a$  is assigned to a class contained in  $C_P(a)$  by all compatible instances of the sorting model. Such an assignment can be deemed as robust or necessary.

Note that the possible assignment  $C_P(a)$  for each alternative  $a \in A$  is a union of intervals, one for each possible type of function. However, since such a union cannot be ensured to be an interval on its own, we cannot guarantee “the no jump property” for the possible assignments [14]. Therefore, in what follows, all possible assignments are represented as sets of classes (e.g.,  $C_P(a_{35}) = \{C_3, C_4, C_5\}$ ) rather than intervals (e.g.,  $C_P(a_{35}) = [C_3, C_5]$ ). In what follows, we provide a detailed discussion on “the no jump property” in the context of the method introduced in this paper.

Let us denote by  $\mathcal{U}$  a set of all possible value functions, by  $\mathcal{T}$  – a set of all possible thresholds vectors and by  $\mathcal{V}$  – a set of all possible binary vectors. Now, let us denote by  $\mathcal{P} \subseteq \mathcal{U} \times \mathcal{T} \times \mathcal{V}$  a set of all triples  $(U, \mathbf{b}, \mathbf{v})$  satisfying constraints in  $E^{AR}$ , that is, all models (value functions, vectors of thresholds, binary vectors) compatible with the preference information provided by the DM.

Let us suppose  $(U_1, \mathbf{t}_1, \mathbf{v}_1), (U_2, \mathbf{t}_2, \mathbf{v}_2) \in \mathcal{P}$  and that  $a$  is assigned to  $C_h$  w.r.t.  $(U_1, \mathbf{t}_1, \mathbf{v}_1)$ , while  $a$  is assigned to  $C_k$  w.r.t.  $(U_2, \mathbf{t}_2, \mathbf{v}_2)$ , with  $h, k \in [1, \dots, p]$  such that  $h > k + 1$ .

We have to distinguish two cases:

- 1)  $\mathbf{v}_1 = \mathbf{v}_2$ : in all criteria, the two functions  $U_1$  and  $U_2$  present the shape and the monotonicity changes exactly in the same characteristic points;
- 2)  $\mathbf{v}_1 \neq \mathbf{v}_2$ : in at least one criterion, the two functions  $U_1$  and  $U_2$  have a different shape or, they are of the same shape but the monotonicity changes in different characteristic points.

Let us prove that for all  $l \in ]h, k[$ , there exists  $(U, \mathbf{b}, \mathbf{v}) \in \mathcal{P}$  such that  $a$  is assigned to  $C_l$  w.r.t.  $(U, \mathbf{b}, \mathbf{v})$ .

**Proposition 1.** Let  $a \in A, (U_1, \mathbf{t}_1, \mathbf{v}_1), (U_2, \mathbf{t}_2, \mathbf{v}_2) \in \mathcal{P}$  and  $h, k \in [1, \dots, p]$ , such that:

- 1)  $a$  is assigned to  $C_h$  w.r.t.  $(U_1, \mathbf{t}_1, \mathbf{v}_1)$ ,
- 2)  $a$  is assigned to  $C_k$  w.r.t.  $(U_2, \mathbf{t}_2, \mathbf{v}_2)$ ,
- 3)  $\mathbf{v}_1 = \mathbf{v}_2$ ,
- 4)  $h > k + 1$ ,

then for all  $l \in ]k, h[$  there exists  $(U, \mathbf{b}, \mathbf{v}_1) \in \mathcal{P}$  such that  $a$  is assigned to  $C_l$  w.r.t.  $(U, \mathbf{b}, \mathbf{v}_1)$ .

**Proof.** The first two hypotheses are equivalent to the following:

$$t_{1,h-1} \leq U_1(a) < t_{1,h} \quad \text{and} \quad t_{2,k-1} \leq U_2(a) < t_{2,k}.$$

Since  $l \in ]k, h[$  and because of the thresholds monotonicity, we have:

$$U_1(a) \geq t_{1,h-1} \geq t_{1,l} > t_{1,l-1}, \quad (4)$$

and

$$t_{2,l} > t_{2,l-1} \geq t_{2,k} > U_2(a). \quad (5)$$

Let  $\alpha \in ]0, 1[$  and let us define the corresponding convex combinations of  $t_{1,l}$  and  $t_{1,l-1}$  on one hand and of  $t_{2,l}$  and  $t_{2,l-1}$  on the other hand, that is,

$$t_{1\alpha l} = \alpha t_{1,l} + (1 - \alpha) t_{1,l-1}$$

and

$$t_{2\alpha l} = \alpha t_{2,l} + (1 - \alpha) t_{2,l-1}.$$

Let us consider the triple  $(\lambda U_1 + (1 - \lambda) U_2, \lambda \mathbf{t}_1 + (1 - \lambda) \mathbf{t}_2, \mathbf{v}_1)$  with  $\lambda \in \mathbb{R}$  such that

$$\lambda U_1(a) + (1 - \lambda) U_2(a) = \lambda t_{1\alpha l} + (1 - \lambda) t_{2\alpha l}$$

from which

$$\lambda = \frac{t_{2\alpha l} - U_2(a)}{(U_1(a) - t_{1\alpha l}) + (t_{2\alpha l} - U_2(a))}.$$

Observing that  $t_{1\alpha l} \in ]t_{1,l-1}, t_{1,l}[$  and  $t_{2\alpha l} \in ]t_{2,l-1}, t_{2,l}[$ , by Eqs. (4) and (5), we get

$$U_1(a) > t_{1\alpha l}$$

and

$$t_{2\alpha l} > U_2(a),$$

from which we get  $\lambda \in ]0, 1[$ . Consequently, since any subset of  $\mathcal{P}$  containing all the triples  $(U, \mathbf{b}, \mathbf{v})$  with  $\mathbf{v} = \bar{\mathbf{v}}$  for some  $\bar{\mathbf{v}}$  is convex, then  $(\lambda U_1 + (1 - \lambda) U_2, \lambda \mathbf{t}_1 + (1 - \lambda) \mathbf{t}_2, \mathbf{v}_1) \in \mathcal{P}$ .

Observing that:

- the component  $l - 1$  of the vector  $\lambda \mathbf{t}_1 + (1 - \lambda) \mathbf{t}_2$  is  $\lambda t_{1,l-1} + (1 - \lambda) t_{2,l-1}$ ,
- the component  $l$  of the vector  $\lambda \mathbf{t}_1 + (1 - \lambda) \mathbf{t}_2$  is  $\lambda t_{1,l} + (1 - \lambda) t_{2,l}$ ,
- $\lambda U_1(a) + (1 - \lambda) U_2(a) = \lambda t_{1\alpha l} + (1 - \lambda) t_{2\alpha l} = \alpha (\lambda t_{1,l} + (1 - \lambda) t_{2,l}) + (1 - \alpha) (\lambda t_{1,l-1} + (1 - \lambda) t_{2,l-1})$ ,
- $\alpha \in ]0, 1[$ ,

then

$$\lambda t_{1,l-1} + (1 - \lambda) t_{2,l-1} \leq \lambda U_1(a) + (1 - \lambda) U_2(a) < \lambda t_{1,l} + (1 - \lambda) t_{2,l}$$

implying that  $a$  is assigned to  $C_l$  w.r.t.  $(\lambda U_1 + (1 - \lambda) U_2, \lambda \mathbf{t}_1 + (1 - \lambda) \mathbf{t}_2, \mathbf{v}_1)$ .  $\square$

**Corollary 1.** If for all  $(U, \mathbf{t}, \mathbf{v}) \in \mathcal{P}$ ,  $\mathbf{v} = \bar{\mathbf{v}}$ , then for all  $a \in A$ ,  $C_P(a)$  is an interval of classes without any jump, that is

$$C_P(a) = \{C_{L(a)}, C_{L(a)+1}, \dots, C_{R(a)}\}$$

where

$$L(a) = \min\{h : C_h \in C_P(a)\},$$

$$R(a) = \max\{h : C_h \in C_P(a)\}.$$

Let us denote by  $C_P^Q(a)$  the set of possible classes to which  $a$  can be assigned by at least one triple  $(U, \mathbf{t}, \mathbf{v})$  in  $Q \subseteq \mathcal{P}$ . In particular,  $C_P(a) = C_P^{\mathcal{P}}(a)$ . The following holds:

**Corollary 2.** Let  $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 \cup \dots \cup \mathcal{P}_r$  where, for all  $i = 1, \dots, r$ , for all  $(U, \mathbf{b}, \mathbf{v}) \in \mathcal{P}_i$ ,  $\mathbf{v} = \bar{\mathbf{v}}_i$ ; then:

- $C_P^{\mathcal{P}_1}(a), C_P^{\mathcal{P}_2}(a), \dots, C_P^{\mathcal{P}_r}(a)$ , are intervals without jumps and  $C_P(a) = C_P^{\mathcal{P}_1}(a) \cup C_P^{\mathcal{P}_2}(a) \cup \dots \cup C_P^{\mathcal{P}_r}(a)$ ,
- if  $\bar{v}_1 = \bar{v}_2 = \dots = \bar{v}_r$ , then  $C_P(a)$  is an interval without jumps.

Since, in general, a union of intervals is not necessarily an interval, if in the previous corollary,  $\bar{v}_i \neq \bar{v}_j$  for some  $i, j \in \{1, \dots, r\}$ , nothing can be said about the presence or absence of jumps in  $C_P(a)$ .

### 3. Application to exposure management of engineered nanomaterials

Engineered nanomaterials (ENMs) are materials with at least one dimension in the range of 1–100 nanometers, though larger ones are usually included in this definition. One distinctive feature of these materials is that their physicochemical properties are significantly different from materials of larger sizes. This makes them suitable for the development of products with enhanced performances in several areas including construction, electronics, environmental management and healthcare [35,44,45]. As a result, the number of workers exposed to such materials is rising [12]. Even though ENMs enable the development of high performance products, there is a lot discussion and concern about their potential impacts on human health and the environment [9]. This motivates the development of risk assessment and management strategies to handle the risks that ENMs can cause. Risk assessment is the tool that has been advanced to assess and manage risks of ENMs and it is composed of a hazard and an exposure assessment part. In this paper, we focus on the latter and contribute to the development of decision support systems to manage exposure to ENMs manufacturing by recommending RMMs [42].

#### 3.1. Decision classes, criteria, and alternatives

We present the model for assessing the need of using a risk management measure (i.e., a respirator) by workers exposed to nanomaterials during their manufacturing. We incorporate real-world data elaborated by Naidu [32], who developed a set of exposure scenarios to ENMs and received expert recommendations on several risk management measures.

**Classes.** The considered problem is formulated in terms of multiple criteria sorting with five preference ordered classes referring to the requirement of precautions:  $C_1$  (required; the least preferred class),  $C_2$  (might be required),  $C_3$  (optional),  $C_4$  (might be optional), and  $C_5$  (not required; the most preferred class).

**Criteria.** We consider a set of exposure scenarios to ENMs, which are characterized with ten descriptors. These criteria refer to the following characteristics of the materials and the exposure conditions:

- Particle size ( $g_1$ ) evaluated on a 6-point ordinal scale. Since most studies suggest that toxicity is higher for smaller sizes [41], but does not differ for greater sizes, we assumed an increase-level marginal value function.
- Toxicity ( $g_2$ ; cost type) defined on a 3-point scale (from low through medium to high) determines what type of effect the ENM has on human health [8].
- Airborne capacity ( $g_3$ ; cost type) – expressed on 4-point scale from none (preferred) to high (not preferred) – characterizes the capacity of the ENMs to spread in the workplace through the air stream [18].
- Detection limit ( $g_4$ ; gain type), defined on a qualitative 4-point scale (none (not preferred), low, moderate, and good (preferred)), relates to the capacity of the exposure assessment tools to identify ENMs.
- Exposure limit ( $g_5$ ; cost type) indicates an assumed level of exposure among five ranges based on asbestos, which is a widely accepted reference [32].
- Quantity ( $g_6$ ; cost type) refers to the quantity (in kg) of ENM handled in the scenario (lesser quantities imply smaller chance of exposure [19]).
- Number of employees ( $g_7$ ) considers the number of people involved in handling of the ENMs. Due to a lack of clear indication how this number affects the exposure management, we consider it as potentially non-monotonic criterion.
- Engineering controls ( $g_8$ ; non-monotonic) indicates the laboratory setting in which the manufacturing tasks are conducted among four possible combinations referring to positive (PP) or negative (NP) pressure as well as open (O) or closed (C) system.
- Duration of exposure ( $g_9$ ; cost type) to the nanomaterials during the manufacturing tasks (the shorter the duration, the lesser the risk of exposure [19]).
- Multiple exposure ( $g_{10}$ ; cost type) concerns the frequency of exposure [43] (unknown value is considered as the least preferred performance).

In Appendix A, we summarize the characteristics of all criteria as well as the encoding of respective performances. Overall, the considered descriptors involve both monotonic and non-monotonic criteria, which justifies an employment of the proposed methodological framework.

**Alternatives.** To demonstrate the framework's applicability, we consider a set of 51 exposure scenarios (for their performances, see Tables 1 and 2). In terms of MCDA, these are interpreted as decision alternatives  $a_1 - a_{51}$ .





### 3.2. Preference information

For thirty scenarios ( $a_1 - a_{30}$ ), we consider the expert input in form of class assignments. The respective classes capture the recommended precaution level for RMM (see Table 1). Overall, the distribution of classes in the reference judgments is as follows:  $C_1 - 5$ ,  $C_2 - 2$ ,  $C_3 - 10$ ,  $C_4 - 2$ , and  $C_5 - 11$ . Hence, the use of a respirator has been judged as optional ( $C_3$ ) or not required ( $C_5$ ) in the context of the greatest number of exposure scenarios. In addition, for better discrimination between the classes the minimal difference between the neighboring thresholds has been assumed to be 0.07.

### 3.3. Results

#### 3.3.1. Representative sorting model

The expert input has been used to develop a model to recommend a precaution level for workers exposed to nano-materials. In this way, we account for the interrelations between the ten descriptors of the exposure scenarios and the recommended risk management measure. Fig. 3 exhibits the marginal value functions which can constitute a part of the representative sorting model. They indicate two changes in the monotonicity, from decreasing to increasing for  $g_7$  (engineering controls) and from increasing to decreasing for  $g_8$  (number of employees). For all remaining criteria, the shape of marginal function adheres to the pre-defined monotonicity constraints. Specifically, for  $g_1$  (particle size) – it is increase-level, for  $g_4$  (detection limit) – it is increasing, whereas for the remaining criteria – it is either strictly decreasing or non-increasing.

Although all criteria contribute to the comprehensive values, one can observe significant differences in the maximal shares of respective marginal value functions. Specifically, the greatest maximal shares correspond to detection limit (0.2682), duration of exposure (0.1776), and airborne capacity (0.1454). This confirms a substantial impact that these criteria have on the classification. On the contrary, the least maximal shares are noted for quantity (0.0209) and frequency of exposure (0.0296), thus indicating their marginal role in deciding upon the sorting of considered scenarios.

When it comes to the variation of marginal values, it also differs vastly from one criterion to another. The greatest difference of marginal values can be observed for:

- airborne capacity ( $g_3$ ) when moving from moderate (2) to low (1) capacity of the nanomaterial to spread in the workplace;
- detection limit ( $g_4$ ) when attaining poor (1) rather than none (0) or good (3) rather than moderate (2) capacity of the exposure assessment tool to identify the nanomaterial;
- number of employees ( $g_8$ ) when moving to an intermediate level (11 – 50) from both lower and higher numbers;
- duration of exposure ( $g_9$ ) when reducing the time from less than one hour (3) to less than 15 minutes (2) and further to incidental occurrence (1).

These differences indicate the transitions where a high gain in the reduction of precaution level can be attained. On the contrary, the least or null differences of marginal values can be observed for particles sizes ( $g_1$ ) greater than 2 nm (2), at least moderate (2) toxicity ( $g_2$ ), intermediate (2 – 4) exposure limits ( $g_5$ ), quantities ( $g_6$ ) less than 10 tons (4), number of employees ( $g_8$ ) not less than 51 (4), and duration exposure ( $g_9$ ) not less than one minute (3). Consequently, changes of performances within these ranges do not influence at all or much the comprehensive values and resulting assignments.

The comprehensive values computed according to a representative value function for the reference exposure scenarios ( $a_1 - a_{30}$ ) are presented in Table 1. They need to be interpreted jointly with the following thresholds which set the boundaries for the ranges of comprehensive values judged as typical for particular classes:

$$t_0 = 0, t_1 = 0.3175, t_2 = 0.3933, t_3 = 0.4691, t_4 = 0.5449. \quad (6)$$

For example, all scenarios with comprehensive values not less than 0.3933 and less than 0.4691 are assigned to class  $C_3$ . Clearly, these thresholds were not pre-defined, but rather constructed by the method to reproduce – when coupled with an additive value function – all 30 assignment examples.

To support understanding of the employed threshold-based value-driven sorting procedure, Fig. 4 presents five example reference alternatives with different assignments along with their marginal values and thresholds separating the classes. Firstly, this figure demonstrates to which degree different criteria contribute to the comprehensive values of particular alternatives. Secondly, it clarifies that the assignment is derived from attaining a comprehensive value in a particular range. Thirdly, it exhibits the differences between the alternatives for which the requirement of precaution is, e.g., obligatory ( $a_9$ ), optional ( $a_3$ ), or not needed ( $a_{18}$ ).

In this perspective, an assignment of alternative  $a_{18}$  to class  $C_5$  ( $U(a_{18}) = 0.5522 \geq t_4 = 0.5449$ ) was largely due to its highly preferred performances on  $g_2$ ,  $g_4$ ,  $g_8$ , and  $g_9$ . In particular, its best performance with respect to the detection limit ( $g_4$ ) contributes already almost half of the comprehensive value needed for the assignment to the most preferred class. Furthermore, alternatives  $a_3$  and  $a_{29}$  attained high marginal value on a subset of criteria (for  $a_3 - g_1, g_4, g_7$ , and  $g_{10}$ ; for  $a_{29} - g_1, g_2, g_3, g_4$  and  $g_9$ ), but scored relatively worse on the remaining criteria (including four criteria with marginal values equal to zero), which justifies their assignment to the intermediate classes. When it comes to  $a_1$ , eight

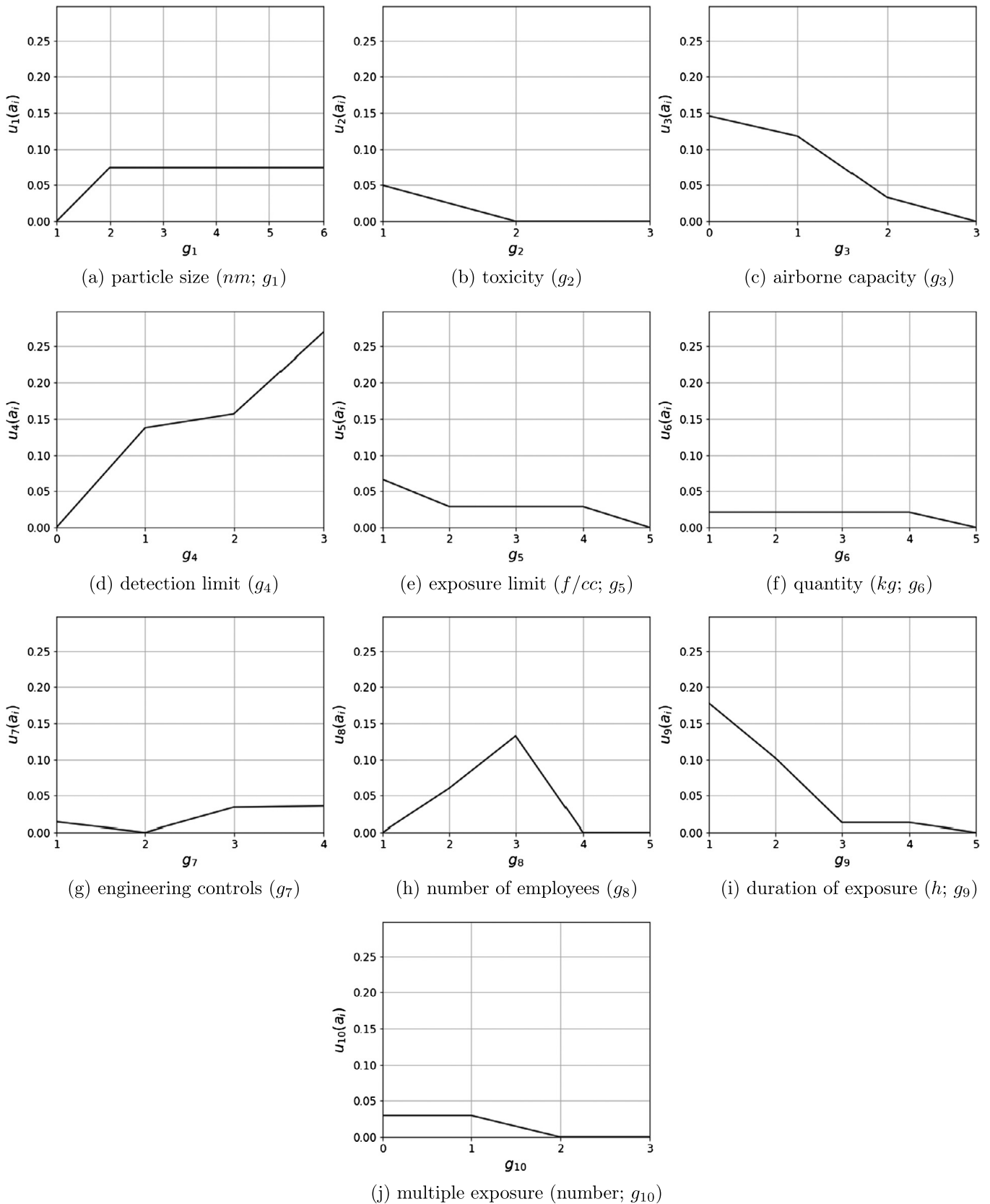


Fig. 3. Marginal value functions for the exposure management of nanomaterials in the context of using a respirator.

criteria contribute to its comprehensive quality with a marginal value greater than zero. However, only for three of them ( $g_1, g_3, g_8$ ), these contributions can be viewed as relatively high. As a result, the comprehensive value of  $a_1$  is rather low and sufficient only for granting a place in class  $C_2$  ( $t_1 = 0.3175 \leq U(a_1) = 0.3858 < t_2 = 0.3933$ ). Finally, alternative  $a_9$

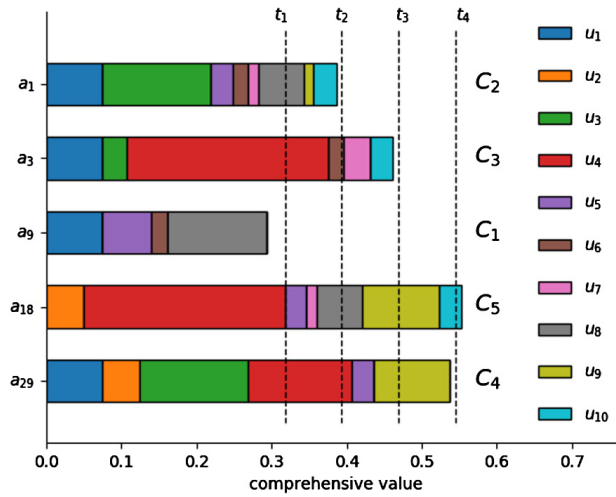


Fig. 4. Marginal and comprehensive values as well as class assignments for the five example reference exposure scenarios. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

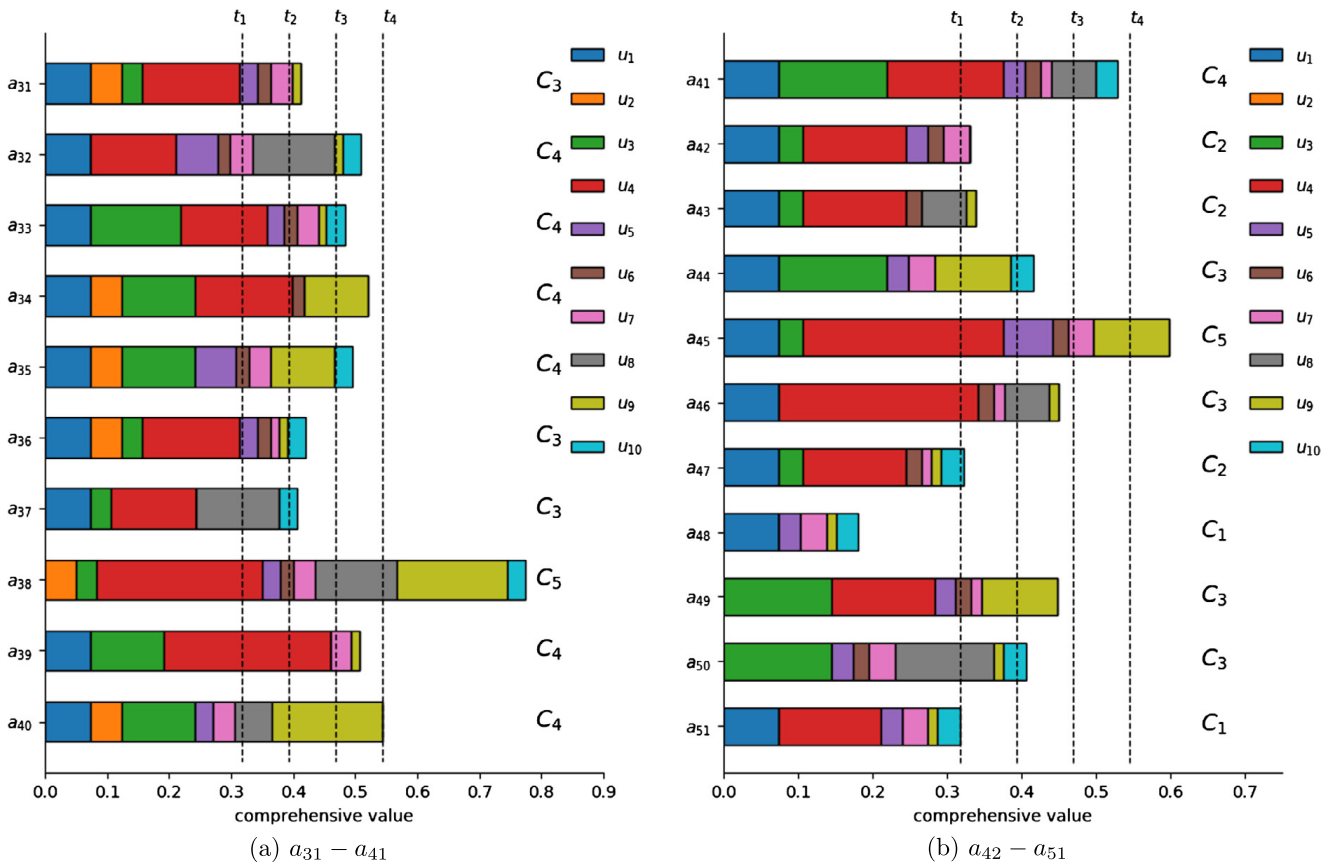


Fig. 5. Marginal and comprehensive values as well as class assignments for the 21 non-reference exposure scenarios.

attained a marginal value of zero on six criteria ( $g_2, g_3, g_4, g_7, g_9,$  and  $g_{10}$ ), which implied its assignment to the least preferred class  $C_1$ .

As far as non-reference exposure scenarios ( $a_{31} - a_{51}$ ) are concerned, their comprehensive values and class assignments are presented in Table 2. The explanation of these assignments is enhanced by Fig. 5, which collates the comprehensive values with the class thresholds, while additionally decomposing them into the marginal values. For the 21 non-reference alternatives, the distribution of class assignments is as follows:  $C_1 - 2, C_2 - 3, C_3 - 7, C_4 - 7,$  and  $C_5 - 2$ . Hence, the precaution level of the greatest number of exposure scenarios is judged as optional ( $C_3$ ) or might be optional ( $C_4$ ), whereas only a pair of alternatives is assigned to either of extreme classes.

Let us explain the classification for a few selected non-reference exposure scenarios by referring to the contribution of particular criteria to their comprehensive values. When it comes to a pair of alternatives ( $a_{48}$  and  $a_{51}$ ) assigned to  $C_1$ , they

perform relatively well only on the less important criteria, while not attaining any positive marginal values on  $g_2$ ,  $g_3$ ,  $g_4$  (only for  $a_{48}$ ),  $g_6$ , and  $g_8$ . This justifies their low comprehensive values and sorting into the least preferred class. When compared with  $a_{48}$ ,  $a_{44}$  is characterized by more advantageous performances on  $g_3$  and  $g_9$ , which is sufficient for attaining an intermediate class ( $C_3$ ). As for the two alternatives ( $a_{38}$  and  $a_{45}$ ) placed in  $C_5$ , 9 and 7 criteria, respectively, contribute to their comprehensive values. However, the main role in exceeding the lower threshold of the most preferred class can be attributed to: for  $a_{38}$  –  $g_2$ ,  $g_4$ ,  $g_8$ , and  $g_9$ , and for  $a_{45}$  –  $g_1$ ,  $g_4$ ,  $g_5$ , and  $g_9$ , on which they reached the maximal values. The alternatives assigned to the least preferred classes either lack any contribution from the significant share of criteria (see, e.g.,  $a_{37}$  –  $C_2$  or  $a_{39}$  –  $C_4$ ), or have rather unbalanced performance profiles with significant contribution from some criteria and relative poor evaluations on the remaining ones (e.g.,  $a_{50}$  –  $C_3$  or  $a_{32}$  –  $C_4$ ), or simply attain average performances on the vast majority of criteria, hence lacking substantial contribution from the most important descriptors (see, e.g.,  $a_{36}$  –  $C_3$  or  $a_{41}$  –  $C_4$ ).

### 3.3.2. Robustness analysis

The recommendation obtained for the non-reference exposure scenarios with the representative instance of the sorting model is validated against the outcomes of robustness analysis. The possible assignments obtained through the analysis of all compatible instances of the sorting model admitting two changes in monotonicity for all marginal value functions are presented in Table 2. These possible assignments are precise only for 3 out of 21 alternatives. Hence, the assignment of  $a_{36}$  to  $C_3$ ,  $a_{38}$  to  $C_5$ , and  $a_{48}$  to  $C_1$  can be judged as robust in view of the incompleteness of the DM's assignment examples and multiplicity of compatible sorting models.

For the remaining non-reference exposure scenarios, the possible assignments are imprecise, thus indicating a hesitation with respect to the recommended class. However, for 17 alternatives just two classes are possible, and only for  $a_{35}$  – three classes can be recommended depending on the choice of a compatible sorting model. The additional classes contained in the possible assignment are more or less preferred than the classes suggested by the representative model for, respectively, 8 (e.g.,  $a_{34}$  and  $a_{46}$ ) and 11 (e.g.,  $a_{31}$  and  $a_{45}$ ) alternatives.

The analysis of such possible assignments allows to indicate the classes that cannot be viewed as an admissible result, because they are not confirmed by any compatible instance of the sorting model. For example, since  $a_{34}$  is possibly assigned to  $C_4$  or  $C_5$ , the recommended precaution is surely not required ( $C_1$ ), not might be required ( $C_2$ ), nor optional ( $C_3$ ). Similarly, since  $C_P(a_{51}) = \{C_1, C_2\}$ , the following requirements of precautions are excluded: optional ( $C_3$ ), might be optional ( $C_4$ ), and not required ( $C_5$ ).

### 3.4. Discussion

The proposed model could be a first tiered solution to exposure management of nanomaterials, similarly to the step-wise strategies proposed for the exposure assessment phase [20,38]. It could be used to provide an initial indication of concern regarding specific tasks performed by the workers. In this way, when the proposed model indicates that the assigned class is at most  $C_3$  (indicating required, potentially required, or optional precaution level), these tasks should be given priority and further investigated as they can be seen as “safety warning flags”. Obviously, the less preferred the class, the greater attention should be paid to the analysis of a respective task. For such alternatives, the choices of the health managers could be directed towards working on the criteria of the model, i.e., characteristics of the materials and the exposure conditions, to see whether any of them can be modified to increase a comprehensive value and to trigger a more preferred class. The analysis of marginal value functions provides directions on which performance changes offer the greatest gains in this regard and which modifications do not lead to significant improvements.

Let us also emphasize that the primary aim of Section 3 was to illustrate the applicability of the proposed method in a standard MCDA setting. In this setting, the DM's preference information is used to construct a preference model compatible with the DM's value system. Such a model is subsequently employed to evaluate the non-reference alternatives that have not been directly judged by the DM, in a way that would be acceptable for him/her, being consistent with his/her preferences. Thus, in typical MCDA applications the objective truth to be discovered does not exist as the true classification for the non-reference alternatives is not known. However, it can be analyzed for the considered study, because the most appropriate assignments for the non-reference exposure scenarios have also been determined by the experts. In this regard, for 12 out of 21 non-reference scenarios ( $a_{36}$ ,  $a_{37}$ ,  $a_{38}$ ,  $a_{42}$ ,  $a_{44}$ ,  $a_{45}$ ,  $a_{46}$ ,  $a_{47}$ ,  $a_{48}$ ,  $a_{49}$ ,  $a_{50}$ , and  $a_{51}$ ) the assignments obtained with a representative sorting model instance agree with the actual ones. Moreover, for the remaining 9 non-reference scenarios, the actual class is contained in the set of possible assignments, hence being confirmed by at least one compatible sorting model instance.

## 4. Conclusions

We proposed a novel approach for multiple criteria sorting incorporating a threshold-based value-driven procedure. The parameters of the constructed model deciding upon the shape of marginal value functions and separating class thresholds are inferred through disaggregation of the assignment examples provided by the DM. This is attained by solving dedicated MILP problems. Apart from accounting for the incomplete preference information, the method allows the DM to specify partial requirements on the assumed type of (non-)monotonicity for the individual criteria. Specifically, we considered gain

and cost attributes, monotonic functions without a pre-defined preference direction, level-monotonic shapes, A- and V-types combining increasing and decreasing trends within a single function, and lack of monotonicity constraints. In this perspective, to control the complexity and interpretability of the inferred model, we minimized the number of changes in monotonicity for all marginal value functions.

The characteristic of the provided results is two-fold. On the one hand, we derive univocal assignments with a representative instance of the sorting model. The analysis of such a model allows capturing the trade-offs between different criteria, assessment of their relative importance, and indication of performance changes that can be viewed as the most advantageous in terms of improving a comprehensive quality. On the other hand, we perform robustness analysis and quantify its results by means of possible assignments. They confirm which classes are admissible for a given alternative for at least one compatible instance of the sorting model. Moreover, they allow to reject the hypotheses concerning the assignments which are not confirmed by any model.

The proposed approach can be seen as a sorting counterpart of the method proposed by Rezaei [34]. However, it does not require to pre-define the exact shapes of marginal value functions, tolerating instead partial information on the monotonicity constraints. Moreover, it extends the algorithm introduced by Kliegr [25] to a broad family of shapes of marginal functions as well as to a robustness analysis which accounts for all compatible instances of the sorting model with the minimal possible complexity.

Apart from the methodological advances, the paper contributes to the literature by exhibiting its applicability to analysis of a real-world sorting problem. Specifically, we considered the problem of exposure management for engineered nanomaterials, and used expert judgments to develop a model for predicting precaution level while handling nanomaterials in certain conditions using a respirator. The model was able to capture the interrelations between ten criteria – including monotonic descriptors, a single increase-level criterion, and a pair of non-monotonic attributes – and the recommended assignments.

The analysis of a representative instance of the sorting model allowed to identify the criteria that significantly affected the recommended sorting. These descriptors involved detection limits, airborne capacity, and duration of exposure. On the contrary, quantity of nanomaterial, frequency of exposure, and engineering controls had the least share in the comprehensive values of exposure scenarios. The classes were well-separated due to a significant difference between the inferred thresholds. In this way, each class accommodated multiple alternatives with diverse characteristics on the individual criteria that considered jointly could be holistically judged as required, optional, or absolutely redundant with respect to the precaution level. The case study also demonstrated how the representative and univocal results can be enriched with the outcomes of robustness analysis. Specifically, we showed the usefulness of possible assignments for capturing the uncertainty related to the recommended classification as a consequence of incompleteness of the DM's preference information.

The potential extensions of the proposed method and the considered case study are five-fold. Firstly, the practical applicability of our approach is limited due to a high number of binary variables. In fact, it depends on the numbers of criteria and unique performances of alternatives. Hence, a potential revision of the method needs to control the model's complexity and impose monotonicity constraints without incorporating binary variables. Secondly, we assumed that the model's complexity can be adjusted to reproduce all assignment examples by increasing the number of changes in monotonicity. In case this is not possible, one can apply the standard procedures for eliminating the minimal number of assignment examples underlying the inconsistency [30,31]. However, since they are based on MILP and associate a unique binary variable with each assignment example, their applicability is also limited to few hundred of holistic judgments. Dealing with larger sets of potentially inconsistent example assignments requires the development of some dedicated heuristic approaches.

Furthermore, the family of shapes of marginal value functions can be extended to account for polynomials and splines, whose interpretability is desirable in many real-world applications [40]. Moreover, the method can be easily adapted to multiple criteria ranking and choice. Instead of incorporating the DM's assignment examples, it should accept pairwise comparisons of reference alternatives.

Finally, motivated by the peculiarity of exposure management of nanomaterials, the method can be enriched to account for a few decision attributes simultaneously. In this specific application, they would represent different risk management measures [32]. For each classification problem, one should construct a dedicated sorting model reproducing the provided assignment examples. However, the individual models should be interrelated to reflect the dependencies between classes desired for the same alternative on various decision attributes.

## Declaration of competing interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

## Acknowledgements

Miłosz Kadziński acknowledges support from the Polish Ministry of Science and Higher Education under grant no. 0296/IP2/2016/74. Marco Cinelli acknowledges funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 743553. Salvatore Corrente and Salvatore Greco acknowledge funding by the FIR of

**Table A.3**

Encoding of performances on ten criteria used for the management of exposure scenarios to nano-materials.

$g_j$	Criterion	Preference	Performance	Code
$g_1$	Particle size (nm)	increase-level	< 2	1
			2 – 10	2
			10 – 100	3
			100 – 500	4
			500 – 1000	5
			> 1000	6
$g_2$	Toxicity	cost	Low	1
			Moderate	2
			High	3
$g_3$	Airborne capacity	cost	None	0
			Low	1
			Moderate	2
$g_4$	Detection limit	gain	None	0
			Poor	1
			Moderate	2
$g_5$	Exposure limit (fiber/cc)	cost	Good	3
			< 0.1	1
			0.1 – 0.2	2
$g_6$	Quantity (kg)	cost	0.2 – 0.5	3
			0.5 – 1.0	4
			> 1.0	5
			< 1	1
			1 – 100	2
$g_7$	Engineering controls	non-monotonic	100 – 1000	3
			1000 – 10000	4
			> 10000	5
			Open-PP	1
			Open-NP	2
$g_8$	Number of employees	non-monotonic	Closed	3
			Closed-NP	4
			1 – 3	1
			3 – 10	2
			11 – 50	3
$g_9$	Duration of exposure (h)	cost	51 – 100	4
			101 – 500	5
			incidental	1
			< 0.25	2
			< 1	3
$g_{10}$	Multiple exposure (number)	cost	1 – 5	4
			5 – 8	5
			none	0
			1 – 3	1
			> 3	2
			unknown	3

the University of Catania BCAEA3 “New developments in Multiple Criteria Decision Aiding (MCDA) and their application to territorial competitiveness”. Salvatore Greco has also benefited of the fund Chance of the University of Catania.

#### Appendix A. Encoding of performances for the case study

Table A.3 summarizes the characteristics of ten criteria used for the management of exposure scenarios to nanomaterials as well as the encoding of respective performances.

#### References

- [1] J. Błaszczyński, S. Greco, R. Słowiński, Inductive discovery of laws using monotonic rules, *Eng. Appl. Artif. Intell.* 25 (2013) 284–294.
- [2] J. Branke, S. Corrente, S. Greco, W. Gutjahr, Efficient pairwise preference elicitation allowing for indifference, *Comput. Oper. Res.* 88 (2017) 175–186.
- [3] K. Ciomek, M. Kadziński, T. Tervonen, Heuristics for prioritizing pair-wise elicitation questions with additive multi-attribute value models, *Omega* 71 (2017) 27–45.
- [4] K. Ciomek, M. Kadziński, T. Tervonen, Heuristics for selecting pair-wise elicitation questions in multiple criteria choice problems, *Eur. J. Oper. Res.* 262 (2) (2017) 693–707.
- [5] S. Corrente, S. Greco, M. Kadziński, R. Słowiński, Robust ordinal regression in preference learning and ranking, *Mach. Learn.* 93 (2–3) (2013) 381–422.



- [6] D.K. Despotis, C. Zopounidis, Building additive utilities in the presence of non-monotonic preferences, in: *Advances in Multicriteria Analysis*, Springer, Boston, 1995, pp. 101–114.
- [7] M. Doumpos, Learning non-monotonic additive value functions for multicriteria decision making, *OR Spektrum* 34 (1) (2012) 89–106.
- [8] D. Fadeel, L. Farcal, B. Hardy, S. Vázquez-Campos, D. Hristozov, A. Marcomini, I. Lynch, E. Valsami-Jones, H. Alenius, K. Savolainen, Advanced tools for the safety assessment of nanomaterials, *Nat. Nanotechnol.* 13 (2018) 537–543.
- [9] M.M. Falinski, D.L. Plata, S.S. Chopra, T.L. Theis, L.M. Gilbertson, J.B. Zimmerman, A framework for sustainable nanomaterial selection and design based on performance, hazard, and economic considerations, *Nat. Nanotechnol.* 13 (2018) 708–714.
- [10] M. Ghaderi, F. Ruiz, N. Agell, Understanding the impact of brand colour on brand image: a preference disaggregation approach, *Pattern Recognit. Lett.* 67 (2015) 11–18.
- [11] M. Ghaderi, F. Ruiz, N. Agell, A linear programming approach for learning non-monotonic additive value functions in multiple criteria decision aiding, *Eur. J. Oper. Res.* 259 (3) (2016) 1073–1084.
- [12] H. Goede, Y. Christopher-de Vries, E. Kuijpers, W. Fransman, A review of workplace risk management measures for nanomaterials to mitigate inhalation and dermal exposure, *Ann. Work. Expo. Heal.* 62 (2018) 907–922.
- [13] S. Greco, V. Mousseau, R. Słowiński, Ordinal regression revisited: multiple criteria ranking using a set of additive value functions, *Eur. J. Oper. Res.* 191 (2) (2008) 415–435.
- [14] S. Greco, V. Mousseau, R. Słowiński, Multiple criteria sorting with a set of additive value functions, *Eur. J. Oper. Res.* 207 (3) (2010) 1455–1470.
- [15] S. Greco, M. Kadziński, R. Słowiński, Selection of a representative value function in robust multiple criteria sorting, *Comput. Oper. Res.* 38 (11) (2011) 1620–1637.
- [16] S. Greco, M. Ehrgott, J.R. Figueira (Eds.), *Multiple Criteria Decision Analysis: State of the Art Surveys*, International Series in Operations Research and Management Science, vol. 233, Springer, New York, 2016.
- [17] M. Guo, X. Liao, J. Liu, A progressive sorting approach for multiple criteria decision aiding in the presence of non-monotonic preferences, *Expert Syst. Appl.* 123 (2018) 1–17.
- [18] S. Hansen, K. Jensen, A. Baun, NanoRiskCat: a conceptual tool for categorization and communication of exposure potentials and hazards of nanomaterials in consumer products, *J. Nanopart. Res.* 16 (2013) 1–25.
- [19] D. Hristozov, S. Gottardo, M. Cinelli, P. Isigonis, A. Zabeo, A. Critto, M. Van Tongeren, L. Tran, A. Marcomini, Application of a quantitative weight of evidence approach for ranking and prioritization of occupational exposure scenarios for titanium dioxide and carbon nanomaterials, *Nanotoxicology* 8 (2014) 117–131.
- [20] P. Isigonis, D. Hristozov, C. Benighaus, E. Giubilato, K. Grieger, L. Pizzol, E. Semenzin, I. Linkov, A. Zabeo, A. Marcomini, Risk governance of nanomaterials: review of criteria and tools for risk communication, evaluation, and mitigation, *Nanomaterials* 9 (2019) 696.
- [21] E. Jacquet-Lagrèze, Y. Siskos, Preference disaggregation: 20 years of MCDA experience, *Eur. J. Oper. Res.* 130 (2) (2001) 233–245.
- [22] M. Kadziński, K. Ciomek, Integrated framework for preference modeling and robustness analysis for outranking-based multiple criteria sorting with ELECTRE and PROMETHEE, *Inf. Sci.* 352 (2016) 167–187.
- [23] M. Kadziński, S. Corrente, S. Greco, R. Słowiński, Preferential reducts and constructs in robust multiple criteria ranking and sorting, *OR Spektrum* 36 (4) (2014) 1021–1053.
- [24] R. Keeney, H. Raiffa, *Decisions With Multiple Objectives: Preferences and Value Tradeoffs*, Wiley, New York, 1976.
- [25] T. Kliegr, UTA-NM: explaining stated preferences with additive non-monotonic utility functions, in: *Preference Learning (PL-09) ECML/PKDD-09 Workshop*, 2009.
- [26] M. Köksalan, S.B. Özpeynirci, An interactive sorting method for additive utility functions, *Comput. Oper. Res.* 36 (9) (2009) 2565–2572.
- [27] V.D. Krishna, K. Wu, D. Su, M.C.J. Cheeran, J.-P. Wang, A. Perez, Nanotechnology: review of concepts and potential application of sensing platforms in food safety, *Food Microbiol.* 75 (2018) 47–54.
- [28] R. Lahdelma, P. Salminen, Stochastic Multicriteria Acceptability Analysis (SMAA), in: M. Ehrgott, J.R. Figueira, S. Greco (Eds.), *Trends in Multiple Criteria Decision Analysis*, Springer, Boston, 2010, pp. 285–315.
- [29] J. Liu, X. Liao, M. Kadziński, R. Słowiński, Preference disaggregation within the regularization framework for sorting problems with multiple potentially non-monotonic criteria, *Eur. J. Oper. Res.* 276 (3) (2019) 1071–1089.
- [30] V. Mousseau, L.C. Dias, J.R. Figueira, Dealing with inconsistent judgments in multiple criteria sorting models, *4OR* 4 (2) (2006) 145–158.
- [31] V. Mousseau, V. Figueira, L.C. Dias, C.G. da Silva, J. Climaco, Resolving inconsistencies among constraints on the parameters of an MCDA model, *Eur. J. Oper. Res.* 147 (1) (2003) 72–93.
- [32] S.R. Naidu, *Towards Sustainable Development of Nanomanufacturing*, PhD thesis, University of Tennessee, 2012, [http://trace.tennessee.edu/cgi/viewcontent.cgi?article=2413&context=utk\\_graddiss](http://trace.tennessee.edu/cgi/viewcontent.cgi?article=2413&context=utk_graddiss).
- [33] C. Oksel, N. Hunt, T. Wilkins, X.Z. Wang, Risk management of nanomaterials - guidelines for the safe manufacture and use of nanomaterials. Sustainable nanotechnologies project, <http://www.sun-fp7.eu>, 2017.
- [34] J. Rezaei, Piecewise linear value functions for multi-criteria decision-making, *Expert Syst. Appl.* 98 (2018) 43–56.
- [35] M.C. Roco, C.A. Markin, M.C. Hersam, *Nanotechnology Research Directions for Societal Needs in 2020: Retrospective and Outlook*, Springer, 2011.
- [36] B. Roy, *Multicriteria Methodology for Decision Aiding*, Kluwer Academic, Dordrecht, 1996.
- [37] A. Salo, R. Hämäläinen, Preference programming – multicriteria weighting models under incomplete information, in: C. Zopounidis, P.M. Pardalos (Eds.), *Handbook of Multicriteria Analysis*, Springer, Berlin, 2010.
- [38] P. Arezes, P. Swuste, 29 – risk management: controlling occupational exposure to nanoparticles in construction, in: F. Pacheco-Torgal, M.V. Diamanti, A. Nazari, C.G. Granqvist, A. Pruna, S. Amirkhanian (Eds.), *Nanotechnology in Eco-Efficient Construction*, second edition, Woodhead Publishing Limited, Cambridge, 2019, pp. 755–784.
- [39] Y. Siskos, E. Grigoroudis, N. Matsatsinis, UTA methods, in: S. Greco, M. Ehrgott, J.R. Figueira (Eds.), *Multiple Criteria Decision Analysis: State of the Art Surveys*, Springer, Boston, 2016, pp. 315–362.
- [40] O. Sobrie, N. Gillis, V. Mousseau, M. Pirlot, UTA-poly and UTA-splines: additive value functions with polynomial marginals, *Eur. J. Oper. Res.* 264 (2) (2018) 405–418.
- [41] S.K. Sohaebuddin, P.T. Thevenot, D. Baker, J.W. Eaton, L. Tang, Nanomaterial cytotoxicity is composition, size, and cell type dependent, *Part. Fibre Toxicol.* 7 (2010) 1–17.
- [42] V. Stone, M. Fuhr, P.H. Feindt, H. Bouwmeester, I. Linkov, S. Sabella, F. Murphy, K. Bizer, L. Tran, M. Agerstrand, C. Fito, T. Andersen, D. Anderson, E. Bergamaschi, J.W. Cherrie, S. Cowan, J.F. Dalemcourt, M. Faure, S. Gabbert, A. Gajewicz, T.F. Fernandes, D. Hristozov, H.J. Johnston, T.C. Lansdown, S. Linder, H.J.P. Marvin, M. Mullins, K. Purnhagen, T. Puzyn, A. Sanchez Jimenez, J.J. Scott-Fordsmann, G. Strefataris, M. van Tongeren, N.H. Voelcker, G. Voyiatzis, S.N. Yannopoulos, P.M. Poortvliet, The essential elements of a risk governance framework for current and future nanotechnologies, *Risk Anal.* 38 (2018) 1321–1331.
- [43] B. Van Duuren-Stuurman, S.R. Vink, K.J.M. Verbist, H.G.A. Heussen, D.H. Brouwer, D.E.D. Kroese, M.F.J. Van Niftrik, E. Tielemans, W. Fransman, *Stoffenmanager nano version 1.0: a web-based tool for risk prioritization of airborne manufactured nano objects*, *Ann. Occup. Hyg.* 56 (2012) 525–541.
- [44] W. Zhang, D. Zhang, Y. Liang, Nanotechnology in remediation of water contaminated by poly- and perfluoroalkyl substances: a review, *Environ. Pollut.* 247 (2019) 266–276.



- [45] X. Zhao, J. E. G. Wu, Y. Deng, D. Han, B. Zhang, Z. Zhang, A review of studies using graphenes in energy conversion, energy storage and heat transfer development, *Energy Convers. Manag.* 184 (2019) 581–599.
- [46] C. Zopounidis, M. Doumpos, PREFDIS: a multicriteria decision support system for sorting decision problems, *Comput. Oper. Res.* 27 (7–8) (2000) 779–797.
- [47] C. Zopounidis, M. Doumpos, Multicriteria classification and sorting methods: a literature review, *Eur. J. Oper. Res.* 138 (2002) 229–246.



## Publication [P2]

M. Kadziński, K. Martyn, M. Cinelli, R. Słowiński, 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*, 218:106879, 2021, DOI: 10.1016/j.knosys.2021.106879.

Number of citations<sup>1</sup>:

- according to Web of Science: 9
- according to Google Scholar: 11

---

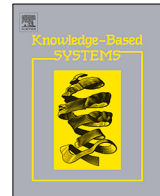
<sup>1</sup>as on June 1, 2023





Contents lists available at ScienceDirect

# Knowledge-Based Systems

journal homepage: [www.elsevier.com/locate/knosys](http://www.elsevier.com/locate/knosys)

## Preference disaggregation method for value-based multi-decision sorting problems with a real-world application in nanotechnology



Miłosz Kadziński<sup>a,\*</sup>, Krzysztof Martyn<sup>a,b</sup>, Marco Cinelli<sup>a</sup>, Roman Słowiński<sup>a,c</sup>,  
Salvatore Corrente<sup>d</sup>, Salvatore Greco<sup>d,e</sup>

<sup>a</sup> Institute of Computing Science, Poznań University of Technology, Poznań, Poland

<sup>b</sup> Poznań Supercomputing and Networking Center, Institute of Bioorganic Chemistry PAS, Poznań, Poland

<sup>c</sup> Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland

<sup>d</sup> Department of Economics and Business, University of Catania, Catania, Italy

<sup>e</sup> CORL, Portsmouth Business School, University of Portsmouth, Portsmouth, United Kingdom

### ARTICLE INFO

#### Article history:

Received 28 August 2020

Received in revised form 2 November 2020

Accepted 16 February 2021

Available online 18 February 2021

#### Keywords:

Multiple criteria sorting

Multiple decisions

Preference disaggregation

Non-monotonic value functions

Nanomaterials

Precaution level

### ABSTRACT

We consider a problem of multi-decision sorting subject to multiple criteria. In the newly formulated decision problem, besides performances on multiple criteria, alternatives get evaluations on multiple interrelated decision attributes involving preference-ordered classes. We propose a dedicated method for dealing with such a problem, incorporating a threshold-based value-driven sorting procedure. The Decision Maker (DM) is expected to holistically evaluate a subset of reference alternatives by indicating the quality or risk level on a pre-defined scale of each decision attribute. Based on these evaluations, we construct a set of interrelated preference models, one for each decision attribute, compatible with intra- and inter-decision constraints imposed by such indirect preference information. We also formulate a new way of dealing with potentially non-monotonic criteria by discovering local monotonicity changes in different performance scale regions. The marginal value functions for criteria with unknown monotonicity are represented as a sum of two value functions assuming opposing preference directions, one non-decreasing and the other non-increasing. This permits to obtain an aggregated marginal value function with an arbitrary non-monotonic shape. The practical usefulness of the approach is demonstrated on a case study concerning risk management related to handling (i.e., production, use, manipulation, and processing) nanomaterials in different conditions. We analyze the expert judgments and discuss the inferred preference models, which can be applied to support health and safety managers in reducing the possible risk associated with the respective exposure scenario.

© 2021 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

### 1. Introduction

Multiple Criteria Decision Analysis (MCDA) concerns decision problems where a set of alternatives are evaluated on a family of criteria, which represent all relevant, heterogeneous viewpoints on the quality of alternatives [1,2]. Many such decision problems fall into the general category of classification, where the alternatives need to be assigned to distinct classes [3]. If the classes are completely ordered, one deals with ordinal classification or, equivalently, sorting problems [4]. They are considered, e.g., in the ABC analysis, which is a type of inventory categorization method where high-, mid-, and low-value alternatives need to be identified [5], in medical diagnosis, where high-risk patients need to be distinguished from the low-risk ones [6], in business

failure prediction, where firms are sorted into healthy, uncertain, and close to bankruptcy [7], or in nanomanufacturing, where synthesis processes of nanomaterials can be sorted according to their greenness level [8].

Decision aiding sorting methods aim to provide recommendations to the Decision Maker (DM) regarding the assignment of alternatives to pre-defined and ordered classes [9]. There are various approaches serving this purpose though differing with respect to the underlying assumptions and characteristics of delivered results. In particular, various methods expect the DM to provide different types of preference information through a co-constructive elicitation process led by a decision analyst. On the one hand, some methods assume that the DM would directly specify values for a set of parameters of an assumed sorting model [10,11]. However, this is a bit unrealistic because (s)he usually has difficulties with keeping consistency between the supplied values and the model output [4]. On the other hand, preference disaggregation procedures have been proposed to prevent such difficulties [12]. They aim at deriving compatible model

\* Correspondence to: Institute of Computing Science, Poznań University of Technology, Piotrowo 2, 60-965 Poznań, Poland.

E-mail address: [miłosz.kadziński@cs.put.poznan.pl](mailto:miłosz.kadziński@cs.put.poznan.pl) (M. Kadziński).

parameters from the analysis of the DM's comprehensive judgments (assignment examples) concerning a subset of reference alternatives. This allows generalizing the DM's policy to an entire set of alternatives through the use of a suitably parametrized sorting model. Specifically, UTADIS disaggregates the DM's assignment examples into marginal value functions and class thresholds separating the consecutive decision classes on a scale of comprehensive value [13,14]. This idea was found appealing in such various fields as finance [15], energy management [16], or stock portfolio analysis [17].

Motivated by the complexity of real-world sorting problems, UTADIS has been extended in various ways. First, different procedures for dealing with inconsistency of assignment examples with an assumed model have been proposed [14,18]. Second, a hierarchical classification approach, called MHDIS, has been introduced in [19]. Third, the Multiple Criteria Hierarchy Process has been adapted to UTADIS to allow handling preference information and deriving recommendations at both comprehensive and intermediate levels of the hierarchy of criteria [20]. Moreover, novel preference modeling procedures have been designed to admit the specification of desired class cardinalities, assignment-based pairwise comparisons [21], or valued assignment examples [22]. Furthermore, the frameworks for robustness analysis have been proposed [4,23,24] to exploit infinitely many instances of the preference model (e.g., value functions and class thresholds) compatible with the DM's holistic decisions. While all approaches mentioned above considered a single DM, [25] introduced a group decision framework, called UTADIS<sup>GMS</sup>-GROUP, investigating the spaces of agreement and disagreement between sorting recommendations obtained for different DMs. Some other recent methodological advancements of the UTADIS method concern the form of an employed value-based model. In this regard, an additive value function has been extended in [26] to account for positively and negatively interacting pairs of criteria. The other stream concerned dealing with the non-monotonicity of preferences on a per-criterion level. In particular, [27] defined a broad spectrum of non-monotonic shapes that could be considered along with the gain- and cost-type criteria. Moreover, [28] and [29] introduced the models admitting non-monotonicity of marginal value functions while not restraining their complexity. In turn, [29,30] and [31] minimized, respectively, the variation in the slope or the number of changes of non-monotonicity in the shape of marginal value functions to ensure the most interpretable sorting model.

The contribution of this paper is three-fold. First, we introduce a new problem of multi-decision sorting in MCDA and propose a dedicated method for dealing with it. In this problem, each alternative is evaluated in terms of multiple decision attributes involving preference-ordered classes. We expect the DM to assign a subset of reference alternatives to classes of each decision attribute by indicating a quality or risk level on a scale pre-defined for all decision attributes. A similar setting has been considered in [32] and [33] in the context of credit rating problems. On the one hand, [32] adapted the UTADIS method to infer a single threshold-based value-driven sorting model compatible with the ratings provided by Moody's and Standard & Poor's, hence providing a precise recommendation based on potentially conflicting inputs for the same alternative. On the other hand, [33] used the three credit rating agencies' recommendations to form an interval rating that was subsequently used as a potentially imprecise reference benchmark to be reproduced by the ELECTRE TRI-nC method [34].

The multi-decision sorting problem and dedicated approach introduced in this paper are original in the sense of requiring construction of a set of interrelated preference models, one for each decision attribute. Such a requirement contrasts with the

inference of a single sorting model that would align with multiple classifications at the same time [32] or an imprecise assignment built on multiple ratings for the same alternative [33]. Specifically, we propose a threshold-based value-driven sorting method. It involves a set of intra- and inter-decision constraints. The former ones ensure appropriate relations between comprehensive values of different alternatives for an individual value function used to derive the assignments for a single decision attribute. The latter correspond to the relations between comprehensive values of the same alternative for multiple value functions employed for classifying this alternative given various decision attributes. This makes sense when the classes of various decision attributes correspond to the same default categories, having the same scope and interpretation.

This paper's second contribution derives from presenting the results of a case study concerning risk management related to handling nanomaterials in different conditions [35]. The production, processing, and use of nanomaterials may lead to health or life exposure. Depending on the particular exposure scenario, different types of precautions or safety measures can be used to counteract the respective risk [36]. Also, some precautions meet general hazards, whereas others are dedicated to deal with some specific dangers. Each of the precaution types (e.g., incorporation of some personal protective equipment, engineering controls, or work practices) can be interpreted as a decision attribute with pre-defined preference-ordered classes representing different levels of risk [37]. When facing hazards that particular nanomaterials carry with them, some precautions are required, and others are optional or unnecessary [31]. However, different precautions are not independent, being defined on the same set of criteria and related in terms of their interpretation.

There is a need for a method dealing with multiple interrelated preference-ordered decision attributes to tackle such a problem. In this regard, MCDA has little to offer. This, in turn, implies that such complex problems would typically be decomposed into independent ones. This would allow for modeling intra-decision dependencies, neglecting the inter-decision relations that could negatively affect results' usefulness. Other ideas are solutions derived from multi-label classification, such as label powerset [38] or probabilistic classifier chains [39]. The label powerset generates a vast number of classes and requires many examples so that each class has a sufficient number of its representatives. The latter is difficult to satisfy for the case study. Probabilistic classifier chains offer different solutions depending on the order of the decisions under consideration and require repeated solving of the same problem. The limitations of the existing approaches motivated the development of a dedicated multiple criteria sorting method. In the context of the considered case study, the information coming from the proposed approach will help the DMs in assessing the risk related to the treatment of nanomaterials in different conditions. Specifically, it can be used for recommending the level of need for the use of specific personal protective equipment, engineering controls, or work practices.

Our third contribution consists of proposing a new way of dealing with potentially non-monotonic criteria. Non-monotonic criteria appear in the MCDA problem when, for some attributes, neither the univocal preference direction could be specified, nor the non-monotonic shape of respective marginal value function could be defined a priori. This happens in our case study. Then, such a shape needs to be inferred from data describing the multiple criteria problem and the DM's holistic judgments. In particular, the method should verify whether a monotonic relationship exists, if it is of gain- or cost-type, or if the monotonicity is not global. The latter scenario could reveal some local positive or negative relationships in different parts of the investigated performance scale [40]. To perform this task, we propose a new

approach that attempts to discover local monotonicity changes without requiring the DM to fix the preference directions for all criteria. Specifically, we represent the marginal value functions of potentially non-monotonic criteria as a sum of marginal value functions assuming opposing preference directions, one non-decreasing and the other non-increasing. This permits to obtain an aggregated marginal value function with an arbitrary non-monotonic shape. However, the two monotonic components remain easy to interpret.

The remainder of the paper is organized in the following way. Section 2 is devoted to the new method dealing with multi-decision sorting problems and handling potentially non-monotonic criteria. Section 3 illustrates the use of the proposed method on a didactic instance. In Section 4, we report the results of a case study concerning the analysis of exposure scenarios related to the treatment of nanomaterials in various conditions. The last section concludes and outlines the ideas for a future work.

## 2. Notation and problem statement

We use the following notation:

- $A = \{a_1, a_2, \dots, a_i, \dots, a_n\}$  – a finite set of  $n$  alternatives;
- $A^R = \{a^*, b^*, \dots\}$  – a finite set of reference alternatives, which are holistically judged by the DM; we assume that  $A^R \subseteq A$ ;
- $G = \{g_1, g_2, \dots, g_j, \dots, g_m\}$  – a finite set of  $m$  criteria,  $g_j : A \rightarrow \mathbb{R}$  for all  $j \in J = \{1, 2, \dots, m\}$ ;
- $X_j = \{x_j \in \mathbb{R} : g_j(a_i) = x_j, a_i \in A\}$  – a set of all different performances on  $g_j, j \in J$ ;
- $x_j^1, x_j^2, \dots, x_j^{n_j(A)}$  – increasingly ordered values of  $X_j, x_j^k < x_j^{k+1}, k = 1, 2, \dots, n_j(A) - 1$ , where  $n_j(A) = |X_j|$  and  $n_j(A) \leq n$ ; consequently,  $X = \prod_{j=1}^m X_j$  is the performance space;
- $\mathcal{D} = \{D_1, D_2, \dots, D_l\}$  – a finite set of  $l$  decision attributes;
- $C_1^{D_s}, C_2^{D_s}, \dots, C_p^{D_s}$  –  $p$  pre-defined preference-ordered classes defined for each decision attribute  $D_s, s = 1, \dots, l$ , where  $C_{h+1}^{D_s}$  is preferred to  $C_h^{D_s}, h = 1, \dots, p - 1$ ; moreover,  $H = \{1, \dots, p\}$ . Remark that the number of pre-defined preference-ordered classes for all  $l$  decision attributes is the same and equal to  $p$ . Qualitative meaning of class  $C_h$  is also the same for all decision attributes  $D_s \in \mathcal{D}$ .
- $b_0^{D_s}, b_1^{D_s}, \dots, b_p^{D_s}$  – thresholds separating the classes on decision attribute  $D_s, s = 1, \dots, l$ , such that  $b_{h-1}^{D_s}$  and  $b_h^{D_s}$  are, respectively, the lower and upper comprehensive values admissible for alternatives assigned to  $C_h^{D_s}, h = 1, \dots, p$ .

In what follows, we discuss the employed preference model and preference information. We present the mathematical constraints that allow dealing with multi-decision sorting problems while originally handling potentially non-monotonic criteria. The latter ones are interpreted as criteria with unknown monotonicity. This means that a decision analyst and the DM cannot specify a preference direction for them, and they admit that such a direction may not exist. Moreover, they accept that the shapes of marginal value functions for these criteria will be inferred through disaggregating the DM's holistic preferences. The resulting shape will determine if the monotonic relation can be imposed in the entire performance space of a given criterion, and, if not, what are the local relationships of monotonicity in different regions of this space.

### 2.1. Preference model

For each decision attribute  $D_s \in \mathcal{D}$ , a comprehensive quality of each alternative  $a_i \in A$  is quantified using an additive value

function defined as the sum of marginal values  $u_j^{D_s}(a_i)$  on all criteria  $g_j, j = 1, \dots, m$ :

$$U^{D_s}(a_i) = \sum_{j=1}^m u_j^{D_s}(a_i) \in [0, 1]. \quad (1)$$

Alternatives are evaluated in terms of three types of criteria: gain  $g_j \in G_g$ , cost  $g_j \in G_c$ , and potentially non-monotonic  $g_j \in G_n$  ( $G_g \cup G_c \cup G_n = G$ ). For the gain-type criteria, greater performances are more preferred than smaller performances. This implies the following monotonicity and normalization constraints:

$$u_j^{D_s}(x_j^k) \geq u_j^{D_s}(x_j^{k-1}), \quad k = 2, \dots, n_j(A), \quad \text{and} \quad u_j^{D_s}(x_j^1) = 0. \quad (2)$$

Analogously, for the cost-type criteria, smaller performances are more preferred:

$$u_j^{D_s}(x_j^k) \leq u_j^{D_s}(x_j^{k-1}), \quad k = 2, \dots, n_j(A), \quad \text{and} \quad u_j^{D_s}(x_j^{n(A)}) = 0. \quad (3)$$

Example marginal value functions for the gain- and cost-type criteria are presented in Fig. 1. Note that these functions are, respectively, non-decreasing and non-increasing.

The marginal value function for the potentially non-monotonic criterion  $g_j \in G_n$  is modeled as the sum of marginal values derived from the non-decreasing and non-increasing components contributing to the comprehensive assessment of alternatives from this particular viewpoint:

$$u_j^{D_s}(x_j^k) = u_{j,nd}^{D_s}(x_j^k) + u_{j,ni}^{D_s}(x_j^k), \quad k = 1, \dots, n_j(A),$$

where the monotonicity of  $u_{j,nd}^{D_s}$  and  $u_{j,ni}^{D_s}$  is modeled in a standard way:

$$u_{j,nd}^{D_s}(x_j^k) \geq u_{j,nd}^{D_s}(x_j^{k-1}), \quad k = 2, \dots, n_j(A), \quad \text{and} \quad u_{j,nd}^{D_s}(x_j^1) = 0, \quad (4)$$

$$u_{j,ni}^{D_s}(x_j^k) \leq u_{j,ni}^{D_s}(x_j^{k-1}), \quad k = 2, \dots, n_j(A), \quad \text{and} \quad u_{j,ni}^{D_s}(x_j^{n(A)}) = 0. \quad (5)$$

In case the method discovers that a monotonic relationship exists, either  $u_{j,nd}^{D_s}(x_j^k)$  or  $u_{j,ni}^{D_s}(x_j^k)$  takes non-negative values for  $k = 1, \dots, n_j(A)$  and the other component is zeroed for all performances. Then, the resulting marginal value function  $u_j^{D_s}$  is also monotonic. In Figs. 2a and b, we present the examples of such non-decreasing and non-increasing functions along with the two components.

When both components  $u_{j,nd}^{D_s}$  and  $u_{j,ni}^{D_s}$  take some positive values over the range from  $x_j^1$  to  $x_j^{n_j(A)}$ , then any non-monotonic shape of the marginal value function  $u_j^{D_s}$  can be obtained. However, this may yield a comprehensive marginal value function, which is not equal to zero for the worst performance on the non-monotonic criterion. Such a situation is undesired because it is hard to interpret such a model, and, moreover, the scale of values attained by the comprehensive model gets reduced. To prevent such a scenario, the marginal value function should be normalized so that  $\exists x_j^k \in X_j$  such that  $u_j^{D_s}(x_j^k) = 0$ . This can be obtained by subtracting a value of bias  $t_j^{D_s} \geq 0$  from the marginal values of  $g_j$ , and adding a constraint that  $u_j^{D_s}(x_j^k)$  should be non-negative for all performances  $x_j^k, k = 1, \dots, n_j(A)$ :

$$u_j^{D_s}(x_j^k) = u_{j,nd}^{D_s}(x_j^k) + u_{j,ni}^{D_s}(x_j^k) - t_j^{D_s}, \quad k = 1, \dots, n_j(A), \quad (6)$$

$$1 \geq t_j^{D_s} \geq 0, \quad (7)$$

$$u_j^{D_s}(x_j^k) \geq 0, \quad k = 1, \dots, n_j(A). \quad (8)$$

In Figs. 2c–f, we present the two components, a value of bias, and the resulting non-monotonic functions of different types: A-, V-, W-, and M-type functions. The elementary components are monotonic, but the marginal functions which aggregate them

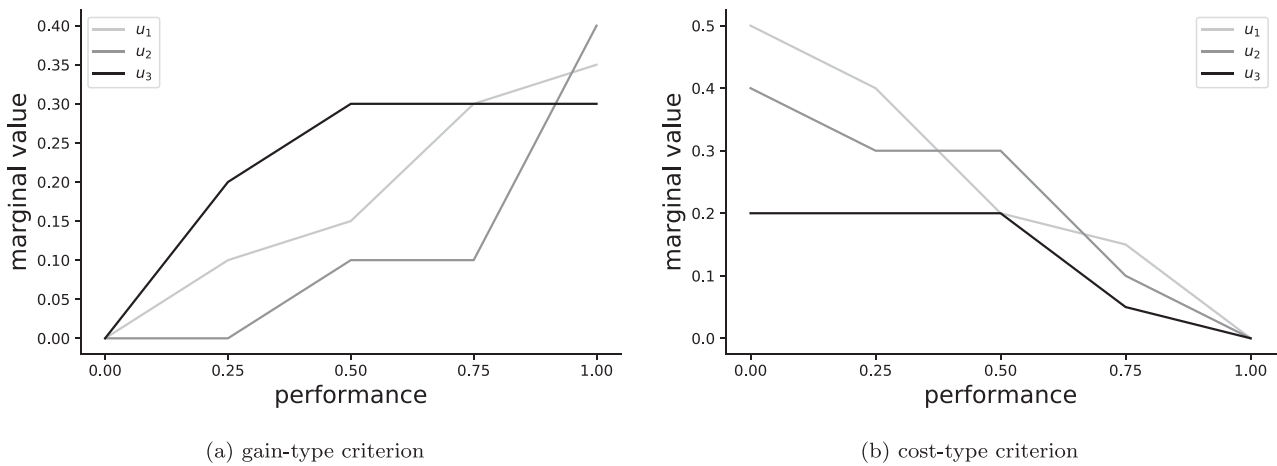


Fig. 1. Example monotonic marginal value functions.

with a bias are arbitrarily non-monotonic. Such functions can capture the local relationships of monotonicity that can be positive in some of the considered performance space and negative in the other part of the same space. Since in the proposed approach, these functions are inferred from the assignment examples, their complexity (i.e., the non-monotonic character, the number of monotonicity changes, or differences in slopes in various regions of the performance space) depends on the dependencies observed in the preferences of alternatives and input preference information.

To assign the alternatives to pre-defined and ordered classes for decision attribute  $D_s \in \mathcal{D}$ , we will apply a threshold-based sorting procedure. It derives the assignment of alternative  $a_i \in A$  from the comparison of  $U^{D_s}(a_i)$  with a set of thresholds  $b_h^{D_s}$ ,  $h = 0, \dots, p$ , such that for  $D_s \in \mathcal{D}$ :

$$b_0^{D_s} = 0, \quad b_{p-1}^{D_s} \leq 1 - \varepsilon, \quad \text{and} \quad b_p^{D_s} = 1 + \varepsilon, \quad (9)$$

$$b_h^{D_s} \geq b_{h-1}^{D_s} + \varepsilon, \quad h = 2, \dots, p - 1, \quad (10)$$

where  $\varepsilon$  is an arbitrarily small positive value. In this way, the values of the worst and the best thresholds are set to, respectively, zero and greater than one. Moreover, there is a difference between the extreme thresholds delimiting each class so that it could accommodate some alternatives. Then,  $a_i$  is assigned to class  $C_h^{D_s}$  iff  $b_{h-1}^{D_s} \leq U^{D_s}(a_i) < b_h^{D_s}$ , i.e., if  $a_i$  is at least as good as the respective lower threshold and strictly worse than the corresponding upper threshold. Such a threshold-based sorting procedure is illustrated in Fig. 3. Eqs. (1)–(10) form a core component of a larger set of linear programming constraints defining a set of instances of an assumed sorting model that are compatible with the DM's preference information. We will refer to it as  $E^{BASE}$ .

## 2.2. Preference information

We expect the DM to specify the desired assignments for a subset of reference alternatives  $a^* \in A^R \subseteq A$  on each decision attribute  $D_s \in \mathcal{D}$ :

$$a^* \in A^R \rightarrow C_{DM}^{D_s}(a^*). \quad (11)$$

Note that the classes provided by the DM for different decision attributes  $D_s \in \mathcal{D}$  can be different for the same reference alternative. Such holistic preference information is used in a two-fold way. On the one hand, we need to reproduce the desired

assignments on each decision attribute, i.e.:

$$\left. \begin{array}{l} \text{for all } a^* \in A^R : \\ v(a^*) \in \{0, 1\}, \\ \text{for all } D_s \in \mathcal{D} : \\ U^{D_s}(a^*) \geq b_{C_{DM}^{D_s}(a^*)-1}^{D_s} - v(a^*), \\ U^{D_s}(a^*) + \varepsilon \leq b_{C_{DM}^{D_s}(a^*)}^{D_s} + v(a^*). \end{array} \right\} E^R(\text{intra} - \mathcal{D}) \quad (12)$$

In case  $v(a^*) = 0$ ,  $U^{D_s}(a^*)$  falls in the range corresponding to class  $C_{DM}^{D_s}(a^*)$ , i.e.,  $[b_{C_{DM}^{D_s}(a^*)-1}^{D_s}, b_{C_{DM}^{D_s}(a^*)}^{D_s})$ . Then, the assignment provided

for  $a^*$  on  $D_s$  is reproduced. If  $v(a^*) = 1$ , the respective constraints are always satisfied, being relaxed. The binary variables  $v(a^*)$ ,  $a^* \in A^R$ , will be subsequently used to minimize the prediction distance of the inferred model from the reference data in case the sorting model would not be able to align with all assignment examples.

On the other hand, in line with the specificity of the multi-decision sorting problem, we will compare the desired assignments for each reference alternative  $a^* \in A^R$  for different pairs of decision attributes. Let us remind that both the number and interpretation of classes are the same for all decision attributes. In this way, the classes specified by the DM determine an order of labels associated with each reference alternative. If  $C_{DM}^{D_s}(a^*)$  is more preferred than  $C_{DM}^{D_t}(a^*)$ , this can be interpreted as the label  $D_s$  being more desired for  $a^*$  than label  $D_t$ . Consequently, a comprehensive value of  $a^*$  associated with  $D_s$  should be greater than its respective value associated with  $D_t$ , i.e.:

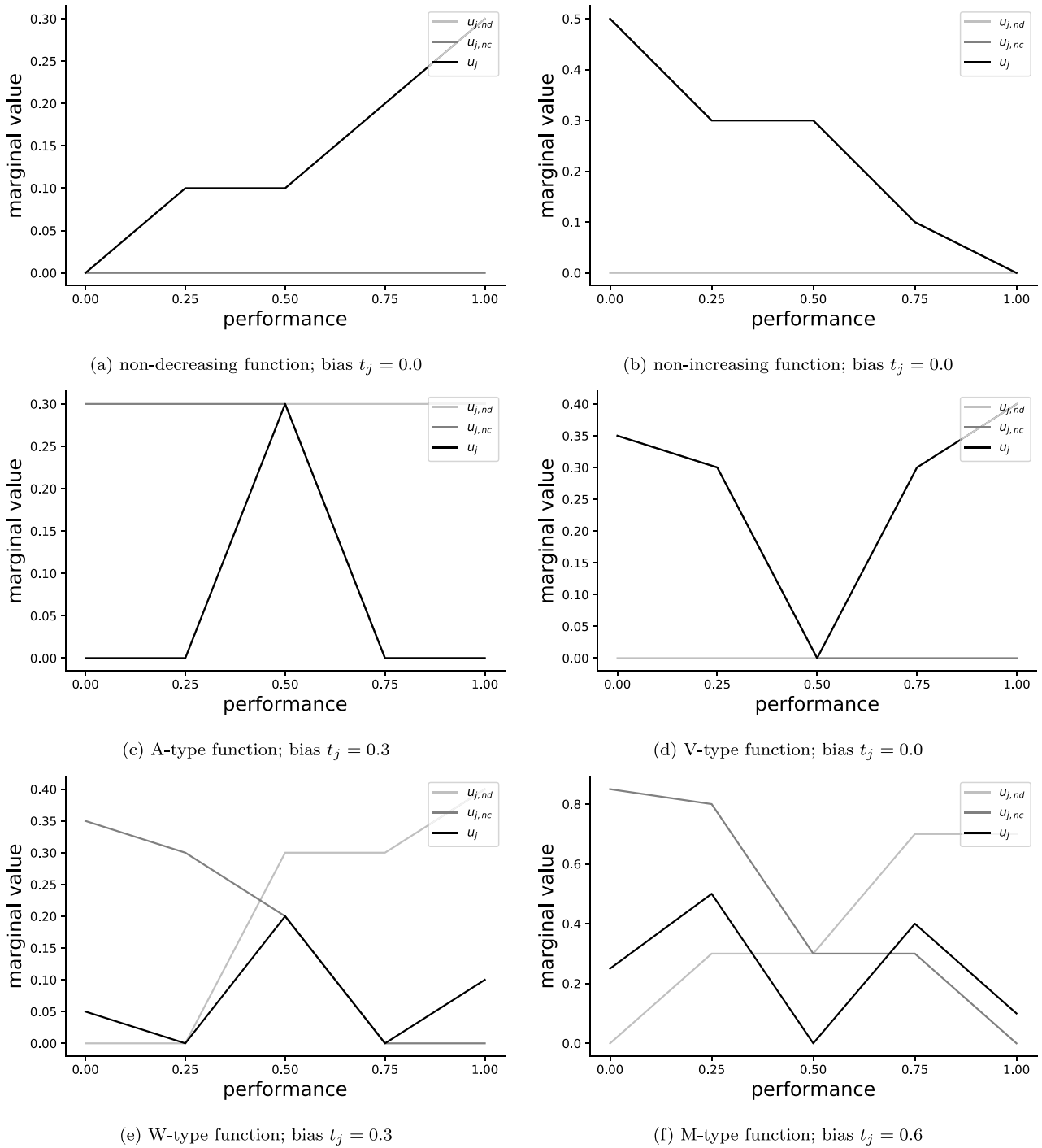
$$\left. \begin{array}{l} \text{for all } a^* \in A^R : \\ \text{if } C_{DM}^{D_s}(a^*) > C_{DM}^{D_t}(a^*) : \\ U^{D_s}(a^*) \geq U^{D_t}(a^*) + \varepsilon - v(a^*). \end{array} \right\} E^R(\text{inter} - \mathcal{D}) \quad (13)$$

Analogously as in  $E^R(\text{intra} - \mathcal{D}_s)$ , binary variable  $v(a^*)$  implies that the respective constraints associated with the assignments of  $a^*$  are either instantiated (when  $v(a^*) = 0$ ) or relaxed (when  $v(a^*) = 1$ ).

## 2.3. Compatible sorting model

We aim to infer a sorting model that would be compatible with the provided assignment examples while respecting the assumptions on additivity, monotonicity, and normalization, as well as intra- and inter-decision constraints. The model is composed of a set of interrelated additive value functions and vectors of class thresholds such that a single function is associated with a single





**Fig. 2.** Example non-decreasing ( $u_{j,nd}$ ) and non-increasing ( $u_{j,ni}$ ) components along with resulting marginal value functions ( $u_j$ ) of different types for the potentially non-monotonic criteria.

vector of thresholds corresponding to each decision attribute. We admit that the reference assignments are burdened with some error, though we would like the model to be compatible with as many assignments of reference alternatives as possible. For this purpose, we solve the following Mixed-Integer Linear Programming (MILP) model:

$$\begin{aligned}
 & \text{Minimize } f_w = (r \cdot l + 1) \sum_{a^* \in A^R} v(a^*) + \sum_{j \in G_n, D_s \in \mathcal{D}} t_j^{D_s}, \\
 & \text{subject to } E^{BASE} \cup E^R(\text{intra} - \mathcal{D}) \cup E^R(\text{inter} - \mathcal{D}). \tag{14}
 \end{aligned}$$

The primary goal of the above objective function  $f_w$  is to minimize the number of reference alternatives for which the desired assignments are inconsistent with an assumed preference model, i.e.,  $\sum_{a^* \in A^R} v(a^*)$ . The secondary goal is to minimize a sum of bias values for all decision attributes and all potentially non-monotonic criteria, i.e.,  $\sum_{j \in G_n, D_s \in \mathcal{D}} t_j^{D_s}$ . To ensure a lexicographic optimization of these two targets, we multiply the first component by  $(r \cdot l + 1)$  and the second by 1. Note that  $(r \cdot l + 1)$  is greater than a maximal possible sum of all bias values. Based on Eqs. (6)

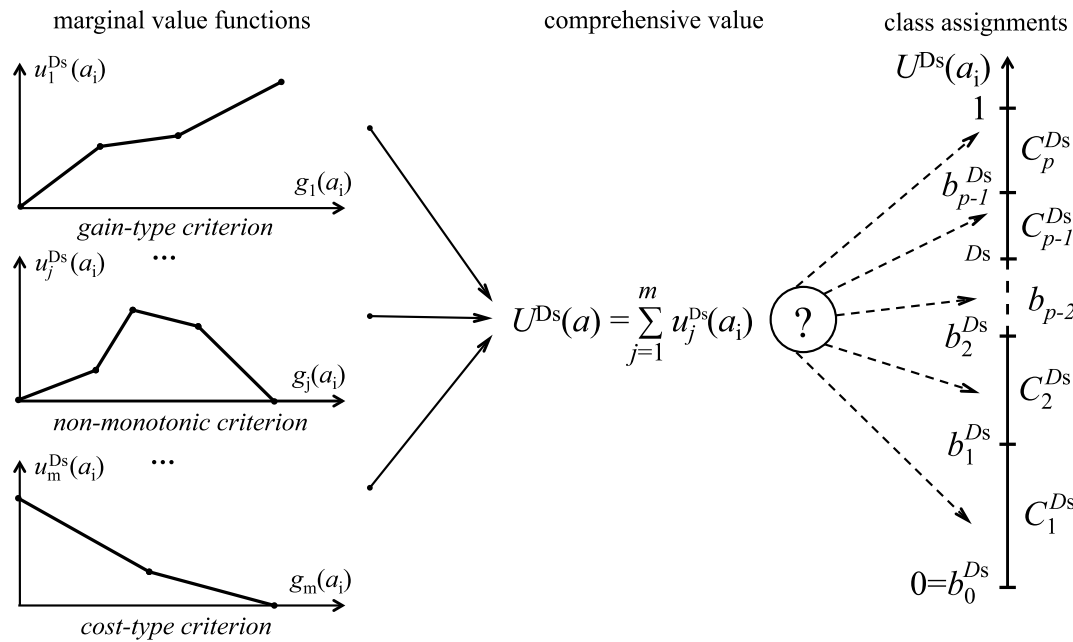


Fig. 3. Threshold-based sorting procedure driven by an additive value function aggregating marginal values corresponding to criteria of different types.

and (7), such a sum is constrained by  $r \cdot l$ , i.e.:

$$\sum_{j \in G_n, D_s \in \mathcal{D}} t_j^{D_s} \leq r \cdot l, \tag{15}$$

where  $r = |G_n|$  is the number of potentially non-monotonic criteria, and  $l$  is the number of decision attributes. As a result, the above model always favors a lesser number of reference alternatives  $a^* \in A^R$  that need to be removed to restore consistency. Specifically, among the models for which such a number is minimal, we favor the one for which the sum of biases on all criteria with unknown monotonicity is as small as possible.

### 3. Illustrative example

In this section, we illustrate the use of the proposed method on a simple didactic example composed of a pair of scenarios, denoted as Scenarios 1 and 2. The considered problem involves ten alternatives, which are evaluated on the following three criteria:  $g_1$  of gain type,  $g_2$  of cost type, and  $g_3$  being potentially non-monotonic. For the performance matrix, see Table 1. The alternatives are comprehensively evaluated using two decision attributes ( $D_1$  and  $D_2$ ) with five preference ordered classes  $C_1 - C_5$  such that  $C_5$  and  $C_1$  are, respectively, the most and the least preferred ones.

Let us first consider Scenario 1 for which the reference assignments are provided in Table 1. For example,  $a_1$  is assigned to  $C_4$  on  $D_1$  and to  $C_3$  on  $D_2$ , whereas the order of classes for  $a_5$  is inverse. The inferred model is able to reconstruct the assignments for eight alternatives (see column  $v(a^*)$  (Scenario 1) of Table 2). The comprehensive judgments for  $a_6$  and  $a_{10}$  could not be reproduced. The assignments of  $a_6$  were contradictory with those of  $a_7$ . Specifically,  $a_7$  is more preferred than  $a_6$  on  $g_1$  and  $g_2$ , while attaining the same performance on  $g_3$ . However, the classes of  $a_7$  are worse than the respective classes of  $a_6$ , which contradicts the dominance principle. Similarly,  $a_{10}$  dominates  $a_1$  while being at least as good on all monotonic criteria and having the same performance on the non-monotonic criterion. However, the desired class of  $a_{10}$  on  $D_1$  is worse than that of  $a_1$ , while they

Table 1 Performance matrix and the reference assignments considered in the two scenarios in the illustrative example.

Alternative	Criteria			Scenario 1		Scenario 2	
	$g_1$	$g_2$	$g_3$	$D_1$	$D_2$	$D_1$	$D_2$
$a_1$	2	1	2	$C_4$	$C_3$	$C_4$	$C_3$
$a_2$	0	2	0	$C_2$	$C_1$	$C_2$	$C_1$
$a_3$	1	3	4	$C_5$	$C_2$	$C_5$	$C_2$
$a_4$	3	2	3	$C_5$	$C_4$	$C_5$	$C_4$
$a_5$	3	3	3	$C_3$	$C_4$	$C_3$	$C_4$
$a_6$	0	4	1	$C_2$	$C_3$	$C_1$	$C_3$
$a_7$	1	3	1	$C_1$	$C_2$	$C_1$	$C_3$
$a_8$	3	0	0	$C_4$	$C_5$	$C_4$	$C_5$
$a_9$	0	3	4	$C_3$	$C_1$	$C_3$	$C_1$
$a_{10}$	2	0	2	$C_1$	$C_3$	$C_1$	$C_3$

are both assigned to the same class on  $D_2$ . Also in this case, the assignments of  $a_{10}$  and  $a_1$  could not be reproduced jointly.

The comprehensive values of all ten alternatives and the assignments generated with the inferred sorting model are presented in Table 2. For the separating class thresholds, see Table 3. Let us first discuss the assignments of reference alternatives which agree with the one specified by the DM. For example on  $D_1$ ,  $a_1$  is assigned to  $C_4$  and  $a_3$  is assigned to  $C_5$ . Not only  $a_3$  attains a greater comprehensive value than  $a_1$ , but also the comprehensive values of these alternatives fall in the ranges delimited by the respective class thresholds (compare Tables 2 and 3). Similarly, on  $D_2$ ,  $a_1$  is assigned to a more preferred class than  $a_3$ , which is reflected in the relationship between their comprehensive values ( $U^{D_2}(a_1) = 0.4762 > U^{D_2}(a_3) = 0.2886$ ). However, the inferred comprehensive values respect also the desired inter-decision relationships. For example,  $a_3$  was assigned to  $C_5$  on  $D_1$  and to  $C_2$  on  $D_2$ . As a result, its comprehensive value on  $D_1$  ( $U^{D_1}(a_3) = 0.6162$ ) is greater than that on  $D_2$  ( $U^{D_2}(a_3) = 0.2886$ ). In the same spirit, since  $a_8$  was assigned to  $C_4$  and  $C_5$  on, respectively,  $D_1$  and  $D_2$ , we have  $U^{D_1}(a_8) = 0.6078 < U^{D_2}(a_8) = 0.6162$ . When it comes to the reference alternatives for which the desired assignments were not fully reproduced, the resulting class of  $a_6$  on  $D_2$  and  $a_{10}$

**Table 2**

Comprehensive values  $U(a^*)$ , respective class assignments on decision attributes  $D_1$  and  $D_2$ , and values of binary variables  $v(a^*)$  for the two scenarios in the illustrative example.

Alternative	Scenario 1					Scenario 2				
	$U^{D_1}(a^*)$	$U^{D_2}(a^*)$	$D_1$	$D_2$	$v(a^*)$	$U^{D_1}(a^*)$	$U^{D_2}(a^*)$	$D_1$	$D_2$	$v(a^*)$
$a_1$	0.4846	0.4762	$C_4$	$C_3$	0	0.1223	0.0057	$C_1$	$C_2$	1
$a_2$	0.2800	0.1402	$C_2$	$C_1$	0	0.1279	0.0001	$C_2$	$C_1$	0
$a_3$	0.6162	0.2886	$C_5$	$C_2$	0	0.5556	0.0056	$C_5$	$C_2$	0
$a_4$	0.6722	0.5408	$C_5$	$C_4$	0	0.5556	0.5001	$C_5$	$C_4$	0
$a_5$	0.4762	0.4846	$C_3$	$C_4$	0	0.4944	0.5000	$C_3$	$C_4$	0
$a_6$	0.0002	0.0002	$C_1$	$C_1$	1	0.0613	0.4444	$C_1$	$C_3$	0
$a_7$	0.1402	0.1486	$C_1$	$C_2$	0	0.1224	0.4500	$C_1$	$C_3$	0
$a_8$	0.6078	0.6162	$C_4$	$C_5$	0	0.5500	0.5556	$C_4$	$C_5$	0
$a_9$	0.4762	0.1402	$C_3$	$C_1$	0	0.4944	0.0001	$C_3$	$C_1$	0
$a_{10}$	0.6722	0.4763	$C_5$	$C_3$	1	0.1224	0.4444	$C_1$	$C_3$	0

**Table 3**

The class thresholds for the two scenarios in the illustrative example.

Scenario	Decision attribute	$b_1$	$b_2$	$b_3$	$b_4$
Scenario 1	$D_1$	0.1445	0.3585	0.4800	0.6100
	$D_2$	0.1445	0.3585	0.4800	0.6100
Scenario 2	$D_1$	0.1251	0.1807	0.4972	0.5527
	$D_2$	0.0028	0.4416	0.4972	0.5527

on  $D_1$  was  $C_1$  as compared to, respectively,  $C_2$  and  $C_5$  in the DM's judgments.

The marginal value functions inferred for Scenario 1 are presented in Fig. 4. The imposed monotonicity constraints are respected for the gain ( $g_1$ ) and cost ( $g_2$ ) criteria and the monotonic components of  $g_3$ . The shapes of marginal value functions on the different decisions are similar. The differences concern slightly greater marginal value assigned to performance 3 on  $g_1$  for  $D_2$  and to performances 0–2 on  $g_2$  and 3–4 on  $g_3$  for  $D_1$ . The non-increasing component for  $g_3$  was the same for both decision attributes. The marginal value function's overall course for  $g_3$  took the V-shape with 1 being the least preferred performance.

As the other scenario (Scenario 2), let us consider slightly modified desired assignments (see Table 1). When compared to Scenario 1, the assignments of  $a_7$  on  $D_2$  and  $a_6$  on  $D_1$  were changed to, respectively,  $C_3$  and  $C_1$ . Now, only assignments of alternative  $a_1$  could not be reproduced with an assumed model (see Table 2, column  $v(a^*)$ ) for Scenario 2). The comprehensive values and the respective assignments are presented in Table 2 and the class thresholds are given in Table 3. Similarly as for Scenario 1, we could observe that the classifications on the two decision attributes and the relationships between comprehensive values attained by each alternative on  $D_1$  and  $D_2$  are preserved. For example,  $a_2$  is assigned to  $C_2$  on  $D_1$  and to  $C_1$  on  $D_2$ , because  $b_2^{D_1} = 0.1807 > U^{D_1}(a_2) = 0.1279 \geq b_1^{D_1} = 0.1251$ ,  $b_1^{D_2} = 0.0028 > U^{D_2}(a_2) = 0.0001$ , and  $U^{D_1}(a_2) > U^{D_2}(a_2)$ . However, the primary motivation for considering Scenario 2 is to show the impact of eliminating a bias for a non-monotonic criterion (for the marginal value functions, see Fig. 5). Indeed, when summing up the non-decreasing and non-increasing components for  $g_3$  for decision  $D_2$ , all performances would be assigned positive marginal values. To ensure that the worst performance on  $g_3$  (for this scenario –  $g_3^{D_2}(a) = 2$ ) was assigned zero, the constructed model subtracted a bias  $t_3^{D_2} = 0.4444$ . In this way, a comprehensive value of the anti-ideal alternative was also zeroed, while not affecting the relative comparison of existing alternatives.

To support the comprehension of different types of constraints defining a set of inter-related sorting models, in Table 4, we illustrate the use of these constraints in the context of an example presented in this section. Specifically, for nine different constraint types, we provide their general form, an example constraint for a

specific decision attribute, alternative, criterion, performance, or class, and the values assigned in this example constraint to the variables by the sorting models inferred for Scenario 2.

#### 4. Multi-decision sorting in the context of exposure management of nanomaterials

Nanomaterials are particles with a size of several dozen nanometers and physicochemical properties being significantly different from the materials of larger sizes composed of the same atoms [35]. Due to these specific properties, they can improve the performance of products in several application areas, including energy production and storage [41], water treatment [42], healthcare [43], and food preservation [44], to name a few. The production of nanomaterials is based on the manipulation of materials at the nanoscale (1–100 nanometers), which requires caution and protective measures to guarantee their safe handling.

Since nanotechnology is a relatively new scientific field, all the potentially harmful effects of individual nanomaterials and threats resulting from their production and employment are not yet precisely known [45,46]. The research on this subject is still ongoing, but the safety standards used in nanomaterials production are currently mainly adopted from similar chemical production processes [37,47,48]. Nevertheless, the safety of nanomaterials production processes is a pressing issue in the area of nanotechnology [48,49]. In this perspective, the development of guidelines for the appropriate selection of precautions for nanomanufacturing would be a beneficial contribution.

##### 4.1. Problem definition

###### 4.1.1. Criteria

When evaluating nanomanufacturing exposure scenarios, there are several characteristics of the nanomaterials and operating conditions that need to be accounted for. In this case study, we will consider the following ten criteria, which are common descriptors for this type of scenarios [31,35,48]:

- *Particle size* ( $g_1$ ) – in general, the smaller the size, the easier the nanomaterial gets through any filter. Nevertheless, since nanomaterials have different toxicological profiles according to their size, it is not yet possible to generalize a monotonic dependency between size and harmfulness [50].
- *Toxicity* ( $g_2$ ) determines type of effect the nanomaterial has on human health [51].
- *Airborne capacity* ( $g_3$ ) characterizes the engineered nanomaterials' capacity to spread in the workplace through the air stream. It is scored on 4-point scale from none to high, with none and high being, respectively, the most and the least preferred performances [52].

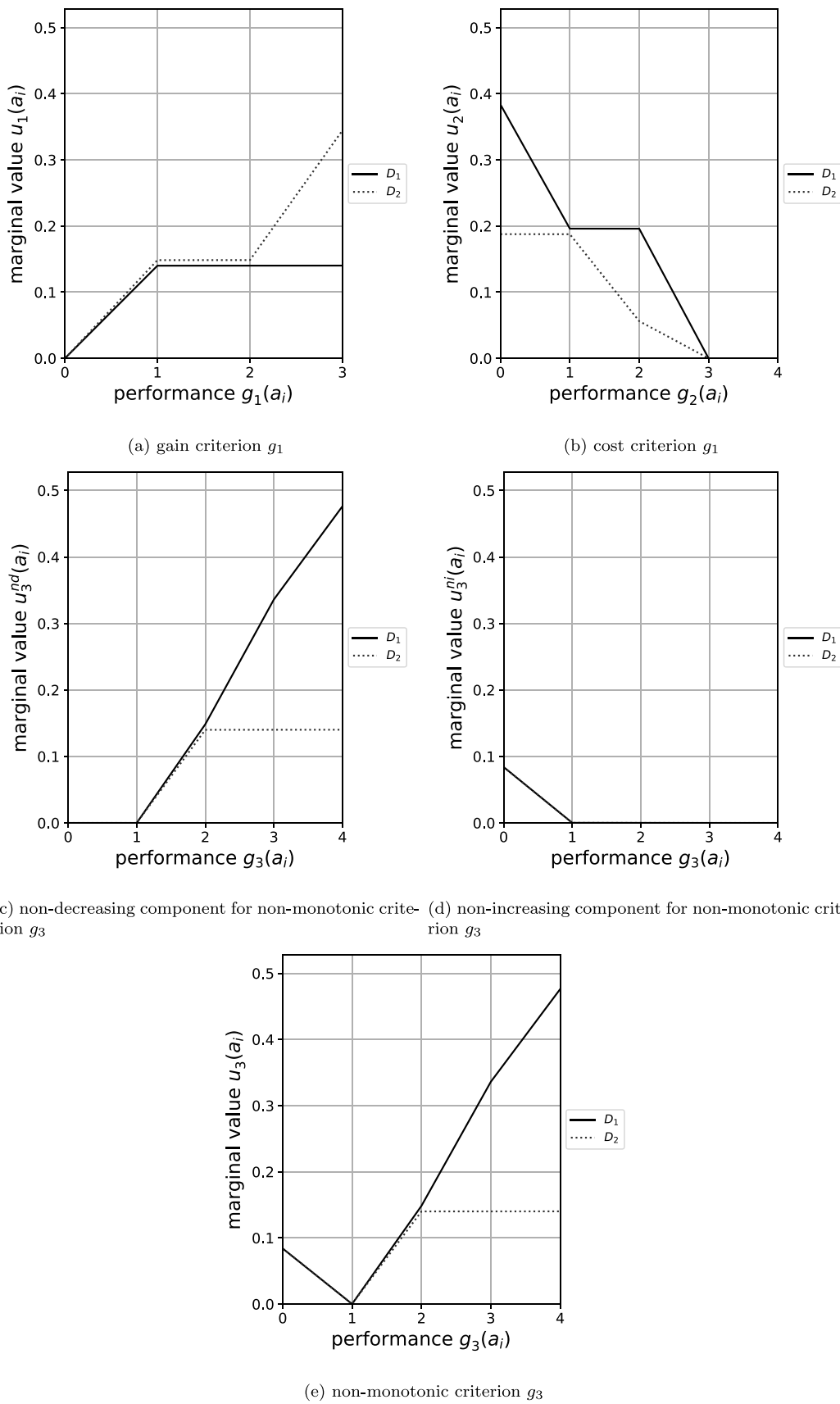


Fig. 4. Marginal value functions for all criteria and decision  $D_1$  and  $D_2$  for Scenario 1 of the illustrative example.

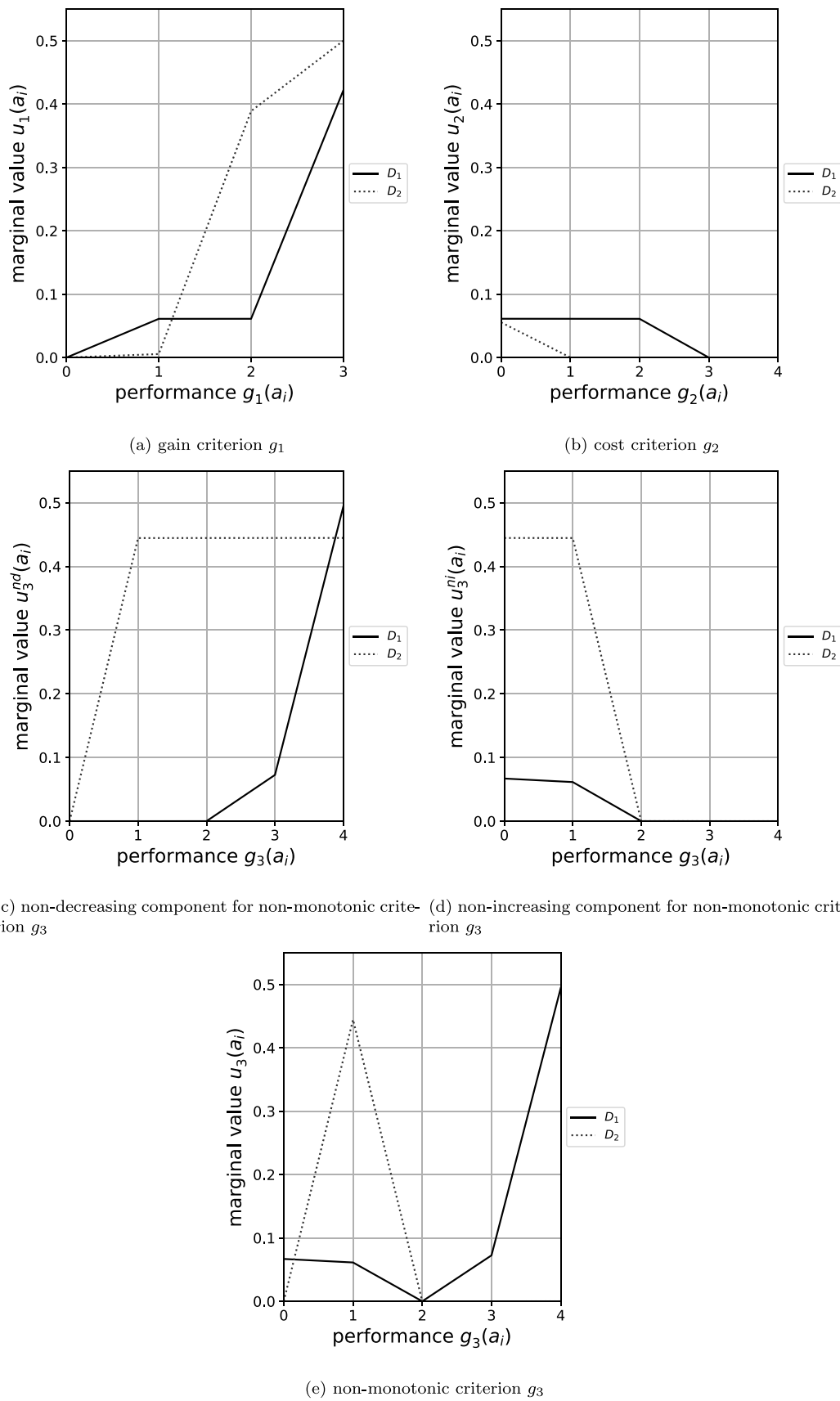


Fig. 5. Marginal value functions for all criteria and decision  $D_1$  and  $D_2$  for Scenario 2 of the illustrative example.

**Table 4**  
Example constraints defining two inter-related sorting models for Scenario 2.

Type	Constraint
Comprehensive value of an alternative defined using an additive value function	
General constraint form	$U^{D_s}(a_i) = \sum_{j=1}^m u_j^{D_s}(a_i)$
Example constraint for $D_2$ and $a_7$	$U^{D_2}(a_7) = u_1^{D_2}(a_7) + u_2^{D_2}(a_7) + u_3^{D_2}(a_7)$
Example values assigned to the model variables	$0.4500 = 0.0056 + 0.0 + 0.4444$
Constraints on a comprehensive value of an alternative	
General constraint form	$1 \geq U^{D_s}(a_i) \geq 0$
Example constraint for $D_2$ and $a_7$	$1 \geq U^{D_2}(a_7) \geq 0$
Example values assigned to the model variables	$1 \geq 0.4500 \geq 0$
The least preferred marginal value for a monotonic criterion of a gain-type	
General constraint form	$u_j^{D_s}(x_j^1) = 0$
Example constraint for $D_1$ , $g_1$ , and $x_1^1 = 0$	$u_1^{D_1}(0) = 0$
Example values assigned to the model variables	$0 = 0$
Monotonicity constraint for a gain-type criterion	
General constraint form	$u_j^{D_s}(x_j^k) \geq u_j^{D_s}(x_j^{k-1})$
Example constraint for $D_2$ , $g_1$ , $x_1^3 = 2$ , and $x_1^2 = 1$	$u_1^{D_2}(2) \geq u_1^{D_2}(1)$
Example values assigned to the model variables	$0.3888 \geq 0.0056$
Monotonicity constraint for a cost-type criterion	
General constraint form	$u_j^{D_s}(x_j^k) \leq u_j^{D_s}(x_j^{k-1})$
Example constraint for $D_2$ , $g_2$ , $x_2^2 = 1$ , and $x_2^1 = 0$	$u_2^{D_2}(1) \leq u_2^{D_2}(0)$
Example values assigned to the model variables	$0.0001 \leq 0.0556$
Marginal value for a potentially non-monotonic criterion	
General constraint form	$u_j^{D_s}(x_j^k) = u_{j,nd}^{D_s}(x_j^k) + u_{j,ni}^{D_s}(x_j^k) - t_j^{D_s}$
Example constraint for $D_2$ , $g_3$ , and $x_3^2 = 1$	$u_3^{D_2}(1) = u_{3,nd}^{D_2}(1) + u_{3,ni}^{D_2}(1) - t_3^{D_2}$
Example values assigned to the model variables	$0.4444 = 0.4445 + 0.4445 - 0.4445$
Constraints between the neighboring thresholds separating decision classes	
General constraint form	$b_i^{D_s} \geq b_{i-1}^{D_s} + \varepsilon$
Example constraint for $D_1$ and $h = 3$	$b_3^{D_1} \geq b_2^{D_1} + \varepsilon$
Example values assigned to the model variables	$0.4972 \geq 0.1807 + 0.0001$
Intra-decision constraints imposed by an assignment example	
General constraint form	$U^{D_s}(a^*) \geq b_{C_{DM}^{D_s}(a^*)-1}^{D_s} - v(a^*)$
Example constraint for $a_5 \rightarrow C_3^{D_1}$	$U^{D_1}(a_5) \geq b_2^{D_1} - v(a_5)$
Example values assigned to the model variables	$0.4944 \geq 0.4416 - 0$
General constraint form	$U^{D_s}(a^*) + \varepsilon \leq b_{C_{DM}^{D_s}(a^*)}^{D_s} + v(a^*)$
Example constraint for $a_5 \rightarrow C_3^{D_1}$	$U^{D_1}(a_5) + \varepsilon \leq b_3^{D_1} + v(a_5)$
Example values assigned to the model variables	$0.4944 + 0.0001 \leq 0.4972 + 0$
Inter-decision constraint imposed by assignment examples	
General constraint form	$U^{D_s}(a^*) \geq U^{D_t}(a^*) + \varepsilon - v(a^*)$
Example constraint for $a_7 \rightarrow C_3^{D_2}$ and $a_7 \rightarrow C_1^{D_1}$	$U^{D_2}(a_7) \geq U^{D_1}(a_7) + \varepsilon - v(a_7)$
Example values assigned to the model variables	$0.4500 \geq 0.1224 + 0.0001 - 0$

- *Detection limit* ( $g_4$ ) specifies the capacity of the instruments used for exposure assessment to detect the nanomaterials. The better the detection limit is, the safer the exposure scenario is assumed to be [35].
- *Exposure limit* ( $g_5$ ) indicates the limit of exposure, expressed on five ranges, for a given exposure scenario. The lower this limit, the less risk concerning a given exposure is [35].
- *Quantity* ( $g_6$ ) of a nanomaterial (in kg) handled in a given scenario. Smaller quantities are preferred as they imply a smaller chance of exposure [53].
- *Engineering controls* ( $g_7$ ) is a potentially non-monotonic attribute, specifying a setting in which the nanomanufacturing tasks are performed. It refers to the combinations of open (O) or closed (C) system and positive (PP) or negative (NP) pressure.
- *Number of employees* ( $g_8$ ) indicates the number of people required to handle a given exposure scenario. One cannot define a priori how this attribute is associated with the risk of an exposure scenario [35].
- *Duration of exposure* ( $g_9$ ) is negatively associated with the risk, i.e., the shorter duration is deemed to be less risky [53].

- *Multiple exposures* ( $g_{10}$ ) is related to the frequency of exposure (a scenario is safer in case the number of exposures is lesser) [54].

The measurement units, preference directions, performance scales, and encoding of performances for all criteria are provided in Table 5. In general, there are six criteria of cost type, a single gain criterion, and three criteria for which the preference direction is unknown.

#### 4.1.2. Alternatives

The considered set of alternatives is composed of exposure scenarios for nanomanufacturing generated by the JMP software [35]. They correspond to the existing and future types of nanomaterials and manufacturing processes. To demonstrate the proposed method's applicability, we consider a set of 45 exposure scenarios, denoted by  $a_1 - a_{45}$ . For their performances, see Tables 6 and 7.

#### 4.1.3. Multi-decision sorting

The alternatives are holistically evaluated in terms of four decision attributes that could be considered individually. However,

**Table 5**  
A set of criteria considered in the risk management of exposure scenarios for nanomanufacturing.

$g_j$	Criterion	Preference	Performance	Code
$g_1$	Particle size (nm)	none	< 2	1
			2–10	2
			10–100	3
			100–500	4
			500–1000	5
			> 1000	6
$g_2$	Toxicity	cost	Low	1
			Moderate	2
			High	3
$g_3$	Airborne capacity	cost	None	0
			Low	1
			Moderate	2
			High	3
$g_4$	Detection limit	gain	None	0
			Poor	1
			Moderate	2
			Good	3
$g_5$	Exposure limit (fiber/cc)	cost	< 0.1	1
			0.1–0.2	2
			0.2–0.5	3
			0.5–1.0	4
			> 1.0	5
$g_6$	Quantity (kg)	cost	< 1	1
			1–100	2
			100–1000	3
			1000–10000	4
			> 10000	5
$g_7$	Engineering controls	none	O-PP	1
			O-NP	2
			C-PP	3
			O-NP	4
$g_8$	Number of employees	none	1–3	1
			3–10	2
			11–50	3
			51–100	4
			101–500	5
$g_9$	Duration of exposure (h)	cost	incidental	1
			< 0.25	2
			< 1	3
			1–5	4
			5–8	5
$g_{10}$	Multiple exposure (number)	cost	none	0
			1–3	1
			> 3	2
			unknown	3

the multiplicity of these attributes, the reference to the same aspect of production, i.e., safety, and the resulting inter-relations between the examined decisions form the basis for multi-decision sorting. Specifically, the decision attributes correspond to four types of precautions that can be used to reduce the risk. They concern three main aspects of safety: *respirator* ( $D_1$ ) represents a personal protective equipment, *fume hood* ( $D_2$ ) and *fume hood with HEPA filter* ( $D_3$ ) stand for the engineering controls, and *HEPA vacuum cleaner* ( $D_4$ ) corresponds to the work practices. Let us note that a respirator is a form of a mask with a filter that protects against dangerous substances in the air. A fume hood is a type of ventilation system protecting against harmful gases and toxins. Finally, High Efficiency Particulate Air (HEPA) filter is a filter that has a very high capacity of retaining particles in the range of several micrometers and above, as well as below.

For each precaution type, we make decisions about its requirement during the nanomanufacturing process. The holistic

preference on each decision attribute includes five preference-ordered classes corresponding to the levels of need for the specific precaution: required ( $C_1$ ), might be required ( $C_2$ ), optional ( $C_3$ ), might be optional ( $C_4$ ), and not required ( $C_5$ ). The reasoning on the decision attributes is the following: if the exposure scenario is deemed as risky, then a given precaution will be indicated as required. If it is not, then the expert would indicate no need for the precaution. A non-risky scenario is preferred to the risky one.

#### 4.2. Preference information

For forty exposure scenarios ( $a_1 - a_{40}$ ), we consider input provided by the health and safety managers in the form of class assignments on four decision attributes [35]. The experts were asked in a survey what precautions should be taken and in what intensity they should be used given a set of production parameters and features of the nanomaterials based on those presented in Table 5. For the scenarios deemed as risky and dangerous by the specialists, a given precaution is required. In the case of safer scenarios, the requirements are lower, and the necessity of some precaution types is not required. For the answers of the experts, see Table 9. The numbers of reference alternatives assigned to each class for the four decisions are presented in Table 8. The most common decisions are “required” ( $C_1$ ), “optional” ( $C_3$ ) and “not required” ( $C_5$ ), whereas the least chosen classes in the survey classes are “might be required” ( $C_2$ ) and “might be optional” ( $C_4$ ).

Let us discuss the performances and assignments for the three example reference alternatives ( $a_{26}$ ,  $a_5$ , and  $a_1$ ). Alternative  $a_{26}$  attains very favorable performances on four criteria of cost type  $g_2$ ,  $g_6$ ,  $g_9$  and  $g_{10}$  and the best performance on gain criterion  $g_4$ . As a result, the assigned classes for *respirator*, *fume hood with HEPA filter* and *HEPA vacuum cleaner* are “not required” and for *fume hood* – “required”. Consequently, the most risky evaluation in terms of *fume hood* is linked to the performances on  $g_3$ ,  $g_5$ , and the potentially non-monotonic criteria. Furthermore,  $a_5$  performs poorly on  $g_2$ ,  $g_3$ ,  $g_4$ ,  $g_5$ ,  $g_9$  and  $g_{10}$ , which was an important reason to classify this scenario to  $C_1$  (“precaution is required”) for all precaution types. Finally,  $a_1$  attains favorable performances on criteria  $g_2$ ,  $g_3$ ,  $g_4$  and  $g_{10}$ , while being less advantageous on criteria  $g_5$ ,  $g_6$  and  $g_9$ . Therefore, its classification for all decisions is between “might be required” ( $C_2$ ) and “optional” ( $C_3$ ).

#### 4.3. Research questions

The research goal consists of understanding under which operational conditions and according to which characteristics of the nanomaterials, different types of precautions can be required, might be required, are optional, might be optional, or not be required. This contribution results in a sorting model capable of providing decision recommendations on multiple risk management measures, corresponding to various precautions, for the same exposure scenario.

We wish to find a set of additive value functions and class thresholds that will describe the allocation to a particular class for each decision attribute based on the scenario described in terms of a set of ten criteria/attributes. Such a preference model is expected to capture the patterns from experts’ choices. Then, we will demonstrate that these discovered regularities can be used to support decision making. Thus, the inferred preference model involving intra- and inter-decision relations will be used to classify a set of non-reference exposure scenarios.



**Table 6**

Performance matrix of the reference exposure scenarios  $a_1 - a_{40}$  (↑ and ↓ indicate gain and cost criteria, respectively; – denotes criteria without a pre-defined preference direction).

	$g_1$ –	$g_2$ ↓	$g_3$ ↓	$g_4$ ↑	$g_5$ ↓	$g_6$ ↓	$g_7$ –	$g_8$ –	$g_9$ ↓	$g_{10}$ ↓		$g_1$ –	$g_2$ ↓	$g_3$ ↓	$g_4$ ↑	$g_5$ ↓	$g_6$ ↓	$g_7$ –	$g_8$ –	$g_9$ ↓	$g_{10}$ ↓
$a_1$	4	1	0	3	3	4	3	5	4	1	$a_{21}$	6	1	2	2	2	3	1	4	4	1
$a_2$	5	2	0	0	2	4	1	2	3	1	$a_{22}$	6	1	3	0	5	3	3	1	2	1
$a_3$	5	2	3	2	4	1	2	5	5	0	$a_{23}$	4	1	0	3	4	2	1	1	3	2
$a_4$	4	1	3	2	3	4	2	4	3	2	$a_{24}$	6	3	2	1	5	5	2	3	5	0
$a_5$	1	3	3	0	3	1	2	2	4	3	$a_{25}$	2	1	2	0	1	4	4	4	2	0
$a_6$	3	1	1	0	4	5	1	4	1	3	$a_{26}$	1	1	2	3	2	1	3	3	1	0
$a_7$	3	1	1	1	2	3	2	3	3	3	$a_{27}$	1	1	2	1	1	2	2	2	3	1
$a_8$	4	3	3	1	1	1	4	3	3	0	$a_{28}$	6	2	2	2	5	4	1	1	1	2
$a_9$	4	2	0	1	5	5	1	1	5	1	$a_{29}$	5	1	1	3	4	3	1	1	1	1
$a_{10}$	2	1	3	2	1	5	1	3	4	3	$a_{30}$	2	3	1	2	5	2	2	1	1	1
$a_{11}$	2	2	3	0	1	3	2	3	5	2	$a_{31}$	6	2	2	2	3	3	4	2	2	3
$a_{12}$	4	2	1	1	1	3	4	1	4	3	$a_{32}$	6	2	0	0	1	2	2	1	3	0
$a_{13}$	1	1	3	3	4	5	1	2	2	0	$a_{33}$	5	1	0	1	3	5	2	4	2	2
$a_{14}$	6	1	0	2	4	1	3	3	4	3	$a_{34}$	1	3	1	2	3	3	2	1	4	0
$a_{15}$	2	3	0	1	3	4	3	1	4	0	$a_{35}$	6	1	2	3	4	2	2	3	3	2
$a_{16}$	6	3	1	3	3	1	1	4	5	0	$a_{36}$	5	3	2	1	2	4	4	5	5	3
$a_{17}$	6	1	0	2	2	2	3	5	4	0	$a_{37}$	5	3	2	3	1	2	3	1	2	3
$a_{18}$	1	1	3	2	1	5	1	5	5	1	$a_{38}$	4	2	3	3	2	1	2	2	2	1
$a_{19}$	1	1	1	2	2	3	2	2	1	0	$a_{39}$	2	1	0	0	5	4	2	5	1	3
$a_{20}$	6	1	1	0	1	1	4	5	2	0	$a_{40}$	6	2	1	0	2	4	1	3	4	2

**Table 7**

Performance matrix of the non-reference exposure scenarios  $a_{41} - a_{45}$  (↑ and ↓ indicate gain and cost criteria, respectively; – denotes criteria without a pre-defined preference direction).

	$g_1$ –	$g_2$ ↓	$g_3$ ↓	$g_4$ ↑	$g_5$ ↓	$g_6$ ↓	$g_7$ –	$g_8$ –	$g_9$ ↓	$g_{10}$ ↓
$a_{41}$	5	2	3	2	3	3	3	3	1	2
$a_{42}$	2	1	0	2	2	2	2	2	2	3
$a_{43}$	5	1	0	1	0	4	1	1	2	3
$a_{44}$	0	2	2	0	3	3	0	3	2	1
$a_{45}$	1	1	2	2	3	4	2	0	4	2

4.4. Results

4.4.1. Marginal value functions

The marginal value functions for the ten criteria and four decision attributes are presented in Figs. 6 and 7. They preserve the imposed monotonicity constraints. In particular, the marginal value function  $u_2$  for the cost-type criterion *toxicity* is non-increasing, i.e., a value assigned to the “moderate” performance is always greater than to the “high” performance and equal or lesser (depending on the decision attribute) than the value corresponding to the “low” toxicity (see Fig. 6). Similarly, the marginal value function  $u_4$  for the gain-type criterion *detection limit* is non-decreasing. It assigns a strictly greater value to “poor” than to “null” performance, and exhibits a stable or a slightly increasing trend from “poor” through “moderate” to “good” detection limit (see Fig. 6). On the contrary, the marginal value functions for the criteria with unknown monotonicity exhibit a non-monotonic trend. For example, the least marginal value on  $u_1$  does not correspond to either of the extreme performances (see Fig. 6). However, the corresponding non-decreasing and non-increasing components adhere to the monotonicity constraints.

The impact of each criterion on the recommended decision can be estimated with the maximal share of each criterion in the comprehensive value (see Table 10). The greatest shares correspond to: for *respirator* – *airborne capacity* and *detection limit*, for *fume hood* – *particle size* and *airborne capacity*, for *fume hood with HEPA filter* – *airborne capacity*, and for *HEPA vacuum cleaner* – *exposure limit* and *airborne capacity*. The values of bias for all non-monotonic criteria are given in Table 11. They allowed normalizing the performance of an anti-ideal alternative to zero, as described in Section 2.

For the marginal value function  $u_1$  for *particle size*, the greatest value is assigned to the size greater than 1000 nm for all decisions but *fume hood with HEPA filter*, for which the greatest value is attained for the size of lesser than 2 nm. The function is of “W” shape for *respirator*, *fume hood with HEPA filter* and *HEPA vacuum cleaner* with a significant peak corresponding to sizes of 10 – 100 nm or 100 – 500 nm. Such a shape is implied by the largest decrease of value for the non-increasing component observed between sizes 10–100 nm, 100–500 nm, and 500–1000 nm, and the largest increase of value for the non-decreasing component observed for sizes between 2–10 nm, 10–100 nm, and 100–500 nm. For *fume hood*, the shape of  $u_1$  is similar to “V”, and the zero value is assigned to the intermediate size.

The value function  $u_2$  for *toxicity* indicates a negligible difference between the low and moderate performances. Such a difference is slightly greater only for *HEPA vacuum cleaner*. Intuitively, the precautions are less required with low toxicity. This criterion has a very low impact on the comprehensive value when considering *respirator* and *fume hood*. This means that this precaution type is needed even with low toxicity.

*Airborne capacity* has a very significant impact on the alternatives’ assignments. The “null” performance vastly contributes to reducing the requirement of a given precaution type. In addition, for *fume hood* the value differences between performances “null” and “low” or “moderate” and “high” are very marginal or non-existing. Thus, in this case, only the difference between “low” and “moderate” matters. For the remaining decision attributes, the difference between “moderate” and “high” and “low” and “moderate” are huge.

The shapes of marginal value functions for *detection limit* ( $u_4$ ) reveal high discrepancy between the decisions, even if they are similar in terms of a general trend. For *fume hood*, this criterion has almost no effect on the comprehensive value, whereas for *respirator* – the impact of  $g_4$  is significant. The main difference in terms of a trend is that for *fume hood with HEPA filter*, the difference between values assigned to “poor” and “moderate” or “good” detection limits is negligible, whereas for the *respirator* and *HEPA vacuum cleaner* it is around 0.025.

Analogously, the slight differences in the shapes of value functions for various decision attributes can be observed for the *exposure limit* ( $u_5$ ). For *respirator*, *fume hood with HEPA filter*, and *HEPA vacuum cleaner*, the greatest value difference is between the performances of < 0.1 and 0.1–0.2, whereas for *fume hood* – the



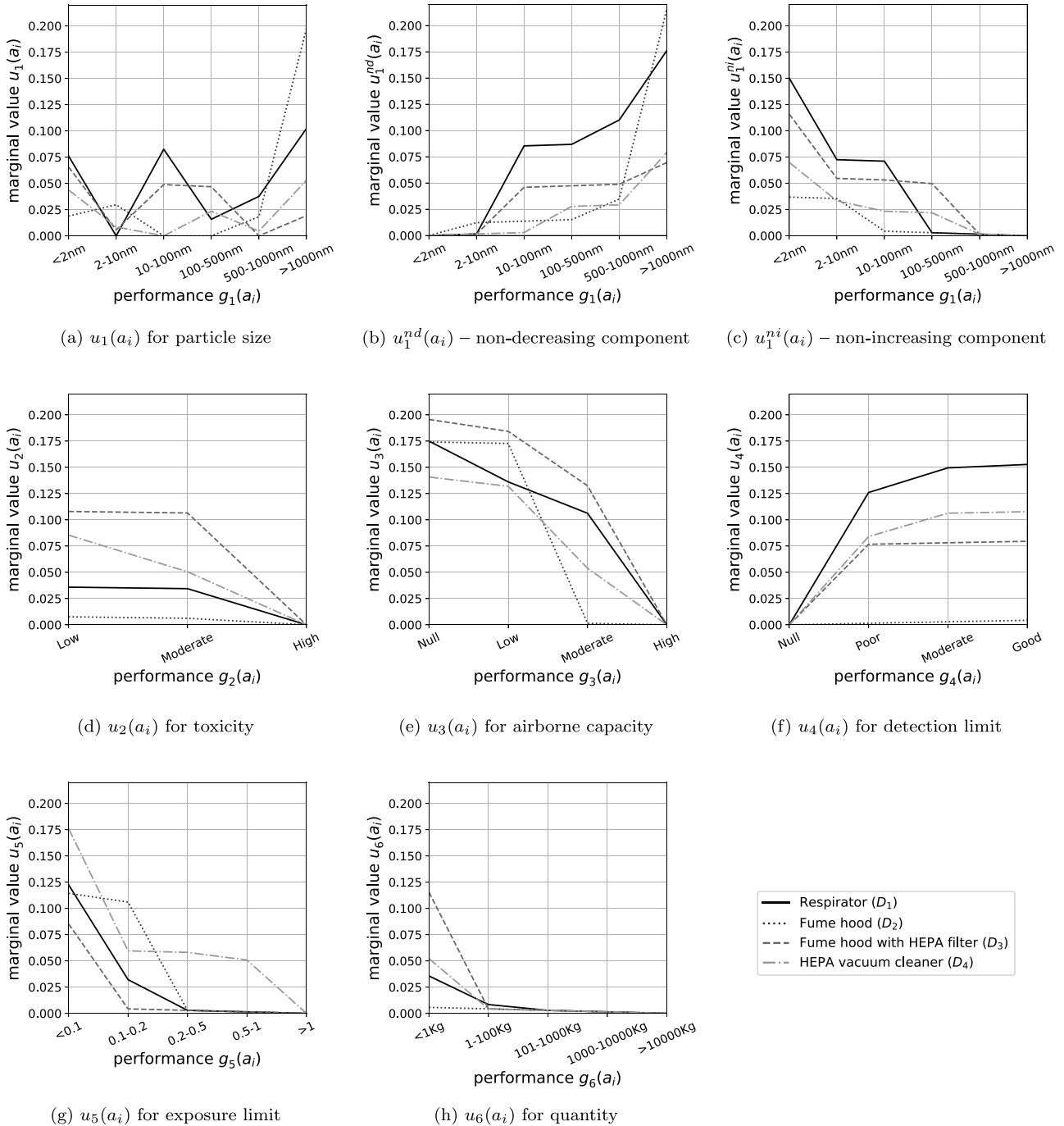


Fig. 6. Marginal value functions for criteria  $g_1 - g_6$  for four decision attributes.

Table 8

The number of reference alternatives assigned to a given class on four decision attributes.

Decision	Respirator ( $D_1$ )	Fume hood ( $D_2$ )	Fume hood with HEPA filter ( $D_3$ )	HEPA vacuum cleaner ( $D_4$ )
"Required" ( $C_1$ )	8	27	10	6
"Might be required" ( $C_2$ )	3	3	4	6
"Optional" ( $C_3$ )	14	5	14	17
"Might be optional" ( $C_4$ )	3	2	2	2
"Not required" ( $C_5$ )	12	3	10	9

most significant difference is between the limits of 0.1 – 0.2 and 0.2–0.5. In addition, for all types of precautions but *HEPA vacuum cleaner*, the marginal value assigned to performances at least 0.2 is close to zero.

The marginal value function for *quantity* ( $u_6$ ) indicates that the production of small quantities of nanomaterial is considered less risky. In contrast, for the mass production above 1kg – the marginal values are close to zero. The production of a

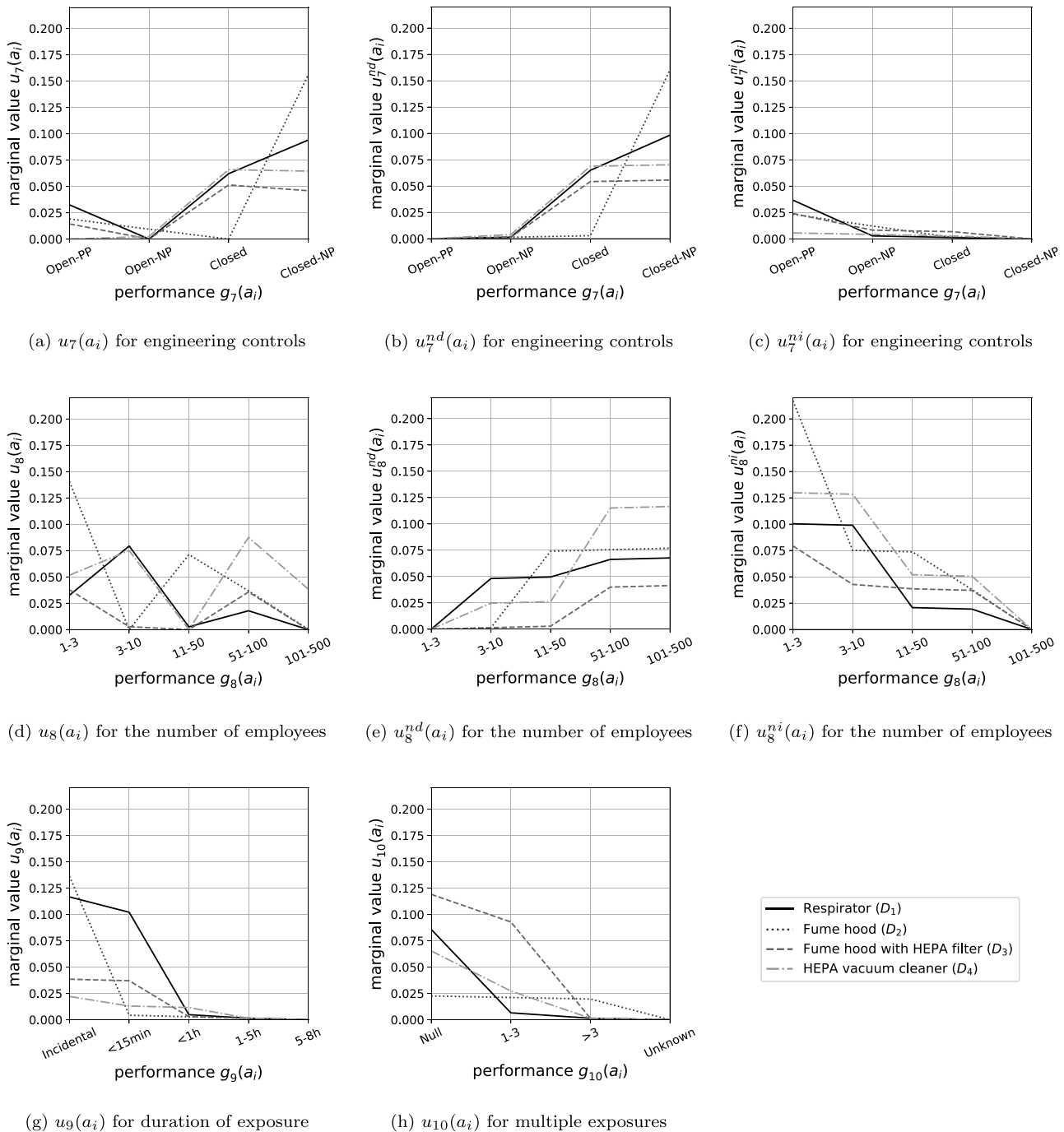


Fig. 7. Marginal value functions for criteria  $g_7 - g_{10}$  for four decision attributes.

small amount of nanomaterial reduces the chance of exposure. At higher manufacturing loads, the chances of accidental or undesired contact are higher. In the context of *fume hood*,  $g_6$  has no significant impact on the comprehensive value, and for each quantity produced, this precaution is required.

The marginal value functions for the *engineering controls* ( $u_7$ ) indicate that the closed systems are safer than open ones, particularly those with the negative pressure. The most desired configuration depends on the decision attribute. *Fume hood* is more required in the closed system, and for the remaining types of precautions, an open system with negative pressure is the most needed. The non-decreasing component ( $u_7^{nd}$ ) is prevailing, implying high marginal values for the closed systems, whereas

the non-increasing component ( $u_7^{ni}$ ) assigns low values to all possible configurations of the *engineering controls*.

The marginal functions for the *number of employees* ( $u_8$ ) reveal a slightly different shape for each decision attribute. We can observe two main peaks corresponding to 3–10 employees for *respirator* and *HEPA vacuum cleaner* or 51–100-employees for *respirator*, *HEPA vacuum cleaner* and *fume hood with HEPA filter*. The performances with the least assigned marginal value are 11–50 employees for *respirator*, *HEPA vacuum cleaner*, and *fume hood with HEPA filter* or 101–500 employees for *respirator*, *fume hood*, and *fume hood with HEPA filter*. For all decision attributes, the values assigned to 101–500 employees are

**Table 9**  
Class assignments provided by the experts for reference alternatives and their comprehensive values for four decision attributes.

$i / j$	$C_{DM}^{D_j}(a_i)$				$U^{D_j}(a_i)$				$i / j$	$C_{DM}^{D_j}(a_i)$				$U^{D_j}(a_i)$			
	1	2	3	4	1	2	3	4		1	2	3	4	1	2	3	4
$a_1$	3	2	2	3	0.4539	0.2134	0.5800	0.5502	$a_{21}$	3	2	3	3	0.4874	0.3970	0.4894	0.4773
$a_2$	2	1	2	2	0.4039	0.3493	0.4207	0.3707	$a_{22}$	1	1	1	1	0.3445	0.3742	0.3493	0.2993
$a_3$	1	1	2	2	0.3445	0.0668	0.4207	0.3707	$a_{23}$	3	1	3	3	0.4607	0.3742	0.4922	0.4773
$a_4$	1	1	1	3	0.2296	0.0835	0.2772	0.3779	$a_{24}$	3	1	1	1	0.4231	0.3037	0.3476	0.2588
$a_5$	1	1	1	1	0.1954	0.0382	0.1881	0.2330	$a_{25}$	5	1	5	5	0.5659	0.3742	0.5708	0.5559
$a_6$	3	1	3	3	0.4231	0.3742	0.4312	0.3779	$a_{26}$	5	1	5	5	0.7060	0.3742	0.7145	0.5559
$a_7$	3	1	3	3	0.4231	0.3742	0.4279	0.3779	$a_{27}$	5	1	5	5	0.5659	0.1814	0.5708	0.5639
$a_8$	3	1	3	3	0.4874	0.3742	0.4922	0.4773	$a_{28}$	4	4	4	4	0.5772	0.5255	0.4300	0.3405
$a_9$	3	1	5	3	0.4231	0.3628	0.5708	0.3779	$a_{29}$	4	4	4	4	0.5551	0.5245	0.5598	0.4845
$a_{10}$	1	1	1	3	0.3445	0.2464	0.2929	0.3779	$a_{30}$	3	3	3	2	0.4500	0.5171	0.4417	0.3548
$a_{11}$	1	1	1	1	0.1639	0.2534	0.2019	0.2420	$a_{31}$	5	1	3	3	0.6738	0.3742	0.4279	0.4773
$a_{12}$	5	5	5	5	0.5659	0.5957	0.5877	0.5867	$a_{32}$	5	5	5	5	0.5659	0.6721	0.5708	0.5559
$a_{13}$	5	1	3	3	0.5659	0.0783	0.4279	0.4411	$a_{33}$	4	1	3	3	0.4988	0.2745	0.4574	0.4773
$a_{14}$	5	3	5	5	0.5659	0.4622	0.5708	0.5559	$a_{34}$	3	1	3	3	0.4874	0.3742	0.4922	0.4640
$a_{15}$	3	1	3	3	0.4874	0.3742	0.4922	0.4773	$a_{35}$	2	1	1	2	0.4159	0.3199	0.3493	0.3707
$a_{16}$	5	3	5	5	0.5659	0.4619	0.5708	0.5559	$a_{36}$	2	1	1	2	0.3975	0.2850	0.2609	0.3064
$a_{17}$	5	3	5	5	0.6526	0.5171	0.5815	0.6206	$a_{37}$	5	1	3	3	0.6244	0.2875	0.4279	0.4773
$a_{18}$	3	1	3	3	0.4231	0.1845	0.4442	0.4773	$a_{38}$	3	1	3	3	0.4589	0.1570	0.4854	0.4114
$a_{19}$	5	3	5	5	0.7145	0.4794	0.6035	0.5948	$a_{39}$	1	1	1	1	0.3290	0.3591	0.3493	0.2993
$a_{20}$	3	2	2	2	0.7145	0.6815	0.7145	0.6801	$a_{40}$	1	5	1	1	0.3445	0.5957	0.3331	0.2993

**Table 10**  
The maximal shares of the individual criteria in the comprehensive values (in %) for four decision attributes.

Criterion	Respirator ( $D_1$ )	Fume hood ( $D_2$ )	Fume hood with HEPA filter ( $D_3$ )	HEPA vacuum cleaner ( $D_4$ )
Particle size ( $g_1$ )	10.22%	20.6%	7.32%	6.18%
Toxicity ( $g_2$ )	3.58%	0.79%	12.04%	9.97%
Airborne capacity ( $g_3$ )	17.5%	18.15%	21.82%	16.44%
Detection limit ( $g_4$ )	15.27%	0.44%	8.87%	12.58%
Exposure limit ( $g_5$ )	12.24%	11.91%	9.5%	20.57%
Quantity ( $g_6$ )	3.56%	0.59%	12.88%	6.09%
Engineering controls ( $g_7$ )	9.4%	16.26%	5.73%	7.7%
Number of employees ( $g_8$ )	7.93%	14.65%	4.22%	10.22%
Duration of exposure ( $g_9$ )	11.66%	14.21%	4.29%	2.58%
Multiple exposure ( $g_{10}$ )	8.58%	2.35%	13.28%	7.63%

**Table 11**  
Values of bias for each non-monotonic criterion for all decision attributes.

Criterion	Respirator ( $D_1$ )	Fume hood ( $D_2$ )	Fume hood with HEPA filter ( $D_3$ )	HEPA vacuum cleaner ( $D_4$ )
Particle size ( $g_1$ )	0.10	0.03	0.07	0.04
Engineering controls ( $g_7$ )	0.01	0.01	0.01	0.01
Number of employees ( $g_8$ )	0.09	0.11	0.06	0.11

smaller than those associated with 51–100 employees. The non-increasing ( $u_8^{ni}$ ) and non-decreasing ( $u_8^{nd}$ ) components explain why the resulting marginal functions differ across various decisions. For *respirator*, the non-decreasing component increases only between 3–10 and 11–50 employees, whereas the functions for the remaining decisions in this performance area are stable. In turn, they increase for the number of employees between 1–3 and 3–10 and between 11–50 and 51–100.

The *duration of exposure* ( $u_9$ ) is particularly important in the context of *respirator* and *fume hood*. When the time exceeds one hour, there is a greater safety concern, and thus all precautions are more required. A short duration of exposure can motivate the reduction of safety requirements.

The analysis of marginal functions for the *number of exposures* ( $u_{10}$ ) indicates that if the exposures are non-existing, the marginal value is high, hence leading to the assignment to a less risky class for all decision attributes. In case there is at least one exposure, the *respirator* is more required. The precautions involving the HEPA filter are required when three exposures are exceeded. In turn, *fume hood* is equally necessary for all values when the *number of exposures* is known, and the marginal functions attain zero when the value of this criterion cannot be determined.

In general, the marginal value functions for the decision attributes concerning the use of the HEPA filter are similar for the *airborne capacity*, *detection limit*, *quantity*, *engineering controls*, *duration of exposure*, and *number of exposures*. In turn, the marginal functions corresponding to the two *fume hoods* differ on all criteria. This may suggest that these two precautions are complementary, and depending on the conditions, one should choose the fume hood either with the filter or without it. Finally, the functions for the *respirator* are more similar to those derived from the analysis for the *HEPA vacuum cleaner* and *fume hood with HEPA filter* than for the *fume hood*.

4.4.2. Class assignments for the reference alternatives

The comprehensive values and class assignments for the forty reference alternatives with respect to the four decision attributes are provided in Table 9. The constructed model reproduced the desired assignments for all reference exposure scenarios but  $a_1$ ,  $a_{20}$ , and  $a_{28}$ . These alternatives form the minimal subset that had to be removed to impose the consistency of the experts' judgments with an assumed preference model. The comparison of desired and resulting assignments for these three exposure scenarios is given in Table 12. For example, the inferred model

**Table 12**

Class assignments derived with the constructed preference model for the reference alternatives, not aligning with the ones desired by the experts.

i / j	$C_{DM}^{D_j}(a_i)$				$C^{D_j}(a_i)$			
	1	2	3	4	1	2	3	4
$a_1$	3	2	2	3	3	1	5	4
$a_{20}$	3	2	2	2	5	5	5	5
$a_{28}$	4	4	4	4	5	4	3	2

**Table 13**

Class thresholds separating the five preference-ordered classes for four decision attributes.

$D_j$	$b_1^{D_j}$	$b_2^{D_j}$	$b_3^{D_j}$	$b_4^{D_j}$
$D_1$	0.3480	0.4195	0.4909	0.5624
$D_2$	0.3778	0.4492	0.5207	0.5921
$D_3$	0.3529	0.4243	0.4958	0.5672
$D_4$	0.3028	0.3743	0.4809	0.5524

evaluated alternative  $a_{20}$  as “not required” ( $C_5$ ) for all decision classes, while the experts indicated that it should be “optional” and “might be required”. This was implied by the most preferred or nearly the best performances on the monotonic criteria  $g_2, g_3, g_5, g_6$  and  $g_{10}$ , as well as the performances on the non-monotonic criteria  $g_1$  and  $g_7$  that were assigned the greatest marginal values according to the constructed model.

The thresholds separating the decision classes on a scale of a comprehensive value for all decisions are given in Table 13. Let us remind that the range delimited by these thresholds in which a comprehensive value of a given alternative falls determines its assignment to the respective class. For example,  $U^{D_1}(a_2) = 0.4039$  is not lesser than  $b_1^{D_1} = 0.3480$  and lesser than  $b_2^{D_1} = 0.4195$ , which allows to reproduce the assignment of  $a_2$  to  $C_2$  provided by the experts. Although these thresholds have similar values for various decision attributes, we can observe that, e.g., on  $D_4$ , they are by 0.04 – 0.07 lower than on  $D_2$ .

4.4.3. Inter-decision relationships

Let us focus on the inter-decision relationships implied by the specificity of the considered multi-decision sorting problem. The impact of the individual criteria on the comprehensive values as well as the relations between the latter ones for different decision attributes are demonstrated in Fig. 8 for the four reference alternatives ( $a_9, a_3, a_{13}$ , and  $a_{33}$ ).

For example,  $a_9$  was assigned to  $C_5$  for  $D_3$ , to  $C_3$  for  $D_1$  and  $D_4$ , and to  $C_1$  for  $D_2$ . This information can be interpreted in such a way that when considering different types of precautions in the context of  $a_9$ , their ranking is the following: fume hood with HEPA filter ( $D_3$ ), respirator ( $D_1$ ) and HEPA vacuum cleaner ( $D_4$ ), fume hood ( $D_2$ ). Such a ranking is reflected in the comprehensive values on the respective decision attributes:  $U^{D_3}(a_9) > U^{D_1}(a_9), U^{D_4}(a_9) > U^{D_2}(a_9)$ . The analysis of marginal values for  $a_9$  indicates that, depending on the decision context, the same performance can yield very different contributions to the comprehensive values. This, in turn, may result in the extreme assignments for various decision attributes (e.g., the most preferred class on  $D_3$  and the least preferred class on  $D_2$ ). The comprehensive value of  $a_9$  for fume hood with HEPA filter ( $D_3$ ) is equal to 0.5707. Such a great value is implied mainly by the following high contributions from the individual criteria:  $u_2^{D_3}(a_9) = 0.1065, u_3^{D_3}(a_9) = 0.1956, u_4^{D_3}(a_9) = 0.0767$ , and  $u_{10}^{D_3}(a_9) = 0.0928$ . In turn, for fume hood, despite a slightly higher value for  $u_8^{D_2}(a_9) = 0.1406$ , the marginal value of  $a_9$  derived from  $u_5^{D_2}(a_9), u_6^{D_2}(a_9)$ , and  $u_9^{D_2}(a_9)$  are nearly zero. As a result, comprehensive value is lower (0.3627) than for other decision attributes.

The desired assignments of  $a_3$  were either  $C_2$  on  $D_3$  and  $D_4$  or to  $C_1$  on  $D_1$  and  $D_2$ . Consequently, its comprehensive values on all decision attributes are relatively low, while being slightly higher for the fume hood with HEPA filter and HEPA vacuum cleaner than for the respirator or fume hood. The assignments to classes representing more risky scenarios are mainly due to the low marginal values from the following criteria: airborne capacity ( $u_3$ ), exposure limit ( $u_5$ ), engineering controls ( $u_7$ ), number of employees ( $u_8$ ), and duration of exposure ( $u_9$ ). The differences in the assignments on  $D_1$  and  $D_3$  can be explained, e.g., with respect to toxicity ( $u_2^{D_1} = 0.0344$  and  $u_2^{D_3} = 0.1065$ ) and quantity ( $u_6^{D_1} = 0.0356$  and  $u_6^{D_3} = 0.1155$ ), making the respirator “required” with a comprehensive value of 0.3444 and fume hood with HEPA filter – “might be required” with a greater comprehensive value of 0.4207. When it comes to fume hood,  $a_3$  attained very low values on all criteria, making it “required”. In case of HEPA vacuum cleaner, a higher value justifying an assignment to  $C_2$  is implied by the significant contributions from the following criteria: toxicity, airborne capacity, detection limit, exposure limit, number of employees, and multiple exposure.

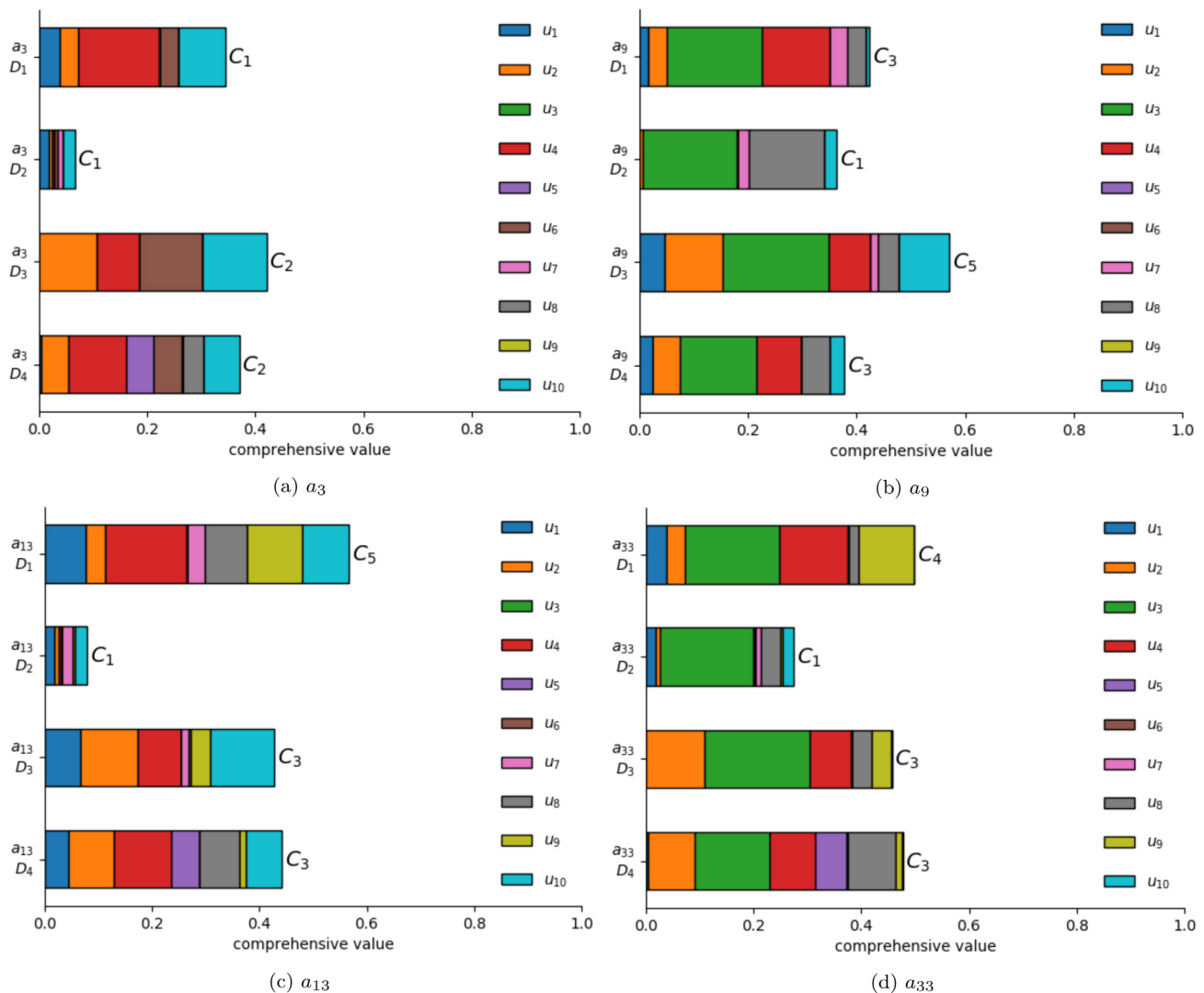
The difference in marginal values assigned to the same performances for various decision attributes as well as the inter-decision relationships between comprehensive values implied by the experts’ assignments can be also observed for  $a_{13}$  and  $a_{33}$  (see Fig. 8). On the one hand, the comprehensive values of  $a_{13}$  range from 0.0783 for  $D_2$  to 0.5659 for  $D_1$ . On the other hand, the large differences in marginal values assigned to  $a_{33}$  on various decision attributes for toxicity, detection limit, number of employees, and duration of exposure imply that it can be assigned to classes ranging from  $C_1$  for  $D_2$  to  $C_4$  for  $D_1$ . As a result, the ranking of precautions associated with  $a_{33}$  in terms of safety requirements (starting from the least required) reproduced by the constructed model is as follows: respirator, fume with HEPA filter and HEPA vacuum cleaner, fume hood.

4.4.4. Intra-decision relationships

To justify the assignments of alternatives to the respective classes, in Fig. 9, we demonstrate the comprehensive values of selected exposure scenarios and class thresholds. For each decision attribute, we depicted a single alternative assigned to each class. The comprehensive values of exposure scenarios assigned to better classes are greater than for the alternatives assigned to the classes associated with greater risk. For example, the following relations between comprehensive values on  $D_1$ :  $U^{D_1}(a_{37}) > U^{D_1}(a_{33}) > U^{D_1}(a_7) > U^{D_1}(a_2) > U^{D_1}(a_4)$  reflect the expert judgments. Let us explain a few example assignments in terms of marginal values attained on the respective criteria and the comparison of comprehensive values with the class thresholds.

When it comes to the evaluation of  $a_2$  in terms of respirator ( $D_1$ ), it attains the greatest marginal values for airborne capacity ( $u_3^{D_1}(a_2) = 0.1750$ ) and number of employees ( $u_8^{D_1}(a_2) = 0.0793$ ). However, its negligible marginal values derived from detection limit ( $u_4^{D_1}(a_2) = 0$ ), quantity ( $u_6^{D_1}(a_2) = 0.0014$ ), duration of exposure ( $u_9^{D_1}(a_2) = 0.0048$ ), and multiple exposure ( $u_{10}^{D_1}(a_2) = 0.0065$ ) imply a relatively small comprehensive value. Since  $b_1^{D_1} = 0.3480 \leq U^{D_1}(a_2) = 0.4039 < b_2^{D_1} = 0.4195$ ,  $a_2$  is assigned to  $C_2$  on  $D_1$ . As far as  $a_4$  is concerned, it attained zero marginal values on a few criteria and very low values on other criteria ( $u_3^{D_1}(a_4) = 0, u_5^{D_1}(a_4) = 0.0028, u_6^{D_1}(a_4) = 0.0014, u_7^{D_1}(a_4) = 0$ , and  $u_{10}^{D_1}(a_4) = 0.0014$ ). Therefore, despite high values for toxicity ( $u_2^{D_1}(a_4) = 0.0358$ ) and detection limit ( $u_4^{D_1}(a_4) = 0.1495$ ), it is assigned to  $C_1$  due to  $U^{D_1}(a_4) = 0.2296 < b_4^{D_1} = 0.3480$ .

When comparing the assignments of  $a_{14}$  and  $a_{40}$  in terms of fume hood ( $D_2$ ), these alternatives perform similarly on  $g_1, g_2$ ,



**Fig. 8.** Marginal and comprehensive values and class assignments demonstrating the inter-decision relationships for four reference exposure scenarios in terms of four decision attributes (Respirator –  $D_1$ , Fume hood –  $D_2$ , Fume hood with HEPA filter –  $D_3$ , and HEPA vacuum cleaner –  $D_4$ ).

$g_3$ , and  $g_8$  (e.g.,  $u_1^{D_2}(a_{14}) = 0.1976$  and  $u_1^{D_2}(a_{40}) = 0.1976$  or  $u_2^{D_2}(a_{14}) = 0.0076$  and  $u_2^{D_2}(a_{40}) = 0.0062$ ). However, the more advantageous performances of  $a_{40}$  on  $g_5$ ,  $g_7$ , and  $g_{10}$  imply that it is assigned to  $C_5$  as compared to  $C_3$  for  $a_{14}$ . Even though  $a_2$  attains comparable marginal values to  $a_{40}$  on six criteria ( $g_2$ ,  $g_3$ ,  $g_4$ ,  $g_5$ ,  $g_6$ , and  $g_7$ ), it is significantly less preferred on  $g_1$  and  $g_8$  (e.g.,  $u_1^{D_2}(a_2) = 0.0182 < u_1^{D_2}(a_{40}) = 0.1976$ ). As a result,  $a_2$  has a very low comprehensive value ( $U^{D_2}(a_2) = 0.3492$ ), justifying the assignment to the least preferred class  $C_1$ .

4.5. Classification of the non-reference alternatives

The model inferred from the analysis of reference alternatives can be used to classify other exposure scenarios. Thus, we first used expert knowledge to build a preference model. The latter is subsequently applied to evaluate other alternatives in a way that is consistent with the experts' value system and hence could be accepted by them. In this way, the proposed method can support nanomaterials' exposure management, suggesting the reasons for concern regarding some nanomanufacturing tasks performed by the workers [47,55].

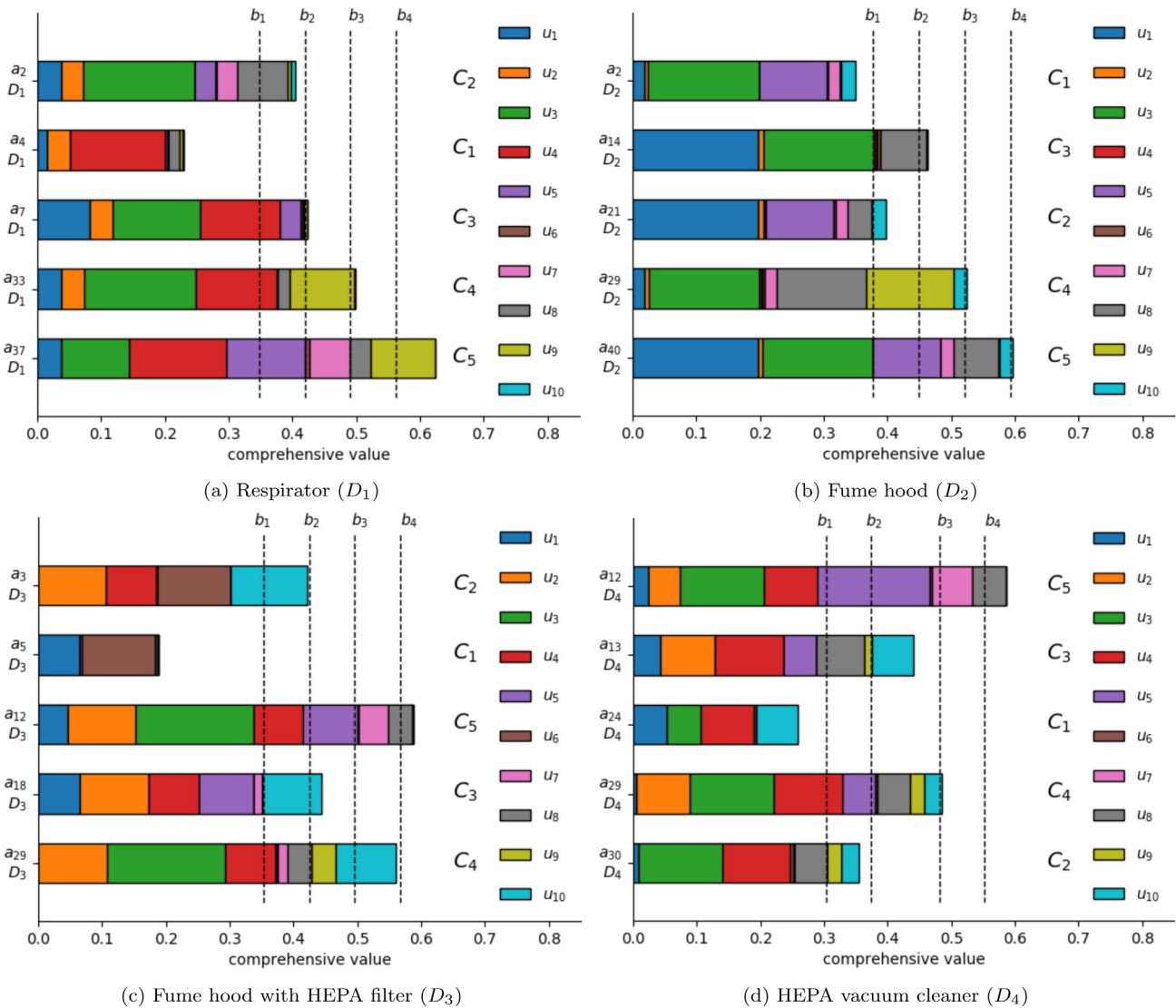
For this purpose, let us consider five non-reference alternatives presented in Table 7. Their comprehensive values and the

**Table 14**  
Comprehensive values and class assignments for the non-references alternatives for the four decision attributes.

$a_i / D_j$	$C^{D_j}(a_i)$				$U^{D_j}(a_i)$			
	1	2	3	4	1	2	3	4
$a_{41}$	3	2	1	3	0.4701	0.4202	0.2204	0.3781
$a_{42}$	4	1	3	3	0.5168	0.2631	0.4892	0.4362
$a_{43}$	5	3	3	5	0.6445	0.5061	0.4890	0.5936
$a_{44}$	1	1	1	1	0.2470	0.1031	0.3468	0.2763
$a_{45}$	2	1	2	3	0.3878	0.2019	0.4150	0.3893

respective class assignments are given in Table 14. Since the comprehensive values attained by these alternatives differ vastly from one decision attribute to another, the suggested classes differ too. For example,  $a_{42}$  is assigned to  $C_1$  on  $D_2$ , to  $C_3$  on  $D_3$  and  $D_4$ , and to  $C_4$  on  $D_1$ , whereas the classes for  $a_{43}$  range from  $C_3$  on  $D_2$  and  $D_3$  to  $C_5$  on  $D_1$  and  $D_4$ . Note, however, that although the comprehensive values of  $a_{44}$  differ with respect to various precaution types, they are all very low, justifying the assignment to  $C_1$  on all decision attributes.

For the five non-reference alternatives, the contribution of the individual criteria in the comprehensive values, as well as the assignments derived from the comparison of comprehensive



**Fig. 9.** Marginal and comprehensive values and class assignments demonstrating the intra-decision relationships for five reference exposure scenarios in terms of four decision attributes.

values with class thresholds, are presented in Fig. 10. Let us justify the assignments obtained for two selected non-reference exposure scenarios ( $a_{43}$  and  $a_{45}$ ).

When it comes to  $a_{43}$ , it is assigned to  $C_3$  on  $D_2$  and  $D_3$  and to  $C_5$  on  $D_1$  and  $D_4$ . This alternative attains the extreme values on different criteria. However, it performs relatively well on the criteria with a significant impact on the classification, i.e.,  $g_3$ ,  $g_4$  and  $g_5$ , which justifies its relatively high comprehensive values. They are slightly lower for *fume hood* ( $D_2$ ) and *fume hood with HEPA filter* ( $D_3$ ) mainly due to either zero ( $u_8^{D_2}(a_{43}) = 0$ ) or negligible ( $u_8^{D_3}(a_{43}) = 0.0027$ ) contribution of the *number of employees* ( $g_8$ ). This criterion forms an example of the direction in which the health managers should work to verify if any of them can be improved to increase a comprehensive value and justify the assignment to a less risky class for all decision attributes.

As far as the evaluation of  $a_{45}$  is concerned, it attains high or moderate marginal values on  $g_2$ ,  $g_3$ ,  $g_4$ ,  $g_7$ , and  $g_8$  when assessed in terms of  $D_1$ ,  $D_3$ , and  $D_4$ . This allows exceeding the lower threshold of class  $C_2$  for these decision attributes. When it comes to  $D_2$ , the significant contribution to the overall quality of  $a_{45}$  is offered only by  $g_8$ , implying the assignment to the least preferred class  $C_1$ . A comprehensive evaluation of  $a_{45}$  as “optional” given

*HEPA vacuum cleaner* ( $D_4$ ) is justified mainly by the value added by quantity ( $g_5$ ). Nevertheless, the indication of class at most  $C_3$  for all decision attributes can be perceived as a “safety warning flag”, suggesting that this exposure scenario should be prioritized. In general, the greater risk associated with the respective class, the greater attention should be paid to its further investigation. The marginal value functions for which the alternative attains very low values should be analyzed to identify the performance modifications offering a significant increase of the comprehensive value.

In the e-Appendix (supplementary material available online), we compare the method introduced in this paper with the one we proposed in [31]. We also collate the outcomes obtained for the case study with both methods. This required a suitable methodological extension of the approach presented in [31] to a multi-decision setting considered in this work.

### 5. Conclusions and future work

We considered and formalized a new problem of multi-decision sorting in Multiple Criteria Decision Analysis. In this



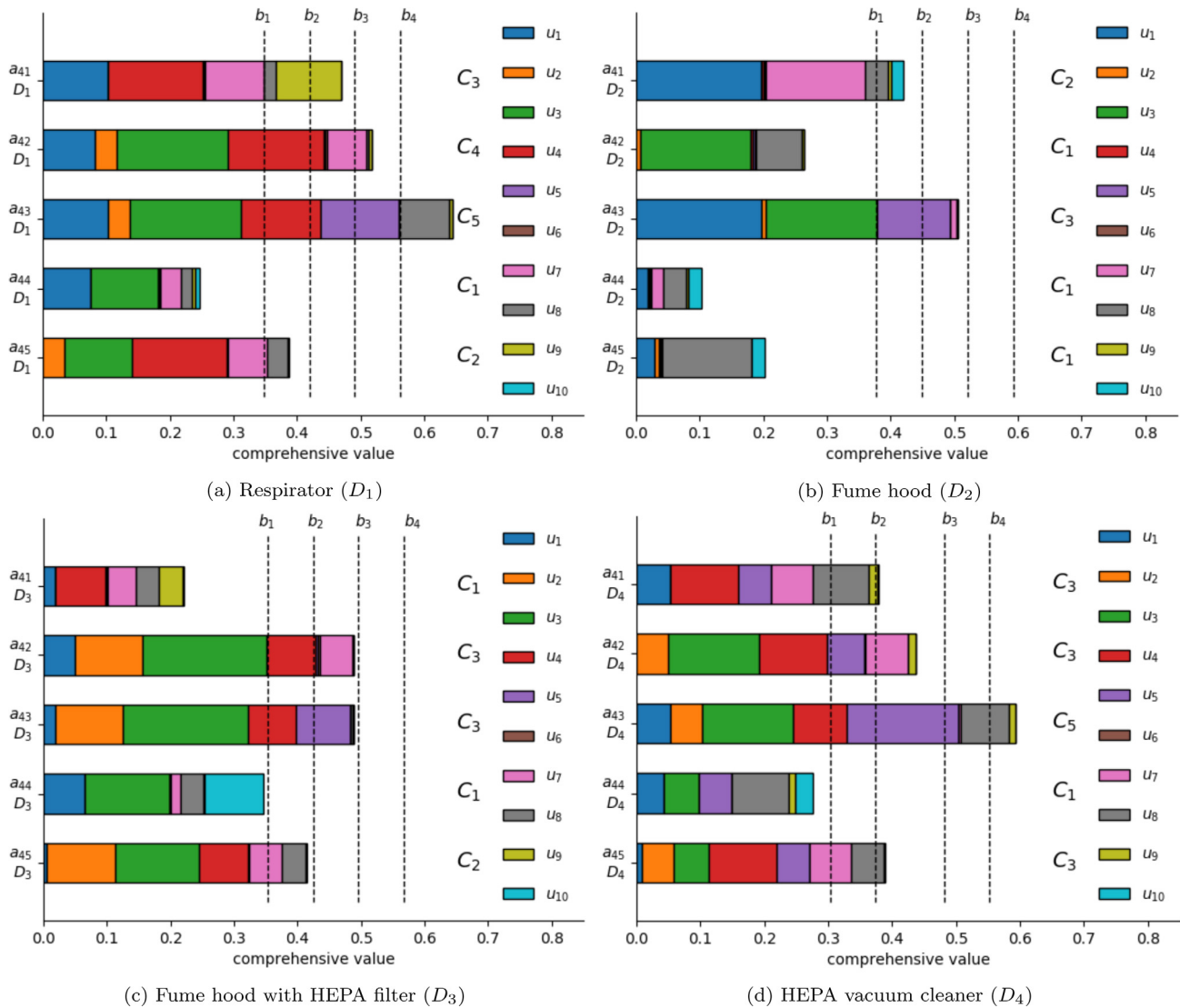


Fig. 10. Marginal and comprehensive values for the five non-reference protocols in terms of four decision attributes.

problem, besides performances on multiple criteria, each alternative gets evaluated in terms of multiple decision attributes involving pre-defined preference-ordered classes. To solve such a problem, one needs to construct a set of individual sorting models, one for each decision attribute. They should reflect both intra-decision dependencies between the assignments of different alternatives and inter-decision dependencies between the classes desired for the same alternative on various decision attributes.

We have proposed a preference disaggregation method for dealing with the multi-decision sorting. In this approach, the DM is expected to assign a subset of reference alternatives to a single class for each decision attribute by indicating the quality or risk level on the pre-defined scale for all decision attributes. Such indirect preference information is used to learn a set of inter-related models composed of an additive value function and class thresholds separating the decision classes on a comprehensive value scale. The preference modeling involves intra- and inter-decision constraints intending to reproduce the assignments of as many reference alternatives as possible.

The proposed framework has been extended with a novel proposal for dealing with criteria for which the preference direction cannot be specified a priori. Explicitly, a marginal value function for each non-monotonic criterion is represented as a sum of

non-decreasing and non-increasing components. In this way, the resulting marginal function can take any arbitrary shape. Hence it can represent local monotonicity relationships in different regions of the performance scale. The interpretability of the model is enhanced by the monotonicity of non-decreasing and non-increasing components, as well as normalization imposed by the subtraction of biases, guaranteeing that an anti-ideal alternative would attain a zero comprehensive value.

The introduction of a new type of multiple criteria problem and the dedicated methodology have been motivated by the peculiarity of nanomaterials' exposure management. In this context, each exposure scenario is described in terms of various characteristics of a given nanomaterial and working conditions related to its production. However, it is also evaluated in terms of different safety measures corresponding to various types of precautions. Each precaution can be modeled as a decision attribute capturing the potential level of concern related to a nanomanufacturing scenario. We have considered four inter-related precautions representing personal protective equipment, engineering controls, and work practices.

The analysis of desired assignments provided by the health and safety managers for forty exposure scenarios allowed us to construct four inter-related classification models. These models

captured some patterns and regularities from experts' judgments for risk management in nanomanufacturing. In particular, the highest maximal share in the comprehensive values of alternatives was attributed to airborne capacity, detection limit, and exposure limit. Furthermore, the high variability of marginal values assigned to different performances on the same criterion indicated the directions for analysis of nanomanufacturing processes to reduce the risk level by vastly increasing the marginal value with a small modification of performance. For example, we can consider changing the toxicity from high to moderate, decreasing the airborne capacity from high to moderate or low, decreasing the exposure limit to less than 0.1 fiber/cc, reducing the quantity to less than 1kg, or nullifying multiple exposures. Even though the shapes of marginal value functions related to various decision attributes were similar for most criteria, some differences revealed the peculiarities of the risk management in the context of incorporating the respirator, fume hood with and without the HEPA filter, or HEPA vacuum cleaner. For example, the marginal functions for the precautions involving the HEPA filter were very similar, which is probably related to the notable reliance on this type of filter to reduce the potential concern during the production processes. On the contrary, the functions for fume hood with or without the HEPA filter were rather different, confirming their complementarity. We have also demonstrated that the constructed model can support decision-making by applying it to the classification of five non-reference exposure scenarios with unknown risk levels. These sorting models could thus be used to provide decision recommendations on multiple risk management measures – corresponding to various types of precautions – for nanomanufacturing processes, especially those where there is still high uncertainty in the operational conditions as well as the physicochemical and toxicological characteristics of the nanomaterials.

The potential extensions of the proposed method are fourfold. The motivation for the first development comes from a large number of constraints imposed by the intra- and inter-decision relationships and the use of binary variables that are needed to find the largest subset of reference alternatives for which an assumed model can reproduce the expert judgments. These factors imply that the proposed method, requiring a mathematical programming solver, cannot be applied in problems with thousands of alternatives and hundreds of decision attributes. An adaptation to such a setting would require the development of heuristic algorithms incorporating the machine learning techniques. As opposed to the proposed framework, they should not attempt to find an accurate, optimal solution, searching instead for a highly satisfactory model in an approximate way.

Secondly, the marginal functions for which the monotonicity direction cannot be pre-defined can be modeled differently without incorporating the non-decreasing and non-increasing components. In particular, the proposed methodology remains valid when the functions for potentially non-monotonic criteria are inferred to minimize either the number of changes in monotonicity [31] (see e-Appendix) or the sum of changes in slopes [30, 56]. Similarly, the framework remains valid with threshold-based sorting procedures incorporating other preference models to compute alternatives' scores than an additive value function (e.g., the Choquet integral [57,58]).

Thirdly, the proposed method can be extended with the robustness analysis framework [4,23,58]. In this approach, one should account for all compatible multi-decision sorting models instead of a single representative one. Then, one should capture the potential variability of assignments for the non-reference alternatives given the multiplicity of analyzed models consistent with the DM's judgments. Such an approach can be

extended to analyze all maximal subsets of reference alternatives for which the provided assignments are consistent with an assumed model [59].

Finally, the idea of evaluating each alternative with a set of inter-related preference models can be adjusted to other problem types. For example, various value functions can be used to assess the suitability of a given alternative to be assigned to the groups of alternatives exhibiting different characteristics and being similar in terms of the DM's preferences. Then, it should be placed in a group for which the attained comprehensive value is the greatest. Such an approach could provide a novel way of dealing with multiple criteria nominal classification [60].

### CRediT authorship contribution statement

**Miłosz Kadziński:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Supervision, Project administration, Funding acquisition. **Krzysztof Martyn:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Visualization. **Marco Cinelli:** Conceptualization, Validation, Formal analysis, Investigation, Resources, Writing - review & editing. **Roman Słowiński:** Conceptualization, Methodology, Writing - review & editing. **Salvatore Corrente:** Conceptualization, Methodology, Writing - review & editing. **Salvatore Greco:** Conceptualization, Methodology, Writing - review & editing.

### 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.

### Acknowledgments

Miłosz Kadziński acknowledges support by the research funds (SBAD in 2021) of Poznan University of Technology, Poland. Krzysztof Martyn acknowledges support from the Polish National Science Center under the SONATA BIS project (grant no. DEC-2019/34/E/HS4/00045). Marco Cinelli acknowledges funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 743553. Salvatore Corrente and Salvatore Greco acknowledge the support of the Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) – PRIN 2017, project “Multiple Criteria Decision Analysis and Multiple Criteria Decision Theory”, grant 2017CY2NCA, and of the research projects “Multicriteria analysis to support sustainable decisions” and “Analysis and measurement of the competitiveness of enterprises, and territorial sectors and systems: a multicriteria approach” of the Department of Economics and Business of the University of Catania, Italy.

### Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.knosys.2021.106879>.

### References

- [1] M. Cinelli, M. Kadziński, M. Gonzalez, R. Słowiński, How to support the application of multiple criteria decision analysis? Let us start with a comprehensive taxonomy, *Omega* 96 (2020) 102261.
- [2] S. Greco, M. Ehrgott, J. Figueira, *Multiple Criteria Decision Analysis – State of the Art Surveys*, in: *International Series in Operations Research & Management Science*, Springer, New York, 2016.
- [3] C. Zopounidis, M. Doumpos, Multicriteria classification and sorting methods: a literature review, *European J. Oper. Res.* 138 (2002) 229–246.



- [4] S. Greco, V. Mousseau, R. Słowiński, Multiple criteria sorting with a set of additive value functions, *European J. Oper. Res.* 207 (3) (2010) 1455–1470.
- [5] J. Liu, X. Liao, W. Zhao, N. Yang, A classification approach based on the outranking model for multiple criteria ABC analysis, *Omega* 61 (2016) 19–34.
- [6] E. Saleh, J. Błaszczyński, A. Moreno, A. Valls, P. Romero-Aroca, S. de la Riva-Fernandez, R. Słowiński, Learning ensemble classifiers for diabetic retinopathy assessment, *Artif. Intell. Med.* 85 (2018) 50–63.
- [7] R. Slowinski, C. Zopounidis, Application of the rough set approach to evaluation of bankruptcy risk, *Intell. Syst. Account. Finance Manag.* 4 (1) (1995) 27–41.
- [8] M. Cinelli, S. Coles, M.N. Nadagouda, J. Błaszczyński, R. Słowiński, R.S. Varma, K. Kirwan, A green chemistry-based classification model for the synthesis of silver nanoparticles, *Green Chem.* 17 (2015) 2825–2839.
- [9] C. Zopounidis, M. Doumpos, *Multicriteria Sorting Methods*, Springer, Boston, 2009, pp. 2379–2396.
- [10] J. Almeida-Dias, J. Figueira, B. Roy, Electre Tri-C: A multiple criteria sorting method based on characteristic reference actions, *European J. Oper. Res.* 204 (3) (2010) 565–580.
- [11] W. Yu, Aide multicritère à la décision dans le cadre de la problématique du tri: méthodes et applications (Ph.D. thesis), LAMSADE, Université Paris Dauphine, Paris, 1992.
- [12] E. Jacquet-Lagréze, Y. Siskos, Preference disaggregation: 20 years of MCDA experience, *European J. Oper. Res.* 130 (2) (2001) 233–245.
- [13] J. Devaud, G. Groussaud, E. Jacquet-Lagréze, UTADIS: Une méthode de construction de fonctions d'utilité additives rendant compte de jugements globaux, in: *European Working Group on MCDA*, Bochum, Germany, 1980.
- [14] C. Zopounidis, M. Doumpos, PREFDIS: a multicriteria decision support system for sorting decision problems, *Comput. & Oper. Res.* 27 (7–8) (2000) 779–797.
- [15] C. Zopounidis, M. Doumpos, A multicriteria decision aid methodology for sorting decision problems: The case of financial distress, *Comput. Econ.* 14 (3) (1999) 197–218.
- [16] D. Diakoulaki, C. Zopounidis, G. Mavrotas, M. Doumpos, The use of a preference disaggregation method in energy analysis and policy making, *Energy* 24 (2) (1999) 157–166.
- [17] K. Pendaraki, C. Zopounidis, M. Doumpos, On the construction of mutual fund portfolios: A multicriteria methodology and an application to the greek market of equity mutual funds, *European J. Oper. Res.* 163 (2) (2005) 462–481.
- [18] V. Mousseau, L.C. Dias, J.R. Figueira, Dealing with inconsistent judgments in multiple criteria sorting models, *4OR* 4 (2) (2006) 145–158.
- [19] C. Zopounidis, M. Doumpos, Building additive utilities for multi-group hierarchical discrimination: the M.H.DIS method, *Optim. Methods Softw.* 14 (3) (2000) 219–240.
- [20] S. Corrente, M. Doumpos, S. Greco, R. Słowiński, C. Zopounidis, Multiple criteria hierarchy process for sorting problems based on ordinal regression with additive value functions, *Ann. Oper. Res.* 251 (1) (2017) 117–139.
- [21] M. Kadziński, K. Ciomek, R. Słowiński, Modeling assignment-based pairwise comparisons within integrated framework for value-driven multiple criteria sorting, *European J. Oper. Res.* 241 (3) (2015) 830–841.
- [22] J. Liu, M. Kadziński, X. Liao, X. Mao, Y. Wang, A preference learning framework for multiple criteria sorting with diverse additive value models and valued assignment examples, *European J. Oper. Res.* 286 (3) (2020) 963–985.
- [23] M. Kadziński, T. Tervonen, Stochastic ordinal regression for multiple criteria sorting problems, *Decis. Support Syst.* 55 (1) (2013) 55–66.
- [24] M. Köksalan, S. Bilgin Özpeynirci, An interactive sorting method for additive utility functions, *Comput. Oper. Res.* 36 (9) (2009) 2565–2572.
- [25] S. Greco, M. Kadziński, V. Mousseau, R. Słowiński, Robust ordinal regression for multiple criteria group decision: UTA-GMS-GROUP and UTADIS-GMS-GROUP, *Decis. Support Syst.* 52 (3) (2012) 549–561.
- [26] S. Greco, V. Mousseau, R. Słowiński, Robust ordinal regression for value functions handling interacting criteria, *European J. Oper. Res.* 239 (3) (2014) 711–730.
- [27] J. Rezaei, Piecewise linear value functions for multi-criteria decision-making, *Expert Syst. Appl.* 98 (2018) 43–56.
- [28] M. Doumpos, Learning non-monotonic additive value functions for multicriteria decision making, *OR Spectrum* 34 (1) (2012) 89–106.
- [29] M. Ghaderi, F. Ruiz, N. Agell, Understanding the impact of brand colour on brand image: A preference disaggregation approach, *Pattern Recognit. Lett.* 67 (2015) 11–18.
- [30] J. Liu, X. Liao, M. Kadziński, R. Słowiński, Preference disaggregation within the regularization framework for sorting problems with multiple potentially non-monotonic criteria, *European J. Oper. Res.* 276 (3) (2019) 1071–1089.
- [31] M. Kadziński, K. Martyn, M. Cinelli, R. Słowiński, S. Corrente, S. Greco, Preference disaggregation for multiple criteria sorting with partial monotonicity constraints: Application to exposure management of nanomaterials, *Internat. J. Approx. Reason.* 117 (2020) 60–80.
- [32] A. Ulucan, K. Atici, A multiple criteria sorting methodology with multiple classification criteria and an application to country risk evaluation, *Technol. Econ. Dev. Econ.* 19 (1) (2013) 93–124.
- [33] M. Doumpos, J. Figueira, A multicriteria outranking approach for modeling corporate credit ratings: An application of the Electre Tri-nC method, *Omega* 82 (2019) 166–180.
- [34] J. Almeida-Dias, J. Figueira, B. Roy, A multiple criteria sorting method where each category is characterized by several reference actions: The electre Tri-nC method, *European J. Oper. Res.* 217 (3) (2012) 567–579.
- [35] S. Naidu, *Towards Sustainable Development of Nanomanufacturing* (Ph.D. thesis), University of Tennessee, Knoxville, 2012.
- [36] H. Goede, Y. Christopher-de Vries, E. Kuijpers, W. Fransman, A review of workplace risk management measures for nanomaterials to mitigate inhalation and dermal exposure, *Ann. Work. Expo. Health* 62 (8) (2018) 907–922.
- [37] C. Oksel, N. Hunt, T. Wilkins, X. Wang, Risk management of nanomaterials - guidelines for the safe manufacture and use of nanomaterials, 2017, Sustainable Nanotechnologies Project, <http://www.sun-fp7.eu>.
- [38] G. Tsoumakas, I. Katakis, Multi-label classification: An overview, *Int. J. Data Warehous. Min. (IJDWM)* 3 (3) (2007) 1–13.
- [39] K. Dembczyński, W. Cheng, E. Hüllermeier, Bayes optimal multilabel classification via probabilistic classifier chains, in: *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICM 10*, (Madison, USA), Omnipress, 2010, pp. 279–286.
- [40] J. Błaszczyński, S. Greco, R. Słowiński, Inductive discovery of laws using monotonic rules, *Eng. Appl. Artif. Intell.* 25 (2) (2012) 284–294.
- [41] K. Zhang, K.O. Kirlikovali, Q.V. Le, Z. Jin, R.S. Varma, H.W. Jang, O.K. Farha, M. Shokouhimehr, Extended metal-organic frameworks on diverse supports as electrode nanomaterials for electrochemical energy storage, *ACS Appl. Nano Mater.* 3 (5) (2020) 3964–3990.
- [42] M. Nasrollahzadeh, M. Sajjadi, S. Irvani, R.S. Varma, Green-synthesized nanocatalysts and nanomaterials for water treatment: Current challenges and future perspectives, *J. Hazard. Mater.* 401 (2021) 123401.
- [43] E.O. Ogunsona, R. Muthuraj, E. Ojogbo, O. Valerio, T.H. Mekonnen, Engineered nanomaterials for antimicrobial applications: A review, *Appl. Mater. Today* 18 (2020) 100473.
- [44] A.M. Abdelmonem, Application of carbon-based nanomaterials in food preservation area, in: K.A. Abd-El Salam (Ed.), *Carbon Nanomaterials for Agri-Food and Environmental Applications*, Elsevier, Micro and Nano Technologies, 2020, pp. 583–593.
- [45] M.M. Falinski, D.L. Plata, S.S. Chopra, T.L. Theis, L.M. Gilbertson, J.B. Zimmerman, A framework for sustainable nanomaterial selection and design based on performance, hazard, and economic considerations, *Nat. Nanotechnol.* 13 (8) (2018) 708–714.
- [46] R.D. Klaper, The known and unknown about the environmental safety of nanomaterials in commerce, *Small* 16 (36) (2020) e2000690.
- [47] P. Isigonis, D. Hristozov, C. Benighaus, E. Giubilato, K.G. Pizzolo, E. Semenzin, I. Linkov, A. Zabeo, A. Marcomini, Risk governance of nanomaterials: Review of criteria and tools for risk communication, evaluation, and mitigation, *Nanomaterials* 9 (5) (2019) 696.
- [48] M.L. Kirkegaard, P. Kines, K.C. Jeschke, K.A. Jensen, Risk perceptions and safety cultures in the handling of nanomaterials in academia and industry, *Ann. Work. Expo. Health* 64 (5) (2020) 479–489.
- [49] V. Stone, M. Fuhr, P.H. Feindt, H. Boutermeester, I. Linkov, S. Sabella, F. Murphy, K. Bizer, L. Tran, M. Agerstrand, C. Fito, T. Andersen, D. Anderson, E. Bergamaschi, J.W. Cherrie, S. Cowan, J.-F. Dalemcourt, M. Faure, S. Gabbert, A. Gajewicz, T.F. Fernandes, D. Hristozov, H.J. Johnston, T.C. Lansdown, S. Linder, H.J.P. Marvin, M. Mullins, K. Purnhagen, T. Puzyn, A. Sanchez Jimenez, J.J. Scott-Fordsmand, G. Streftaris, M. van Tongeren, N.H. Voelcker, G. Voyiatzis, S.N. Yannopoulos, P.M. Poortvliet, The essential elements of a risk governance framework for current and future nanotechnologies, *Risk Anal.* 38 (7) (2018) 1321–1331.
- [50] A. Rahi, N. Sattarahmady, H. Heli, Toxicity of nanomaterials-physicochemical effects, *Austin J. Nanomed. & Nanotechnol.* 2 (2014) 1034.
- [51] B. Fadeel, L. Farcial, B. Hardy, S. Vázquez-Campos, D. Hristozov, A. Marcomini, I. Lynch, E. Valsami-Jones, H. Alenius, K. Savolainen, Advanced tools for the safety assessment of nanomaterials, *Nat. Nanotechnol.* 13 (7) (2018) 537–543.
- [52] S.F. Hansen, K.A. Jensen, A. Baun, Nanoriskcat: a conceptual tool for categorization and communication of exposure potentials and hazards of nanomaterials in consumer products, *J. Nanopart. Res.* 16 (1) (2013) 2195.
- [53] D. Hristozov, S. Gottardo, M. Cinelli, P. Isigonis, A. Zabeo, A. Critto, M.V. Tongeren, L. Tran, A. Marcomini, Application of a quantitative weight of evidence approach for ranking and prioritising occupational exposure scenarios for titanium dioxide and carbon nanomaterials, *Nanotoxicology* 8 (2) (2014) 117–131.
- [54] B.V. Duuren-Stuurman, S. Vink, K. Verbist, H. Heussen, D. Brouwer, D. Kroese, M.V. Niftrik, E. Tieleman, W. Fransman, *Stoffenmanager nano* version 1.0: a web-based tool for risk prioritization of airborne manufactured nano objects, *Ann. Occup. Hyg.* 56 (5) (2012) 525–541.

- [55] F. Silva, P. Arezes, P. Swuste, Risk management: Controlling occupational exposure to nanoparticles in construction, in: F. Pacheco-Torgal, M.V. Diamanti, A. Nazari, C.G. Granqvist, A. Pruna, S. Amirkhanian (Eds.), *Nanotechnology in Eco-Efficient Construction*, second ed., Woodhead Publishing, 2019, pp. 755–784.
- [56] M. Ghaderi, F. Ruiz, N. Agell, A linear programming approach for learning non-monotonic additive value functions in multiple criteria decision aiding, *European J. Oper. Res.* 259 (3) (2017) 1073–1084.
- [57] S. Angilella, P. Catalfo, S. Corrente, A. Giarlotta, S. Greco, M. Rizzo, Robust sustainable development assessment with composite indices aggregating interacting dimensions: The hierarchical-SMAA-choquet integral approach, *Knowl. Based Syst.* 158 (2018) 136–153.
- [58] G. Beliakov, J.-Z. Wu, D. Divakov, Towards sophisticated decision models: Nonadditive robust ordinal regression for preference modeling, *Knowl. Based Syst.* 190 (2020) 105351.
- [59] M. Kadziński, M. Cinelli, K. Ciomek, S. Coles, M. Nadagouda, R. Varma, K. Kirwan, Co-constructive development of a green chemistry-based model for the assessment of nanoparticles synthesis, *European J. Oper. Res.* 264 (2) (2018) 472–490.
- [60] A. Costa, J. Figueira, J. Borbinha, A multiple criteria nominal classification method based on the concepts of similarity and dissimilarity, *European J. Oper. Res.* 271 (1) (2018) 193–209.

## Publication [P3]

K. Martyn and M. Kadziński. Deep preference learning for multiple criteria decision analysis. *European Journal of Operational Research*, 305(2):781–805, 2023, DOI: 10.1016/j.ejor.2022.06.053.

Number of citations<sup>1</sup>:

- according to Web of Science: 5
- according to Google Scholar: 6

---

<sup>1</sup>as on June 1, 2023



Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

# European Journal of Operational Research

journal homepage: [www.elsevier.com/locate/ejor](http://www.elsevier.com/locate/ejor)

## Decision Support

# Deep preference learning for multiple criteria decision analysis

Krzysztof Martyn\*, Miłosz Kadziński

Institute of Computing Science, Poznań University of Technology, Piotrowo 2, Poznań 60-965, Poland



## ARTICLE INFO

### Article history:

Received 7 September 2021

Accepted 24 June 2022

Available online 30 June 2022

### Keywords:

Multiple criteria decision aiding

Preference learning

Artificial neural networks

Multiple criteria sorting

Preference disaggregation

## ABSTRACT

We propose preference learning algorithms for inferring the parameters of a threshold-based sorting model from large sets of assignment examples. The introduced framework is adjusted to different scores originally used in Multiple Criteria Decision Analysis (MCDA). They include Ordered Weighted Average, an additive value function, the Choquet integral, a distance from the ideal and anti-ideal alternatives, and Net Flow Scores built on the results of outranking-based pairwise comparisons. As a concrete application of these models, we use Artificial Neural Networks with up to five hidden layers. Their components and architecture are designed to ensure high interpretability, which supports the models' acceptance by domain experts. To learn the most favorable values of all parameters at once, we use a variant of a gradient descent optimization algorithm called AdamW. In this way, we make the MCDA methods suitable for handling vast, inconsistent information. The extensive experiments on various benchmark problems indicate that the introduced algorithms are competitive in predictive accuracy quantified in terms of Area Under Curve and the 0/1 loss. In this regard, some approaches outperform the state-of-the-art algorithms, including generalizations of logistic regression, mathematical programming, rule ensemble and tree induction algorithms, or dedicated heuristics.

© 2022 Elsevier B.V. All rights reserved.

## 1. Introduction

The need to process data to conclusions or arguments that support more informed and better decision-making is growing each year (Liu, Kadziński, Liao, & Mao, 2021). Consequently, one of the main trends in today's information technology is developing intelligent decision support systems. Their successful application depends on the quality of being believable or trustworthy (Linkov, Galaitsi, Trump, Keisler, & Kott, 2020). The need to explain the decisions made by computer systems (Doshi-Velez & Kim, 2017) is reflected in the legal regulations of the European Union (Goodman & Flaxman, 2017).

Multiple Criteria Decision Aiding (MCDA) and Machine Learning (ML) belong to the most important and fastest developing disciplines within Artificial Intelligence (AI) (Corrente, Greco, Kadziński, & Słowiński, 2013; Doumpos & Zopounidis, 2011). They offer methods that support humans in decision-making processes. Within the scope of this paper, we focus on multiple criteria sorting (Alvarez, Ishizaka, & Martínez, 2021) or instance ranking (Fürnkranz & Hüllermeier, 2011) problems. They aim at assigning a set of alternatives to preference ordered classes, labels, or degrees in the

presence of multiple attributes with pre-defined preference directions. Moreover, we limit our interest to learning ordered classification models from decision examples. In MCDA, they are treated as the DM's indirect preference information in the form of assignment examples (Liu, Liao, Kadziński, & Słowiński, 2019; Zopounidis & Doumpos, 2000), whereas in ML – they form a training set in the task of supervised learning (Doumpos & Zopounidis, 2011). The goal is to find the model for classifying all alternatives, including the ones that have not been judged directly by the Decision Maker (DM) nor considered in the reference set (Doumpos & Zopounidis, 2018).

Even though the paradigm of learning by example is handled by both MCDA and ML, there are notable differences between these two disciplines (Corrente et al., 2013; Doumpos & Zopounidis, 2011; Waegeman, De Baets, & Boullart, 2009). On the one hand, MCDA is user-oriented. It exploits Decision Makers' knowledge or expertise and aims at the DMs to learn about their preferences and the problem at hand. On the contrary, ML is model-oriented, being focussed on data analysis, information extraction, and preference discovery. These various aims are, in turn, reflected in different forms of incorporated models, the amount of processed information, techniques used for arriving at a final result, and the role of users.

The preference models used in MCDA are highly interpretable and explainable (Corrente et al., 2013). Their primary role is to en-

\* Corresponding author.

E-mail addresses: [krzysztof.martyn@cs.put.poznan.pl](mailto:krzysztof.martyn@cs.put.poznan.pl) (K. Martyn), [milosz.kadziński@cs.put.poznan.pl](mailto:milosz.kadziński@cs.put.poznan.pl) (M. Kadziński).

courage the involvement of the DMs (Roy, 2010) through gaining insights on the role of different criteria, the character of alternatives, and the influence of particular performances on the decision. On the contrary, ML has mainly focused on the development of non-linear models, offering higher predicting ability and the possibility of capturing complex interdependencies (Corrente et al., 2013). However, this results in limited ability to determine which data influences a decision and, consequently, less confidence in the model's employment by the users who need to interpret and understand the underlying process (Waegeman et al., 2009).

The traditional MCDA methods have been designed for learning from a small set of decision examples for a subset of reference alternatives (Doumpos & Zopounidis, 2018). Typically, the translation of assignment examples into compatible values of an assumed preference model has been conducted with mathematical programming techniques that aim at reconstructing the DM's judgments as faithfully as possible. However, when the DM's preference information is rich and highly inconsistent, most approaches cannot deal efficiently with preference disaggregation (Liu et al., 2019). Some exceptions that have in-built mechanisms for dealing efficiently with large sets of inconsistent preferences include variants of the Dominance-based Rough Set Approach (DRSA) (Greco, Matarazzo, & Słowiński, 2001) and UTADIS (Zopounidis & Doumpos, 2000). On the contrary, ML has always been focused on dealing with large, inconsistent sets of training data (Doumpos & Zopounidis, 2011). These are usually composed of historical data, preferences collected over time, or observations of past decisions. In ML, some advanced statistical models or optimization algorithms are used to exploit the parameters space in search of the values that minimize some classification error.

Over the years, MCDA and ML have been developing separately while fostering their interests mentioned above. Nonetheless, the availability of large data resources as well as the need for both explainable models and interpretable decision-making processes have motivated the cross-fertilization of the two disciplines. Individuals, companies, organizations, and governments have accumulated a vast quantity of data, and its analysis has exceeded the reach of human processing capacity. However, it needs to be exploited in a way that allows verifying whether a model focuses on the relevant aspects, offers arguments and knowledge for decision-making, and involves the DM to take part in the process actively. Consequently, one has developed the algorithms that scale up well with an increasing number of assignment examples, at the same time incorporating the intuitive models originally proposed in MCDA (Cinelli, Kadziński, Miebs, Gonzalez, & Słowiński, 2022).

The research at the crossroads of MCDA and ML is called preference learning (Fürnkranz & Hüllermeier, 2011). Within this field, some MCDA methods have been adjusted to deal with large data, leading to the elaboration of intuitive classification methods. In what follows, we list the representative algorithms aimed at multiple criteria sorting and instance ranking problems. In particular, Chandrasekaran, Ryu, Jacob, & Hong (2005) proposed linear programming models based on isotonic separation, and Kotłowski & Słowiński (2013) introduced a family of classifiers exploiting the class of all monotonic functions, not making any additional assumptions about the model apart from the monotonicity constraints. Then, Tehrani, Cheng, Dembczyński, & Hüllermeier (2012) generalized logistic regression to learn the parameters of the Choquet integral, Liu et al. (2021) formulated optimization models for learning additive value functions augmented with components for handling the interactions between criteria, whereas Kadziński & Szczepański, (2022) proposed a variety of methods for learning the parameters of a sorting model with characteristic class profiles. Furthermore, Dembczyński, Kotłowski, & Słowiński (2009) introduced an algorithm based on the variant of

DRSA for generating a monotonic rule ensemble and Dembczyński, Kotłowski, & Słowiński (2006) extended DRSA by considering an additive function model resulting from rough approximations. Also, a few approaches have been proposed to learn the parameters of an outranking-based sorting model used in the ELECTRE TRI-B method or its simplified variant called MR-Sort. They include an evolutionary algorithm (Doumpos, Marinakis, Marinaki, & Zopounidis, 2009) or linear programming models combined with simulated annealing (Olteanu & Meyer, 2014) or a dedicated metaheuristic (Sobrie, Mousseau, & Pirlot, 2019).

This paper proposes to use Artificial Neural Networks (ANNs) for preference learning in the context of highly interpretable MCDA models. ANNs are versatile learners that can be applied to nearly any learning task, where input and output data are well-understood, yet the process that relates the input to the output is highly complex. Over the last years, ANNs have been successfully applied in the context of data analysis, control systems, speech and pattern recognition, and computer games. This is mainly due to the development of Deep Learning (DL) (i.e., efficient learning algorithms for ANNs with multiple hidden layers) that has revolutionized the field of AI and its applicability in the context of big data (Deng & Yu, 2014). However, the employment of ANNs in MCDA has been scarce. In particular, Malakooti & Zhou (1994) used an Adaptive Feedforward Adaptive Feedforward ANN to learn the utility function based on a set of training patterns in the form of alternatives with their associated evaluations by the DM and then applied it to rank a discrete set of alternatives. Moreover, Hu (2009) proposed a single-layer perceptron for multiple criteria classification problems based on pairwise comparisons among alternatives conducted in the spirit of an ELECTRE-based outranking relation. Furthermore, Hanne (1997) suggested the use of ANNs as a part of an MCDA network, in which they can be applied to standardize and aggregate performances from different criteria or even to choose the most relevant method from a pre-defined pool of a few approaches. Finally, Guo, Zhang, Liao, Chen, & Zeng (2021) proposed the NN-MCDA method that combines an additive value model with potentially non-monotonic marginal functions and a fully connected deep neural network.

We introduce the preference learning algorithms that use ANNs to infer parameters of the threshold-based sorting procedure from large sets of assignment examples. In this procedure, following UTADIS (Zopounidis & Doumpos, 2000), the frontiers between classes are delimited by the thresholds on a scale of a comprehensive score that reflects the quality of each alternative from all relevant viewpoints considered jointly. We adjust the introduced framework to different types of scores. In particular, we consider aggregation of the performances on various criteria using OrderedWeighted Average (OWA) operator (Yager, 1988), an additive value function initially employed in UTADIS (Zopounidis & Doumpos, 2000), and the Choquet integral (Angilella, Corrente, Greco, & Słowiński, 2013). These scores are able to capture different compensation levels or interactions between criteria. Moreover, we account for a model postulated in TOPSIS that builds on the distances of a given alternative from the ideal and anti-ideal options (Hwang & Yoon, 1981). Also, we consider the Net Flow Score (NFS) procedures that aggregate the results of pairwise comparisons between all alternatives. The comparisons are conducted in the spirit of the PROMETHEE (Brans & De Smet, 2016) and ELECTRE (Figueira, Greco, Roy, & Słowiński, 2013) methods, exploiting either preference degrees or the outcomes of concordance and discordance tests.

The ANNs have been originally designed to capture complex transformations of inputs (in our case, performances on all criteria) to outputs (in our case, class assignments). We have designed their architecture and adjusted the characteristics of individual units to derive sorting models that are flexible enough to fit the learn-



ing data and sufficiently interpretable due to being inspired by the MCDA methods. This is in line with the recent trend in ML, which postulates making prediction models and their decisions interpretable (Molnar, 2020).

When learning the sorting models, we minimize the loss function defined as an average of regrets for all reference alternatives. The choice of ANNs as a computation technique for conducting preference disaggregation allowed us to use a variety of tools supporting the optimization. In particular, to learn the most favorable value parameters, we employ a variant of a gradient descent optimization algorithm called Adam (Kingma & Ba, 2014). The optimization is enhanced with techniques such as data augmentation to increase the noise resistance, regularization to prevent model overfitting, and batch optimization to reduce the impact of the information processing order on the attained results. The networks deriving the parameters of the OWA-, Choquet-, and distance-based models are shallow. However, the ANNs proposed for UTADIS, PROMETHEE, and ELECTRE can be classified as deep learning models (Deng & Yu, 2014) due to many hidden layers and considering different levels related to the data processing (e.g., criteria, alternatives, pairs of alternatives, and assignments). Hidden layers are required to learn complex models inspired by the value- and outranking-based MCDA methods. However, the raw weight values of multiple layers, some of which conduct non-linear transformations of data, are hardly interpretable for the users. Therefore, we ensure that users are exhibited only with the final models of ANN-UTADIS, ANN-PROMETHEE, and ANN-ELECTRE. These models summarize the comprehensive contribution of individual criteria, resulting from the transformations conducted by various layers, activation performed with non-linear activations functions, and normalization to an easily interpretable range of alternatives' scores.

We conduct a thorough experimental verification of the proposed algorithms on a set of benchmark sorting problems. Its results are quantified in terms of two quality measures for different proportions between the sizes of the training and testing sets. The multiplicity of proposed methods allows indicating which model is most appropriate for a given problem. We also compare the obtained results with the performance of the existing preference learning approaches. These include the Choquistic (Tehrani et al., 2012) and logistic (Hosmer, Lemeshow, & Sturdivant, 2000) regression, Kernel Logistic Regression (KLR) with polynomial and Gaussian kernels, rule ensemble (MORE) (Dembczyński et al., 2009) and tree induction (LMT) (Landwehr, Hall, & Frank, 2003) algorithms, value-based UTADIS model (Zopounidis & Doumpos, 2000), and outranking-based methods incorporating mathematical programming (MIP) (Leroy, Mousseau, & Pirlot, 2011) or a dedicated meta-heuristic (META) (Sobrie et al., 2019).

The remainder of the paper is organized in the following way. Section 2 reminds a threshold-based sorting procedure. In Section 3, we discuss the novel preference learning algorithms that incorporate different scores for judging a comprehensive quality of alternatives. Section 4 provides details of the employed optimization techniques. In Section 5, we illustrate the use of the proposed methods on a selected multiple criteria sorting problem for which a large set of assignment examples is available. Section 6 discusses the results of computational experiments, comparing the predictive capabilities of our ANN-based approaches and the state-of-the-art methods. The last section concludes and provides avenues for future research.

## 2. Threshold-based score-driven multiple criteria sorting

The following notation is used in the paper:

- $A = \{a_1, a_2, \dots, a_i, \dots, a_n\}$  – a finite set of  $n$  alternatives;

- $A^R = \{a_1^*, a_2^*, \dots, a_i^*, \dots\} \subseteq A$  – a finite set of reference alternatives, which the DM accepts to critically judge in a holistic way;
- $G = \{g_1, g_2, \dots, g_j, \dots, g_m\}$  – a finite set of  $m$  evaluation criteria,  $g_j : A \rightarrow \mathbb{R}$  for all  $j \in J = \{1, \dots, m\}$ ;
- $X_j = \{x_j \in \mathbb{R} : g_j(a_i) = x_j, a_i \in A\}$  – a set of all different performances on  $g_j$ ,  $j \in J$ ; as typical in the field of preference learning, we assume that all performances on  $g_j$ ,  $j = 1, \dots, m$ , are scaled to the  $[0,1]$  interval;
- $x_j^1, x_j^2, \dots, x_j^{n_j(A)}$  – increasingly ordered values of  $X_j$ ,  $x_j^k < x_j^{k+1}$ ,  $k = 1, 2, \dots, n_j(A) - 1$ , where  $n_j(A) = |X_j|$  and  $n_j(A) \leq n$ ;
- $C_1, C_2, \dots, C_p$  –  $p$  pre-defined, preference ordered classes, where  $C_{h+1}$  is preferred to  $C_h$ ,  $h = 1, \dots, p - 1$  ( $H = \{1, \dots, p\}$ ).

We consider the problem of sorting imposed by the use of function  $f : R^m \rightarrow H$  that maps alternative  $a_i \in A$  evaluated in terms of  $m$  criteria to one of the decision classes  $C_h$ ,  $h = \{1, \dots, p\}$ . To aggregate performances on multiple criteria, we use a function assigning a comprehensive score  $Sc(a_i)$  to  $a_i \in A$ . The maximal score is assigned to an ideal alternative  $a^+$  with the most preferred performances on all criteria, whereas the minimal score is associated with an anti-ideal alternative  $a^-$ . The range  $[Sc(a^-), Sc(a^+)]$  may differ depending on the applied method. Moreover, the scale of a comprehensive score is divided by a set of class thresholds  $t_h$ ,  $h = 1, \dots, p - 1$ , which delimit the intervals implying an assignment to particular decision classes (Köksalan & Özpeynirci, 2009):

$$\begin{aligned} Sc(a_i) < t_1 &\Rightarrow a_i \in C_1, \\ t_{h-1} \leq Sc(a_i) < t_h &\Rightarrow a_i \in C_h, \quad \text{for } h = 2, \dots, p - 1, \\ Sc(a_i) \geq t_{p-1} &\Rightarrow a_i \in C_p. \end{aligned} \tag{1}$$

To avoid direct specification of the parameter values, we assume indirect preference information is available or specified by the DM. It has the form of desired class assignments  $C_{DM}(a_i^*)$  for reference alternatives  $a_i^* \in A^R$ . When constructing or training the sorting model, we will disaggregate holistic preferences to respect the reference assignments in the following way (Doumpos & Zopounidis, 2004):

$$\left. \begin{aligned} &\text{for all } a_i^* \in A^R : \\ Sc(a_i^*) &\geq t_{C_{DM}(a_i^*)-1}, \quad \text{if } C_{DM}(a_i^*) > 1, \\ Sc(a_i^*) + \epsilon &\leq t_{C_{DM}(a_i^*)}, \quad \text{if } C_{DM}(a_i^*) < p, \end{aligned} \right\} \tag{2}$$

where  $\epsilon$  is an arbitrarily small positive value. When numerous assignment examples are considered, they might not be reproduced simultaneously. Therefore, in the optimization phase, we will consider the following loss function defined as an average of regrets for all reference alternatives:

$$\text{Minimize : } loss = \frac{1}{|A^R|} \sum_{a_i^* \in A^R} regret(a_i^*), \tag{3}$$

where  $regret$  is equal to the distance from thresholds delimiting the desired class in case an alternative is misclassified or to zero, otherwise:

$$regret(a_i^*) = \max\{t_{C_{DM}(a_i^*)-1} - Sc(a_i^*), Sc(a_i^*) - t_{C_{DM}(a_i^*)}, 0\}. \tag{4}$$

In the following section, we discuss a variety of scoring procedures that will be incorporated in the ANN-based preference learning algorithms. For each of them, the scoring function  $Sc$  is defined differently.

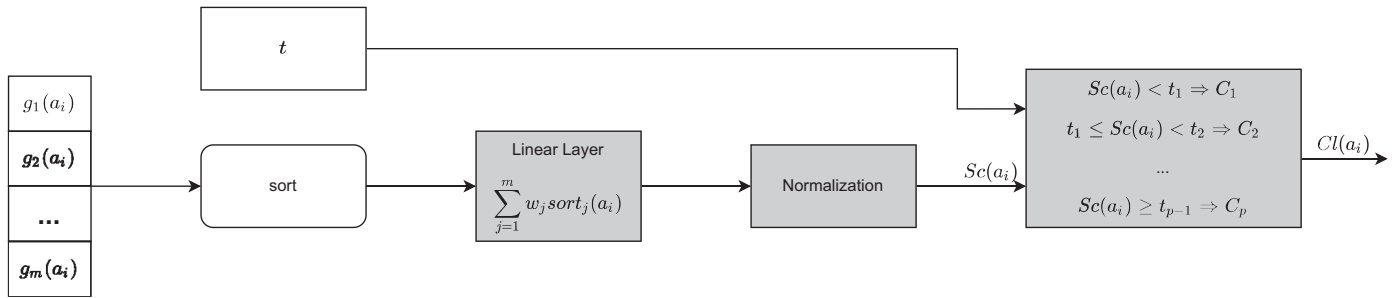


Fig. 1. The architecture of the neural network employed by the ANN-OWA method.

### 3. Preference learning with artificial neural networks and MCDA-inspired preference models

In this section, we present the MCDA-inspired approaches that learn parameters of the sorting models from large sets of assignment examples. For this purpose, they apply Artificial Neural Networks. We will discuss a variety of methods that implement different strategies for deriving the comprehensive scores of alternatives. Nonetheless, for all of them, the derived model remains easily interpretable, and the sorting results are explainable for a human DM.

#### 3.1. ANN-OWA: preference learning with ordered weighted average and ANN

OWA is an aggregation function generalizing other operators such as min, max, average, median, or sum (Yager, 1988). It aggregates performances using a revised weighted sum:

$$OWA(a_i) = \sum_{j=1}^m w_j \text{sort}_j(a_i), \tag{5}$$

where  $\text{sort}_j(a_i)$  is the  $j$ -th largest performance of alternative  $a_i$  on any criterion and  $w_j$  is the weight linked with the  $j$ -th position in sorted performance vector of  $a_i$ . We assume that  $w_j \in \mathbb{R}_{\geq 0}$ .

**ANN-OWA** starts with sorting the performances of each alternative in a non-increasing order (see Fig. 1). Then, a single linear layer aggregates the performances using the OWA operator with non-negative weights. Since the value of OWA can be, in general, arbitrarily large, to increase interpretability of the results, we apply normalization to the  $[0,1]$  range by dividing the scores by the sum of weights  $w_j$ :

$$Sc_{ANN-OWA}(a_i) = \frac{\sum_{j=1}^m w_j \text{sort}_j(a_i)}{\sum_{j=1}^m w_j}. \tag{6}$$

Such a score is compared against the thresholds  $t = [t_1, t_2, \dots, t_{p-1}]$  to determine the class assignments using Eq. (1) and calculate the regret that is considered when optimizing the network parameters, i.e., weights  $w_j$  and thresholds  $t$ .

The last component of the ANN responsible for the comparison of a comprehensive score with class thresholds to derive the assignment is the same for all methods presented in the following subsections. Thus, we will not mention it when describing these approaches, instead focussing on the computation of scores in line with the assumptions of different methods. Nevertheless, the thresholds and the underlying sorting procedure will always be depicted in the figures representing the architectures of neural networks.

#### 3.2. ANN-Ch: preference learning with the Choquet integral and ANN

The Choquet integral model is an additive aggregation method, dealing with interactions between criteria (Angilella et al., 2013).

It takes the form of a weighted sum over all subsets of criteria  $T \subseteq G$ , where the performance for  $T$  is the minimum over the performances on criteria contained in  $T$ :

$$Ch_{\mu}(a_i) = \sum_{T \subseteq G} w_T \cdot \min_{j \in T} g_j(a_i), \tag{7}$$

where  $\sum_{T \subseteq G} w_T = 1$ . We limit the considered interactions to pairs of criteria by referring to the 2-additive Möbius transform (Tehrani et al., 2012):

$$Ch_{\mu,2}(a_i) = \sum_{j=1}^m w_j g_j(a_i) + \sum_{\{j,l\} \subseteq G} w_{\{j,l\}} \min(g_j(a_i), g_l(a_i)). \tag{8}$$

To respect the pre-defined preference directions for all criteria, we assume that the weights are non-negative:

$$w_j \geq 0, \forall j \in \{1, \dots, m\}. \tag{9}$$

Moreover, we consider the positive and negative interactions, though limiting their impact on the attained scores in the following way:

$$w_{\{j,l\}} + w_j \geq 0, \forall j \in \{1, \dots, m\}, \forall l \in \{1, \dots, m\} \setminus \{j\}. \tag{10}$$

The variant of the method respecting such constraints will be denoted as **ANN-Ch-Constr**. In the pre-processing phase, we perform the Möbius transform of a 2-order additive measure of the input data (see Fig. 2). Then, two linear layers are responsible for aggregating pre-criteria performances using non-negative weights respecting Eq. (9) and interaction components using weights associated with pairs of criteria that respect Eq. (10). Their outputs are summed and normalized to the  $[0,1]$  range as follows:

$$Sc_{ANN-Ch-Constr}(a_i) = \frac{Ch_{\mu,2}(a_i)}{\sum_{j=1}^m w_j + \sum_{\{j,l\} \subseteq G} w_{\{j,l\}}}. \tag{11}$$

The parameters optimized by ANN are weights of both linear layers ( $w_j$  and  $w_{j,l}$ ) and class thresholds  $t$ .

The other two variants of the Choquet integral-based method incorporate different assumptions. The first one, called **ANN-Ch-Pos.**, considers only positive interactions, hence limiting the weights for individual criteria and pairs to non-negative values. The other variant, called **ANN-Ch-Uncons.**, does not impose any constraints on the weights. Moreover, both variants apply normalization of scores with the sigmoid function as proposed in Tehrani et al. (2012):

$$Sc_{ANN-Ch-Sig}(a_i) = \text{sigmoid}(Ch_{\mu,2}(a_i) + \text{bias}). \tag{12}$$

A diagram showing the network operations for these variants is presented in Fig. 3. First, we perform the Möbius transform. Since there are no constraints involving different weights, per-criteria performances and interaction components can be aggregated using a single linear layer. It performs the calculations defined by Eq. (8) and adds a bias value as defined by Eq. (12). The bias allows the sigmoid function to be shifted, and the lack of restrictions on the sum of weights allows for an arbitrary adjustment of



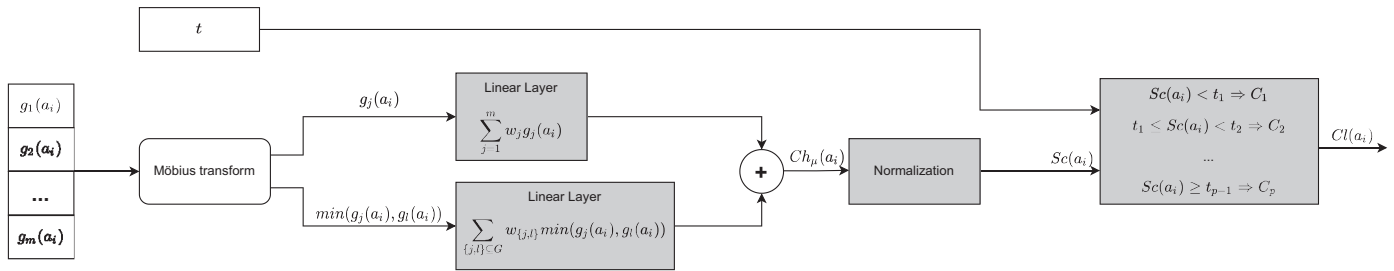


Fig. 2. The architecture of the neural network employed by the ANN-Ch-Constr. method.

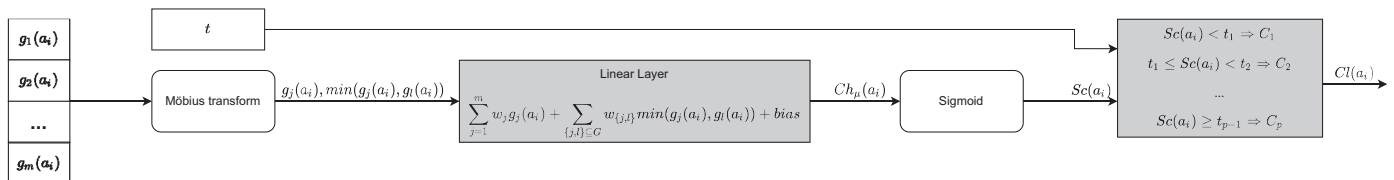


Fig. 3. The architecture of the neural network employed by the ANN-Ch-Pos. and ANN-Ch-Uncons. methods.

the sigmoid function's argument scale. In **ANN-Ch-Pos.**, all weights need to be non-negative. The result from the linear layer is processed by a sigmoid activation function. It ensures that the score for each alternative is in the [0,1] range.

3.3. ANN-TOPSIS: preference learning with TOPSIS and ANN

Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) considers the ideal  $a^+$  and anti-ideal  $a^-$  alternatives with the following performances on each criterion  $g_j \in G$  (Hwang & Yoon, 1981):

$$g_j(a^+) = \max_{a_i \in A}(g_j(a_i)) \text{ and } g_j(a^-) = \min_{a_i \in A}(g_j(a_i)). \quad (13)$$

The closer alternative  $a_i \in A$  is to  $a^+$  and the further it is from  $a^-$ , the more preferred it is. The respective distances can be computed as follows:

$$d^+(a_i) = \left( \sum_{j=1}^n w_j y_j^+(a_i) \right)^{\frac{1}{z}} \text{ and } d^-(a_i) = \left( \sum_{j=1}^n w_j y_j^-(a_i) \right)^{\frac{1}{z}}, \quad (14)$$

where  $w_j = |w_j|^z$ ,  $w_j \in \mathbb{R}_{\geq 0}$  is the weight associated with criterion  $g_j \in G$ ,  $y_j^+(a_i) = |g_j(a^+) - g_j(a_i)|^z$  and  $y_j^-(a_i) = |g_j(a_i) - g_j(a^-)|^z$ , for  $j = 1, \dots, m$ . Overall, the comprehensive score for  $a_i$  is computed in the following way:

$$R(a_i) = \frac{d^-(a_i)}{d^-(a_i) + d^+(a_i)}. \quad (15)$$

In this paper, we assume  $z = 1$ . Thus  $w_j$  can be interpreted as the weight of criterion  $g_j$  without any additional transformations.

The architecture of the neural network performing the respective calculations for **ANN-TOPSIS** is presented in Fig. 4. In the pre-processing stage, we compute  $y_j^+(a_i)$  and  $y_j^-(a_i)$  values for each alternative  $a_i \in A$ . The linear layer calculates the distances from the ideal and anti-ideal alternatives while using non-negative weights  $w_j$ . It is followed by aggregation according to Eq. (15). The parameters subject to optimization are weights  $w_j$  and class thresholds  $t$ . The neural networks for ANN-OWA, all variants of ANN-Ch, and ANN-TOPSIS share the same number of layers, including one input layer, one hidden layer, and one output layer responsible for sorting.

3.4. Modelling monotonic functions with ANNs

To construct ANNs suitable for conducting calculations of more complex MCDA methods, it is necessary to define a monotonic

function. It can be seen as transforming per-criteria performances or performance differences, maintaining the pre-defined preference directions. We consider two monotonic functions: non-decreasing and non-increasing for gain- and cost-type criteria, respectively. The transformation of a function from non-decreasing to non-increasing is conducted by negating the function. We define a non-decreasing function as a neural network with a single hidden layer and a continuous sigmoidal activation function with positive weights. According to Cybenko (1989), for an arbitrary continuous sigmoid function  $\sigma$ , function  $u(\mathbf{x})$  of vector  $\mathbf{x} \in \mathbb{R}^N$ :

$$u(\mathbf{x}) = \sum_{k=1}^L \alpha_k \sigma(y_k^T \mathbf{x} + \theta_k), \quad (16)$$

where  $\alpha_k, \theta_k \in \mathbb{R}$  and  $y_k \in \mathbb{R}^N$ , can approximate any  $N$ -dimensional continuous function with precision depending on the number of components  $L$ . Also,  $u(\mathbf{x})$  is equivalent to a neural network with a single hidden layer (Cybenko, 1989).

In what follows, we build on the following two observations. On the one hand, if  $F$  is a family of monotonic functions, then  $\sum_{f(x) \in F} f(x)$  is also a monotonic function. On the other hand, the linear transformation  $\alpha f(x) + \beta$  of a monotonic function  $f$ , where  $\alpha \in \mathbb{R}_{\geq 0}$  and  $\beta \in \mathbb{R}$ , is a monotonic function. Assuming  $\alpha_k \in \mathbb{R}_{\geq 0}$ ,  $y_k \in \mathbb{R}_{\geq 0}^N$ ,  $\theta_k \in \mathbb{R}$ , and  $\sigma$  is a monotonic continuous sigmoidal function, then  $u(\mathbf{x})$  is also a monotonic function. The values of  $\alpha_k$ ,  $y_k$ , and  $\theta_k$  will be optimized using an algorithm described in Section 4 by iteratively refining parameter values with function gradients. The major monotonic continuous sigmoidal functions are sigmoid and hard sigmoid functions. However, to avoid a problem of gradient vanishing, in the learning process, we will consider the non-decreasing monotonic function *LeakyHardSigmoid* (see Fig. 5):

$$\text{Leaky Hard Sigmoid}(x) = \begin{cases} \delta x, & \text{if } x < 0, \\ x, & \text{if } 0 \leq x \leq 1, \\ \delta(x - 1) + 1, & \text{if } x > 1, \end{cases} \quad (17)$$

where  $\delta$  is a slope factor, being a very small value in the range [0,1). The above function is not a continuous sigmoidal function and cannot be used to approximate any non-decreasing monotonic function. For example, it cannot represent the level segments. However, it is possible to decrease the value of a slope during training to zero. Then, *LeakyHardSigmoid* will be equal to hard sigmoid function. We will consider a one-dimensional space of  $x$  and

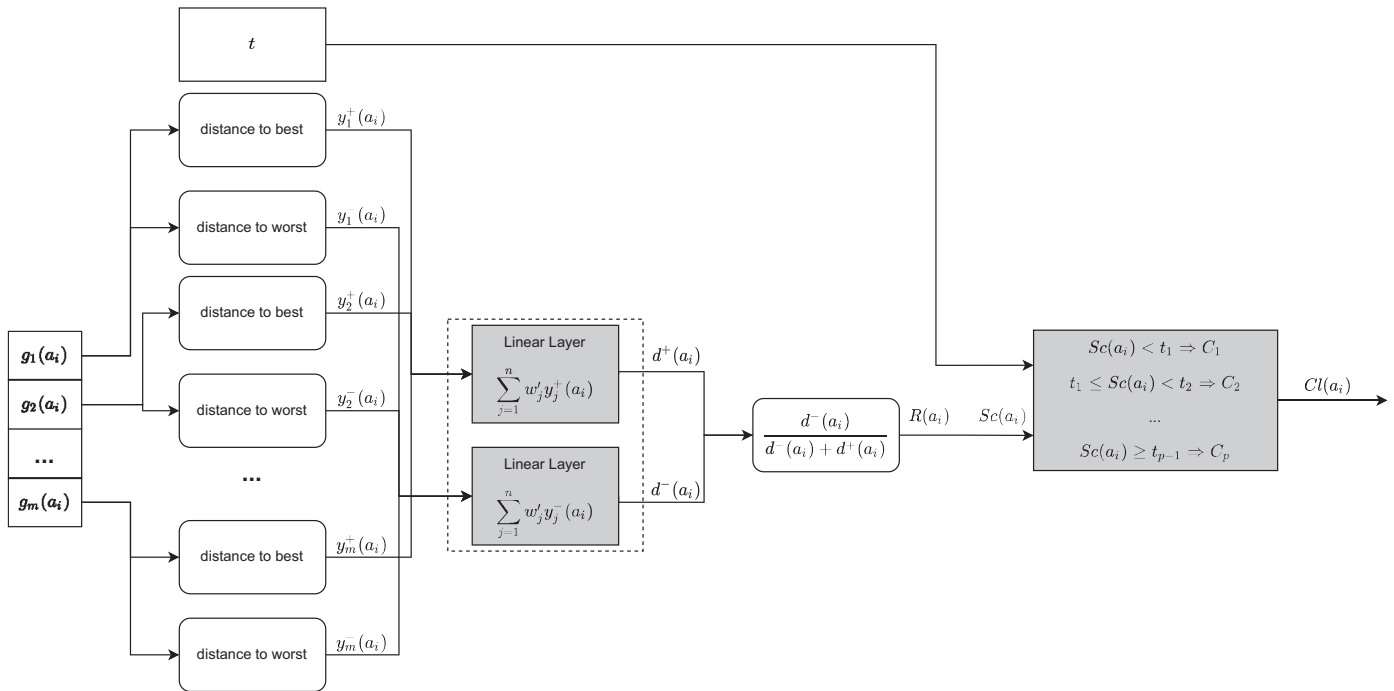


Fig. 4. The architecture of the neural network employed by the ANN-TOPSIS method.

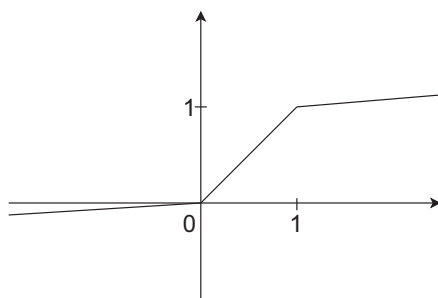


Fig. 5. The LeakyHardSigmoid function.

y. Thus, for the sake of simplicity, we assume that:

$$u(x) = \sum_{k=1}^L \alpha_k \sigma(y_k x + \theta_k), \tag{18}$$

where  $\sigma$  is LeakyHardSigmoid with a slope 0.01,  $L$  is the number of components of function  $u$ , and  $y_k, \alpha_k \in \mathbb{R}_{\geq 0}$ . Function  $u(x)$  can be considered as a line segment function with ends designated by the individual components. It changes slope only at the characteristic points resulting from the applied  $\sigma$  function. Each component has two characteristic points:  $(-\frac{\theta_k}{y_k}, 0)$  and  $(\frac{-\theta_k+1}{y_k}, \alpha_k)$ , which are projected onto  $u(x)$ . Such a projection from the component functions on the output model for a single argument  $x$  is presented in Fig. 6. Function  $u(x)$  is marked with a solid line resulting from the combination of three components marked with dashed lines. The transformation conducted by Monotonic Block is general, not imposing the limits on the ranges of its output values. This means that, in particular,  $u_j(0) \in \mathbb{R}$  and  $u_j(1) \in \mathbb{R}_{\geq 0}$ . To ensure that the results are interpretable, subsequent normalization to the desired range, e.g.,  $[0, 1]$ , is needed.

Function  $u(x)$ , defined by Eq. (18), can be presented as a neural network with a single input value  $x$ . This value is copied  $L$  times and passed as the input to the linear layers, where it is scaled by weights  $y_k$  and shifted by bias  $\theta_k$ . Then, the output from the input

layer is transformed by the LeakyHardSigmoid function and passed to the next linear layer. It must be ensured that the weights in all layers are greater than zero to maintain the function's monotonicity. The weights  $\alpha_k$  are initialized with positive values. If during training some value falls below  $\varepsilon$  being an arbitrarily small positive value, it is set to  $\varepsilon$ . In what follows, we will refer to the network representing function  $u(x)$  as Monotonic Block (see Fig. 7). It will be used as a component of the three preference learning methods that are presented in the following subsections.

### 3.5. ANN-UTADIS: preference learning with UTADIS and ANN

UTADIS is a preference disaggregation method that quantifies a comprehensive quality of each alternative using an additive value function (Zopounidis & Doumpos, 2000):

$$U(a_i) = \sum_{j=1}^m w_j u_j(g_j(a_i)), \tag{19}$$

where  $u_j \in [0, 1]$  is a marginal value function and  $w_j$  is a weight associated with criterion  $g_j$ . Function  $U(a_i)$  takes values in the  $[0,1]$  range, delimited by  $U(a^-) = 0$  and  $U(a^+) = 1$  for anti-ideal and ideal alternatives, respectively. In UTADIS,  $u_j$  is piecewise linear with  $n_j(A)$  pre-defined characteristic points  $x_j^k$  such that:

$$u_j(x_j^k) \leq u_j(x_j^{k+1}), \forall k \in \{1, \dots, n_j(A) - 1\}, \text{ and } \forall j \in \{1, \dots, m\}. \tag{20}$$

The marginal values between these points are computed using linear interpolation. In UTADIS, the marginal values  $u_j(x_j^k)$  in the characteristic points and weights  $w_j$  are determined using mathematical programming based on a set of assignment examples (Zopounidis & Doumpos, 2000). In turn, we will employ ANN for deriving weights and the shape of marginal value functions without having to specify characteristic points. In this way, the method offers greater flexibility in fitting the learning data.

The neural network used by ANN-UTADIS is shown in Fig. 8. The performance on each criterion is transformed using Monotonic

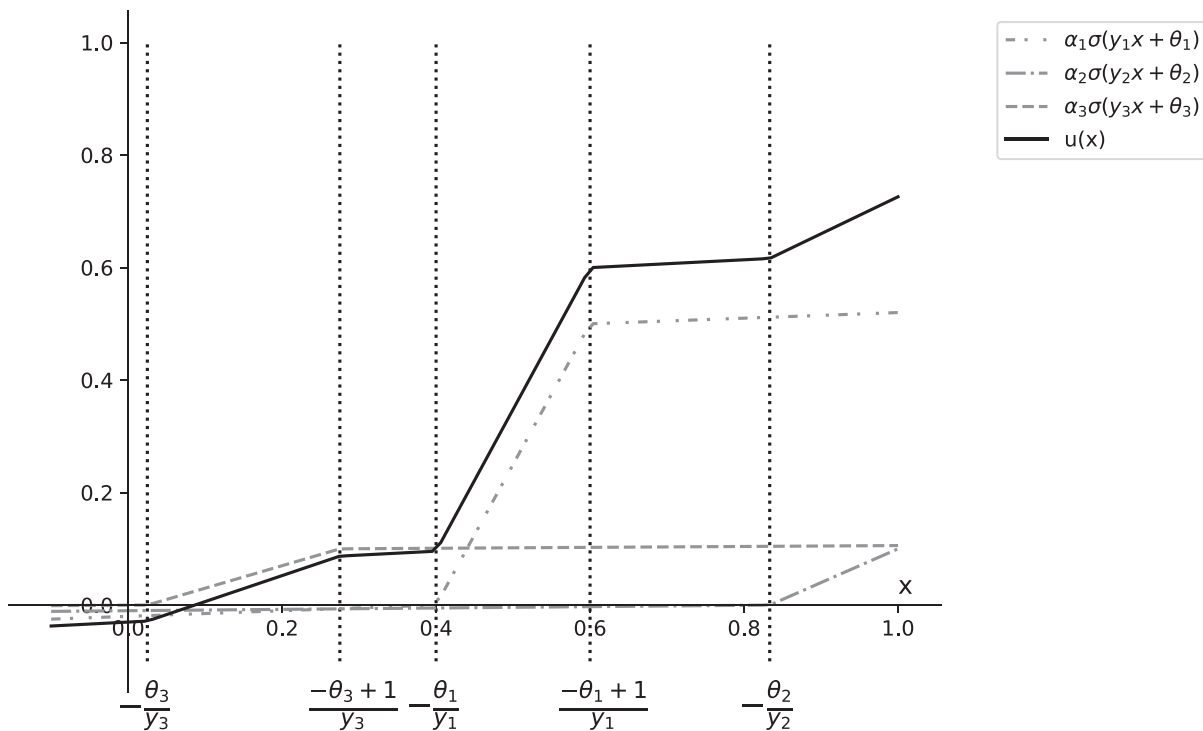


Fig. 6. Function  $u(x)$  representing the transformation conducted by the *Monotonic Block* with three ( $L = 3$ ) components.

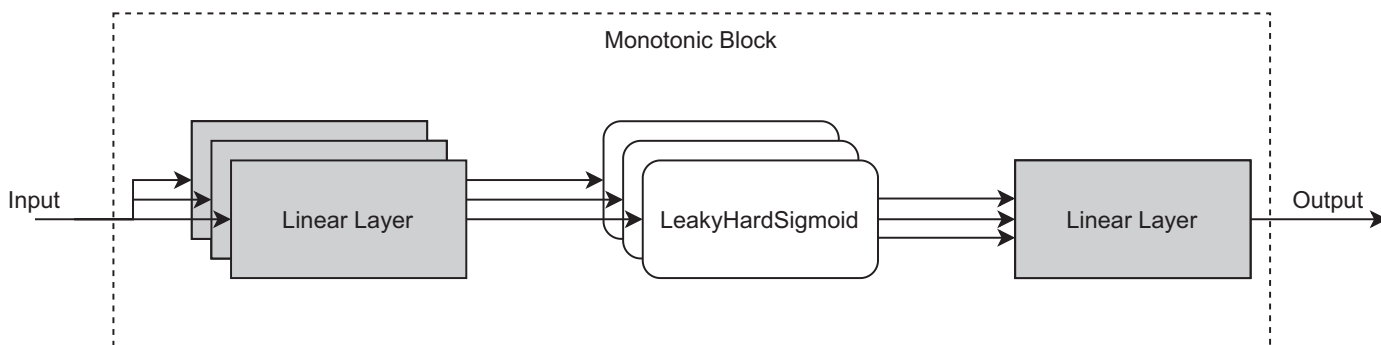


Fig. 7. The *Monotonic Block* used in the preference learning algorithms based on ANNs.

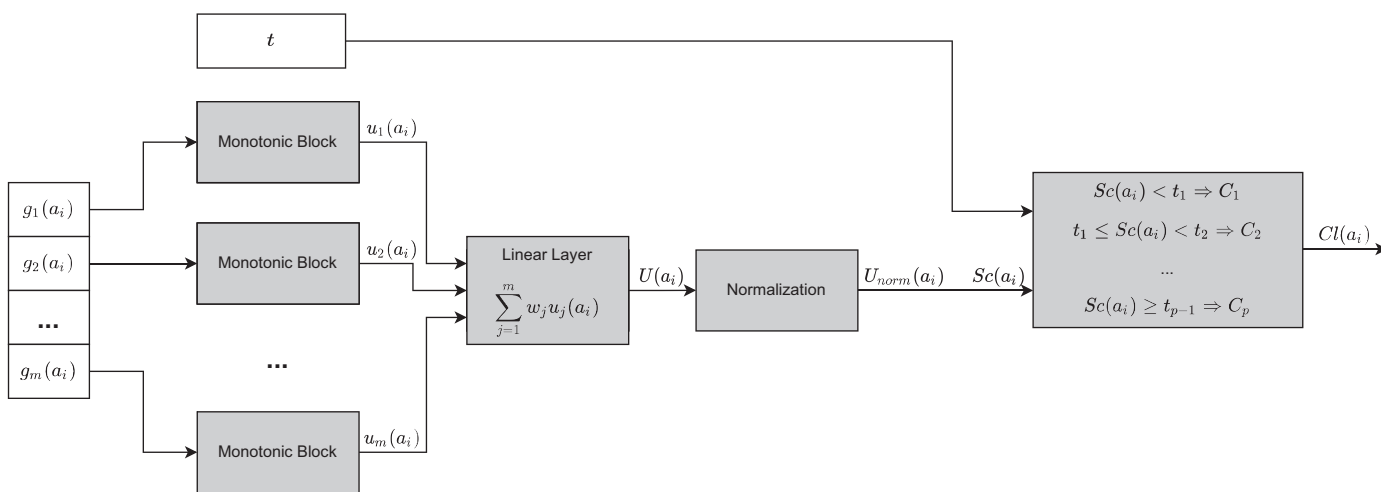


Fig. 8. The architecture of the neural network employed by the ANN-UTADIS method.

Blocks according to Eq. (18), adhering to the monotonicity constraint. To use each *Monotonic Block*, it is required to provide the number  $L$  of components. It constrains the maximum number of breakpoints for the resulting function (note that some components may become inactive during optimization, i.e., when  $\alpha_k = 0$ ). Then, the per-criterion marginal values are aggregated into a comprehensive value in line with Eq. (19) by a linear layer. Its weights  $w_j$  are constrained to positive values to preserve the pre-defined preference directions. For the sake of normalization, we apply the min-max scaling of comprehensive scores:

$$S_{\text{ANN-UTADIS}}(a_i) = \frac{U(a_i) - U(a^-)}{U(a^+) - U(a^-)}. \tag{21}$$

The neural network used by ANN-UTADIS optimizes weights  $w_j$ , class thresholds  $t$ , and parameters incorporated in the *Monotonic Blocks*. In general, a marginal value function for each criterion may be modeled with a different number  $L$  of components. However, we will use the same value of  $L$  for all criteria, which allows for a more straightforward parallelization of calculations in Eq. (18) by operations on tensors rather than on individual scalars. Overall, the network for ANN-UTADIS involves one input layer, three hidden layers, and one output layer.

### 3.6. ANN-PROMETHEE: preference learning with PROMETHEE and ANN

The PROMETHEE method aggregates the results of pairwise comparisons of each alternative against all remaining ones into a comprehensive measure of desirability (Brans & De Smet, 2016). For each pair  $(a_i, a_k) \in A \times A$  and each criterion  $g_j \in G$ , the marginal preference degree is computed as follows:

$$F_j(a_i, a_k) = P_j(d_j(a_i, a_k)), \tag{22}$$

where  $P_j$  is a marginal preference function and  $d_j(a_i, a_k) = g_j(a_i) - g_j(a_k)$  is the performance difference. In PROMETHEE, six pre-defined types of  $P_j$  are considered. However, the most commonly used is the following:

$$F_j(a_i, a_k) = \begin{cases} 0, & \text{if } d_j(a_i, a_k) \leq q_j, \\ \frac{d_j(a_i, a_k) - q_j}{p_j - q_j}, & \text{if } 0 < d_j(a_i, a_k) \leq p_j, \\ 1, & \text{if } d_j(a_i, a_k) > p_j, \end{cases} \tag{23}$$

where  $q_j$  is an indifference threshold defining the maximal performance difference that is negligible and  $p_j$  is a preference threshold specifying the minimal performance difference justifying a strict preference. All preference functions in PROMETHEE are non-decreasing. Also, they are normalized so that  $F_j(a_i, a_k) = 0$  for  $d_j(a_i, a_k) \leq 0$  and their largest value is one. The function type and the respective parameter values for each criterion need to be provided by the DM. The outcomes from the individual criteria are aggregated into a comprehensive preference index  $\pi(a_i, a_k)$  using a weighted sum:

$$\pi(a_i, a_k) = \sum_{j=1}^m w_j F_j(a_i, a_k), \tag{24}$$

where  $w_j \geq 0$  is a weight associated with criterion  $g_j$  and  $\sum_{j=1}^m w_j = 1$ . As a result,  $\pi(a_i, a_i) = 0$ ,  $a_i \in A$  and  $\pi(a^+, a^-) = 1$ , where  $a^+$  and  $a^-$  are the ideal and anti-ideal alternatives. Such preference degrees are further aggregated into the positive  $NFS^+(a_i)$  and negative  $NFS^-(a_i)$  flows, using the NFS procedure:

$$\begin{aligned} NFS^+(a_i) &= \frac{1}{n-1} \sum_{k=1}^n \pi(a_i, a_k) \text{ and} \\ NFS^-(a_i) &= \frac{1}{n-1} \sum_{k=1}^n \pi(a_k, a_i). \end{aligned} \tag{25}$$

The arguments in favour and against alternative  $a_i$  are finally aggregated into a comprehensive flow:

$$NFS(a_i) = NFS^+(a_i) - NFS^-(a_i). \tag{26}$$

In the proposed ANN-PROMETHEE method, we use monotonic marginal preference functions that are automatically adjusted to the training data, not requiring the specification of type, weights, or comparison thresholds. The architecture of the underlying neural network is presented in Fig. 9. Following the assumptions of PROMETHEE, we first compute the performance differences  $d_j(a_i, a_k)$  on each criterion. The negative differences are clipped to zero via the ReLU function:

$$ReLU(x) = \max(x, 0). \tag{27}$$

In this way, the non-positive performance differences will be assigned the same value of the preference index. The values of marginal preference functions  $F_j$  are computed using the *Monotonic Block* which ensures both monotonicity and flexibility of shape adjustment:

$$F_j(a_i, a_k) = u_j(\max(d_j(a_i, a_k), 0)). \tag{28}$$

The marginal preference degrees are aggregated into a comprehensive preference index using a linear layer with non-negative weights. Since weights and parameters of the *Monotonic Block* are not constrained from the top, we normalize the comprehensive indices as follows:

$$\pi_{\text{norm}}(a_i, a_k) = \frac{\pi(a_i, a_k) - \pi(a^-, a^-)}{\pi(a^+, a^+) - \pi(a^-, a^-)}. \tag{29}$$

Then, the outcomes of pairwise comparisons are aggregated over all alternatives into positive, negative, and comprehensive flows using the Net Flow Score procedure:

$$\begin{aligned} S_{\text{ANN-PROMETHEE}}(a_i) &= NFS^+(a_i) - NFS^-(a_i) \\ &= \frac{1}{n-1} \left[ \sum_{k=1}^n \pi_{\text{norm}}(a_i, a_k) - \pi_{\text{norm}}(a_k, a_i) \right]. \end{aligned} \tag{30}$$

The use of NFS implies that the preference degrees for all pairs of alternatives need to be computed in a batch. Moreover, similar to the ANN-UTADIS, ANN-PROMETHEE requires specification of the number of components for each *Monotonic Block*. However, it is recommended to use the same number  $L$  for all such blocks. Overall, the network for ANN-PROMETHEE involves one input layer, four hidden layers, and one output layer.

### 3.7. ANN-ELECTRE: preference learning with ELECTRE and ANN

The ELECTRE method compares the alternatives pairwise through an outranking relation (Figueira et al., 2013). In what follows, we discuss its adaptation for scoring the alternatives based on aggregating the sufficiently great outranking credibilities using the NFS procedure. We will consider two tests to compute the credibility for pair  $(a_i, a_k) \in A \times A$ . The concordance test quantifies the arguments in favor of  $a_i$  being at least as good as  $a_k$ . The marginal concordance index for criterion  $g_j$  is computed as follows:

$$c_j(a_i, a_k) = \begin{cases} 1, & \text{if } g_j(a_i) \geq g_j(a_k) - q_j, \\ \frac{g_j(a_i) + p_j - g_j(a_k)}{p_j - q_j}, & \text{if } g_j(a_i) < g_j(a_k) - q_j \\ & \text{and } g_j(a_i) \geq g_j(a_k) - p_j, \\ 0, & \text{if } g_j(a_i) < g_j(a_k) - p_j, \end{cases} \tag{31}$$

where  $q_j$  and  $p_j$  are, respectively, indifference and preference thresholds. Whichever the threshold values,  $c_j(a_i, a_k) = 1$  for  $g(a_i) \geq g(a_k)$ . Moreover,  $c_j(a_i, a_k)$  is a monotonic and piecewise

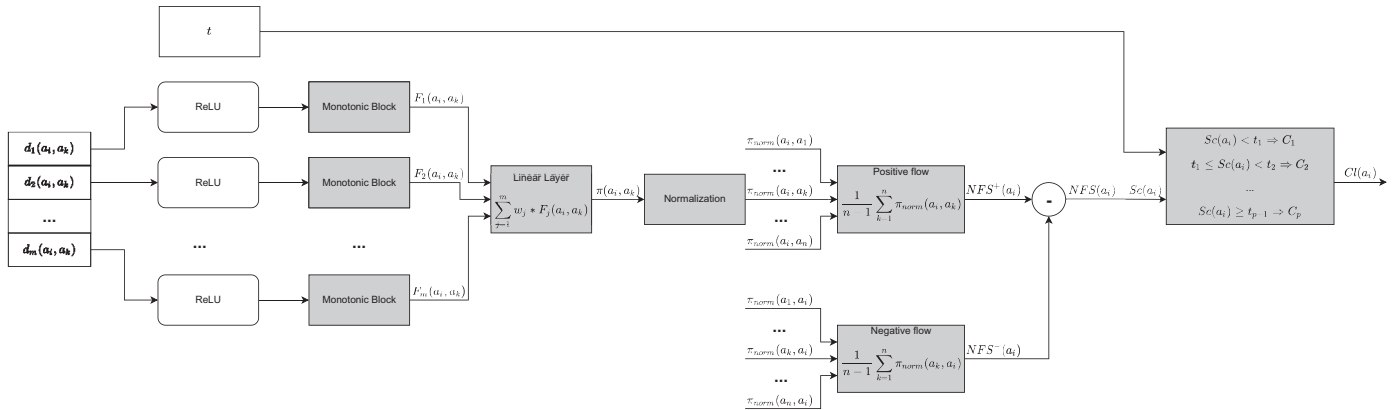


Fig. 9. The architecture of the neural network employed by the ANN-PROMETHEE method.

linear function. The per-criteria results are aggregated into a comprehensive concordance index  $C(a_i, a_k)$  using a weighted sum:

$$C(a_i, a_k) = \sum_{j=1}^m w_j c_j(a_i, a_k), \tag{32}$$

where  $w_j$  is a weight associated with  $g_j$  and  $\sum_{j=1}^m w_j = 1$ . Index  $C(a_i, a_k)$  is interpreted as the strength of the coalition of criteria supporting the outranking. In turn, the discordance test verifies the strength of arguments against the outranking. In particular, a marginal discordance index is defined as follows:

$$D_j(a_i, a_k) = \begin{cases} 1, & \text{if } g_j(a_i) \leq g_j(a_k) - v_j, \\ \frac{g_j(a_k) - p_j - g_j(a_i)}{v_j - p_j}, & \text{if } g_j(a_i) > g_j(a_k) - v_j \\ & \text{and } g_j(a_i) \leq g_j(a_k) - p_j, \\ 0, & \text{if } g_j(a_i) > g_j(a_k) - p_j, \end{cases} \tag{33}$$

where  $v_j$  is a veto threshold interpreted as the minimal performance difference implying a complete discordance. The thresholds need to respect the following constraints:  $0 \leq q_j \leq p_j < v_j$ . Note that the discordance effect does not to be considered for all  $g_j \in G$  because the power to veto against the outranking is usually attributed only to the most important criteria. We consider the aggregation of partial discordances into a comprehensive one using the following function (Mousseau & Dias, 2004):

$$D(a_i, a_k) = 1 - \max_{j=1, \dots, m} D_j(a_i, a_k). \tag{34}$$

Hence the maximal partial discordance over all criteria decides upon the comprehensive strength of arguments against the hypothesis that  $a_i$  outranks  $a_k$ . Finally, the credibility degree is computed by multiplying the comprehensive concordance and discordance:

$$\sigma(a_i, a_k) = C(a_i, a_k) \cdot D(a_i, a_k). \tag{35}$$

Thus the greater the arguments in favor and the lesser the arguments against the outranking, the greater the credibility. To compute the score for each alternative, we will consider only sufficiently great credibilities to avoid compensation between marginal arguments in favor or against  $a_i$  being a favorable alternative. Specifically, we will consider only  $\sigma(a_i, a_k)$  which are at least as good as cutting level  $\lambda$  such that  $0.5 \leq \lambda \leq 1$ . Finally, similar to the PROMETHEE method, we compute the Net Flow Score for each alternative  $a_i \in A$ :

$$\begin{aligned} NFS(a_i) &= NFS^+(a_i) - NFS^-(a_i) \\ &= \frac{1}{n-1} \left[ \sum_{k=1}^n \sigma_{NFS}(a_i, a_k) - \sum_{k=1}^n \sigma_{NFS}(a_k, a_i) \right], \end{aligned} \tag{36}$$

where  $\sigma_{NFS}(a_i, a_k) = \sigma(a_i, a_k) - \lambda$  if  $\sigma(a_i, a_k) \geq \lambda$  and  $\sigma_{NFS}(a_i, a_k) = 0$ , otherwise. Note that other realizations of  $\sigma_{NFS}(a_i, a_k)$  would also be possible. However, we opted for a variant that keeps the spirit of ELECTRE while being intuitively useful in computing comprehensive scores of alternatives via NFS.

In the proposed ANN-ELECTRE, we avoid direct specification of thresholds ( $q_j, p_j$  and  $v_j$ ), weights  $w_j$ , and cutting level  $\lambda$ . In turn, the parameters of an outranking-based sorting model are inferred indirectly using the neural network whose architecture is presented in Fig. 10. In the preprocessing phase, ANN computes the performance differences. Then, the calculations are split into two parts responsible for conducting the concordance and discordance tests. These parts share the value of preference thresholds  $p_j, j = 1, \dots, m$ , to prevent the simultaneous occurrence of concordance and discordance. These thresholds are optimized when training the ANN while ensuring that  $p_j \in [0, 1]$ .

In part responsible for the concordance test, the performance differences are truncated to positive values by the ReLU function (see Eq. (27)), and their order is reversed by subtracting them from one. Since the performances on individual criteria are normalized in the [0,1] range, after the above transformation, we will get one (corresponding to the maximal value of the concordance index) if  $g_j(a_i) \geq g_j(a_k)$ , or a value in the [0,1] range, otherwise. The obtained value is processed by the marginal concordance function  $u_j^c$  implemented by Monotonic Block, allowing for its monotonic and flexible transformation as depicted in Fig. 11(a). The marginal concordance should be zero if the performance difference exceeds the preference threshold  $p_j$ . This can be attained by subtracting the value of  $u_j^c$  attained for  $1 - p_j$ , i.e.,  $u_j^c(1 - p_j)$  from  $u_j^c(1 - ReLU(g_j(a_k) - g_j(a_i)))$ . The resulting difference should be truncated to positive values, e.g., using the ReLU function. However, the lack of a gradient for the negative arguments of this function makes it difficult to optimize values of preference thresholds  $p_j, j = 1, \dots, m$ . For this reason, we use the LeakyReLU function instead which has a non-zero gradient for negative values equal to  $\delta$ :

$$LeakyReLU(x) = \max(x, \delta x), \tag{37}$$

where  $\delta$  is a slope angle for the negative part of the function. It should take a small value and can be minimized to zero during optimization. The result of these operations is shown in Fig. 11(b). Overall, the marginal concordance index  $c_j(a_i, a_k)$  is computed as follows:

$$c_j(a_i, a_k) = LeakyReLU_p(u_j^c(1 - ReLU(g_j(a_k) - g_j(a_i))) - u_j^c(1 - p_j)). \tag{38}$$

Comprehensive concordance index  $C(a_i, a_k)$  is calculated using Eq. (32) by a linear layer that incorporates criteria weights  $w_j \geq 0$ .

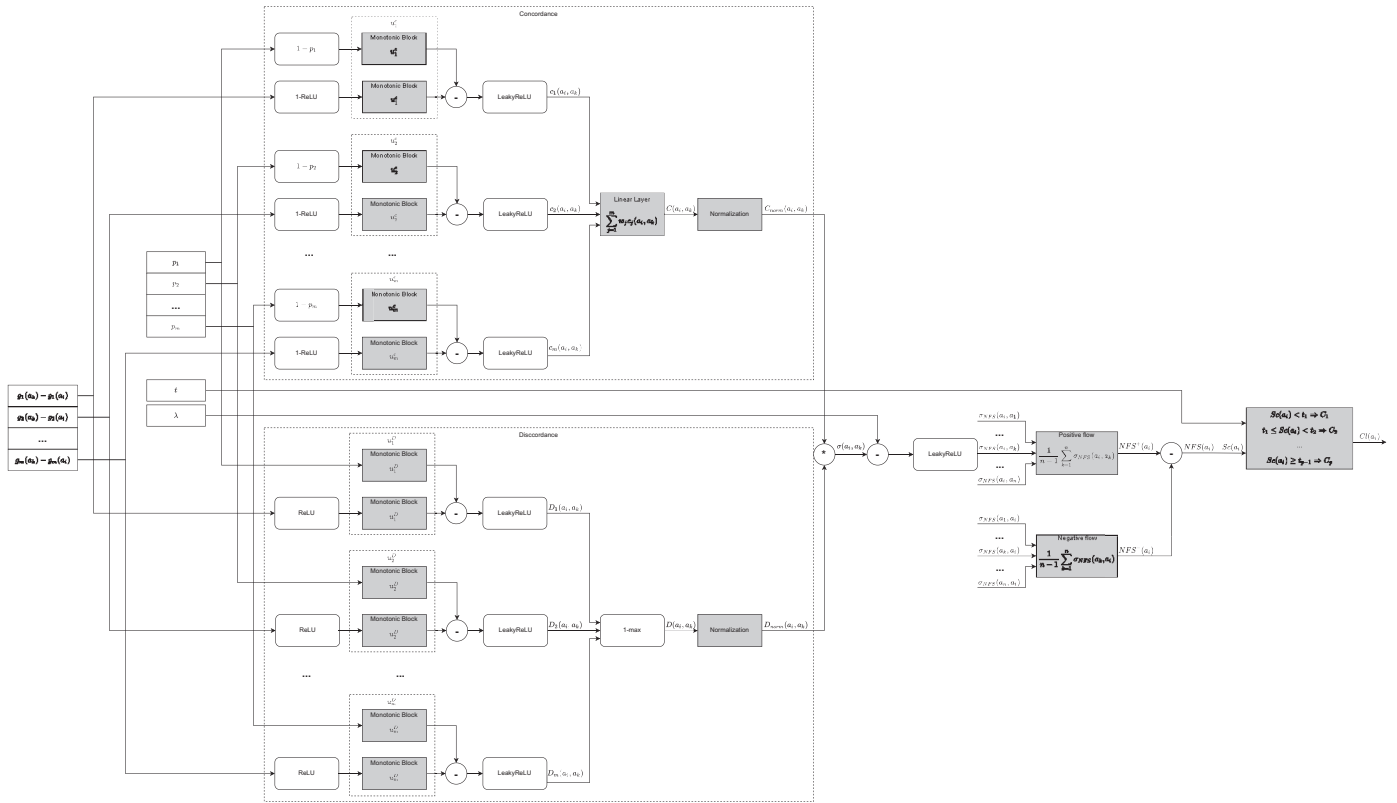


Fig. 10. The architecture of the neural network employed by the ANN-ELECTRE method.

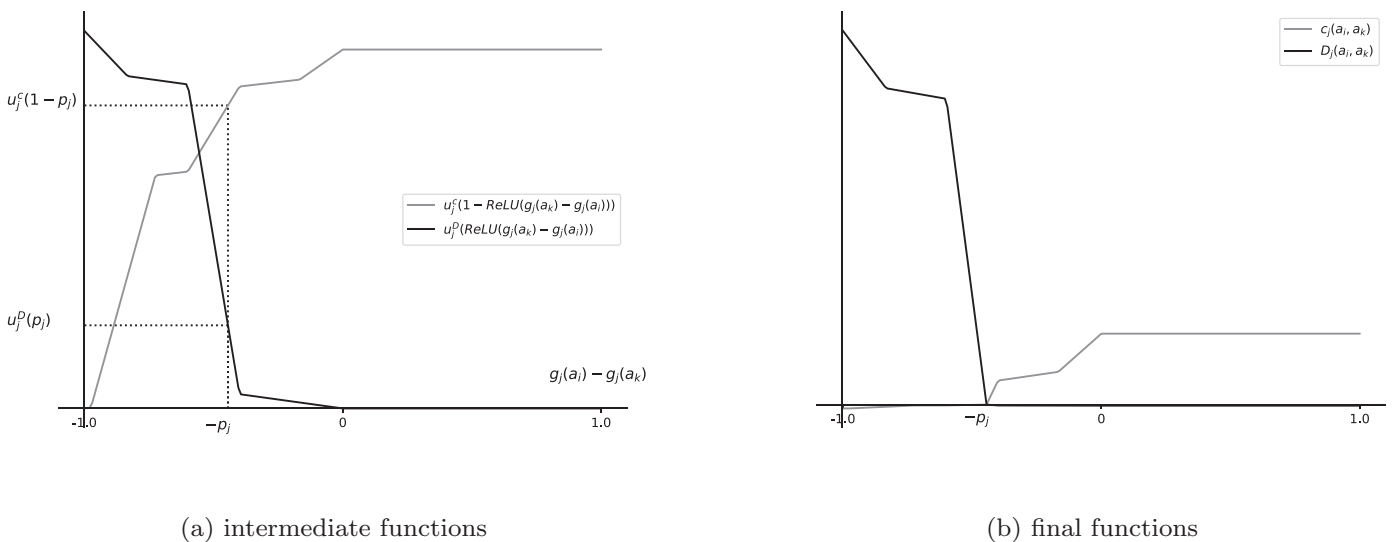


Fig. 11. The marginal concordance and discordance functions for the ANN-ELECTRE method before (a) and after (b) subtracting the value attained for preference threshold  $p_j$  and after transformation by LeakyReLU with  $\delta = 0.01$ .

Finally, values of  $C(a_i, a_k)$  are normalized to the [0,1] range, using the min-max scaling:

$$C_{norm}(a_i, a_k) = \frac{C(a_i, a_k) - C(a^-, a^+)}{C(a^+, a^-) - C(a^-, a^+)}. \tag{39}$$

The other part of the ANN-ELECTRE network is responsible for conducting the discordance test. It first truncates the performance differences to positive values, i.e., these for which  $g_j(a_k) \geq g_j(a_i)$ . Then, the result of such an operation is processed by function  $u_j^D$  modeled by the *Monotonic Block* to obtain marginal discordance in-

dex (see Fig. 11a). To account for the preference threshold  $p_j$  and reduce the discordances to zero for performance differences below this threshold, we subtract the value of  $u_j^D$  attained for  $p_j$ , i.e.,  $u_j^D(p_j)$ , from  $u_j^D(ReLU(g_j(a_k) - g_j(a_i)))$ . Finally, the resulting difference is processed using the LeakyReLU function (see Fig. 11b) in the following way:

$$D_j(a_i, a_k) = LeakyReLU(u_j^D(ReLU(g_j(a_k) - g_j(a_i))) - u_j^D(p_j)). \tag{40}$$



Comprehensive discordance index  $D(a_i, a_k)$  is computed in line with Eq. (34) and normalized to the  $[0,1]$  range:

$$D_{norm}(a_i, a_k) = \frac{D(a_i, a_k) - D(a^+, a^-)}{D(a^-, a^+) - D(a^+, a^-)}. \quad (41)$$

Overall, the largest value of the marginal discordance is one, which allows the method to adjust the test in such a way that the discordance is not necessarily modeled on all criteria.

The results from the two parts of ANN responsible for the concordance and discordance tests are combined into the outranking credibility  $\sigma(a_i, a_k)$  using Eq. (35) in the form of a multiplication layer. To consider only sufficiently great credibilities, we should decrease them by cutting level  $\lambda$  and transform the resulting negative values to zero. However, since cutting level  $\lambda$  is a parameter learned during training, to allow for its more efficient optimization, we decided to transform the negative credibilities to values close to zero using the LeakyReLU function with a very small  $\delta$  equal to 0.001:

$$\sigma_{NFS}(a_i, a_k) = \text{LeakyReLU}(\sigma(a_i, a_k) - \lambda). \quad (42)$$

The positive and negative flows as well as comprehensive scores, denoted by  $Sc_{ANN-ELECTRE}(a_i)$ , for all alternatives  $a_i \in A$  are computed in line with Eq. (36).

The hyperparameters of ANN-ELECTRE are the slope values  $\delta$  for the LeakyReLU function and the number  $L$  of components for *Monotonic Blocks*. Similar to the previously discussed methods, to speed up the optimization process, we use the same value of  $L$  for all criteria in the concordance and discordance parts of the network. Overall, the network for ANN-ELECTRE involves one input layer, five hidden layers, and one output layer. Hence its architecture involves the greatest number of layers and units among all introduced methods.

#### 4. Optimization

In this section, we discuss the process of determining parameter values for the presented sorting models along with all the supporting techniques that accelerate this process. The role of optimization is to determine an optimal model highly consistent with the supplied/available assignment examples. Due to non-linear transformations, numerous relationships between values of different parameters, and a large number of objects to be scored (particularly for methods based on pairwise comparisons), the use of contemporary mathematical programming solvers is excluded because of their insufficient efficiency. Therefore, to determine the values of model parameters, we use the iterative optimization methods based on Gradient Descent (GD). There are many different techniques, called optimizers, used in ANN that are based on GD. In this paper, we employ AdamW, which is the Adam optimizer (Kingma & Ba, 2014) with decoupled weight decay regularization (Loshchilov & Hutter, 2018).

The AdamW optimizer employs the following hyperparameters having a significant impact on the training process, speed, and quality of an identified solution:

- $\alpha$  – a learning rate that affects the size of the parameter correction in an optimization step. Too low values imply slow learning and the possibility of getting stuck in the local optimum too early, while too high values make it possible to omit the optimum and prevent the optimization from converging.
- $\beta_1$  and  $\beta_2$  – momentum factors determining the impact of historical improvement of parameters on the current step. Momentum is used to speed up and improve the optimization process by drawing conclusions from previous steps to determine a more stable optimization direction and less dynamic response to perturbations during training.

- $\epsilon$  – a small value added to the denominator to stabilize the calculations.
- $w_\tau$  – a weight decay factor.

The entire optimization process is presented as Algorithm 1.

**Algorithm 1** Optimization algorithm using AdamW (adapted after Loshchilov & Hutter, 2018).

```

1: given  $\alpha \in (0, 1)$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 10^{-8}$ ,  $w_\tau = 0.01$ ,  $\xi \in (0, 1)$ 
2: initialize epoch number  $\tau \leftarrow 0$ , parameter vector  $\mathbf{x}_{\tau=0} \in \mathbb{R}^n$ , first moment vector  $\mathbf{m}_{\tau=0} \leftarrow 0$ , second moment vector  $\mathbf{v}_{\tau=0} \leftarrow 0$ 
3:  $evaluations \leftarrow g(A^R)$ 
4:  $input \leftarrow \text{Preprocessing}(evaluations)$ 
5: repeat
6:    $\tau \leftarrow \tau + 1$ 
7:    $input_{noised} \leftarrow input + \mathcal{N}(0, \xi)$ 
8:    $\mathbf{g}_\tau \leftarrow \nabla \text{Loss}(Sc(\mathbf{x}_{\tau-1}, input_{noised}))$ 
9:    $\mathbf{m}_\tau \leftarrow \beta_1 \mathbf{m}_{\tau-1} + (1 - \beta_1) \mathbf{g}_\tau$ 
10:   $\mathbf{v}_\tau \leftarrow \beta_2 \mathbf{v}_{\tau-1} + (1 - \beta_2) \mathbf{g}_\tau^2$ 
11:   $\hat{\mathbf{m}}_\tau \leftarrow \mathbf{m}_\tau / (1 - \beta_1^\tau)$ 
12:   $\hat{\mathbf{v}}_\tau \leftarrow \mathbf{v}_\tau / (1 - \beta_2^\tau)$ 
13:   $\mathbf{x}_\tau \leftarrow \mathbf{x}_{\tau-1} - (\alpha \hat{\mathbf{m}}_\tau / (\sqrt{\hat{\mathbf{v}}_\tau} + \epsilon) + w_\tau \mathbf{x}_{\tau-1})$ 
14: until stopping criterion is met
    
```

First, all parameter values  $\mathbf{x}_{\tau=0}$  are initialized randomly according to the constraints imposed on specific parameter types. These parameters can be, e.g., weights  $w_j$ , interaction coefficients  $w_{\{j,l\}}$  and class thresholds  $t$  for the ANN-Ch methods, whereas for ANN-ELECTRE – these are  $\alpha_k, y_k, \theta_k$  from each *Monotonic Block*, weights  $w_j$ , preference thresholds  $p_j$ , cutting level  $\lambda$ , and class thresholds  $t$ . At the same time, all auxiliary variables for the optimization process, including an epoch number and moment vectors, are initialized (see line 2).

We used two optimization techniques aimed at accelerating optimization. The first one is Batch Gradient Descent (BGD), which calculates loss, gradient, and modifications of network parameter values at once after processing all alternatives in  $A^R$  (see line 3). It speeds up the entire optimization process and makes the final model independent from the order of processing the alternatives. If it is impossible to use BGD, it is recommended to employ Mini Batch Gradient Descent (Ruder, 2016). This technique divides the training set into subsets in each epoch and trains this subset at once. In this case, the order of processing alternatives may affect the final result, but this impact will be negligible with sufficiently large batches.

The other method for reducing processing time is to prepare the input data in the preprocessing stage so that only operations using network parameters are performed in each epoch (see line 4). For example, one assumes that the entry gets alternatives with performances converted to the 0–1 range via min-max scaling.

After the input data preprocessing stage, the actual optimization process takes place. It consists of the iterative improvement of the model parameters to minimize a comprehensive classification error. To increase the noise resistance, robustness of the model, and its generalization capabilities, we used data augmentation (Zheng, Song, Leung, & Goodfellow, 2016). It is a technique mainly used to reduce overfitting (Shorten & Khoshgoftaar, 2019). It is about creating new training objects from the transformations of the original objects. The basic change is to add noise, e.g., in the form of Gaussian noise  $\mathcal{N}(0, \xi)$ , where  $\xi$  is the standard deviation, being an additional hyperparameter of the optimization process. Its application implies a slight change in alternatives performances, different in each epoch (see line 7).

**Table 1**  
Values of criteria weights obtained for the ANN-based methods for the illustrative example concerning the ERA dataset.

Method	$w_1$	$w_2$	$w_3$	$w_4$
ANN-OWA	0.4257	0.0055	0.2225	0.3464
ANN-Ch-Constr.	0.0693	0.0433	0.0000	0.0255
ANN-Ch-Uncons.	0.0030	0.0039	0.0018	-0.0029
ANN-Ch-Pos.	0.0060	0.0048	0.0021	0.0003
ANN-TOPSIS	0.5799	0.7987	0.6676	0.5701
ANN-UTADIS	0.3251	0.1663	0.4217	0.0869
ANN-PROMETHEE	0.2126	0.4573	0.1591	0.1709
ANN-ELECTRE (concordance)	0.5139	0.1955	0.2726	0.0180
ANN-ELECTRE (discordance)	0.3029	0.0000	1.0000	0.2110

By propagating the input with noise through the successive layers of the network in iteration  $\tau$  with the current parameter values  $x_{\tau-1}$ , scores  $S_c$  are computed for all reference alternatives  $A^R$ . The resulting class assignments are compared with the desired ones, and the respective loss is computed. Then, the loss is backpropagated across all network layers, leading to gradient vectors  $g_\tau$  (see line 8). Subsequently, gradient transformation is performed for each parameter to improve the optimization process (see lines 9–12). The AdamW algorithm employs an adaptive learning rate for each method parameter, using squared gradients to scale the learning rate and moving momentum average.

Finally, a new parameter value  $x_\tau$  is computed by combining the current value with the identified correction. For this purpose, AdamW considers the previously prepared auxiliary variables, learning rate, and weight decay (see line 13). The latter parameter controls the model’s regularization, imposing an additional optimization goal that prevents the construction of accurate, though incorrect, solutions with poor generalization capabilities. This may occur in the case of overfitting the model for the training data or assigning parameter values that are hard to interpret (in the case of ANNs, these are usually prohibitively large values). The weight decay mechanism adds a penalty, controlled by  $w_\tau$ , for the value of the parameters in each optimization step.

Processing all alternatives and modifying the parameter values is called an epoch. Such a process is performed multiple times until the stopping condition is met. In our case, it occurs after 200 training epochs. The final parameter values are those for which the model obtained the lowest error for the validation set during optimization.

### 5. Illustration of preference models inferred with neural networks

In this section, we illustrate the preference models inferred with the proposed ANN-based methods. We consider a two-class problem called ERA (Employee Rejection / Acceptance) (Hall et al., 2009). It concerns a student survey regarding the willingness to hire an employee based on four features of a candidate, such as, e.g., experience and verbal skills. All criteria are of gain type and have been pre-processed as described in Section 4. The models were obtained by training the methods on 80% randomly chosen alternatives. The criteria weights obtained for all methods are presented in Table 1, whereas the interaction coefficients for the ANN-Ch algorithms are given in Table 2.

**ANN-OWA** By applying the ANN-OWA method, we obtained a model parameterized with the weights shown in Table 1. They reflect the impact of each position in the sorted performance vector on the comprehensive score and assignment of each alternative. The highest performance on any criterion has the greatest impact on the results (almost 43%), and the lowest performance is the second most important factor (almost 35%). In contrast, the second-best performance has a negligible impact on the recommended as-

signment (below 1%). The two classes considered in the ERA problem are separated by threshold  $t_1 = 0.4114$  with OWA taking values between 0 and 1.

In what follows, we provide the models derived with different variants of the Choquet integral-based algorithms. Unlike in the ANN-OWA method, the weights from the linear layer correspond to the weights of individual criteria and interaction coefficient for pairs of criteria.

**ANN-Ch-Constr** Let us first consider the variant in which the criteria weights need to be positive, interactions can be either positive or negative, but the negative interaction coefficients cannot be greater than the weights of the criteria involved in a given pair. The analysis of weights (see Table 1) indicates that the greatest impact is attributed to the first criterion, whereas the third criterion has the least influence on the attained score. The values of the interaction coefficients are given in Table 2. All coefficients but  $w_{\{1,4\}}$  are positive. The greatest synergy effect is observed for  $g_1$  and  $g_2$  as well as  $g_3$  and  $g_4$ . This means that the simultaneous presence of highly preferred performances on these criteria pairs gives the alternative a bonus. Note that the weights retain the required dependencies and fulfill the constraint defined by Eq. (10) (e.g.,  $w_1 + w_{\{1,4\}} = 0.0693 + -0.0255 \geq 0$ ).

The actual significance of criterion  $g_i$  in the Choquet integral can be represented by the Shapley value  $\varphi(i)$  defined as follows (Angilella et al., 2013):

$$\varphi(i) = w_i + \sum_{\{i,l\} \subseteq G} \frac{w_{\{i,l\}}}{2}. \tag{43}$$

For the considered model, we obtained the following coefficients:  $\varphi(1) = 0.2454$ ,  $\varphi(2) = 0.3409$ ,  $\varphi(3) = 0.2148$ , and  $\varphi(4) = 0.1989$ . They indicate that  $g_1$  and  $g_4$  are the most and the least important criteria, respectively. In addition,  $g_3$  has a relatively high significance level  $\varphi(3)$  despite its zero weight  $w_3$ . However, it is involved in multiple interacting pairs of criteria. Finally, the separating class threshold is  $t_1 = 0.6117$ .

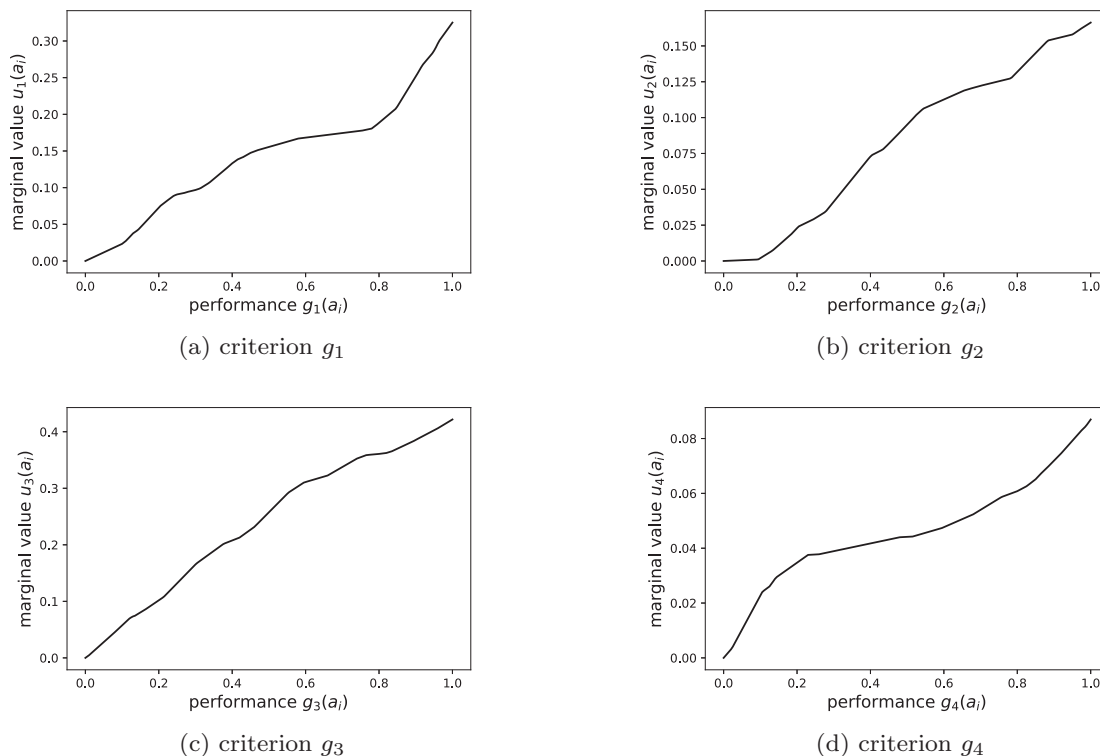
**ANN-Ch-Uncons** For the variant of ANN-Ch that considers both positive and negative interactions, while allowing for a change in the direction of preference for a given criterion, the results are quite different. Based on the inferred weights (see Table 1), we conclude that  $g_2$  and  $g_3$  have, respectively, the greatest and the least individual impacts on the attained scores. Moreover,  $g_4$  is assigned a negative weight, meaning that preference learning led to the inversion of preference direction from gain to cost for this criterion. This may indicate possible inconsistencies in the data or suggest the need for incorporating additional constraints in the model. The interaction coefficients for all pairs of criteria are shown in Table 2. Pair  $\{g_1, g_2\}$  has the greatest positive impact on the attained score, giving a great bonus to alternatives with high performances on both  $g_1$  and  $g_2$ . On the other extreme,  $w_{\{1,3\}}$  is very low, implying that the benefit from the coexistence of high values on  $g_1$  and  $g_3$  is marginal. Furthermore, negative interactions can be observed for  $\{g_1, g_4\}$  and  $\{g_2, g_4\}$ . This suggests that it is beneficial for alternatives to have a low value on at least one criterion in these two pairs, which most likely relates to  $g_4$ , whose individual weight was already negative. The value of bias is 0.4486, serving to shift the sigmoid function and having no direct interpretation. In this case, the value of a separating class threshold is  $t_1 = 0.6117$ .

**ANN-Ch-Pos** The last variant of ANN-Ch assumed that both individual weights and interaction coefficients need to be positive. This excludes, e.g., negating the preference direction of  $g_4$ , as suggested by the previous model. The analysis of weights (see Table 1) indicates the  $g_1$  and  $g_2$  are the most important criteria, whereas the impact of  $g_4$  is negligible. The crucial role of the first two criteria is emphasized by the highest value of the interaction coefficient for this pair. On the other extreme,  $g_3$  and  $g_4$  are not interacting,



**Table 2**  
Values of criteria interaction coefficients obtained for the ANN-based methods using the Choquet integral for the illustrative example concerning the ERA dataset.

Method	$w_{\{1,2\}}$	$w_{\{1,3\}}$	$w_{\{1,4\}}$	$w_{\{2,3\}}$	$w_{\{2,4\}}$	$w_{\{3,4\}}$
ANN-Ch-Constr.	0.2859	0.0919	-0.0255	0.1374	0.1720	0.2003
ANN-Ch-Uncons.	0.0042	0.0002	-0.0007	0.0022	-0.0021	0.0006
ANN-Ch-Pos.	0.0043	0.0012	0.0008	0.0030	0.0006	0.0000



**Fig. 12.** Marginal value functions scaled by criteria weights constructed by ANN-UTADIS for the ERA dataset.

meaning that the coexistence of high or low values on these criteria has no impact on the attained score. The precise value of interaction coefficients are shown in Table 2. All above weights translate into the following normalized Shapley values:  $\varphi(1) = 0.3962$ ,  $\varphi(2) = 0.3794$ ,  $\varphi(3) = 0.1819$ , and  $\varphi(4) = 0.0425$ . They confirm that  $g_1$  and  $g_2$  are the most influential criteria, whereas the role of  $g_4$  is negligible. The threshold separating the two considered classes on a scale of the Choquet integral is  $t_1 = 0.6173$ .

**ANN-TOPSIS** TOPSIS investigates the distance of each alternative from the ideal and anti-ideal alternatives. For the considered problem, the performances of these alternatives are as follows:  $a^+ = [1, 1, 1, 1]$  and  $a^- = [0, 0, 0, 0]$ . Criterion  $g_2$  has the greatest impact on the distances, whereas the influence of  $g_4$  is the least (see Table 1). However, the ratios between the criteria weights are much lesser than in the case of the Choquet integral-based models, meaning that in TOPSIS, the importances of all criteria are more balanced. The threshold separating the less and more preferred classes on the considered distance scale from 0 to 1 is  $t_1 = 0.4601$ .

**ANN-UTADIS** The value-based model inferred by ANN-UTADIS consists of marginal value functions for all criteria. Their shapes can be visualized based on the characteristic points of the *Monotonic Blocks*, weights for the linear layer aggregating marginal values, and the normalization constraint for the weights. We used 20 component functions (neurons in the hidden layer) in each of the *Monotonic Blocks*. Thus the constructed functions can have up to 40 characteristic points. The plots can be reconstructed by querying relevant parts of the ANN for the value assigned to artificially generated input data.

The marginal value functions are shown in Fig. 12. The greatest maximal share in the comprehensive value is assigned to  $g_3$ , whereas the lowest maximal share corresponds to  $g_4$  (see Table 1). The marginal function for  $g_1$  reveals minor differences for the performances ranging from 0.6 to 0.8. In contrast, above 0.8, there is a rapid increase in marginal values, indicating a high preference for alternatives with the most preferred values on  $g_1$ . For  $g_2$ , the marginal values assigned to performances lesser than 0.1 are close to zero. Above this level, the function's shape, similar to the function corresponding to  $g_3$ , is nearly linear. In turn, for  $g_4$ , the differences between the marginal values are significant for very low or very high performances, whereas the slope is less steep in the mid-range. The threshold separating the two classes on a scale of a comprehensive value from 0 to 1 is  $t_1 = 0.4909$ .

**ANN-PROMETHEE** In the PROMETHEE-based method, the parameter values of the network refer to pairwise comparisons of alternatives, providing evidence on how much one of them is preferred to the other. In this case, we used 20 component functions in each of the *Monotonic Blocks* and reconstructed the marginal preference functions similarly as for ANN-UTADIS. The plots in Fig. 13 are already scaled by the criteria weights.

The weight of  $g_2$  is the greatest, whereas the importance coefficients of  $g_3$  and  $g_4$  are much lesser (see Table 1). For all criteria and the non-positive performance differences, preference degrees are zero. Moreover, a small advantage of one alternative over another does not imply the preference or the preference degree is very marginal. For example, for  $g_4$  – the preference functions starts to increase for difference greater than 0.12. Hence this value can be

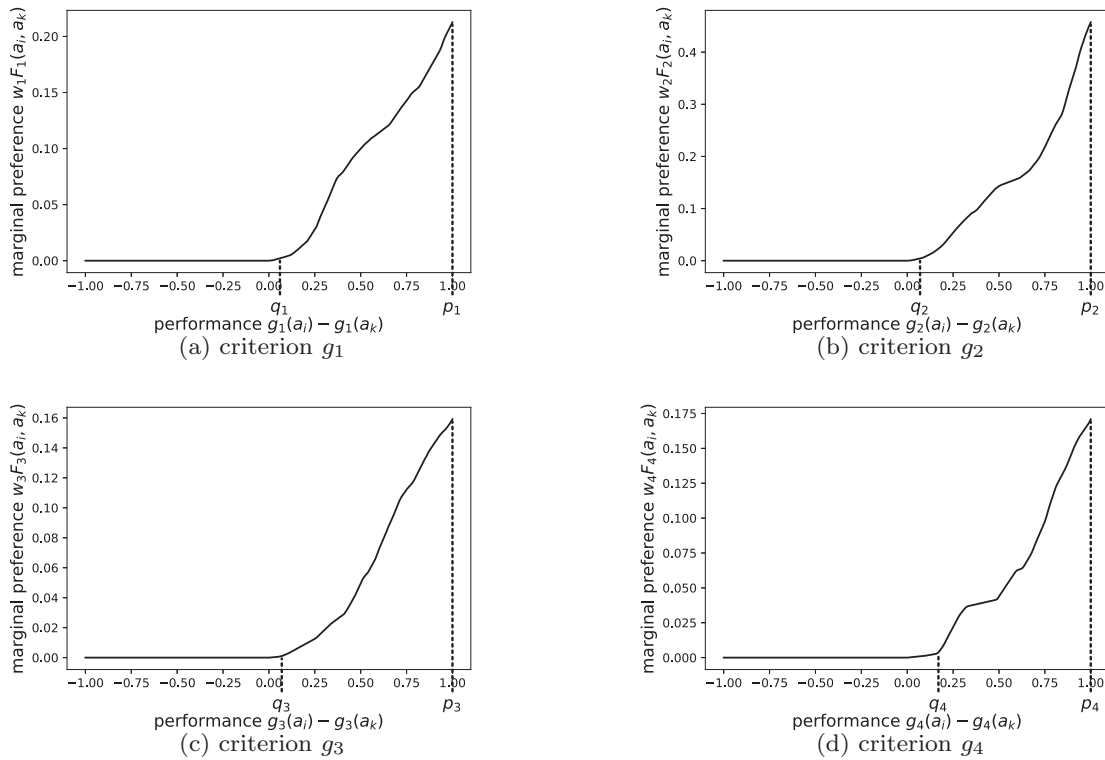


Fig. 13. Marginal preference functions scaled by criteria weights constructed by ANN-PROMETHEE for the ERA dataset.

interpreted as an indifference threshold. Furthermore, we do not observe any level (constant) part above a certain value. Once the function starts to increase, this trend is maintained till the very end. Hence the preference threshold for all criteria is equal to one. The plots show that the greatest increase in the preference degree occurs for the largest differences ( $> 0.75$ ), but for  $g_1$  and  $g_4$ , such a steep slope is also observed for differences between 0.2 to 0.3. The threshold separating the two classes on a Net Flow Score scale from  $-1$  to  $1$  is  $t_1 = 0$ .

**ANN-ELECTRE** For the ELECTRE-like method, we analyze the concordance and discordance functions for each criterion. In this approach, 30 components were used in each of the *Monotonic Blocks*. However, as can be seen in Fig. 14, most of them were deactivated during training. This led to easily interpretable functions with clearly distinguished thresholds for the performance difference, implying the maximal value of either concordance or discordance.

The marginal concordance and discordance functions presented in Fig. 14 were already normalized. Moreover, the concordance functions were scaled by the weights (see Table 1). For ANN-ELECTRE, there is no univocal information on the importance of different criteria because the methods assigned different weights to the arguments in favor and against the outranking deriving from the same criterion. On the one hand,  $g_1$  has the greatest impact in terms of supporting the truth of outranking, whereas the concordance weight of  $g_4$  is the least. On the other hand,  $g_3$  may have a very negative impact by strongly supporting discordance in case of large performance differences against the outranking. The maximal discordance on  $g_3$  is one, hence zeroing the outranking credibility in case one alternative is vastly worse than another on this criterion. Furthermore, the discordance does not occur for  $g_2$ , which can be interpreted as the lack of power of this criterion to veto against the outranking.

When it comes to the marginal functions, for performance differences greater or equal to zero, the concordance indices take

the maximal value of one (if the plot is unscaled) or concordance weight assigned to a given criterion (when considering a scaled plot as depicted in Fig. 14). For all criteria, an indifference threshold is close to zero. It is also possible to distinguish the preference and veto thresholds. When the performance difference exceeds the negated preference threshold, the concordance becomes positive, whereas if the performance difference is lesser than this threshold, the discordance occurs (when veto is admitted for a given criterion). For  $g_3$ , this threshold has a value of 0.2236. In turn, for  $g_1$ , there is a large zone with no or very marginal concordance and discordance. The concordance becomes positive for marginally negative performance differences, whereas the discordance is above zero only when one alternative is worse than another by at least  $p_1 = 0.5940$ . The preference thresholds directly optimized by the ANN for  $g_2$  and  $g_4$  are 0.3329 and 0.3354. Finally, when the performance difference is greater than the veto threshold, the maximal discordance on a given criterion occurs. The values of this threshold for  $g_1$ ,  $g_3$ , and  $g_4$  are, respectively, around 0.93, 0.38, and 0.61.

An important parameter inferred by ANN-ELECTRE is the cutting level  $\lambda$ . It was assigned a very high value of 0.95. This means that the arguments supporting the outranking need to be very strong, and the arguments against the outranking need to be none or negligible to support the inclusion of credibility in the Net Flow Score computations performed by the method. The threshold separating the rejection and acceptance classes on the scale between  $-1$  and  $1$  is  $t_1 = 0$ .

## 6. Computational experiments

To investigate the performance of the proposed methods, they were applied to a set of binary sorting problems (see Table 3). The datasets come from the UCI repository (<http://archive.ics.uci.edu/ml/>) and the WEKA software (Hall et al., 2009). The number of criteria is between four and eight, whereas the number of alternatives is ranging from several dozen to several hundred. In Table 3,

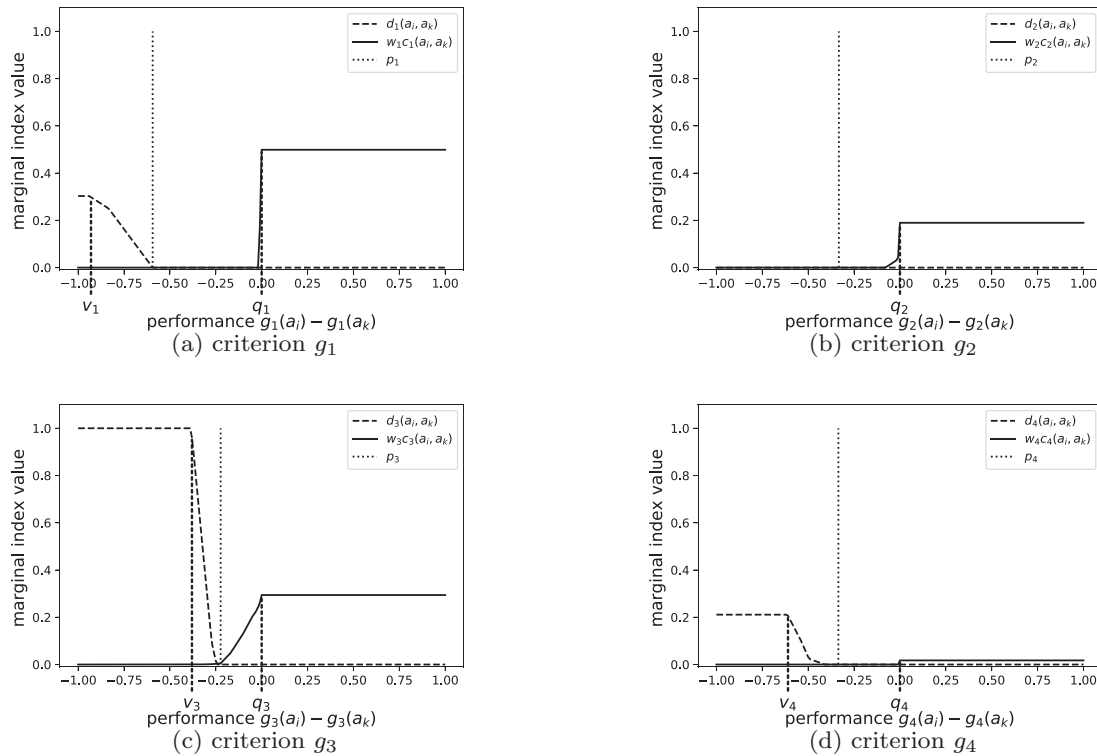


Fig. 14. Marginal concordance and discordance functions constructed by ANN-ELECTRE for the ERA dataset.

Table 3

Datasets considered in the experimental evaluation.

Name	Code	No. of alternatives	No. of criteria	No. of pairwise comparisons
Den Bosch	DBS	120	8	14,280
Computer Processing Units	CPU	209	6	43,472
Breast Cancer	BCC	286	7	81,510
Auto MPG	MPG	392	7	153,272
Employee Selection	ESL	488	4	237,656
Mammographic	MMG	961	5	922,560
Employee Rejection/Acceptance	ERA	1000	4	999,000
Lecturers Evaluation	LEV	1000	4	999,000
Car Evaluation	CEV	1728	6	2,984,256

we include the information on the number of pairwise comparisons that appear as input in outranking-based approaches such as ANN-PROMETHEE and ANN-ELECTRE.

The same set of problems was considered in Tehrani et al. (2012) and Sobrie et al. (2019). For a detailed description of each set, see Tehrani et al. (2012). Some of them (MPG, MMG, and BCC) involve nominal attributes that have been transformed into monotonic criteria according to Tehrani et al. (2012). This increases the difficulty of the preference learning task for such problems as the methods respecting the pre-defined preference directions need to deal with an arbitrarily imposed order which reduces their flexibility in fitting the model.

To quantify the algorithms' performance, we use two classification quality measures. The first one is a standard misclassification error (0/1 loss), referring to the number of alternatives in  $A^C \subseteq A$  that the inferred model classifies incorrectly:

$$0/1 \text{ loss} = \frac{1}{|A^C|} \sum_{a_i \in A^C} \text{CL error}(a_i), \tag{44}$$

where:

$$\text{CL error}(a_i) = \begin{cases} 1, & \text{if } Sc(a_i) < t_{C_{DM}(a_i)}, \text{ or } Sc(a_i) \geq t_{C_{DM}(a_i)+1}, \\ 0, & \text{otherwise.} \end{cases} \tag{45}$$

The other measure is AUC, which – for a binary classification involving classes  $C_1$  and  $C_2$  – takes the following form:

$$AUC = \frac{\sum_{a_i \in A_{C_1}} \sum_{a_j \in A_{C_2}} \mathbb{1}[Sc(a_i) < Sc(a_j)]}{|A_{C_1}| |A_{C_2}|}, \tag{46}$$

where:

$$\mathbb{1}[Sc(a_i) < Sc(a_j)] = \begin{cases} 1, & \text{iff } Sc(a_i) < Sc(a_j), \\ 0, & \text{else.} \end{cases} \tag{47}$$

AUC builds on the number of pairs of alternatives from different classes for which the order of classes is reflected in the respective scores, i.e., a comprehensive score of  $a_i$  from the less preferred class than the class of  $a_j$  is lesser than  $Sc(a_j)$ . The measure is normalized by the number of all pairs of alternatives from different classes. Thus AUC indicates how many changes in the ranking imposed by the comprehensive scores are needed to obtain an entirely consistent outcome.

In the following subsection, we report the experimental results for eight algorithms proposed in this paper. We compare them against the following state-of-the-art preference learning methods:

- logistic regression (LR), which is a well-established statistical classification method, using the linear model of the input attributes (Hosmer et al., 2000); while estimating the

parameters of the weighted sum model, it optimizes the log-likelihood function capturing the probability of observing the desired classification for alternatives given the input data and the model;

- Choquistic regression (CR), i.e., a generalization of LR in which the Choquet integral is used as the preference model (Tehrani et al., 2012); when estimating values of its parameter, the algorithm also optimizes the log-likelihood function using a sequential quadratic programming approach implemented in Matlab;
- kernel logistic regression with the polynomial kernel (KLR-ply) and a degree equal to two so that it models low-level interactions of criteria (Tehrani et al., 2012);
- kernel logistic regression with the Gaussian kernel (KLR-rbf) able to capture interactions of higher-order; note that KLR methods are extensions of LR that are flexible but not necessarily monotonic in the sense of preserving pre-defined preference directions (Tehrani et al., 2012);
- the MORE algorithm that learns rule ensembles, adhering to monotonicity constraints, in which a single rule is treated as a subsidiary base classifier (Dembczyński et al., 2009); rule induction is performed by minimizing the sigmoid 0–1 loss function;
- the LMT algorithm that induces tree-structured models containing logistic regression functions at the leaves (Landwehr et al., 2003), while accounting for the least squared misclassification error;
- the UTADIS method, which employs linear programming provided by the IBM ILOG CPLEX solver (Sobrie et al., 2019) to infer a threshold-based value-driven sorting model using piecewise linear marginal functions with three segments (Zopounidis & Doumpos, 2000); it optimizes a misclassification error defined as an average distance of alternatives' comprehensive values from the value ranges delimited by the thresholds associated with their desired classes; the proposed ANN-based algorithms minimize the same objective function;
- the Mixed-Integer Program (MIP) for learning the parameters of MR-Sort, which is a simplified variant of ELECTRE TRI-B, using a majority rule and boundary class profiles (Leroy et al., 2011); the model parameters are selected by minimizing the 0/1 loss using the IBM ILOG CPLEX solver;
- the metaheuristic (META) for learning the parameters of MR-Sort (Sobrie et al., 2019) which uses evolutionary algorithms and mathematical programming to select parameter values minimizing the 0/1 loss.
- UTADIS-G, i.e., UTADIS employing general marginal value functions with the characteristic points corresponding to all unique performances (Greco, Mousseau, & Słowiński, 2010); the optimized objective is the same as for the standard UTADIS; the method has been implemented by the authors of this paper using the GLPK solver.

### 6.1. Estimation of hyperparameter values

In Section 4, we discussed the process of optimizing parameter values taking into account hyperparameters. This section is devoted to estimating the values of these hyperparameters as well as other hyperparameters involved in the operations of the proposed preference learning algorithms that are needed to train the models successfully.

To find the optimal values, we performed a grid search to verify the classification quality for different values. Specifically, we tested three hyperparameters:

- learning rate  $\alpha \in \{0.001, 0.002, 0.005, 0.01, 0.02, 0.05\}$  (in addition, for ANN-OWA, all variants of ANN-Ch, and ANN-TOPSIS, we considered  $\{0.1, 0.2, 0.5\}$ );
- the number  $L \in \{10, 20, 30\}$  of components used by *Monotonic Block* for ANN-UTADIS, ANN-PROMETHEE, and ANN-ELECTRE – it is the only parameter whose value needs to be provided before training for these methods;
- standard deviation  $\xi \in \{0, 0.01, 0.02, 0.05\}$  of Gaussian noise used in data augmentation, where 0 means there is no additional noise added to the input data in each optimization step.

The range of a learning rate for ANN-OWA, ANN-Ch, and ANN-TOPSIS was extended due to the existing trend in the preliminary tests. They indicated that better results could be obtained for higher values of  $\alpha$ . However, the extended tests revealed that this trend was valid only for a specific range of values, and after exceeding a certain threshold, the classification outcomes deteriorated.

In a single test, we considered precise values for each of the above hyperparameters. The test was repeated 100 times for three sizes of the training and test sets. They correspond to the scenarios where (i) the training set is small compared to the test set (20% vs. 80%), (ii) both sets are equal in size (50% vs. 50%), and (iii) the training set contains a significant number of alternatives compared to the test set (80% vs. 20%), which is the most common setting. In each run, the allocation of alternatives to the training and test sets was performed randomly and independently. The selected values of hyperparameters are the ones for which the best average value of the performance measure was obtained for the training set in a hundredfold experiment described above.

In the main paper, we present the results obtained for the ERA dataset for 80% of training data and the AUC measure (see Fig. 15). The results for ANN-UTADIS were similar for different hyperparameter values, ranging from 0.7807 to 0.7935. The highest average score was obtained for  $\alpha = 0.02$ ,  $L = 20$ , and  $\xi = 0.02$ . However, they cannot be claimed as the best hyperparameter values unanimously. The Student's T-test with a confidence level of 0.95 indicated that the AUC mean was statistically indistinguishable for 7 out of 72 configurations. On the other extreme, the lowest AUC value was observed for  $\alpha = 0.001$ ,  $L = 10$ , and  $\xi = 0.05$ . There are no strict trends here, however, it can be observed that the results for  $L = 20$  and  $L = 30$  are more often better than for  $L = 10$ .

For ANN-PROMETHEE, we observe a trend indicating that better results are achieved for lesser values of learning rate and standard deviation of the noise. The best AUC score (0.7840) is attained for  $\alpha = 0.005$ ,  $L = 10$ ,  $\xi = 0.0$ , being, however, statistically indistinguishable for 41 out of 72 configurations. In turn, the best results for the ANN-ELECTRE are achieved for a learning rate of 0.01 and 0.02. At the same time, the greater the learning rate and lower noise std, the better the results. The number  $L$  of components has no significant influence on the results. The best combination of parameters is  $\alpha = 0.01$ ,  $L = 30$ , and  $\xi = 0.0$  with mean AUC 0.7678 (we noted 20 other statistically indistinguishable configurations).

For ANN-Ch-Uncons., we observe the greatest differences in classification outcomes among all methods. For different values of hyperparameters, AUC ranges between 0.6387 and 0.7872. The best outcomes are obtained for learning rates between 0.01 and 0.2. The highest average score was obtained for  $\alpha = 0.05$  and  $\xi = 0.02$ . However, no statistically sound difference between means was observed for the other 13 out of 36 vectors of hyperparameter values. Also, for ANN-Ch-Uncons, we did not observe a noticeable impact of the input noise on the final results.

Similar trends occur for the remaining methods, i.e., the value of a learning rate for which the best results are obtained is: for ANN-Ch-Pos. – between 0.02 and 0.1, for ANN-TOPSIS – between

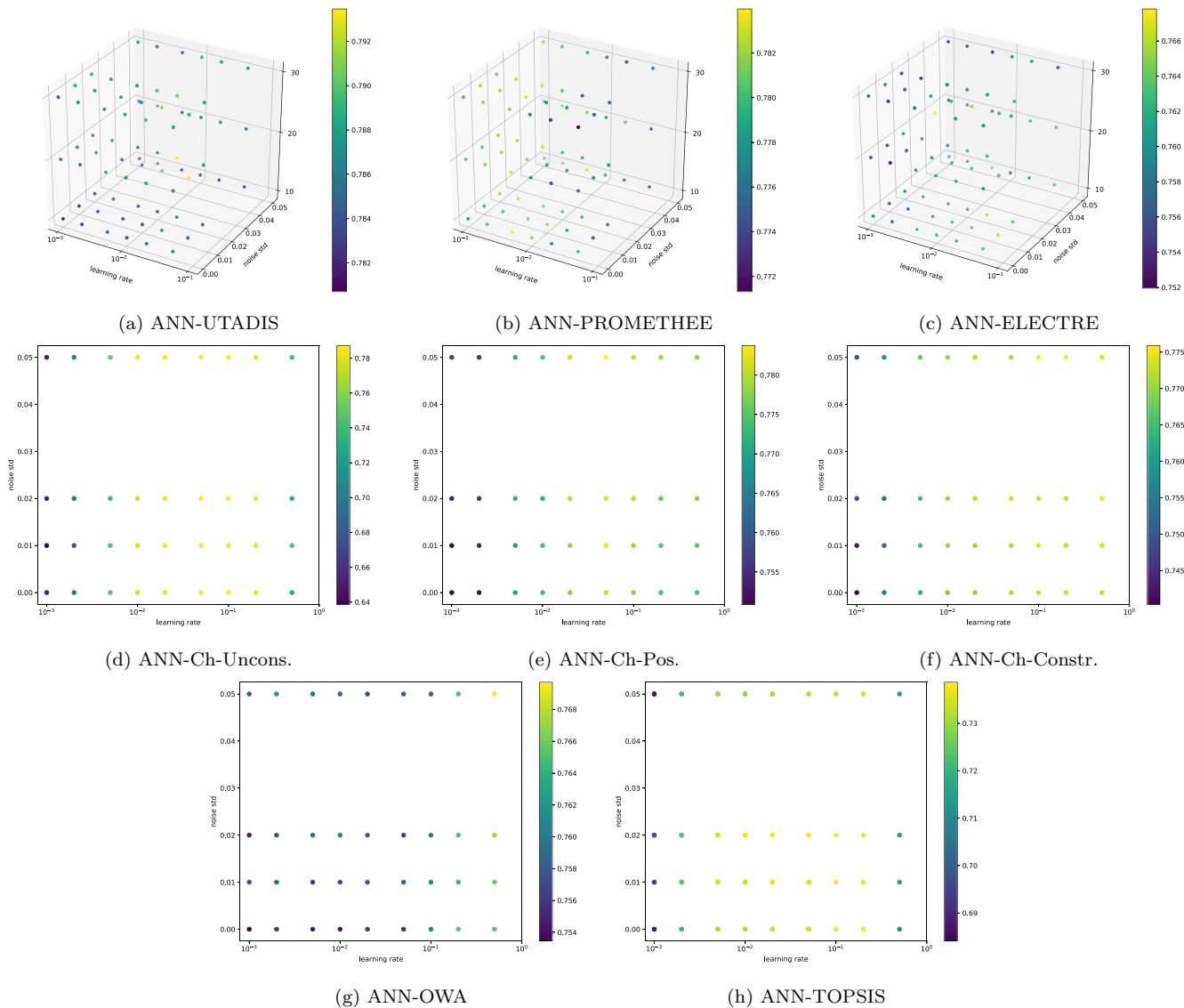


Fig. 15. The AUC value attained for the training set by various methods for different hyperparameter values for the ERA dataset.

0.05 and 0.2, for ANN-Ch-Constr – it is above 0.1, and for ANN-OWA – it is equal to 0.5. The best configurations for these methods are: for ANN-Ch-Pos. –  $\alpha = 0.05$  and  $\xi = 0.05$ , for ANN-TOPSIS –  $\alpha = 0.1$  and  $\xi = 0.02$ , for ANN-Ch-Constr –  $\alpha = 0.2$  and  $\xi = 0.05$ , and for ANN-OWA –  $\alpha = 0.5$  and  $\xi = 0.05$ .

The above conclusions hold only for the ERA dataset. For some other sets, the dependencies differed. The respective figures are presented in the e-Appendix (supplementary material available online).

### 6.2. Experimental results in terms of AUC and 0/1 loss

In this section, we report the experimental results for 17 approaches, including eight proposed in this paper. All experiments were carried out on a single CPU 2300MHz Intel(R) Xeon(R) E5-2650 v3 using Python 3.6 and the Pytorch 1.2.0 library. The training times are shown in the e-Appendix. The outcomes for the state-of-the-art methods are derived from [Tehrani et al. \(2012\)](#) and [Sobrie et al. \(2019\)](#).

In [Tables 4–6](#), we report the mean AUC values for nine benchmark datasets and different proportions of the training and test sets. For each approach, we provide the standard deviation, rank

according to the mean for a given problem, and an average rank for all datasets (see the last column). A few missing values in the tables for MIP indicate that this approach was not able to find a solution within a pre-defined time limit. In what follows, we will discuss in detail the results obtained for 80% share of the training set (see [Table 6](#)). Then, we will indicate the major differences for the remaining two settings.

Let us start by discussing the specificity of different datasets. In general, the best AUC values were attained for CPU, ESL, DBS, and CEV. For example, the mean AUC values for ANN-UTADIS for these four datasets were 0.9998, 0.9885, 0.9676, and 0.9410, whereas the respective means attained by ANN-ELECTRE were 0.9998, 0.9600, 0.9893, and 0.8786. Such high-quality scores for CPU or ESL indicate that the best-performing approaches assigned such comprehensive scores to the alternatives that inversed the original preference relation only for a few or several pairs in the testing sets. On the other extreme, the least AUC values were observed for BCC and ERA. For these problems, ANN-UTADIS attained average values equal to 0.7830 and 0.7957, whereas for ANN-ELECTRE – these were 0.7497 and 0.7695. This means that the input and output orders were not consistent for about 20–25% of pairs in the test set. Such differences confirm that the considered datasets posed

**Table 4**  
Classification performance in terms of the mean and standard deviation of AUC for 20% training data and 80% test data.

Method	DBS	CPU	BCC	MPG	ESL	ERA	LEV	CEV	MMG	Avg. rank
ANN-UTADIS	0.9159 ± 0.0230 (8)	0.9979 ± 0.0024 (1)	0.7513 ± 0.0158 (4)	0.8870 ± 0.0086 (9)	0.9839 ± 0.0030 (4)	0.7844 ± 0.0081 (1)	0.8955 ± 0.0066 (2)	0.9384 ± 0.0041 (8)	0.8769 ± 0.0118 (13)	5.56
ANN-PROMETHEE	0.9289 ± 0.0224 (3)	0.9918 ± 0.0089 (3)	0.7524 ± 0.0162 (2)	0.8750 ± 0.0088 (10)	0.9840 ± 0.0030 (3)	0.7801 ± 0.0087 (2)	0.8923 ± 0.0075 (5)	0.8919 ± 0.0060 (15)	0.8869 ± 0.0162 (8)	5.67
ANN-Ch-Uncons.	0.9181 ± 0.0150 (6)	0.9798 ± 0.0082 (9)	0.7292 ± 0.0211 (8)	0.9640 ± 0.0089 (6)	0.9835 ± 0.0037 (5)	0.7758 ± 0.0081 (4)	0.8930 ± 0.0069 (3)	0.9685 ± 0.0026 (6)	0.8866 ± 0.0075 (11)	6.44
ANN-Ch-Pos.	0.9202 ± 0.0161 (5)	0.9751 ± 0.0072 (11)	0.7495 ± 0.0187 (5)	0.7394 ± 0.0614 (17)	0.9834 ± 0.0029 (6)	0.7771 ± 0.0076 (3)	0.8929 ± 0.0059 (4)	0.9284 ± 0.0038 (11)	0.8868 ± 0.0128 (9)	7.89
ANN-Ch-Constr	0.9164 ± 0.0242 (7)	0.9806 ± 0.0073 (8)	0.7515 ± 0.0169 (3)	0.8451 ± 0.0147 (12)	0.9848 ± 0.0030 (2)	0.7721 ± 0.0094 (5)	0.8918 ± 0.0059 (6)	0.9344 ± 0.0071 (9)	0.8920 ± 0.0081 (4)	6.22
ANN-ELECTRE	0.9285 ± 0.0210 (4)	0.9971 ± 0.0038 (2)	0.7325 ± 0.0177 (6)	0.8540 ± 0.0219 (11)	0.9854 ± 0.0023 (1)	0.7640 ± 0.0094 (9)	0.8852 ± 0.0077 (10)	0.8753 ± 0.0160 (16)	0.8960 ± 0.0119 (2)	6.78
ANN-OWA	0.9077 ± 0.0161 (10)	0.9411 ± 0.0110 (17)	0.7533 ± 0.0180 (1)	0.6514 ± 0.0159 (18)	0.9808 ± 0.0029 (7)	0.7654 ± 0.0074 (8)	0.8688 ± 0.0064 (14)	0.7240 ± 0.0070 (17)	0.8897 ± 0.0057 (5)	10.78
ANN-TOPSIS	0.8919 ± 0.0191 (12)	0.9130 ± 0.0191 (18)	0.7243 ± 0.0139 (9)	0.9533 ± 0.0046 (7)	0.7806 ± 0.0112 (18)	0.7369 ± 0.0076 (14)	0.8137 ± 0.0073 (17)	0.9655 ± 0.0028 (7)	0.8527 ± 0.0080 (17)	13.22
CR	0.9290 ± 0.0322 (2)	0.9822 ± 0.0121 (5)	0.6400 ± 0.0641 (18)	0.9788 ± 0.0160 (1)	0.9670 ± 0.0074 (12)	0.7669 ± 0.0334 (6)	0.8971 ± 0.0098 (1)	0.9825 ± 0.0080 (3)	0.8867 ± 0.0123 (10)	6.44
LR	0.8866 ± 0.0511 (14)	0.9806 ± 0.0124 (7)	0.6970 ± 0.0411 (12)	0.9675 ± 0.0068 (5)	0.9721 ± 0.0060 (8)	0.7602 ± 0.0331 (11)	0.8905 ± 0.0081 (7)	0.9332 ± 0.0033 (10)	0.8962 ± 0.0080 (1)	8.33
KLR-ply	0.9359 ± 0.0218 (1)	0.9716 ± 0.0072 (13)	0.6509 ± 0.0568 (17)	0.9704 ± 0.0075 (4)	0.9638 ± 0.0106 (13)	0.7555 ± 0.0139 (12)	0.8870 ± 0.0094 (8)	0.9818 ± 0.0058 (5)	0.8552 ± 0.0203 (16)	9.89
KLR-rbf	0.9053 ± 0.0433 (11)	0.9843 ± 0.0116 (4)	0.7124 ± 0.0290 (11)	0.9741 ± 0.0055 (3)	0.9705 ± 0.0099 (9)	0.7662 ± 0.0098 (7)	0.8860 ± 0.0128 (9)	0.9821 ± 0.0076 (4)	0.8938 ± 0.0121 (3)	6.78
MORE	0.8731 ± 0.0481 (16)	0.9749 ± 0.0235 (12)	0.6639 ± 0.0567 (15)	0.9501 ± 0.0263 (8)	0.9466 ± 0.0484 (17)	0.7198 ± 0.0329 (17)	0.8137 ± 0.0621 (18)	0.9888 ± 0.0063 (2)	0.8754 ± 0.0274 (14)	13.22
LMT	0.9151 ± 0.0228 (9)	0.9816 ± 0.0113 (6)	0.7310 ± 0.0675 (7)	0.9753 ± 0.0092 (2)	0.9696 ± 0.0086 (11)	0.7619 ± 0.0160 (10)	0.8797 ± 0.0182 (11)	0.9902 ± 0.0042 (1)	0.8890 ± 0.0259 (6)	7.00
META	0.8761 ± 0.0462 (15)	0.9531 ± 0.0247 (15)	0.6810 ± 0.0458 (13)	0.8337 ± 0.0291 (13)	0.9569 ± 0.0114 (15)	0.7256 ± 0.0238 (16)	0.8530 ± 0.0258 (15)	0.8968 ± 0.0116 (14)	0.8828 ± 0.0129 (12)	14.22
MIP	0.8637 ± 0.0463 (17)	0.9497 ± 0.0262 (16)	0.7155 ± 0.0365 (10)	0.8215 ± 0.0368 (15)	0.9510 ± 0.0166 (16)	0.7182 ± 0.0328 (18)	0.8424 ± 0.0291 (16)	-	0.8877 ± 0.0151 (7)	14.78
UTADIS	0.8886 ± 0.0496 (13)	0.9789 ± 0.0283 (10)	0.6650 ± 0.0527 (14)	0.8162 ± 0.0335 (16)	0.9704 ± 0.0095 (10)	0.7409 ± 0.0175 (13)	0.8707 ± 0.0146 (12)	0.9235 ± 0.0183 (13)	0.8650 ± 0.0294 (15)	12.89
UTADIS-G	0.8564 ± 0.0507 (18)	0.9552 ± 0.0366 (14)	0.6617 ± 0.0489 (16)	0.8314 ± 0.0328 (14)	0.9636 ± 0.0122 (14)	0.7307 ± 0.0233 (15)	0.8705 ± 0.0134 (13)	0.9269 ± 0.0149 (12)	0.8474 ± 0.0284 (18)	14.89

**Table 5**  
Classification performance in terms of the mean and standard deviation of AUC for 50% training data and 50% test data.

Method	DBS	CPU	BCC	MPG	ESL	ERA	LEV	CEV	MMG	Avg. rank
ANN-UTADIS	0.9399 ± 0.0292 (4)	0.9991 ± 0.0017 (1)	0.7632 ± 0.0307 (3)	0.8911 ± 0.0164 (9)	0.9859 ± 0.0047 (3)	0.7880 ± 0.0144 (1)	0.8996 ± 0.0100 (4)	0.9395 ± 0.0060 (8)	0.8815 ± 0.0162 (14)	5.22
ANN-PROMETHEE	0.9446 ± 0.0266 (2)	0.9971 ± 0.0044 (3)	0.7636 ± 0.0336 (2)	0.8746 ± 0.0180 (10)	0.9856 ± 0.0042 (4)	0.7839 ± 0.0136 (2)	0.8946 ± 0.0131 (8)	0.8960 ± 0.0099 (14)	0.8874 ± 0.0168 (11)	6.22
ANN-Ch-Uncons.	0.9301 ± 0.0234 (8)	0.9890 ± 0.0060 (8)	0.7517 ± 0.0252 (6)	0.9713 ± 0.0084 (6)	0.9855 ± 0.0045 (5)	0.7823 ± 0.0163 (3)	0.8974 ± 0.0110 (5)	0.9707 ± 0.0036 (6)	0.8923 ± 0.0124 (8)	6.11
ANN-Ch-Pos.	0.9303 ± 0.0254 (7)	0.9821 ± 0.0082 (13)	0.7560 ± 0.0359 (5)	0.7538 ± 0.0489 (16)	0.9844 ± 0.0048 (6)	0.7816 ± 0.0148 (4)	0.8963 ± 0.0101 (6)	0.9302 ± 0.0060 (13)	0.8878 ± 0.0167 (10)	8.89
ANN-Ch-Constr	0.9299 ± 0.0299 (9)	0.9865 ± 0.0056 (11)	0.7641 ± 0.0314 (1)	0.8494 ± 0.0224 (12)	0.9870 ± 0.0039 (1)	0.7769 ± 0.0138 (5)	0.8957 ± 0.0105 (7)	0.9357 ± 0.0088 (10)	0.8952 ± 0.0113 (7)	7.00
ANN-ELECTRE	0.9416 ± 0.0251 (3)	0.9988 ± 0.0020 (2)	0.7318 ± 0.0368 (9)	0.8536 ± 0.0218 (11)	0.9864 ± 0.0042 (2)	0.7652 ± 0.0150 (11)	0.8869 ± 0.0110 (11)	0.8751 ± 0.0192 (16)	0.9019 ± 0.0128 (1)	7.33
ANN-OWA	0.9117 ± 0.0296 (15)	0.9447 ± 0.0138 (17)	0.7568 ± 0.0336 (4)	0.6575 ± 0.0281 (17)	0.9816 ± 0.0049 (7)	0.7665 ± 0.0150 (10)	0.8714 ± 0.0119 (15)	0.7236 ± 0.0113 (17)	0.8920 ± 0.0112 (9)	12.33
ANN-TOPSIS	0.9082 ± 0.0284 (16)	0.9193 ± 0.0176 (18)	0.7402 ± 0.0293 (7)	0.9545 ± 0.0097 (8)	0.7844 ± 0.0262 (18)	0.7416 ± 0.0171 (14)	0.8203 ± 0.0125 (17)	0.9662 ± 0.0033 (7)	0.8569 ± 0.0119 (16)	13.44
CR	0.9341 ± 0.0228 (5)	0.9920 ± 0.0073 (6)	0.6912 ± 0.0469 (15)	0.9818 ± 0.0075 (1)	0.9720 ± 0.0084 (12)	0.7705 ± 0.0310 (9)	0.9098 ± 0.0103 (1)	0.9912 ± 0.0024 (4)	0.9003 ± 0.0132 (2)	6.11
LR	0.9191 ± 0.0293 (11)	0.9914 ± 0.0056 (7)	0.7184 ± 0.0367 (11)	0.9803 ± 0.0084 (3)	0.9764 ± 0.0062 (8)	0.7633 ± 0.0241 (12)	0.8935 ± 0.0113 (9)	0.9362 ± 0.0071 (9)	0.8972 ± 0.0125 (5)	8.33
KLR-ply	0.9492 ± 0.0198 (1)	0.9771 ± 0.0109 (14)	0.7001 ± 0.0396 (12)	0.9776 ± 0.0083 (4)	0.9726 ± 0.0080 (11)	0.7740 ± 0.0148 (7)	0.8999 ± 0.0120 (3)	0.9950 ± 0.0019 (2)	0.8962 ± 0.0140 (6)	6.67
KLR-rbf	0.9174 ± 0.0316 (13)	0.9925 ± 0.0056 (5)	0.7294 ± 0.0344 (10)	0.9752 ± 0.0068 (5)	0.9754 ± 0.0070 (9)	0.7745 ± 0.0141 (6)	0.9012 ± 0.0128 (2)	0.9907 ± 0.0031 (5)	0.8995 ± 0.0091 (3)	6.44
MORE	0.9179 ± 0.0403 (12)	0.9873 ± 0.0149 (10)	0.6980 ± 0.0586 (13)	0.9563 ± 0.0313 (7)	0.9557 ± 0.0301 (17)	0.7215 ± 0.0381 (17)	0.8185 ± 0.0580 (18)	0.9921 ± 0.0042 (3)	0.8839 ± 0.0305 (13)	12.22
LMT	0.9259 ± 0.0289 (10)	0.9883 ± 0.0077 (9)	0.7387 ± 0.0656 (8)	0.9814 ± 0.0074 (2)	0.9707 ± 0.0120 (14)	0.7719 ± 0.0144 (8)	0.8920 ± 0.0164 (10)	0.9977 ± 0.0017 (1)	0.8976 ± 0.0153 (4)	7.33
META	0.9074 ± 0.0366 (17)	0.9701 ± 0.0140 (15)	0.6929 ± 0.0398 (14)	0.8337 ± 0.0231 (14)	0.9640 ± 0.0099 (15)	0.7366 ± 0.0233 (16)	0.8721 ± 0.0147 (14)	0.8960 ± 0.0073 (15)	0.8862 ± 0.0138 (12)	14.67
MIP	0.8998 ± 0.0336 (18)	0.9645 ± 0.0194 (16)	-	-	0.9563 ± 0.0114 (16)	0.7167 ± 0.0274 (18)	0.8511 ± 0.0219 (16)	-	-	17.33
UTADIS	0.9325 ± 0.0345 (6)	0.9940 ± 0.0131 (4)	0.6650 ± 0.5270 (16)	0.8272 ± 0.0243 (15)	0.9747 ± 0.0116 (10)	0.7437 ± 0.0211 (13)	0.8746 ± 0.0137 (12)	0.9339 ± 0.0138 (11)	0.8667 ± 0.0385 (15)	11.33
UTADIS-G	0.9117 ± 0.0332 (14)	0.9830 ± 0.0201 (12)	0.6571 ± 0.0524 (17)	0.8456 ± 0.0205 (13)	0.9714 ± 0.0069 (13)	0.7388 ± 0.0187 (15)	0.8738 ± 0.0134 (13)	0.9329 ± 0.0114 (12)	0.8439 ± 0.0253 (17)	14.00



**Table 6**  
Classification performance in terms of the mean and standard deviation of AUC for 80% training data and 20% test data.

Method	DBS	CPU	BCC	MPG	ESL	ERA	LEV	CEV	MMG	Avg. rank
ANN-UTADIS	0.9676 ± 0.0319 (1)	0.9998 ± 0.0007 (1)	0.7830 ± 0.0656 (4)	0.9034 ± 0.0307 (9)	0.9885 ± 0.0086 (3)	0.7957 ± 0.0290 (1)	0.9044 ± 0.0218 (3)	0.9410 ± 0.0114 (8)	0.8856 ± 0.0280 (14)	4.89
ANN-PROMETHEE	0.9574 ± 0.0433 (4)	0.9988 ± 0.0029 (4)	0.7896 ± 0.0584 (2)	0.8794 ± 0.0328 (10)	0.9885 ± 0.0078 (4)	0.7891 ± 0.0291 (3)	0.8980 ± 0.0202 (9)	0.8977 ± 0.0168 (14)	0.8943 ± 0.0249 (10)	6.67
ANN-Ch-Uncons.	0.9492 ± 0.0507 (7)	0.9937 ± 0.0072 (8)	0.8074 ± 0.0595 (1)	0.9788 ± 0.0123 (5)	0.9880 ± 0.0077 (5)	0.7911 ± 0.0284 (2)	0.9035 ± 0.0207 (4)	0.9726 ± 0.0067 (6)	0.8988 ± 0.0200 (7)	5.00
ANN-Ch-Pos.	0.9368 ± 0.0534 (11)	0.9892 ± 0.0092 (13)	0.7712 ± 0.0636 (5)	0.7650 ± 0.0655 (16)	0.9867 ± 0.0091 (6)	0.7856 ± 0.0268 (4)	0.9019 ± 0.0197 (6)	0.9303 ± 0.0119 (13)	0.8923 ± 0.0228 (11)	9.44
ANN-Ch-Constr	0.9522 ± 0.0474 (5)	0.9905 ± 0.0101 (12)	0.7865 ± 0.0632 (3)	0.8503 ± 0.0394 (14)	0.9886 ± 0.0075 (2)	0.7812 ± 0.0300 (5)	0.8998 ± 0.0206 (7)	0.9376 ± 0.0131 (10)	0.8981 ± 0.0233 (8)	7.33
ANN-ELECTRE	0.9600 ± 0.0393 (3)	0.9998 ± 0.0011 (2)	0.7497 ± 0.0621 (8)	0.8582 ± 0.0388 (12)	0.9893 ± 0.0086 (1)	0.7695 ± 0.0284 (10)	0.8880 ± 0.0230 (11)	0.8786 ± 0.0245 (16)	0.9066 ± 0.0185 (2)	7.22
ANN-OWA	0.9293 ± 0.0548 (14)	0.9531 ± 0.0271 (17)	0.7660 ± 0.0674 (6)	0.6614 ± 0.0610 (17)	0.9838 ± 0.0102 (7)	0.7696 ± 0.0300 (9)	0.8726 ± 0.0222 (14)	0.7304 ± 0.0234 (17)	0.8954 ± 0.0214 (9)	12.22
ANN-TOPSIS	0.9322 ± 0.0517 (13)	0.9318 ± 0.0363 (18)	0.7573 ± 0.0548 (7)	0.9598 ± 0.0180 (7)	0.7911 ± 0.0615 (18)	0.7499 ± 0.0337 (13)	0.8251 ± 0.0233 (17)	0.9684 ± 0.0054 (7)	0.8600 ± 0.0305 (16)	12.89
CR	0.9427 ± 0.0443 (9)	0.9971 ± 0.0063 (6)	0.7349 ± 0.0692 (9)	0.9855 ± 0.0108 (1)	0.9766 ± 0.0150 (10)	0.7670 ± 0.0290 (11)	0.9122 ± 0.0202 (1)	0.9959 ± 0.0027 (3)	0.9135 ± 0.0233 (1)	5.67
LR	0.9224 ± 0.0514 (15)	0.9907 ± 0.0085 (11)	0.7253 ± 0.0715 (12)	0.9843 ± 0.0138 (2)	0.9722 ± 0.0167 (13)	0.7630 ± 0.0281 (12)	0.8928 ± 0.0234 (10)	0.9352 ± 0.0095 (12)	0.9048 ± 0.0237 (4)	10.11
KLR-ply	0.9608 ± 0.0347 (2)	0.9827 ± 0.0167 (14)	0.7071 ± 0.0720 (13)	0.9797 ± 0.0121 (4)	0.9746 ± 0.0141 (12)	0.7731 ± 0.0293 (8)	0.9048 ± 0.0201 (2)	0.9970 ± 0.0018 (4)	0.9011 ± 0.0199 (5)	7.11
KLR-rbf	0.9495 ± 0.0459 (6)	0.9984 ± 0.0052 (5)	0.7335 ± 0.0690 (11)	0.9771 ± 0.0142 (6)	0.9782 ± 0.0126 (8)	0.7759 ± 0.0315 (6)	0.9031 ± 0.0172 (5)	0.9970 ± 0.0013 (2)	0.8991 ± 0.0255 (6)	6.11
MORE	0.9409 ± 0.0539 (10)	0.9909 ± 0.0167 (9)	0.7042 ± 0.0853 (15)	0.9551 ± 0.0372 (8)	0.9507 ± 0.0508 (17)	0.7228 ± 0.0475 (18)	0.8078 ± 0.0661 (18)	0.9936 ± 0.0046 (5)	0.8889 ± 0.0363 (12)	12.44
LMT	0.9343 ± 0.0479 (12)	0.9959 ± 0.0078 (7)	0.7342 ± 0.0791 (10)	0.9841 ± 0.0106 (3)	0.9713 ± 0.0176 (14)	0.7735 ± 0.0296 (7)	0.8996 ± 0.0222 (8)	0.9993 ± 0.0017 (1)	0.9063 ± 0.0215 (3)	7.22
META	0.9019 ± 0.0606 (18)	0.9721 ± 0.0219 (15)	0.7056 ± 0.0864 (14)	0.8613 ± 0.0341 (11)	0.9613 ± 0.0170 (15)	0.7379 ± 0.0351 (16)	0.8963 ± 0.0265 (15)	0.8941 ± 0.0135 (15)	0.8860 ± 0.0265 (13)	14.67
MIP	0.9080 ± 0.0673 (17)	0.9656 ± 0.0237 (16)	–	–	0.9568 ± 0.0165 (16)	0.7242 ± 0.0477 (17)	0.8499 ± 0.0332 (16)	–	–	17.11
UTADIS	0.9476 ± 0.0401 (8)	0.9989 ± 0.0030 (3)	0.6651 ± 0.0659 (16)	0.8210 ± 0.0434 (15)	0.9778 ± 0.0117 (9)	0.7497 ± 0.0402 (14)	0.8741 ± 0.0217 (13)	0.9399 ± 0.0111 (9)	0.8682 ± 0.0470 (15)	11.33
UTADIS-G	0.9127 ± 0.0467 (16)	0.9909 ± 0.0214 (10)	0.6309 ± 0.0723 (17)	0.8507 ± 0.0365 (13)	0.9748 ± 0.0109 (11)	0.7448 ± 0.0300 (15)	0.8752 ± 0.0227 (12)	0.9373 ± 0.0110 (11)	0.8288 ± 0.0278 (17)	13.56

various challenges to the preference learning algorithms. In the e-Appendix, we discuss various characteristics that partially explain such results. For example, CPU involves six criteria, each with at least several different performances, and no single violation of the dominance or indistinguishability relation in the desired assignments. Analogously, the desired assignments for ESL agree with the dominance relation for the vast majority of pairs of alternatives, and only a tiny share of pairs are inconsistent with the dominance or indistinguishability. On the other extreme, the seven criteria for BCC involve just a few different performances, and almost 7% of all pairs of alternatives assigned to different classes violate the dominance.

Also, some datasets differentiated the considered sorting methods better than others. The greatest differences between mean AUC values were observed for MPG (for CR – 9855 and for ANN-OWA – 6614), CEV (for LMT – 0.9993 and for ANN-OWA – 0.7304), ESL (for ANN-ELECTRE – 0.9893 and for ANN-TOPSIS – 0.7911). This confirms that their specificity posed a significantly greater challenge to some approaches. On the contrary, the least differences were noted for DBS (for ANN-UTADIS – 0.9676 and for META – 0.9019) and CPU (for ANN-UTADIS – 0.9998 and for ANN-TOPSIS – 0.9318). Still, even for these benchmark problems, it was possible to distinguish the subsets of clearly better- or worse-performing methods.

The most favorable average ranks implied by the mean AUC measures for the nine datasets are attained by:

- ANN-UTADIS (4.89), which attains the best results for DBS, CPU, and ERA, positions in the top four for other three problems, and is ranked outside the top ten only for MMG;
- ANN-Ch-Uncons. (5.00), which is the most advantageous for BCC, while never dropping outside the upper half of the ranking; note that this method has a competitive advantage of not having to respect the pre-defined preference directions, which is particularly useful for datasets such as BCC (1st rank), MPG (5th rank), and MMG (7th rank), for which some originally nominal attributes have been arbitrarily transformed to monotonic criteria in [Tehrani et al. \(2012\)](#);
- CR (5.67), which attains the highest mean AUC for MPG, LEV, and MMG, while being ninth or lower for four other datasets;
- KLR-rbf (6.11), attaining ranks between second for CEV and eleventh for BCC;
- ANN-PROMETHEE (6.67), ranked in the top four for most datasets.

On the other extreme, the worst average ranks are attained by MIP (17.11), META (14.67), UTADIS-G (13.56), ANN-TOPSIS (12.89), MORE (12.44), ANN-OWA (12.22), and UTADIS (11.33). Hence, only ANN-OWA and ANN-TOPSIS achieved relatively worse results among the proposed algorithms. This can be attributed to simple preference models employed by these methods.

Following [Tehrani et al., \(2012\)](#), we applied the statistical tests to verify the significance of the performance differences. The Friedman test allowed us to reject the null hypothesis on all methods performing equally for all sizes of the training set and both considered measures (AUC and 0/1 loss). The detailed outcomes of a post hoc analysis for all pairs of algorithms conducted using the Nemenyi and Wilcoxon tests with a confidence level of 90% are discussed in the e-Appendix. In what follows, we directly compare pairwise only the approaches using similar preference models. When claiming that some performance difference in terms of AUC is significant, this is confirmed by the result of the Nemenyi test applied to a subset of algorithms using related models.

ANN-UTADIS performs significantly better than UTADIS (the Wilcoxon test) and UTADIS-G (the Wilcoxon and Nemenyi tests) based on mathematical programming. The reasons are as follows. First, minimizing the sum of regrets by UTADIS and UTADIS-G does

not correspond to the perspective captured by AUC. Also, the use of *Monotonic Block* by ANN-UTADIS gives a chance for inferring very flexible marginal value functions with characteristic points better fitting the input data. In turn, data augmentation prevents the model overfitting that occurs with UTADIS-G.

When it comes to outranking-based methods, ANN-PROMETHEE significantly outperforms MIP and META, which learn the parameters of the MR-Sort model using, respectively, Mixed Integer Programming and a dedicated heuristic. Furthermore, ANN-ELECTRE attains significantly better results than MIP. The ANN-based methods proposed in this paper use the NFS procedure, threshold-based sorting method, and flexible marginal preference, concordance, and discordance functions. In turn, the model used in MR-Sort is more complex with the boundary profiles whose performances need to be determined by the method and concordance functions without zones of indifference and weak preference, hence offering lesser flexibility.

The results attained for all algorithms using the Choquet integral model (i.e., three variants of ANN-Ch and CR) are very similar for DBS, CPU, ESL, LEV, and MMG. For BBC and ERA, CR was worse than the ANN-based methods. In turn, ANN-Ch-Constr. and ANN-Ch-Pos. were outperformed by CR on MPG and CEV. The variant without any constraint on the weights performed better for these challenging datasets because it could fit the data even better by inverting the pre-defined preference directions via assigning the negative weights. Overall, the Nemenyi test confirmed that ANN-Ch-Uncons. and CR were significantly better than ANN-Ch-Pos.

When it comes to logistic regression methods, KLR-ply and KLR-rbf perform, on average, better than LR. This is due to the non-monotonic KLR methods being able to capture low- (ply) or high-level (rbf) interactions. However, according to the Wilcoxon test, the statistically significant difference is observed only for KLR-rbf and LR. Moreover, the slight advantage of KLR methods is not implied by admitting non-monotonicity for datasets that originally involved nominal criteria (e.g., for MPG and MMG, LR attains better results than both KLR-rbf and KLR-ply).

The observations, rankings, and trends for other proportions of the training and test sets (see Tables 4 and 5) are very similar to the outcomes discussed above for the 80/20 division. However, with the decrease in the number of alternatives in the training set, the AUC decreases by a few percent for the ANN-based methods. For example, ANN-UTADIS attains an average AUC equal to 0.9676, 0.9399, and 0.9159 for DBS with 80/20, 50/50, and 20/80 shares of the training and test sets, whereas the analogous results attained by ANN-Ch-Constr. for BCC are 0.7865, 0.7641, and 0.7515. No or marginal performance deterioration is observed for ANN-PROMETHEE and ANN-ELECTRE for datasets with a larger number of alternatives, i.e., MPG, ERA, LEV, and CEV. For example, for ANN-PROMETHEE and MPG, AUC is 0.8794 for 80% training set, 0.8746 for 50%, and 0.8750 for 20%. As a result, the average ranks for these approaches are slightly better for the least size of training data than for more numerous learning sets. In fact, for the 20/80 division, ANN-PROMETHEE shares the best average rank with ANN-UTADIS. In the same spirit, the average ranks for ANN-Ch-Constr., ANN-Ch-Pos., and ANN-OWA get slightly better with the decrease of the training set's share. The opposite trend is observed for ANN-UTADIS and ANN-Ch-Uncons. The greatest improvement of ranks for smaller training data among the state-of-the-art algorithms is observed for LR and META. In contrast, the most significant deterioration is noted for KLR-ply, UTADIS, and UTADIS-G.

In Tables 7–9, we report the mean values of 0/1 loss for nine benchmark datasets and different proportions of the training and test sets. Unlike for AUC, lesser values of 0/1 loss are more favorable. Let us first focus on the results for 80% share of the training set (see Table 9). They confirm the conclusions derived from AUC analysis on the challenge posed by different datasets to the

**Table 7**  
Classification performance in terms of the mean and standard deviation of 0/1 loss for 20% training data and 80% test data.

Method	DBS	CPU	BCC	MPG	ESL	ERA	LEV	CEV	MMG	Avg. rank
ANN-UTADIS	0.1460 ± 0.0316 (6)	0.0185 ± 0.0104 (2)	0.2371 ± 0.0104 (2)	0.1795 ± 0.0130 (9)	0.0620 ± 0.0077 (2)	0.2688 ± 0.0077 (2)	0.1648 ± 0.0077 (6)	0.1118 ± 0.0053 (4)	0.1787 ± 0.0160 (8)	5.33
ANN-PROMETHEE	0.1421 ± 0.0196 (5)	0.1716 ± 0.0528 (18)	0.2659 ± 0.0317 (7)	0.2052 ± 0.0135 (12)	0.0849 ± 0.0081 (12)	0.2959 ± 0.0081 (12)	0.1830 ± 0.0081 (15)	0.2362 ± 0.0087 (15)	0.1894 ± 0.0112 (16)	12.56
ANN-Ch-Uncons.	0.1406 ± 0.0237 (4)	0.0674 ± 0.0205 (4)	0.2406 ± 0.0167 (3)	0.0868 ± 0.0086 (7)	0.0638 ± 0.0080 (3)	0.2707 ± 0.0076 (7)	0.1686 ± 0.0077 (9)	0.0803 ± 0.0060 (6)	0.1795 ± 0.0082 (11)	6.00
ANN-Ch-Pos.	0.1340 ± 0.0289 (3)	0.0805 ± 0.0197 (8)	0.2488 ± 0.0130 (5)	0.2789 ± 0.0384 (18)	0.0652 ± 0.0074 (4)	0.2671 ± 0.0057 (4)	0.1706 ± 0.0069 (12)	0.1306 ± 0.0050 (11)	0.1742 ± 0.0152 (6)	7.89
ANN-Ch-Constr	0.1276 ± 0.0333 (1)	0.0706 ± 0.0151 (5)	0.2362 ± 0.0142 (1)	0.2071 ± 0.0138 (13)	0.0608 ± 0.0081 (1)	0.2716 ± 0.0074 (8)	0.1695 ± 0.0073 (11)	0.1222 ± 0.0051 (9)	0.1660 ± 0.0092 (1)	5.56
ANN-ELECTRE	0.1278 ± 0.0223 (2)	0.0175 ± 0.0144 (1)	0.3411 ± 0.0131 (18)	0.2348 ± 0.0270 (16)	0.0714 ± 0.0101 (7)	0.3073 ± 0.0080 (16)	0.1835 ± 0.0065 (16)	0.2569 ± 0.0298 (16)	0.1743 ± 0.0153 (7)	11.00
ANN-OWA	0.1477 ± 0.0208 (7)	0.1287 ± 0.0186 (16)	0.2455 ± 0.0125 (4)	0.2628 ± 0.0107 (17)	0.0708 ± 0.0062 (6)	0.2671 ± 0.0068 (5)	0.1816 ± 0.0066 (14)	0.2638 ± 0.0034 (17)	0.1802 ± 0.0076 (12)	10.89
ANN-TOPSIS	0.1671 ± 0.0231 (8)	0.1515 ± 0.0238 (17)	0.2551 ± 0.0157 (6)	0.1066 ± 0.0106 (8)	0.2674 ± 0.0133 (18)	0.2926 ± 0.0078 (11)	0.2270 ± 0.0086 (18)	0.0892 ± 0.0048 (7)	0.2125 ± 0.0073 (18)	12.33
CR	0.1713 ± 0.0424 (10)	0.0811 ± 0.0103 (9)	0.2775 ± 0.0335 (10)	0.0709 ± 0.0193 (1)	0.0682 ± 0.0129 (5)	0.2889 ± 0.0273 (9)	0.1499 ± 0.0122 (1)	0.0448 ± 0.0089 (3)	0.1725 ± 0.0120 (4)	5.78
LR	0.2124 ± 0.0650 (17)	0.0711 ± 0.0312 (6)	0.2893 ± 0.0240 (16)	0.0832 ± 0.0151 (6)	0.0733 ± 0.0107 (8)	0.2902 ± 0.0317 (10)	0.1655 ± 0.0082 (6)	0.1410 ± 0.0079 (12)	0.1729 ± 0.0122 (5)	9.56
KLR-ply	0.1695 ± 0.0437 (9)	0.0996 ± 0.0231 (15)	0.2760 ± 0.0243 (9)	0.0788 ± 0.0097 (4)	0.1488 ± 0.0278 (17)	0.3001 ± 0.0130 (15)	0.1627 ± 0.0119 (3)	0.0663 ± 0.0130 (5)	0.1960 ± 0.0160 (17)	10.44
KLR-rbf	0.1883 ± 0.0536 (12)	0.0802 ± 0.0292 (7)	0.2787 ± 0.0237 (11)	0.0772 ± 0.0107 (2)	0.0756 ± 0.0167 (9)	0.2934 ± 0.0112 (12)	0.1691 ± 0.0125 (10)	0.0618 ± 0.0151 (4)	0.1791 ± 0.0133 (10)	8.56
MORE	0.1932 ± 0.0511 (14)	0.0829 ± 0.0379 (10)	0.2827 ± 0.0255 (13)	0.0811 ± 0.0119 (5)	0.0838 ± 0.0241 (11)	0.3155 ± 0.0150 (17)	0.1707 ± 0.0186 (13)	0.0339 ± 0.0076 (1)	0.1764 ± 0.0137 (8)	10.22
LMT	0.1779 ± 0.0420 (11)	0.0850 ± 0.0256 (12)	0.2884 ± 0.0306 (15)	0.0773 ± 0.0148 (3)	0.0771 ± 0.0148 (10)	0.2963 ± 0.0126 (14)	0.1672 ± 0.0140 (7)	0.0432 ± 0.0116 (2)	0.1803 ± 0.0171 (13)	9.67
META	0.1897 ± 0.0423 (13)	0.0994 ± 0.0323 (14)	0.2824 ± 0.0273 (12)	0.2025 ± 0.0356 (11)	0.1042 ± 0.0171 (15)	0.2136 ± 0.0205 (2)	0.1674 ± 0.0187 (8)	0.1488 ± 0.0135 (13)	0.1697 ± 0.0087 (2)	10.00
MIP	0.1977 ± 0.0481 (15)	0.0900 ± 0.0345 (13)	0.2678 ± 0.0276 (8)	0.2080 ± 0.0326 (14)	0.1075 ± 0.0158 (16)	0.2093 ± 0.0174 (1)	0.1608 ± 0.0173 (2)	-	0.1716 ± 0.0140 (3)	10.00
UTADIS	0.2008 ± 0.0533 (16)	0.0652 ± 0.0362 (3)	0.2915 ± 0.0307 (17)	0.2225 ± 0.0318 (15)	0.0889 ± 0.0160 (13)	0.2368 ± 0.0187 (3)	0.1654 ± 0.0160 (5)	0.1300 ± 0.0142 (10)	0.1840 ± 0.0184 (15)	10.78
UTADIS-G	0.2136 ± 0.0460 (18)	0.0846 ± 0.0448 (11)	0.2852 ± 0.0245 (14)	0.2006 ± 0.0398 (10)	0.0903 ± 0.0153 (14)	0.3393 ± 0.0561 (18)	0.1943 ± 0.0247 (17)	0.1679 ± 0.0536 (14)	0.1820 ± 0.0188 (14)	14.44



Table 8

Classification performance in terms of the mean and standard deviation of 0/1 loss for 50% training data and 50% test data.

Method	DBS	CPU	BCC	MPG	ESL	ERA	LEV	CEV	MMG	Avg. rank
ANN-UTADIS	0.1093 ± 0.0353 (2)	0.0104 ± 0.0118 (2)	0.2276 ± 0.0165 (3)	0.1735 ± 0.0197 (9)	0.0546 ± 0.0099 (1)	0.2640 ± 0.0135 (4)	0.1566 ± 0.0110 (8)	0.1126 ± 0.0080 (8)	0.1717 ± 0.0192 (10)	5.22
ANN-PROMETHEE	0.1210 ± 0.0374 (6)	0.1381 ± 0.0569 (18)	0.2709 ± 0.0374 (11)	0.2031 ± 0.0232 (13)	0.0812 ± 0.0122 (14)	0.2900 ± 0.0138 (13)	0.1768 ± 0.0119 (14)	0.2273 ± 0.0138 (15)	0.1812 ± 0.0222 (16)	13.33
ANN-Ch-Uncons.	0.1178 ± 0.0306 (4)	0.0430 ± 0.0200 (4)	0.2196 ± 0.0199 (1)	0.0756 ± 0.0137 (7)	0.0582 ± 0.0114 (3)	0.2653 ± 0.0116 (7)	0.1605 ± 0.0113 (10)	0.0731 ± 0.0060 (6)	0.1711 ± 0.0138 (9)	5.67
ANN-Ch-Pos.	0.1207 ± 0.0313 (5)	0.0637 ± 0.0202 (11)	0.2391 ± 0.0254 (4)	0.2691 ± 0.0346 (17)	0.0605 ± 0.0121 (5)	0.2642 ± 0.0130 (5)	0.1648 ± 0.0125 (12)	0.1277 ± 0.0078 (10)	0.1687 ± 0.0161 (5)	8.22
ANN-Ch-Constr	0.1032 ± 0.0323 (1)	0.0562 ± 0.0155 (9)	0.2201 ± 0.0235 (2)	0.2015 ± 0.0207 (12)	0.0559 ± 0.0104 (2)	0.2673 ± 0.0117 (8)	0.1639 ± 0.0122 (11)	0.1187 ± 0.0083 (9)	0.1596 ± 0.0132 (1)	6.11
ANN-ELECTRE	0.1120 ± 0.0299 (3)	0.0101 ± 0.0111 (1)	0.3363 ± 0.0298 (17)	0.2335 ± 0.0390 (15)	0.0668 ± 0.0097 (6)	0.3075 ± 0.0150 (17)	0.1809 ± 0.0116 (16)	0.2568 ± 0.0187 (16)	0.1653 ± 0.0186 (2)	10.33
ANN-OWA	0.1363 ± 0.0311 (8)	0.1207 ± 0.0230 (16)	0.2395 ± 0.0216 (5)	0.2577 ± 0.0175 (16)	0.0677 ± 0.0112 (7)	0.2651 ± 0.0119 (6)	0.1787 ± 0.0104 (15)	0.2629 ± 0.0065 (17)	0.1770 ± 0.0124 (14)	11.56
ANN-TOPSIS	0.1480 ± 0.0343 (11)	0.1374 ± 0.0235 (17)	0.2453 ± 0.0196 (6)	0.1020 ± 0.0148 (8)	0.2678 ± 0.0371 (18)	0.2871 ± 0.0141 (11)	0.2246 ± 0.0119 (18)	0.0880 ± 0.0066 (7)	0.2093 ± 0.0130 (17)	12.56
CR	0.1572 ± 0.0416 (14)	0.0464 ± 0.0281 (5)	0.2687 ± 0.0282 (10)	0.0577 ± 0.0251 (1)	0.0601 ± 0.0126 (4)	0.2844 ± 0.0306 (9)	0.1372 ± 0.0125 (1)	0.0376 ± 0.0059 (4)	0.1667 ± 0.0144 (3)	5.67
LR	0.1708 ± 0.0380 (18)	0.0626 ± 0.0247 (10)	0.2799 ± 0.0245 (14)	0.0654 ± 0.0150 (2)	0.0704 ± 0.0113 (10)	0.2851 ± 0.0303 (10)	0.1651 ± 0.0133 (13)	0.1360 ± 0.0101 (12)	0.1701 ± 0.0158 (8)	10.78
KLR-ply	0.1333 ± 0.0333 (7)	0.0835 ± 0.0264 (15)	0.2591 ± 0.0287 (7)	0.0728 ± 0.0159 (4)	0.1023 ± 0.0225 (17)	0.2926 ± 0.0151 (14)	0.1520 ± 0.0160 (5)	0.0328 ± 0.0057 (3)	0.1721 ± 0.0164 (11)	9.22
KLR-rbf	0.1692 ± 0.0382 (17)	0.0547 ± 0.0233 (7)	0.2599 ± 0.0301 (8)	0.0744 ± 0.0151 (5)	0.0682 ± 0.0121 (8)	0.2882 ± 0.0142 (12)	0.1493 ± 0.0165 (4)	0.0463 ± 0.0086 (5)	0.1693 ± 0.0130 (7)	8.11
MORE	0.1457 ± 0.0413 (9)	0.0489 ± 0.0226 (6)	0.2640 ± 0.0288 (9)	0.0751 ± 0.0178 (6)	0.0695 ± 0.0139 (9)	0.3037 ± 0.0180 (16)	0.1486 ± 0.0157 (3)	0.0215 ± 0.0053 (2)	0.1691 ± 0.0140 (6)	7.33
LMT	0.1473 ± 0.0406 (10)	0.0674 ± 0.0243 (13)	0.2717 ± 0.0295 (12)	0.0672 ± 0.0164 (3)	0.0709 ± 0.0135 (11)	0.2956 ± 0.0148 (15)	0.1545 ± 0.0142 (6)	0.0174 ± 0.0069 (1)	0.1671 ± 0.0167 (4)	8.33
META	0.1623 ± 0.0469 (15)	0.0675 ± 0.0237 (14)	0.2750 ± 0.0317 (13)	0.1781 ± 0.0237 (11)	0.1004 ± 0.0186 (15)	0.2056 ± 0.0173 (2)	0.1592 ± 0.0122 (9)	0.1483 ± 0.0095 (14)	0.1732 ± 0.0151 (12)	11.67
MIP	0.1627 ± 0.0426 (16)	0.0640 ± 0.0239 (12)	-	-	0.1018 ± 0.0155 (16)	0.1958 ± 0.0137 (1)	0.1422 ± 0.0154 (2)	-	-	13.22
UTADIS	0.1480 ± 0.0421 (12)	0.0230 ± 0.0238 (3)	0.2854 ± 0.0246 (16)	0.2090 ± 0.0236 (14)	0.0783 ± 0.0163 (13)	0.2342 ± 0.0171 (3)	0.1556 ± 0.0132 (7)	0.1324 ± 0.0117 (11)	0.1758 ± 0.0152 (13)	10.22
UTADIS-G	0.1553 ± 0.0413 (13)	0.0555 ± 0.0328 (8)	0.2850 ± 0.0219 (15)	0.1753 ± 0.0251 (10)	0.0771 ± 0.0148 (12)	0.3305 ± 0.0491 (18)	0.1877 ± 0.0247 (17)	0.1430 ± 0.0436 (13)	0.1796 ± 0.0271 (15)	13.44

Table 9

Classification performance in terms of the mean and standard deviation of 0/1 loss for 80% training data and 20% test data.

Method	DBS	CPU	BCC	MPG	ESL	ERA	LEV	CEV	MMG	Avg. rank
ANN-UTADIS	0.0645 ± 0.0542 (1)	0.0046 ± 0.0137 (1)	0.2056 ± 0.0389 (3)	0.1587 ± 0.0324 (9)	0.0436 ± 0.0180 (1)	0.2527 ± 0.0210 (4)	0.1447 ± 0.0144 (4)	0.1081 ± 0.0154 (8)	0.1608 ± 0.0302 (7)	4.22
ANN-PROMETHEE	0.0932 ± 0.0580 (6)	0.1080 ± 0.0775 (17)	0.2656 ± 0.0591 (11)	0.1949 ± 0.0378 (13)	0.0757 ± 0.0251 (14)	0.2814 ± 0.0317 (11)	0.1706 ± 0.0212 (14)	0.2234 ± 0.0195 (15)	0.1691 ± 0.0235 (11)	12.44
ANN-Ch-Uncons.	0.0864 ± 0.0540 (3)	0.0266 ± 0.0265 (5)	0.1816 ± 0.0348 (1)	0.0614 ± 0.0218 (4)	0.0482 ± 0.0186 (3)	0.2556 ± 0.0260 (6)	0.1517 ± 0.0204 (8)	0.0672 ± 0.0119 (6)	0.1595 ± 0.0263 (6)	4.67
ANN-Ch-Pos.	0.0909 ± 0.0526 (5)	0.0385 ± 0.0261 (9)	0.2191 ± 0.0456 (5)	0.2669 ± 0.0470 (17)	0.0500 ± 0.0207 (4)	0.2552 ± 0.0245 (5)	0.1518 ± 0.0217 (9)	0.1238 ± 0.0147 (11)	0.1595 ± 0.0295 (5)	7.78
ANN-Ch-Constr	0.0673 ± 0.0516 (2)	0.0380 ± 0.0285 (8)	0.1909 ± 0.0412 (2)	0.1853 ± 0.0393 (12)	0.0455 ± 0.0178 (2)	0.2587 ± 0.0252 (7)	0.1538 ± 0.0208 (10)	0.1124 ± 0.0148 (9)	0.1486 ± 0.0221 (1)	5.89
ANN-ELECTRE	0.0868 ± 0.0553 (4)	0.0061 ± 0.0116 (2)	0.3200 ± 0.0423 (17)	0.2242 ± 0.0486 (15)	0.0593 ± 0.0207 (7)	0.3010 ± 0.0397 (17)	0.1777 ± 0.0205 (16)	0.2492 ± 0.0281 (16)	0.1551 ± 0.0243 (2)	10.67
ANN-OWA	0.1064 ± 0.0604 (7)	0.0973 ± 0.0433 (16)	0.2169 ± 0.0399 (4)	0.2583 ± 0.0426 (16)	0.0569 ± 0.0216 (6)	0.2589 ± 0.0249 (8)	0.1740 ± 0.0250 (15)	0.2588 ± 0.0144 (17)	0.1670 ± 0.0239 (10)	11.00
ANN-TOPSIS	0.1076 ± 0.0626 (8)	0.1180 ± 0.0461 (18)	0.2224 ± 0.0340 (6)	0.0890 ± 0.0271 (8)	0.2469 ± 0.0554 (18)	0.2789 ± 0.0236 (9)	0.2172 ± 0.0238 (18)	0.0814 ± 0.0086 (7)	0.1987 ± 0.0268 (17)	12.11
CR	0.1416 ± 0.0681 (13)	0.0212 ± 0.0301 (4)	0.2496 ± 0.0485 (7)	0.0551 ± 0.0160 (1)	0.0542 ± 0.0218 (5)	0.2813 ± 0.0280 (10)	0.1314 ± 0.0176 (1)	0.0273 ± 0.0089 (4)	0.1584 ± 0.0251 (3)	5.33
LR	0.1616 ± 0.0743 (17)	0.0640 ± 0.0335 (14)	0.2773 ± 0.0548 (14)	0.0611 ± 0.0263 (2)	0.0660 ± 0.0203 (10)	0.2843 ± 0.0302 (12)	0.1627 ± 0.0249 (13)	0.1328 ± 0.0173 (12)	0.1657 ± 0.0232 (9)	11.39
KLR-ply	0.1265 ± 0.0663 (10)	0.0754 ± 0.0372 (15)	0.2569 ± 0.0506 (8)	0.0727 ± 0.0268 (5)	0.0922 ± 0.0279 (15)	0.2918 ± 0.0290 (15)	0.1472 ± 0.0231 (5)	0.0286 ± 0.0075 (5)	0.1741 ± 0.0246 (15)	10.33
KLR-rbf	0.1343 ± 0.0672 (12)	0.0405 ± 0.0284 (10)	0.2598 ± 0.0529 (10)	0.0740 ± 0.0284 (7)	0.0657 ± 0.0229 (9)	0.2905 ± 0.0312 (13)	0.1496 ± 0.0233 (7)	0.0239 ± 0.0066 (3)	0.1696 ± 0.0271 (12)	9.22
MORE	0.1242 ± 0.0609 (9)	0.0412 ± 0.0299 (11)	0.2570 ± 0.0463 (9)	0.0737 ± 0.0269 (6)	0.0661 ± 0.0219 (11)	0.2988 ± 0.0276 (16)	0.1397 ± 0.0214 (3)	0.0190 ± 0.0070 (2)	0.1645 ± 0.0235 (8)	8.33
LMT	0.1433 ± 0.0667 (14)	0.0338 ± 0.0352 (6)	0.2707 ± 0.0554 (13)	0.0614 ± 0.0251 (3)	0.0691 ± 0.0228 (12)	0.2910 ± 0.0290 (14)	0.1474 ± 0.0232 (6)	0.0089 ± 0.0047 (1)	0.1595 ± 0.0283 (4)	8.11
META	0.1592 ± 0.0698 (16)	0.0640 ± 0.0304 (14)	0.2677 ± 0.0547 (12)	0.1686 ± 0.0369 (11)	0.1001 ± 0.0297 (16)	0.2031 ± 0.0250 (2)	0.1616 ± 0.0222 (12)	0.1506 ± 0.0166 (14)	0.1698 ± 0.0279 (13)	12.17
MIP	0.1480 ± 0.0811 (15)	0.0598 ± 0.0315 (12)	-	-	0.1008 ± 0.0247 (17)	0.1856 ± 0.0260 (1)	0.1359 ± 0.0185 (2)	-	-	13.22
UTADIS	0.1280 ± 0.0501 (11)	0.0152 ± 0.0214 (3)	0.2913 ± 0.0510 (15)	0.2080 ± 0.0388 (14)	0.0744 ± 0.0235 (13)	0.2356 ± 0.0292 (3)	0.1572 ± 0.0222 (11)	0.1336 ± 0.0167 (13)	0.1734 ± 0.0265 (14)	10.78
UTADIS-G	0.1683 ± 0.0667 (18)	0.0356 ± 0.0386 (7)	0.3016 ± 0.0478 (16)	0.1617 ± 0.0383 (10)	0.0656 ± 0.0228 (8)	0.3259 ± 0.0567 (18)	0.1781 ± 0.0253 (17)	0.1166 ± 0.0217 (10)	0.1778 ± 0.0246 (16)	13.33

preference learning algorithm and their ability to differentiate between these approaches. For example, the 0/1 loss values attained by ANN-UTADIS for CPU, ESL, and DBS are 0.0046, 0.0436, and 0.0645, indicating the inconsistencies in the suggested assignments only for a marginal share of test data. On the other extreme, these values for ERA and BCC are 0.2527 and 0.2056, respectively, confirming an incorrect classification for a significant share of alternatives. When it comes to the differences between average 0/1 losses for the best and worst-performing algorithms, they are the least for MMG, LEV, and DBS, while being the greatest for CEV, MPG, ESL, and BCC.

The most favorable average ranks implied by the 0/1 loss for the nine datasets are attained by:

- ANN-UTADIS (4.22), which has the least 0/1 loss for DBS, CPU, and ESL, while being ranked in the upper half of the ranking for all problems;
- ANN-Ch-Uncons. (4.67), which is at the top for BCC, while being ranked in the top six for 8 out of 9 datasets;
- CR (5.33), which attains the lowest mean of 0/1 loss for LEV and MPG,
- ANN-Ch-Constr. (5.89) ranked first for MMP and second for BDS, CPU, and ESL.

On the other extreme, the worst average ranks are attained by UTADIS-G (13.33), MIP (13.22), ANN-PROMETHEE (12.44), META (12.17), ANN-TOPSIS (12.11), LR (11.39), ANN-OWA (11.00), UTADIS (10.78), and ANN-ELECTRE (10.67). Note that the differences between the average ranks for the approaches in the lower half of the ranking are lesser than in the case of AUC.

When it comes to the direct comparison of the approaches using similar preference methods in terms of the 0/1 loss, ANN-UTADIS performs better than UTADIS for all datasets except ERA and better than UTADIS-G for all considered problems; ANN-ELECTRE is more advantageous than META only for 4 out of 9 problems, whereas the algorithms using the Choquet integral attain similar results for CPU, ESL, LEV, and MMG. Moreover, CR was worse than the ANN-Ch methods on DBS and BCC, whereas ANN-Ch-Constr. and ANN-Ch-Pos. were underperforming for MPG and CEV. On average, the latter approach attained the worst average rank among these four methods, most likely due to the least flexible model admitting only positive interactions for pairs of criteria.

With the decrease in the number of alternatives in the training set relative to the test set, the 0/1 loss increases for almost all methods (see Tables 7 and 8). For example, for ANN-UTADIS and DBS, its values are equal to 0.0645 for 80% training data, 0.1093 for 50%, and 0.1460 for 20%. The analogous results attained by ANN-Ch-Constr. for BCC are 0.1816, 0.2196, and 0.2406. The least performance deterioration can be observed for ANN-PROMETHEE and ANN-ELECTRE for BCC, MPG, ERA, LEV, and CEV. In particular, for PROMETHEE-ANN and BCC, the respective 0/1 losses are 0.2656 for 80/20, 0.2709 for 50/50, and 0.2659 for 20/80. In general, the average ranks for ANN-UTADIS, ANN-Ch-Uncons., and LR get slightly worse with the decrease of the training set's share, whereas the ranks for LR, META, and MIP exhibit an inverse trend. In the case of META and MIP, this can be explained by the greater efficiency of these algorithms when dealing with smaller data sizes. For example, for the 20/80 division, MIP identified the solutions for 8 out of 9 datasets, whereas for greater training sets, it failed to identify a sorting model for the additional three problems.

The conclusions derived from the analysis of the 0/1 loss agree with the ones formulated for AUC. On the one hand, ANN-UTADIS, ANN-Ch-Uncons., and CR are the best performing algorithms, whereas MIP, META, TOPSIS, OWA, UTADIS, and UTADIS-G attain the least advantageous results. A noticeable difference concerns the performance of ANN-ELECTRE and ANN-PROMETHEE, which are among the best approaches in terms of AUC but are

rated poorly when considering the 0/1 loss. This means that these two outranking-based methods correctly reproduce the preference relations for the vast majority of pairs of alternatives while making more mistakes concerning their classification. It can be explained given the nature of these methods and the learning process. ANN-ELECTRE and ANN-PROMETHEE incorporate the NFS procedure with a score for each alternative derived from pairwise comparisons against all remaining alternatives. However, these scores are transformed into assignments by comparing them with the class thresholds. It turns out that the threshold inferred for the training set might not generalize well for the test set, leading to the misclassification of alternatives, which attain scores close to the threshold. This is confirmed by Fig. 16, which indicates that for ERA, changing the threshold value for the test set rather than using the one inferred from the learning data might improve the 0/1 loss even by a few percent.

In the e-Appendix, we report the experimental results for the ANN-based algorithms in terms of the F1 score as well as the outcomes given different performance measures obtained for the training set.

## 7. Conclusions and future work

The availability of data resources helps individuals and groups mine helpful information and make better-informed decisions. The spectrum of practical problems that emphasize handling large quantities of data becomes more extensive. This requires the development of dedicated techniques. In recent years, an often emphasized aspect is that such methods should support both the explainability of recommended decisions and the interpretability of the entire decision-making process.

In this paper, we have considered the problem of processing data into explainable and interpretable models. This has been done in the context of preference learning. It consists of training the models on a set of alternatives for which the preferences are known/available and predicting the preferences for all other options. Specifically, we considered learning the parameters of monotonic sorting models from large sets of assignment examples. In this kind of problem, alternatives need to be assigned to predefined, preference-ordered classes in the presence of multiple, potentially conflicting criteria.

We have advocated the use of intuitive models inspired by the development in the field of MCDA. This is consistent with the recent trends in Machine Learning (Rudin, 2019). The considered models offer measures for (i) quantifying the role of individual criteria and subsets of criteria, (ii) understanding the impact of particular performances on the decision, (iii) gaining insights on which performance differences are negligible, significant, or critical, and (iv) capturing the strength of criteria coalitions sufficient for claiming that one alternative is at least as good as another. Moreover, the applied operators offer a mathematically sound and elegant manner for aggregating the arguments supporting each alternative's strengths and weaknesses. Also, the considered threshold-based sorting procedure is easily understandable and transparent in deriving the assignments by comparing alternatives' comprehensive scores with the separating class thresholds.

As a concrete Machine Learning application of these models, we have proposed Artificial Neural Networks as a computation technique for conducting preference disaggregation. ANNs have been used before for classification in the context of extensive data. However, the non-linear models they derived could not be interpreted by human Decision Makers nor accepted by domain experts. Thanks to the suitably adjusted components, units, and architecture, we have made ANNs suitable for learning highly explainable models.

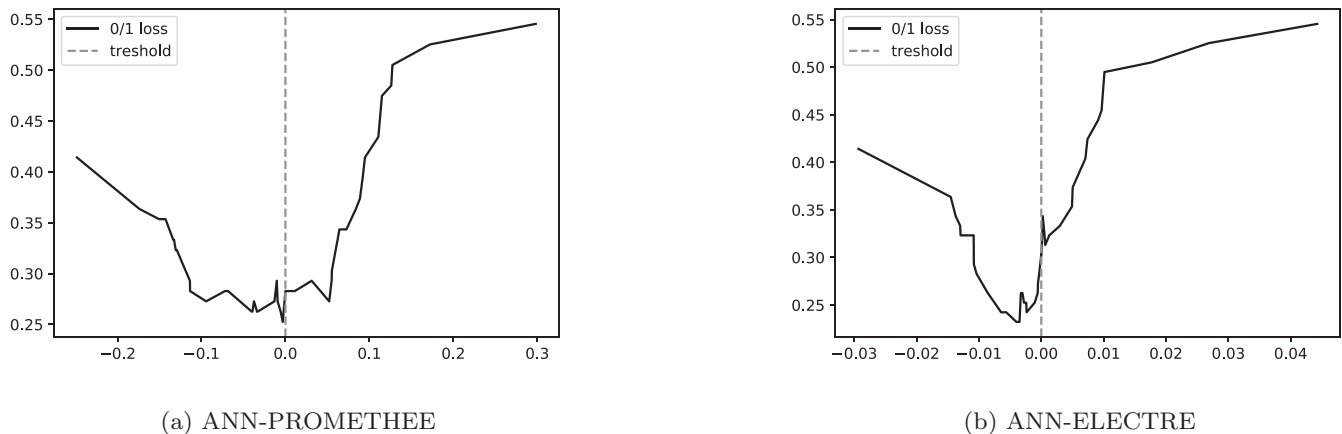


Fig. 16. The values of 0/1 loss (y-axis) for different separating class thresholds (x-axis) for the ERA problem.

The main benefits of the proposed preference learning algorithms are three-fold. First, we infer the parameters of the sorting models from decision examples, not requiring the Decision Maker to specify their values directly. We allow for simultaneous inference of all parameters of the sorting model, such as, e.g., criteria weights, concordance and discordance functions, and the comparison, veto, credibility, and separating class thresholds. This cannot be done efficiently with mathematical programming techniques that are traditionally applied in MCDA. Also, we avoid an arbitrary indication of meta-parameters such as shapes of preference functions or characteristic points of marginal values functions. In turn, we apply more general per-criterion (value, preference, concordance, or discordance) functions that offer greater flexibility in fitting the input data while maintaining the original spirit of MCDA.

Second, we contribute to the stream of making the MCDA methods suitable for handling inconsistent preference information that is too large to be dealt with by most traditional methods within an acceptable time. Sets of alternatives traditionally considered in MCDA consist of modestly-sized collections (Wallenius et al., 2008) and the development of the algorithms scaling up well with the number of alternatives has not been at the core of MCDA (Corrente et al., 2013). For example, the basic MCDA algorithms for dealing with inconsistency in the provided preference information are based on Mixed-Integer Linear Programming (MILP). Nonetheless, some existing MCDA and preference learning methods are capable of dealing with large inconsistent sets of assignment examples (see, e.g., Chandrasekaran et al., 2005; Dembczyński et al., 2009; Greco et al., 2001; Kotłowski & Słowiński, 2013; Manthoulis, Doumpos, Zopounidis, & Galariotis, 2020; Sobrie et al., 2019; Tehrani et al., 2012; Zopounidis & Doumpos, 2000). In this spirit, we demonstrate the feasibility of the proposed ANN-based approaches to the collections of over one thousand alternatives or the problems requiring comparing a few million pairs of alternatives. We know that the volume of datasets considered in some other sub-fields of ML is far more significant than in our experiments. Hence, demonstrating the usability of the proposed methods in areas typical for the ML applications remains a subject for future research. These include, e.g., finance, medicine, economy, and information retrieval, in which even some MCDA methods have been already used in the context of data sets with sizes exceeding those considered in this paper (e.g., bank failure prediction (Manthoulis et al., 2020), prognosis for hospice referral (Gil-Herrera et al., 2015), and recommender systems in numerous application domains (Manouselis & Costopoulou, 2007)).

Third, the extensive experiments on various benchmark problems indicate that the introduced algorithms are competitive in

terms of predictive accuracy. This is particularly true for the three approaches called ANN-UTADIS, ANN-Ch-Uncons., and ANN-PROMETHEE. They incorporate preference models in the form of an additive value function with generalized marginal functions, 2-additive Choquet integral admitting significant variability of weights, and an outranking relation combined with the Net Flow Score procedure. These methods perform well in terms of the AUC measure, which focuses on preserving pairwise preference relations. In addition, ANN-UTADIS and ANN-Ch-Uncons. score favorably also on the 0/1 loss, which is directly related to the classification accuracy. On average, the predictions made by these algorithms were slightly more accurate than the recommendations delivered by the state-of-the-art methods, including logistic regression and its generalizations, rule ensemble methods, approaches based on mathematical programming, and a dedicated metaheuristic for an outranking-based classification model. The advantage of the ANN-based methods derives from a few factors, including incorporating more general preference functions, efficient optimization methods, and techniques for increasing noise resistance, preventing overfitting, and reducing the impact of the information processing order on the attained results.

From a broader perspective, the variability of different algorithms proposed in this paper gives a chance for adjusting the sorting model to the provided preference information, as postulated in Hanne (1997). In particular, we considered score-, distance-, and outranking-based approaches that admit different compensation levels, interactions between criteria, or per-criterion risk attitudes or curvatures of marginal functions. In MCDA, such factors need to be considered when selecting a single method a priori. However, in the preference learning context, all presented neural networks can be aggregated in a single ANN that would, in the end, activate only the part and underlying approach leading to the most advantageous results that fit the available indirect preferences in the best way.

The directions for future research can be divided into experimental and methodological. The former ones derive from the limitations of our study. First, some data sets considered in the experimental comparison involve nominal attributes arbitrarily transformed into monotonic criteria as described in Tehrani et al. (2012). While this increases the difficulty of the preference learning task, such an interpretation neglects the original performance scales without preference directions. In this perspective, we perceive the need to further test the preference learning algorithms on real-world data with correctly defined criteria and increase the variety of publicly available properly designed benchmark data sets. Second, when testing the performance of algorithms, we run only those originally proposed in this paper. For the remaining

methods, we recalled the results reported in the respective works (e.g., Sobrie et al., 2019; Tehrani et al., 2012) on the same benchmark problems. This could be questioned concerning the optimization of hyperparameters which is an essential component of the experimental study. We performed it differently than in Sobrie et al. (2019) and Tehrani et al. (2012). In particular, the performance of some algorithms (e.g., UTADIS) for which the results were reported in other works could be improved if their hyperparameters were set more carefully. Given this limitation, we want to emphasize the need for adopting proper processes for optimizing the hyperparameters of MCDA methods in future studies that will focus on performing comparative analyses. In our understanding, successfully implementing this postulate requires making the source code of all so far proposed methods in the preference learning stream publicly available. Third, when optimizing the parameters of the sorting model, one could investigate the impact of other misclassification errors than a sum of regrets or different techniques than AdamW.

Regarding future research related to the development of other methods, we envisage the following four directions. First, we will propose neural preference learning algorithms for other intuitive MCDA approaches. The most appealing ones include the ELECTRE (Costa, Rui Figueira, Vieira, & Vieira, 2019) and PROMETHEE (Pelissari, Oliveira, Amor, & Abackerli, 2019) methods with boundary or characteristic class profiles and value-based approaches admitting interactions between criteria (Liu et al., 2021) and non-monotonicity (Liu et al., 2019) of marginal value functions. Second, it is possible to combine different methods within a single neural network and aggregate their results into a comprehensive quality measure. The form of an aggregation operator and the weights associated with scores delivered by various approaches could be learned during the optimization process (Hanne, 1997). Third, it would be interesting to verify the impact of using an ensemble of models that attained a pre-defined threshold of the classification error. In this paper, we only used the model that performed the best during learning. However, some other models were only slightly worse, and their joint use on the test set could increase the robustness of recommended assignments. Finally, an appealing idea consists of adjusting the preference learning algorithms to an online setting (Sahoo, Pham, Lu, & Hoi, 2018). Unlike batch learning applied in this paper, it assumes preferences are provided in sequential order, and the method needs to update the classification model at each step. This would correspond to a common MCDA scenario in which the DM provides preferences in successive iterations.

## Acknowledgments

The authors acknowledge financial support from the Polish National Science Center under the SONATA BIS project (grant no. DEC-2019/34/E/HS4/00045).

## Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2022.06.053

## References

- Alvarez, P. A., Ishizaka, A., & Martínez, L. (2021). Multiple-criteria decision-making sorting methods: A survey. *Expert Systems with Applications*, 183, 115368.
- Angilella, S., Corrente, S., Greco, S., & Słowiński, R. (2013). Multiple criteria hierarchy process for the Choquet integral. In R. Purshouse, P. Fleming, C. Fonseca, S. Greco, & J. Shaw (Eds.), *Evolutionary multi-criterion optimization – 7th international conference, EMO 2013, Sheffield, UK, March 19–22, 2013. Proceedings* (pp. 475–489). Springer.
- Brans, J. P., & De Smet, Y. (2016). PROMETHEE methods. In S. Greco, M. Ehrgott, & J. R. Figueira (Eds.), *Multiple criteria decision analysis: State of the art surveys* (pp. 187–219). New York, NY: Springer.
- Chandrasekaran, R., Ryu, Y. U., Jacob, V. S., & Hong, S. (2005). Isotonic separation. *INFORMS Journal on Computing*, 17(4), 462–474.
- Cinelli, M., Kadziński, M., Miebs, G., Gonzalez, M., & Słowiński, R. (2022). Recommending multiple criteria decision analysis methods with a new taxonomy-based decision support system. *European Journal of Operational Research*, 302(2), 633–651.
- Corrente, S., Greco, S., Kadziński, M., & Słowiński, R. (2013). Robust ordinal regression in preference learning and ranking. *Machine Learning*, 93(2), 381–422.
- Costa, A. S., Rui Figueira, J., Vieira, C. R., & Vieira, I. V. (2019). An application of the ELECTRE TRI-c method to characterize government performance in OECD countries. *International Transactions in Operational Research*, 26(5), 1935–1955.
- Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4), 303–314.
- Dembczyński, K., Kotłowski, W., & Słowiński, R. (2006). Additive preference model with piecewise linear components resulting from dominance-based rough set approximations. In L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, & J. M. Zurada (Eds.), *Artificial intelligence and soft computing – ICAISC 2006* (pp. 499–508). Berlin, Heidelberg: Springer.
- Dembczyński, K., Kotłowski, W., & Słowiński, R. (2009). Learning rule ensembles for ordinal classification with monotonicity constraints. *Fundamenta Informaticae*, 94, 163–178.
- Deng, L., & Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4), 197–387.
- Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv:1702.08608.
- Doumpos, M., Marinakis, Y., Marinaki, M., & Zopounidis, C. (2009). 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.
- Doumpos, M., & Zopounidis, C. (2004). Developing sorting models using preference disaggregation analysis: An experimental investigation. *European Journal of Operational Research*, 154(3), 585–598.
- Doumpos, M., & Zopounidis, C. (2011). Preference disaggregation and statistical learning for multicriteria decision support: A review. *European Journal of Operational Research*, 209(3), 203–214.
- Doumpos, M., & Zopounidis, C. (2018). Disaggregation approaches for multicriteria classification: An overview. In N. Matsatsinis, & E. Grigoroudis (Eds.), *Preference disaggregation in multiple criteria decision analysis: Essays in honor of Yannis Siskos* (pp. 77–94). Cham: Springer.
- Figueira, J., Greco, S., Roy, B., & Słowiński, R. (2013). An overview of ELECTRE methods and their recent extensions. *Journal of Multi-Criteria Decision Analysis*, 20(1–2), 61–85.
- Fürnkranz, J., & Hüllermeier, E. (2011). Preference learning: An introduction. In J. Fürnkranz, & E. Hüllermeier (Eds.), *Preference learning* (pp. 1–17). Berlin, Heidelberg: Springer.
- Gil-Herrera, E., Aden-Buie, G., Yalcin, A., Tsalatsanis, A., Barnes, L. E., & Djulbegovic, B. (2015). Rough set theory based prognostic classification models for hospice referral. *BMC Medical Informatics and Decision Making*, 15(1), 98.
- Goodman, B., & Flaxman, S. (2017). European union regulations on algorithmic decision-making and a “right to explanation”. *AI Magazine*, 38(3), 50–57.
- Greco, S., Matarazzo, B., & Słowiński, R. (2001). Rough sets theory for multicriteria decision analysis. *European Journal of Operational Research*, 129(3), 1–47.
- Greco, S., Mousseau, V., & Słowiński, R. (2010). Multiple criteria sorting with a set of additive value functions. *European Journal of Operational Research*, 207, 1455–1470.
- Guo, M., Zhang, Q., Liao, X., Chen, F. Y., & Zeng, D. D. (2021). A hybrid machine learning framework for analyzing human decision-making through learning preferences. *Omega*, 101, 102263.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1), 10–18.
- Hanne, T. (1997). Decision support for MCDM that is neural network-based and can learn. In J. Clímaco (Ed.), *Multicriteria analysis* (pp. 401–410). Berlin, Heidelberg: Springer.
- Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2000). *Applied logistic regression*. New York: Wiley.
- Hu, Y. C. (2009). Bankruptcy prediction using ELECTRE-based single-layer perceptron. *Neurocomputing*, 72(13–15), 3150–3157.
- Hwang, C. L., & Yoon, K. (1981). Multiple attribute decision making: Methods and applications a state-of-the-art survey. In *Methods for multiple attribute decision making* (pp. 58–191). Berlin, Heidelberg: Springer.
- Kadziński, M., & Szczepański, A. (2022). Learning the parameters of an outranking-based sorting model with characteristic class profiles from large sets of assignment examples. *Applied Soft Computing*, 116, 108312.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv:1412.6980.
- Köksalan, M., & Özpeynirci, S. B. (2009). An interactive sorting method for additive utility functions. *Computers and Operations Research*, 36(9), 2565–2572.
- Kotłowski, W., & Słowiński, R. (2013). On nonparametric ordinal classification with monotonicity constraints. *IEEE Transactions on Knowledge and Data Engineering*, 25(11), 2576–2589.
- Landwehr, N., Hall, M., & Frank, E. (2003). Logistic model trees. In N. Lavrac, D. Gamberger, H. Blockeel, & L. Todorovski (Eds.), *Machine learning: ECML 2003 – 14th european conference on machine learning, Cavtat-Dubrovnik, Croatia, September 22–26, 2003. Proceedings* (pp. 241–252). Springer.
- Leroy, A., Mousseau, V., & Pirlot, M. (2011). Learning the parameters of a multiple

- criteria sorting method. In R. Brafman, F. Roberts, & A. Tsoukias (Eds.), *Algorithmic decision theory – second international conference, ADT 2011, Piscataway, NJ, USA, October 26–28, 2011. Proceedings* (pp. 219–233). Springer.
- Linkov, I., Galaitis, S., Trump, B. D., Keisler, J. M., & Kott, A. (2020). Cybertrust: From explainable to actionable and interpretable artificial intelligence. *Computer*, 53(9), 91–96.
- Liu, J., Kadziński, M., Liao, X., & Mao, X. (2021). Data-driven preference learning methods for value-driven multiple criteria sorting with interacting criteria. *INFORMS Journal on Computing*, 33(2), 586–606.
- Liu, J., Liao, X., Kadziński, M., & Stowiński, R. (2019). Preference disaggregation within the regularization framework for sorting problems with multiple potentially non-monotonic criteria. *European Journal of Operational Research*, 276(3), 1071–1089.
- Loshchilov, I., & Hutter, F. (2018). Fixing weight decay regularization in adam. <https://openreview.net/forum?id=rk6qdGgCZ>.
- Malakooti, B., & Zhou, Y. Q. (1994). Feedforward artificial neural networks for solving discrete multiple criteria decision making problems. *Management Science*, 40(11), 1542–1561.
- Manouselis, N., & Costopoulou, C. (2007). Analysis and classification of multi-criteria recommender systems. *World Wide Web*, 10(4), 415–441.
- Manthoulis, G., Doumpos, M., Zopounidis, C., & Galariotis, E. (2020). An ordinal classification framework for bank failure prediction: Methodology and empirical evidence for US banks. *European Journal of Operational Research*, 282(2), 786–801.
- Molnar, C. (2020). *Interpretable machine learning*. Lulu Press.
- Mousseau, V., & Dias, L. (2004). Valued outranking relations in ELECTRE providing manageable disaggregation procedures. *European Journal of Operational Research*, 156(2), 467–482.
- Olteanu, A. L., & Meyer, P. (2014). Inferring the parameters of a majority rule sorting model with vetoes on large datasets. In V. Mousseau, & M. Pirlot (Eds.), *DA2PL 2014: From multicriteria decision aid to preference learning* (pp. 87–94).
- Pelissari, R., Oliveira, M. C., Amor, S. B., & Abackerli, A. J. (2019). A new flow-sort-based method to deal with information imperfections in sorting decision-making problems. *European Journal of Operational Research*, 276(1), 235–246.
- Roy, B. (2010). Two conceptions of decision aiding. *International Journal of Multicriteria Decision Making*, 1(1), 74–79.
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv:1609.04747.
- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), 206–215.
- Sahoo, D., Pham, Q., Lu, J., & Hoi, S. C. (2018). Online deep learning: Learning deep neural networks on the fly. In *Proceedings of the 27th international joint conference on artificial intelligence* (pp. 2660–2666).
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), 1–48.
- Sobrie, O., Mousseau, V., & Pirlot, M. (2019). Learning monotone preferences using a majority rule sorting model. *International Transactions in Operational Research*, 26(5), 1786–1809.
- Tehrani, A. F., Cheng, W., Dembczyński, K., & Hüllermeier, E. (2012). Learning monotone nonlinear models using the Choquet integral. *Machine Learning*, 89(1), 183–211.
- Waegeman, W., De Baets, B., & Boullart, L. (2009). Kernel-based learning methods for preference aggregation. *4OR*, 7(2), 169–189.
- Wallenius, J., Dyer, J. S., Fishburn, P. C., Steuer, R. E., Zionts, S., & Deb, K. (2008). Multiple criteria decision making, multiattribute utility theory: Recent accomplishments and what lies ahead. *Management Science*, 54(7), 1336–1349.
- Yager, R. R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1), 183–190.
- Zheng, S., Song, Y., Leung, T., & Goodfellow, I. (2016). Improving the robustness of deep neural networks via stability training. In L. Agapito, T. Berg, J. Kosecka, & L. Zelnik-Manor (Eds.), *2016 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4480–4488). IEEE.
- Zopounidis, C., & Doumpos, M. (2000). PREFDIS: A multicriteria decision support system for sorting decision problems. *Computers and Operations Research*, 27(7–8), 779–797.



## Publication [P4]

K. Martyn, M. Martyn, and M. Kadziński. PrefRank: a family of multiple criteria decision analysis methods for exploiting a valued preference relation. *Expert Systems With Applications*, 2023. Submitted





# PrefRank: a family of multiple criteria decision analysis methods for exploiting a valued preference relation

Krzysztof Martyn<sup>a,\*</sup>, Magdalena Martyn<sup>a</sup>, Miłosz Kadziński<sup>a</sup>

<sup>a</sup>*Institute of Computing Science, Faculty of Computing and Telecommunications, Poznan University of Technology, Piotrowo 2, 60-965 Poznań, Poland. e-mails: krzysztof.martyn@cs.put.poznan.pl, magdalena.martyn@cs.put.poznan.pl, milosz.kadziński@cs.put.poznan.pl*

---

## Abstract

We propose a family of approaches, called PrefRank, for exploiting a valued preference relation. They are inspired by the algorithms originally conceived for scoring the websites. The methods derive the strength and weakness of each alternative by analyzing how favorably it compares with the remaining ones. The introduced variants differ concerning the implemented weighting schemes when aggregating the out- or in-going preference degrees. We compare the results computed with the three variants of PrefRank and the state-of-the-art PROMETHEE methods on a broad spectrum of simulated problems. The similarity in the generated recommendations is quantified in view of the top-ranked alternatives, incomplete rankings, and complete orders. To demonstrate the methods' usability, we discuss the results of two studies concerning the fleet selection problem and the ranking of special economic zones in Poland. The latter is an original problem aiming to identify the zones that make the best use of their area and funds to provide significant financial profit and create many businesses or jobs.

*Keywords:* Multiple criteria decision analysis, Ranking, Valued preference relation, Exploitation procedure, Special economic zones

---

## 1. Introduction

The problem statement determines what type of recommendation is expected by the Decision Maker (DM) facing a particular decision challenge Cinelli et al. (2020). Choice and ranking belong to the most frequent real-world problems. The former aims at identifying the most preferred subset of alternatives, usually limited to one or a few options. The example choice problems concern supplier selection Govindan et al. (2017), engineering design of a chemical reactor Jaszkiwicz and Słowiński (1994), or siting of a nuclear power plant Keeney and Robilliard (1977). In turn, ranking is oriented toward ordering a set of alternatives from the best to the worst, using a relative comparison approach. Typical problems of this kind include ranking study programs or schools Wedlin (2007), ordering reuse strategies for adaptive heritage Bottero et al. (2019), or prioritizing countries in terms of their advancement in using information and communication technologies Siskos et al. (2014).

An inherent feature of the above problems is a multiple criteria evaluation of decision alternatives. For example, the suppliers are judged in terms of several economic, environmental, and social criteria Govindan et al. (2017), whereas constructing a ranking of study programs requires considering their reputation, quality of the university, and alumni career progress Wedlin (2007). Since the criteria are usually in conflict, one needs an operational aid to handle the complexity of real-world decision problems. Multiple Criteria Decision Analysis (MCDA) offers a plethora of approaches that differ in intuitiveness, properties, assumptions, and context use Cinelli et al. (2022); Watróbski et al.

---

\*Corresponding author: Institute of Computing Science, Faculty of Computing and Telecommunications, Poznan University of Technology, Piotrowo 2, 60-965 Poznań, Poland. e-mail: krzysztof.martyn@cs.put.poznan.pl; Tel. +48 61 665 3022; Fax: +48 61 8771525.

(2019). However, the most distinguishing aspect of these methods is the convention they employ for mathematical aggregation and exploitation of performances, reflecting the quality of alternatives on multiple criteria.

The aggregation methods can be described in view of their formal properties and empirical consequences. Two major techniques can be distinguished in the context of ranking and choice. On the one hand, some approaches derive a comprehensive score of each alternative by aggregating its performances using a value-, utility-, or distance-based procedure Cinelli et al. (2020). On the other hand, one may perform pairwise comparisons and establish a preference relation in the set of alternatives. Such a relation may be either crisp or valued Szelaĝ et al. (2014). The former represents only the presence or absence of preference, whereas the latter allows degrees of membership, indicating how strongly the relation is established.

Let us focus on a valued preference relation. Its interpretation differs depending on the context use Szelaĝ et al. (2014). In particular, it may capture the percentage of voters or stakeholders declaring that one alternative is better than another Liu et al. (2020). Furthermore, it may reflect the share of feasible instances of the preference model confirming the preference Kadziński and Tervonen (2013). Alternatively, it may quantify the strength or credibility of the assertion “ $a$  is at least as good or more preferred to  $b$ ” with the proviso that the relation’s fuzziness may result from applying multiple criteria and/or uncertain performances in the form of, e.g., fuzzy numbers Brans and De Smet (2016); Figueira et al. (2016); Geldermann et al. (2000). Ultimately, the quality of each alternative depends on its relations with all the other alternatives Szelaĝ et al. (2014). Hence the matrix of a valued relation needs to be exploited to develop the final recommendation given a ranking or a choice problem statement.

There exist various procedures for exploiting valued preference relations. The most appealing one computes the entering and leaving flows, which summarize the strength and weakness of each alternative in comparison with all remaining ones Brans and De Smet (2016). There exist other scoring procedures, which differ in terms of three aspects: (i) accounting for arguments only in favor or against each alternative, or both of them jointly, (ii) aggregation operator (min, max, or sum) used to aggregate the results of elementary pairwise comparisons into a comprehensive measure of desirability, and (iii) iterative application of the choice function to break ties between subsets of alternatives attaining the same score Szelaĝ et al. (2014). Furthermore, distillation procedures compute the quality of each alternative and iteratively add them to the constructed order until considering the entire set Corrente et al. (2017). In the downward (upward) distillation, the ranking is constructed in a top-down (bottom-up) fashion, retaining alternatives with the greatest (least) quality first. In Dias and Lamboray (2010), the prudence principle has been extended in the exploitation model, constructing a ranking that maximizes the weakest support for its implicit pairwise comparison. Moreover, Leyva-Lopez and Aguilera-Contreras (2005) proposed an evolutionary algorithm that minimizes the number of alternatives that are ranked higher despite smaller values of an outranking relation when compared to some lower-ranked alternatives. In some scenarios, a set of valued relations is considered, e.g., graded outranking relations Szelaĝ et al. (2014) or the validity of preference, indifference, and incomparability Wang (2001). They can be analyzed independently or aggregated into a single fuzzy relation over a set of alternatives. Other exploitation procedures can be found, e.g., in Perny and Roy (1992), Szelaĝ et al. (2014), and Wang and Li (2018).

In this paper, we focus on a valued preference relation capturing the strength of a coalition of criteria supporting that one alternative is more preferred to another. Moreover, we consider the Net Flow Rule that derives the strength and weakness of each alternative by summarizing how strongly it is preferred to other alternatives and the degree to which other alternatives are preferred to it. Such a setting has been adopted in the PROMETHEE method Brans et al. (1984). It has been applied to a variety of real-world problems such as, e.g., evaluation of urban regeneration processes, prioritization of green suppliers, assessment of development scenarios for the power generation sector, or ranking of enterprises according to their business efficiency level. For a comprehensive review of applications of PROMETHEE, see Behzadian et al. (2010). Also, the method has been revised in numerous ways to handle more complex decision scenarios. The most notable extensions concern admitting fuzzy performances and weights Geldermann et al. (2000); Ziemba (2021), providing visual and interactive decision aiding tools Mareschal and De Smet (2009), tolerating imprecise or indirect preferences and conducting robustness analysis Corrente et al. (2014a,b); Kadziński et al. (2012); Lolli

et al. (2019); Pelissari and Duarte (2022); Ziemba (2021), handling hierarchical structures Arcidiacono et al. (2018), interactions between criteria Corrente et al. (2014a), and discordance effect Hu and Chen (2011), and addressing sorting Angilella and Pappalardo (2021); Pelissari and Duarte (2022) or group decision making Macharis et al. (1998).

The numerous practical applications and methodological advancements confirm the status of PROMETHEE as one of the most important methods in MCDA. Moreover, its aggregation procedure enjoys desirable theoretical properties Dejaegere et al. (2022). However, intuitively, the computation of entering and leaving flows by simply aggregating the preference degrees derived from pairwise comparisons with all remaining alternatives can be criticized. The Net Flow Rule does not consider the preference graph's structure. In particular, being preferred to a relatively good alternative to the same degree as to a relatively worse solution adds the same value to the alternative's comprehensive strength. In the same spirit, being worse than some highly favorable and clearly disadvantageous options counts equally to the alternative's overall weakness. Hence, the results of all pairwise comparisons are assigned the same discriminative power that is proportional only to the preference index values.

This paper presents a novel family of procedures, called *PrefRank*, exploiting a value preference relation. When computing the strength and weakness of each alternative, they account for both the preference degrees when compared to other alternatives and the relative qualities of these alternatives. In this way, the impact of arguments in favor of or opposing a given option is differentiated based on the strength and weakness of the alternative it is compared to. This general idea is implemented differently in three approaches.

First, when computing the strength of each alternative, we appreciate being preferred over relatively good rather than bad alternatives. In turn, when calculating the weakness, we perceive it as a greater disadvantage to be outranked by relatively weak rather than strong alternatives. Second, we postulate that a strong alternative should be heavily preferred over weak solutions. Analogously, a weak option is the one vastly outranked by highly favorable alternatives. Thirdly, we propose that the alternative's power is great if it is preferred to alternatives that are outranked by other good solutions. In turn, the alternative's flaw is high if it is outranked by solutions that are preferred to other weak options.

The above ideas are inspired by the algorithms originally conceived for scoring web pages: PageRank Page et al. (1999), HITS Kleinberg (1999), and SALSA Lempel and Moran (2000). For many years, PageRank served as the basis for evaluating pages in the Google search engine. It exploits the links between the websites and considers their qualities to estimate how important the website is. The algorithm promotes these pages, which are linked by many other important pages. This idea has been subsequently adapted to various science fields ranging from exact sciences and medicine through monitoring systems, scientometrics, and sociology to sports and robotics Coppola et al. (2019); Gleich (2015); Kwak et al. (2010). Furthermore, HITS distinguishes two roles each website can play: hub or authority Kleinberg (1999). The algorithm assigns two scores for each page, exhibiting a mutually reinforcing relationship. The authority score quantifies the value of the page's content. A good authority needs to be linked by many good hubs, being regarded as a meaningful source for a particular topic. The hub value captures the value of each page's links to other sites. A good hub points to many good authorities. SALSA is similar to HITS in terms of computing authorities and hubs scores Lempel and Moran (2000). However, its computational procedure is inspired by PageRank. It considers a bipartite graph with one set containing hub pages and the other containing authority pages but admitting each page belongs to both sets. The resulting scores are based on the contributions of second-degree neighbors, which makes the authority and hub assessments more independent. The proposed methods are used to derive the strengths and weaknesses of all alternatives, hence associating a pair of numerical values with each option. They can be combined into a partial or a complete ranking, similarly as in PROMETHEE I and II Brans et al. (1984).

Our second contribution consists of developing open-source software implementing the introduced methods. This makes them ready for use by a wide spectrum of users. Since the programming module reads input in a dedicated XML format, called XMCDAs Meyer and Bigaret (2012), we also benefit from the flexibility in designing the methodological flows by combining various MCDA procedures. In particular, the criteria weights needed to establish valued preference relation can be derived from various methods, including the revised Simos procedure Figueira and Roy (2002), the Best-

Worst Method Rezaei (2015), Analytical Hierarchy Process Saaty (1990), or surrogate weights de Almeida Filho et al. (2018). In the same spirit, the comprehensive preference degrees can be derived as initially proposed in PROMETHEE. However, they can be enriched by considering the reinforced preference effect Roy and Słowiński (2008), interactions between criteria Figueira et al. (2009), or discordance effect Hu and Chen (2011). In this way, the users can adjust the components to their needs.

Finally, we illustrate the use of the proposed methods in a study concerning the comprehensive evaluation of special economic zones. There are fourteen such areas in Poland offering favorable investment conditions. We consider their performances given five criteria: total area, capital expenditures, number of jobs, business permits, and financial results. They are aggregated into a comprehensive ranking using the PrefRank procedures. Moreover, we report the results of an extensive comparison of PrefRank and PROMETHEE. We consider problems with different numbers of alternatives and criteria and simulated DM's preferences in the form of preference functions and criteria weights. The outcomes indicate the similarities between the considered methods in terms of recommending only the most preferred alternative or the entire ranking.

The remainder of this paper is organized as follows. Section 2 introduces the notations and reminds the PROMETHEE methods. In Section 3, we discuss the novel scoring methods. Section 4 deals with measures used for comparing the recommendations obtained with various approaches. In Section 5, we illustrate the use of PrefRank in a real-world example concerning the prioritization of bus models. Section 6 discusses the results of a comparison between PrefRank and PROMETHEE. In Section 7, we present the software and demonstrate its use in the evaluation of Polish special economic zones. The last section concludes the paper.

## 2. Notation and reminder on PROMETHEE

We consider a finite set  $A = \{a_1, \dots, a_n\}$  of  $n$  alternatives evaluated on a family  $G = \{g_1, \dots, g_m\}$  of  $m$  criteria. The DM aims at ranking the alternatives from the best to the worst according to their performances  $g_j(a_i)$  on  $g_j : A \rightarrow \mathbb{R}$ ,  $j \in J = \{1, \dots, m\}$ . We assume, without loss of generality, that the larger  $g_j(a_i)$ , the more preferred the alternative  $a_i$  on  $g_j$ .

The PROMETHEE methods compare the alternatives pairwise. For each criterion  $g_j \in G$  and each pair  $(a_i, a_k) \in A \times A$ , the performance difference is translated into the marginal preference index (degree):

$$\pi_j(a_i, a_k) = F_j(g_j(a_i) - g_j(a_k)) \in [0, 1]. \quad (1)$$

A typical marginal preference function, incorporating a pair of thresholds, is shown in Figure 1. An indifference threshold  $q_j$  is the maximal performance difference for which a pair of alternatives is indifferent. In contrast, a preference threshold  $p_j$  is the minimal performance difference inducing a strict preference of one alternative over another. For other standard types of preference functions used in PROMETHEE, see Brans and De Smet (2016).

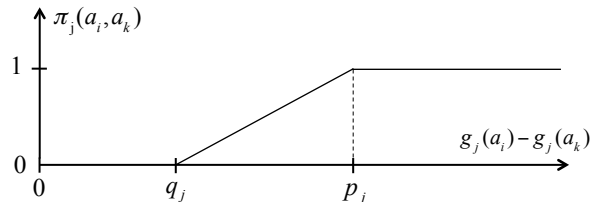


Figure 1: Typical marginal preference function used in PROMETHEE.

The marginal preference degrees are aggregated using a weighted average into a comprehensive preference degree:

$$\pi(a_i, a_k) = \frac{\sum_{j=1}^m w_j \pi_j(a_i, a_k)}{\sum_{j=1}^m w_j}, \quad (2)$$

where  $w_j$  is a weight associated with criterion  $g_j$ ,  $j \in J$ . Thus computed  $\pi(a_i, a_k)$  can be interpreted as a valued preference relation. The greater  $\pi(a_i, a_k)$ , the more significant the support of the family of criteria  $G$  to the assertion that  $a_i$  is preferred to  $a_k$ . For each pair of alternatives  $(a_i, a_k) \in A \times A$ ,  $\pi(a_i, a_k) \in [0, 1]$  and  $0 \leq \pi(a_i, a_k) + \pi(a_k, a_i) \leq 1$ . Moreover, for each  $a_i \in A$ ,  $\pi(a_i, a_i) = 0$ .

Finally, PROMETHEE computes the positive  $\phi^+(a_i)$  and negative  $\phi^-(a_i)$  flows for each  $a_i \in A$  by aggregating arguments supporting its strength and weakness, respectively:

$$\phi^+(a_i) = \frac{1}{n-1} \sum_{k=1}^n \pi(a_i, a_k), \quad (3)$$

$$\phi^-(a_i) = \frac{1}{n-1} \sum_{k=1}^n \pi(a_k, a_i). \quad (4)$$

The positive flow  $\phi^+(a_i)$  indicates how much, on average,  $a_i$  is preferred to other alternatives, while the negative flow  $\phi^-(a_i)$  captures how much, on average, other alternatives are preferred to  $a_i$ . The results of the above aggregations depend only on the degrees of comprehensive preference. In particular, they do not consider whether the alternatives over which  $a_i$  is preferred or those that are preferred to  $a_i$  are strong or weak. In other words, PROMETHEE neglects the difficulty or easiness of being preferred to other alternatives. Still, the method employs the positive and negative flows to impose a complete or partial order on the set of alternatives.

### 3. PrefRank: a family of scoring methods exploiting valued preference relation

In this section, we present a family of procedures exploiting a valued preference relation  $\pi(a_i, a_k)$ ,  $a_i, a_k \in A$ , called *PrefRank*. They compute the strength  $S^+(a_i)$  and weakness  $S^-(a_i)$  for each alternative  $a_i \in A$  by aggregating the preference degrees, similarly to the positive and negative flows in PROMETHEE. These factors can be used to rank the alternatives or aggregated into a comprehensive quality seen as the balance between strength and weakness:

$$S(a_i) = S^+(a_i) - S^-(a_i). \quad (5)$$

However, there are a few notable differences when compared to PROMETHEE. The strengths and weaknesses of all alternatives in PrefRank are normalized to sum up to one. In other words, there is a unary pool of strength or weakness to be distributed. From a computational viewpoint, we consider elementary strength  $\phi^+(a_i)$  and weakness  $\phi^-(a_i)$  for each  $a_i \in A$ , which are normalized by the sum of strengths and weaknesses attained by all considered options:

$$S^+(a_i) = \frac{\phi^+(a_i)}{\sum_{k=1}^n \phi^+(a_k)} \text{ and } S^-(a_i) = \frac{\phi^-(a_i)}{\sum_{k=1}^n \phi^-(a_k)}. \quad (6)$$

More importantly, when calculating  $\phi^+(a_i)$  and  $\phi^-(a_i)$ , we aggregate the preference degrees using weighted sums rather than a simple sum of preference degrees. At this stage, we refer to the relative qualities of alternatives to which a given option is compared. Precisely, the impact of arguments in favor of or in opposition to a given option is differentiated based on the strengths and weaknesses of the alternatives it is compared against, i.e.:

$$\phi^+(a_i) = \sum_{k=1}^n \pi(a_i, a_k) \cdot \omega^+(a_k) \text{ and } \phi^-(a_i) = \sum_{k=1}^n \pi(a_k, a_i) \cdot \omega^-(a_k), \quad (7)$$

where  $\omega^+(a_k)$  and  $\omega^-(a_k)$  are weights assigned to each  $a_k \in A$ , expressing the relative difficulty in other alternatives being preferred to  $a_k$  and the power of  $a_k$  in being preferred to other alternatives, respectively. In this way, when aggregating the arguments from one-against-one comparisons, some preference degrees can be strengthened, while others can be weakened.

This general idea is implemented differently in three approaches, varying in how the weights  $\omega^+(a_k)$  and  $\omega^-(a_k)$  are computed. In what follows, we discuss the idea underlying the variants of PrefRank, distinguished with Roman numerals (I, II, and III). We illustrate these approaches on a didactic problem.

### 3.1. PrefRank I

When calculating the strength of each alternative in PrefRank I, we appreciate being preferred to relatively good rather than bad alternatives. That is, if an alternative is better than some other option which, in turn, is preferred – to a significant degree – over all or the majority of other solutions, this should imply some bonus. On the contrary, being preferred to a relatively poor alternative that, on its own, does not prove its superiority over other solutions, should not significantly increase the alternative’s strength.

When calculating the weakness of each alternative, we perceive it as a more significant disadvantage to be outranked by relatively weak rather than strong alternatives. In this interpretation, if an alternative is worse than some other option which, in turn, is strongly outpreferred by many other solutions, this should lead to a significant penalty. However, proving worse than some strong alternative revealing no or limited deficiencies when other options are compared against it should not add much to the alternative’s weakness. These desired effects can be attained by multiplying the valued outranking relation by the following weights:

$$\omega^+(a_k) = S^+(a_k) \text{ and } \omega^-(a_k) = S^-(a_k). \quad (8)$$

Overall, the strength of each alternative depends on the degrees confirming its preference over other alternatives and the strengths of these options. In turn, its weakness derives from how much other alternatives are preferred to it and the weaknesses of these solutions.

The interpretation of strengths and weaknesses in PrefRank I can be explained from the perspective of a valued preference graph in which the nodes correspond to the alternatives and the arcs are associated with out- or in-going preference degrees. The strength is equivalent to the probability of reaching a given vertex by following a random walk in a preference graph such that the probability of moving from  $a_i$  to  $a_k$  is equal to  $\pi(a_i, a_k)$ . Optionally,  $S^+(a_i)$  can also be interpreted as a result of an alternative voting system where the voting powers are equal to the preference degrees. Analogously, the weakness derives from a random walk in a graph with the move’s probability between  $a_i$  and  $a_k$  equal to  $\pi(a_k, a_i)$ . This interpretation is consistent with the one initially proposed in PageRank Page et al. (1999), where nodes stand for the web pages or documents and the arcs correspond to the links.

Due to the interdependence between the strengths and weaknesses of various alternatives, they are computed using an iterative method. In the beginning, we assume that they are equal for all alternatives, i.e.,  $S^+(a_i) = S^-(a_i) = \frac{1}{n}$ . Then, the computations are conducted as indicated by Eqs. 6, 7, and 8, and repeated with the updated values until the differences between the strengths and weaknesses in the successive iterations are negligible (i.e., lower than pre-defined accuracy threshold).

To illustrate the idea underlying PrefRank I, let us consider an example preference matrix for five alternatives  $a_1 - a_5$  (see Table 1). The obtained strengths, weaknesses, and qualities are reported in Table 2. It also contains intermediate (non-normalized) factors. In what follows, we explain the interdependencies between various strengths and weaknesses while focusing on  $a_1$  and  $a_5$ .

Table 1: Example preference matrix for the problem involving five alternatives.

$\pi(a_i, a_k)$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$a_1$	0.0	1.0	1.0	0.0	0.0
$a_2$	0.0	0.0	1.0	0.0	0.0
$a_3$	0.0	0.0	0.0	0.0	1.0
$a_4$	0.0	0.0	0.0	0.0	0.5
$a_5$	0.0	0.0	0.0	0.5	0.0

The non-normalized strength of  $a_1$  is expressed as follows:

$$\begin{aligned}\phi^+(a_1) &= \pi(a_1, a_1) \cdot S^+(a_1) + \pi(a_1, a_2) \cdot S^+(a_2) + \pi(a_1, a_3) \cdot S^+(a_3) + \pi(a_1, a_4) \cdot S^+(a_4) + \pi(a_1, a_5) \cdot S^+(a_5) = \\ &= 0 \cdot S^+(a_1) + 1 \cdot S^+(a_2) + 1 \cdot S^+(a_3) + 0 \cdot S^+(a_4) + 0 \cdot S^+(a_5) = S^+(a_2) + S^+(a_3) = 0.2 + 0.1 = 0.3.\end{aligned}$$

Hence it derives from the strengths of  $a_2$  and  $a_3$ , which are fully outperformed by  $a_1$ . In turn, its non-normalized weakness  $\phi^-(a_1)$  is zero because no other alternative is preferred to  $a_1$  with a positive degree. Computing the final strength and weakness of  $a_1$  requires normalizing through the sum of strengths and weaknesses of all alternatives:

$$S^+(a_1) = \frac{\phi^+(a_1)}{\sum_{k=1}^n \phi^+(a_k)} = \frac{0.3}{0.3 + 0.1 + 0.05 + 0.025 + 0.025} = 0.6,$$

$$S^-(a_1) = \frac{\phi^-(a_1)}{\sum_{k=1}^n \phi^-(a_k)} = \frac{0.0}{0.0 + 0.0 + 0.0 + 0.25 + 0.25} = 0.0.$$

As a result, the quality  $S$  of  $a_1$  is  $S(a_1) = S^+(a_1) - S^-(a_1) = 0.6 - 0.0 = 0.6$ . When it comes to  $a_5$ , its non-normalized strength is  $\phi^+(a_5) = S^+(a_4) \cdot \pi(a_5, a_4) = 0.05 \cdot 0.5 = 0.025$ . After normalization, it amounts to:

$$S^+(a_5) = \frac{\phi^+(a_5)}{\sum_{k=1}^n \phi^+(a_k)} = \frac{0.025}{0.3 + 0.1 + 0.05 + 0.025 + 0.025} = 0.05.$$

In turn, the non-normalized weakness of  $a_5$  is:

$$\begin{aligned}\phi^-(a_5) &= \pi(a_1, a_5) \cdot S^-(a_1) + \pi(a_2, a_5) \cdot S^-(a_2) + \pi(a_3, a_5) \cdot S^-(a_3) + \pi(a_4, a_5) \cdot S^-(a_4) + \pi(a_5, a_5) \cdot S^-(a_5) = \\ &= 0 \cdot S^-(a_1) + 0 \cdot S^-(a_2) + 1 \cdot S^-(a_3) + 0.5 \cdot S^-(a_4) + 0 \cdot S^-(a_5) = S^-(a_3) + 0.5 \cdot S^-(a_4) = 0 + 0.5 \cdot 0.5 = 0.25.\end{aligned}$$

Its normalized counterpart is computed in the following way:

$$S^-(a_5) = \frac{\phi^-(a_5)}{\sum_{k=1}^n \phi^-(a_k)} = \frac{0.25}{0.0 + 0.0 + 0.0 + 0.25 + 0.25} = 0.5$$

Finally, the quality of  $a_5$  is  $S(a_5) = S^+(a_5) - S^-(a_5) = 0.05 - 0.5 = -0.45$ .

Table 2: Strengths, weaknesses, and qualities derived by the three variants of PrefRank for the example problem.

Algorithm	Value	Normalized					Value	Non-normalized				
		$a_1$	$a_2$	$a_3$	$a_4$	$a_5$		$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
PrefRank I	$S^+$	0.600	0.200	0.100	0.050	0.050	$\phi^+$	0.300	0.100	0.050	0.025	0.025
	$S^-$	0.000	0.000	0.000	0.500	0.500	$\phi^-$	0.000	0.000	0.000	0.250	0.250
	$S$	0.600	0.200	0.100	-0.450	-0.450						
PrefRank II	$S^+$	0.618	0.382	0.000	0.000	0.000	$\phi^+$	1.000	0.618	0.000	0.000	0.000
	$S^-$	0.000	0.382	0.618	0.000	0.000	$\phi^-$	0.000	0.618	1.000	0.000	0.000
	$S$	0.618	0.000	-0.618	0.000	0.000						
PrefRank III	$S^+$	0.267	0.133	0.267	0.133	0.200	$\phi^+$	0.267	0.133	0.267	0.133	0.200
	$S^-$	0.000	0.167	0.333	0.250	0.250	$\phi^-$	0.000	0.167	0.333	0.250	0.250
	$S$	0.267	-0.033	-0.067	-0.117	-0.050						

### 3.2. PrefRank II

The weighting scheme implemented in PrefRank II is inverse with respect to the one adopted in PrefRank I. On the one hand, a strong alternative should be heavily preferred over weak solutions. Precisely, the alternative's strength is computed as the weighted sum of preference degrees with weights interpreted as the weaknesses of solutions it is compared against. Thus the greater the arguments confirming the advantage of the evaluated alternative and the weaker the options with which it is confronted, the stronger the alternative. On the other hand, a weak alternative is

the one vastly outranked by strong alternatives. Hence the alternative's weakness is computed as the weighted sum of preference degrees with weights interpreted as the strengths of solutions that are compared with it. As a result, the weakness increases with the increase in both the in-going preference degrees and the strengths of other solutions. These effects can be attained by using the following weights:

$$\omega^+(a_k) = S^-(a_k) \text{ and } \omega^-(a_k) = S^+(a_k). \quad (9)$$

The above interpretation derives from the HITS algorithm Kleinberg (1999), originally analyzing the links between web pages. This approach distinguishes two roles each page can play: an authority (an authoritative source of information) and a hub (a compilation of a broad catalog of information that lead users directly to other authoritative pages). In this perspective, a page is a good hub if it points to good authorities, and it is a good authority when linked by good hubs. In our adaptation, the alternative's strength is similar to a hub score, and the weakness is similar to an authority score. The computations are conducted using an iterative method according to Eqs. 6, 7, and 9.

Let us illustrate the use of PrefRank II while referring to an example problem introduced in Section 3.1. The results are presented in Table 2. The non-normalized strength of  $a_1$  is expressed as follows:

$$\begin{aligned} \phi^+(a_1) &= \pi(a_1, a_1) \cdot S^-(a_1) + \pi(a_1, a_2) \cdot S^-(a_2) + \pi(a_1, a_3) \cdot S^-(a_3) + \pi(a_1, a_4) \cdot S^-(a_4) + \pi(a_1, a_5) \cdot S^-(a_5) = \\ &= 0 \cdot S^-(a_1) + 1 \cdot S^-(a_2) + 1 \cdot S^-(a_3) + 0 \cdot S^-(a_4) + 0 \cdot S^-(a_5) = S^-(a_2) + S^-(a_3) = 0.382 + 0.618 = 1.0. \end{aligned}$$

The normalized strength is  $S^+(a_1) = \frac{1.0}{1.0+0.618+0.0+0.0+0.0} = 0.618$ . Hence  $a_1$  is judged relatively strong because the alternatives over which it is preferred are relatively weak. In turn, its weakness  $S^-(a_1)$  is zero because no other alternative is preferred to  $a_1$ . The quality of  $a_1$  is  $S(a_1) = S^+(a_1) - S^-(a_1) = 0.618 - 0.0 = 0.618$ .

As far as  $a_5$  is concerned, its non-normalized strength is nullified due to the zero weakness of  $a_4$  that is the only alternative to which  $a_5$  is preferred with a positive degree, i.e.,  $\phi^+(a_5) = \pi(a_5, a_4) \cdot S^-(a_4) = 0.5 \cdot 0 = 0$ . In turn, the non-normalized weakness of  $a_5$  is:

$$\begin{aligned} \phi^-(a_5) &= \pi(a_1, a_5) \cdot S^+(a_1) + \pi(a_2, a_5) \cdot S^+(a_2) + \pi(a_3, a_5) \cdot S^+(a_3) + \pi(a_4, a_5) \cdot S^+(a_4) + \pi(a_5, a_5) \cdot S^+(a_5) = \\ &= 0 \cdot S^+(a_1) + 0 \cdot S^+(a_2) + 1 \cdot S^+(a_3) + 0.5 \cdot S^+(a_4) + 0 \cdot S^+(a_5) = S^+(a_3) + 0.5 \cdot S^+(a_4) = 0 + 0.5 \cdot 0 = 0. \end{aligned}$$

Hence  $a_5$  is not judged weak because the only alternatives that are preferred over it are not judged strong either. Consequently, the final quality of  $a_5$  is  $S(a_5) = S^+(a_5) - S^-(a_5) = 0 - 0 = 0$ .

### 3.3. PrefRank III

PrefRank III extends the idea underlying PageRank I taking into account an overall difficulty in being preferred to some alternative estimated by the analysis of its relations with all other alternatives. On the one hand, an alternative's great strength derives from being highly preferred to the alternatives which are outranked by other good solutions. Thus, the strength of  $a_i$  is defined as a weighted sum of its out-going preference degrees  $\pi(a_i, a_k)$  with the following weights:

$$\omega^+(a_k) = \frac{1}{\sum_{i^*=1}^n \pi(a_{i^*}, a_k)} \sum_{l=1}^n \left[ \frac{\pi(a_l, a_k)}{\sum_{k^*=1}^n \pi(a_l, a_{k^*})} S^+(a_l) \right]. \quad (10)$$

The above weight depends on the degrees to which other alternatives are preferred to  $a_k \in A$  and the strengths of these solutions. On the other hand, an alternative's high weakness is implied by being vastly outranked by alternatives that are preferred to other weak solutions. Hence the weakness of  $a_i$  is a weighted sum of its in-going preference degrees  $\pi(a_k, a_i)$  with the following weights:

$$\omega^-(a_k) = \frac{1}{\sum_{i^*=1}^n \pi(a_k, a_{i^*})} \sum_{l=1}^n \left[ \frac{\pi(a_k, a_l)}{\sum_{k^*=1}^n \pi(a_{k^*}, a_l)} S^-(a_l) \right]. \quad (11)$$



In this case, the weight depends on the degrees with which  $a_k$  is preferred over other alternatives and the weaknesses of these solutions.

PrefRank III is inspired by the SALSA algorithm Lempel and Moran (2000), which exploits the information on the existence of links connecting pairs of pages as well as the number of in- and outgoing links. We adopt this approach to deal with a valued preference relation. Hence we consider a bipartite graph constructed based on the preference matrix. The nodes in one part of this graph correspond to the alternatives' weaknesses, and the nodes in the other part stand for the alternatives' strengths. Since each alternative has its strength and weakness, it is represented with a node in both parts. The preference of  $a_i$  over  $a_k$  is modeled with an arc from  $a_i$  in the strong part to  $a_k$  in the weak part. The strengths and weaknesses in PrefRank III are equivalent to the probability of randomly reaching the node in this bipartite graph representing each alternative's strong and weak points.

The illustration of the computations conducted by PrefRank III is again based on the example introduced in Section 3.1 and refers to the results given in Table 2. The non-normalized strength of  $a_1$ , disregarding the preference indices equal to zero, is:

$$\begin{aligned}
\phi^+(a_1) &= \pi(a_1, a_2) \cdot \omega^+(a_2) + \pi(a_1, a_3) \cdot \omega^+(a_3) = \\
&= \pi(a_1, a_2) \cdot \frac{1}{\pi(a_1, a_2)} \cdot \frac{\pi(a_1, a_2)}{\pi(a_1, a_2) + \pi(a_1, a_3)} \cdot S^+(a_1) \\
&+ \pi(a_1, a_3) \cdot \frac{1}{\pi(a_1, a_3) + \pi(a_2, a_3)} \cdot \left[ \frac{\pi(a_1, a_3)}{\pi(a_1, a_3) + \pi(a_2, a_3)} \cdot S^+(a_1) + \frac{\pi(a_2, a_3)}{\pi(a_2, a_3)} \cdot S^+(a_2) \right] = \\
&= 1 \cdot \frac{1}{1} \cdot \frac{1}{1+1} \cdot 0.267 + 1 \cdot \frac{1}{1+1} \cdot \left[ \frac{1}{1+1} \cdot 0.267 + \frac{1}{1} \cdot 0.133 \right] = 0.267.
\end{aligned}$$

Since these strengths are already normalized, to sum up to one for all alternatives,  $\phi^+(a_1) = S^+(a_1)$ . Similarly to PrefRanks I and II, the weakness for  $\phi^-(a_1) = S^-(a_1) = 0$ . Then, the quality of  $a_1$  is  $S(a_1) = S^+(a_1) - S^-(a_1) = 0.267 - 0.0 = 0.267$ . The strength of  $a_5$  is:

$$\begin{aligned}
\phi^+(a_5) &= \pi(a_5, a_4) \cdot \omega^+(a_4) = \pi(a_5, a_4) \cdot \frac{1}{\pi(a_5, a_4)} \cdot \frac{\pi(a_5, a_4)}{\pi(a_5, a_4)} \cdot S^+(a_5) = \\
&= 0.5 \cdot \frac{1}{0.5} \cdot \frac{0.5}{0.5} \cdot 0.2 = 0.5 \cdot 0.4 = 0.2.
\end{aligned}$$

In turn, the weakness of  $a_5$  is:

$$\begin{aligned}
\phi^-(a_5) &= \pi(a_4, a_5) \cdot \omega^-(a_4) + \pi(a_3, a_5) \cdot \omega^-(a_3) = \\
&= \pi(a_4, a_5) \cdot \frac{1}{\pi(a_4, a_5)} \cdot \frac{\pi(a_4, a_5)}{\pi(a_3, a_5) + \pi(a_4, a_5)} \cdot S^-(a_5) \\
&+ \pi(a_3, a_5) \cdot \frac{1}{\pi(a_3, a_5)} \cdot \frac{\pi(a_3, a_5)}{\pi(a_3, a_5) + \pi(a_4, a_5)} \cdot S^-(a_5) = \\
&= 0.5 \cdot \frac{1}{0.5} \cdot \frac{0.5}{0.5+1} \cdot 0.25 + 1 \cdot \frac{1}{1} \cdot \frac{1}{0.5+1} \cdot 0.25 = 0.25.
\end{aligned}$$

The weaknesses of all alternatives are also normalized, and hence  $\phi^-(a_5) = S^-(a_5)$ . Consequently, the quality of  $a_5$  is  $S(a_5) = S^+(a_5) - S^-(a_5) = 0.2 - 0.25 = -0.05$ .

The computational procedure used by all variants of PrefRank to derive the strengths and weaknesses of all alternatives is presented in the e-Appendix (supplementary material available online).

### 3.4. Ranking construction

A few procedures can be used to construct a ranking of alternatives based on the results of PrefRank. An incomplete ranking can be established by following the principles of PROMETHEE I and considering the strengths and weaknesses

separately. The conditions justifying the preference  $P$ , indifference  $I$ , and incomparability  $R$  relations are as follows:

$$\begin{aligned}
 a_i P a_k & \text{ iff } (S^+(a_i) > S^+(a_k) \text{ and } S^-(a_i) < S^-(a_k)) \\
 & \text{ or } (S^+(a_i) > S^+(a_k) \text{ and } S^-(a_i) = S^-(a_k)) \\
 & \text{ or } (S^+(a_i) = S^+(a_k) \text{ and } S^-(a_i) < S^-(a_k)); \\
 a_i I a_k & \text{ iff } (S^+(a_i) = S^+(a_k) \text{ and } S^-(a_i) = S^-(a_k)); \\
 a_i R a_k & \text{ otherwise.}
 \end{aligned}$$

Such incomplete rankings obtained for the illustrative problem using PROMETHEE I and the three variants of PrefRank are provided in Figure 2. In all these rankings,  $a_1$  is preferred to the remaining alternatives. However, there are also some noticeable differences. For example,  $a_4$  is preferred to  $a_5$  according to PROMETHEE I; the relation is inverse for PrefRank III, whereas, for PrefRank I and II,  $a_4$  and  $a_5$  are indifferent.

In turn, a complete ranking can be established, following the assumptions of PROMETHEE II, by analyzing the comprehensive qualities of alternatives:

$$\begin{aligned}
 a_i P a_k & \text{ iff } S(a_i) > S(a_k); \\
 a_i I a_k & \text{ iff } S(a_i) = S(a_k).
 \end{aligned}$$

Other exploitation procedure is mentioned in the concluding section.

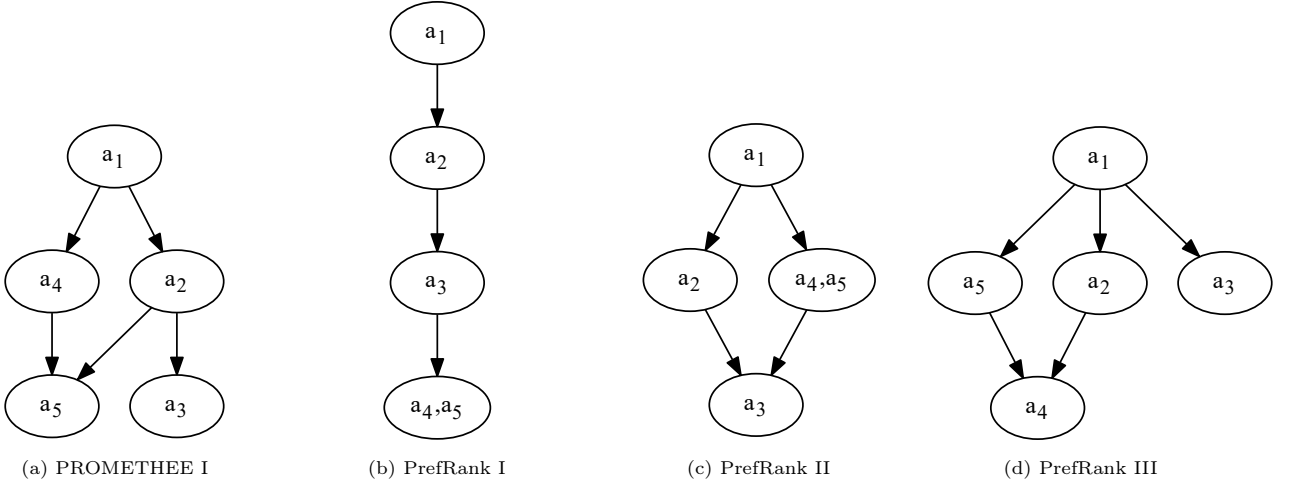


Figure 2: Incomplete rankings obtained for the illustrative problem according to the strengths and weaknesses computed with different methods.

#### 4. Measures used for comparing the recommendation

In this section, we discuss measures used for comparing the recommendation derived with a pair of methods  $M'$  and  $M''$  Kadziński and Michalski (2016). We start with the metrics quantifying the similarity between ranking positions. The rank ( $rank(M, a_i)$ ) of alternative  $a_i \in A$  according to method  $M \in \{M', M''\}$  is defined as the number of alternatives that are strictly preferred to  $a_i$  increased by one. We denote a subset of alternatives that, according to  $M$ , are ranked in the  $r$ -th position by:

$$M(r) = \{a_i \in A : rank(M, a_i) = r\}, \text{ for } r = 1, \dots, n. \quad (12)$$

When considering the rankings produced by  $M'$  and  $M''$  in the context of rank  $r$ , the rank agreement measure  $RA(M', M'', r)$  is defined as the share of alternatives that are assigned  $r$ -th rank by both methods as compared to a

subset of alternatives that are ranked  $r$ -th by either method:

$$RA(M', M'', r) = \frac{|M'(r) \cap M''(r)|}{|M'(r) \cup M''(r)|}. \quad (13)$$

The  $RA$  measure builds on Jaccard's coefficient. Given the first rank ( $r = 1$ ),  $RA$  is called the Normalized Hit Ratio:

$$NHR(M', M'') = RA(M', M'', 1) = \frac{|M'(1) \cap M''(1)|}{|M'(1) \cup M''(1)|}. \quad (14)$$

It focuses on the top-ranked alternatives, hence capturing the agreement in the choice recommendation. When  $NHR$  is equal to one, both methods rank the same subset of alternatives at the top, whereas  $NHR = 0$  means these subsets are disjoint.

Other measures for comparing a pair of rankings are based on pairwise comparisons. They account for the similarity of relations observed for all pairs of alternatives. Table 3 provides such distances for the four possible relations: preference  $P$ , inverse preference  $P^-$ , indifference  $I$ , and incomparability  $R$  Miebs and Kadziński (2021); Roy and Slowinski (1993). The rank difference measure aggregates such distances over all pairs of different alternatives:

$$RD(M', M'') = \sum_{i=1}^n \sum_{k=1, k \neq i}^n RD(M', M'', a_i, a_k). \quad (15)$$

For the incomplete rankings, admitting all four relations, we will consider the normalized variant of such a distance:

$$NRD(M', M'') = \frac{RD(M', M'')}{2n \cdot (n - 1)}. \quad (16)$$

It takes values between 0 and 1, where 0 means that the relations observed in the rankings constructed with  $M'$  and  $M''$  are the same for all pairs of alternatives. We will use  $NRD$  to report the similarity between incomplete rankings.

As far as complete orders are concerned, we will refer to Kendall's  $\tau$ , which is well established in the literature. Let us remind that complete rankings do not admit incomparability. Moreover, one assumes that the distance between preference and inverse preference is twice as great as between preference and indifference. Overall, Kendall's  $\tau$  is defined in the following way by bringing  $NRD$  to the  $[-1, 1]$  range:

$$\tau(M', M'') = 1 - 2 \cdot NRD(M', M''). \quad (17)$$

The value of 1 means that the two rankings are the same, whereas  $-1$  means that all relations in one ranking are inverse in the other.

Table 3: The distances  $RD(M', M'', a_i, a_k)$  between relations observed for pair  $(a_i, a_k)$  in the rankings determined by methods  $M'$  and  $M''$ .

$RD(M', M'', a_i, a_k)$	$a_i P^{M''} a_k$	$a_i I^{M''} a_k$	$a_i R^{M''} a_k$	$a_i P^{-, M''} a_k$
$a_i P^{M'} a_k$	0	2	3	4
$a_i I^{M'} a_k$	2	0	2	2
$a_i R^{M'} a_k$	3	2	0	3
$a_i P^{-, M'} a_k$	4	2	3	0

## 5. Illustrative case study concerning fleet ranking problem

We illustrate the applicability of the methods on a problem of ranking single-stage, single-deck buses by a transport company Żak (2005). Nine models are considered as decision alternatives evaluated in terms of the following five criteria:

- *price* ( $g_1$ ; to be minimized): a net price of the vehicle (in thousands Euro) in the variant indicated by the company while assuming that 20 buses of the same type would be purchased;
- *exploitation cost* ( $g_2$ ; to be minimized): costs of the service, repairs, and fuel (in thousands of PLN per 100,000 kilometers);
- *comfort* ( $g_3$ ; to be maximized): a score given on a 0 – 10 scale aggregating many factors affecting the travel’s comfort (e.g., the luggage capacity or ergonomics of the seats);
- *safety* ( $g_4$ ; to be maximized): a score given on a 0 – 10 scale aggregating many factors affecting the safety and easiness of driving (e.g., quality of the break and steering system);
- *modernity* ( $g_5$ ; to be maximized): a score given on a 0 – 10 scale aggregating multiple factors related to the consistency with current design and technological trends.

The performance matrix is given in Table 4. The parameters needed for comparing the buses pairwise are provided in Table 5.

Table 4: The performance matrix for the fleet ranking problem.

Code	Bus model	Price [euro]	Exploit. cost [PLN]	Comfort [pts]	Safety [pts]	Modernity [pts]
$a_1$	Autosan A 402 T Cezar (A)	209.0	87.5	7.64	9.04	7.8
$a_2$	Bova FHD 12-370 Futura (B)	231.0	88.0	7.74	8.39	8.8
$a_3$	Ikarus EAG E 98 (I)	207.0	92.0	5.67	4.44	5.6
$a_4$	Jelcz T 120/3 MB (J)	102.0	79.7	2.75	5.23	3.9
$a_5$	Man RH 402 Lion’s Star (M)	239.0	83.4	5.18	7.07	4.8
$a_6$	Mercedes 0.350 RHD (E)	229.0	85.9	6.54	9.52	8.5
$a_7$	Neoplan N 316 SHD Transliner (N)	246.0	84.6	9.17	7.28	9.2
$a_8$	Scania Irizar Century (R)	231.0	94.2	8.94	6.34	9.1
$a_9$	Volvo Droegmoeller B 12–600 (V)	263.0	86.2	6.87	8.32	9.3

Table 5: Criteria weights and comparison thresholds for the fleet ranking problem.

Criterion	Weight $w_j$	Indifference threshold $q_j$	Preference threshold $p_j$
Price [euro]	9.6	5.0	20.0
Expl. cost [PLN]	8.8	3.0	10.0
Comfort [pts]	3.8	0.3	1.4
Safety [pts]	6.3	0.2	1.0
Modernity [pts]	2.5	0.3	1.8

The comprehensive preference degrees for all pairs of buses are provided in Table 6. For most pairs, the degrees are positive on either side (e.g.,  $\pi(a_1, a_4) = 0.406$  and  $\pi(a_4, a_1) = 0.504$ ). Their analysis indicates the buses that compare quite favorably with all the remaining ones. For example, the minimal preference degrees of  $a_1$  and  $a_4$  when collated with any other bus are 0.399 and 0.338, respectively. On the other extreme, some other buses attained less favorable results. In particular, for  $a_3$ , the preference degrees range from 0 to 0.358. The matrix of preference degrees is exploited by the PrefRank and PROMETHEE methods to construct the rankings.

Table 6: Comprehensive preference degrees for the fleet ranking problem.

Bus	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$
$a_1$	0.000	0.424	0.467	0.406	0.716	0.399	0.513	0.663	0.494
$a_2$	0.038	0.000	0.447	0.406	0.468	0.100	0.410	0.333	0.373
$a_3$	0.000	0.310	0.000	0.198	0.358	0.310	0.310	0.310	0.310
$a_4$	0.504	0.525	0.743	0.000	0.338	0.439	0.387	0.594	0.452
$a_5$	0.045	0.065	0.430	0.358	0.000	0.000	0.041	0.419	0.310
$a_6$	0.093	0.203	0.473	0.406	0.505	0.000	0.451	0.418	0.513
$a_7$	0.182	0.144	0.585	0.406	0.206	0.144	0.000	0.456	0.370
$a_8$	0.165	0.100	0.406	0.406	0.265	0.139	0.206	0.000	0.432
$a_9$	0.065	0.011	0.498	0.406	0.406	0.030	0.203	0.406	0.000

The positive, negative, and comprehensive flows for PROMETHEE and the normalized strengths, weaknesses, and qualities for the three variants of PrefRank are presented in Table 7. In what follows, we focus on explaining the results attained for the top-ranked alternatives.

The average outgoing comprehensive degree is the greatest for  $a_1$ , which results in the most advantageous positive flow  $\phi^+$ . The same alternative attains the least average ingoing degree, i.e., the negative flow  $\phi^-$ . As a result, it is ranked at the top by PROMETHEE I and II. It is followed by  $a_6$  and  $a_4$ . The former attains the third highest positive flow and the second least negative flow. The latter is second according to  $\phi^+$  while attaining only the fifth best  $\phi^-$ . Consequently,  $a_6$  and  $a_4$  are incomparable according to PROMETHEE I, and  $a_6$  is ranked better than  $a_4$  by PROMETHEE II. Also, due to its high negative flow,  $a_4$  is incomparable with  $a_2$  and  $a_7$  in the PROMETHEE I ranking. However, its highly favorable positive flow makes it preferred over these two alternatives in the complete ranking delivered by PROMETHEE II.

Compared to PROMETHEE, PrefRank I considers the strength of the alternatives over which a given option is preferred and the weakness of the alternatives which are preferred to a given option. This modifies the relative comparison for some pairs of alternatives. For example,  $a_4$  has higher strength than  $a_1$ . This derives from its relatively high preference degrees over other strong alternatives, including  $a_1$ ,  $a_2$ , and  $a_6$ . In turn, even though the average outgoing preference degree is greater for  $a_1$  than  $a_4$ ,  $a_1$  is preferred – to the greatest extent – to  $a_5$ ,  $a_7$ ,  $a_8$ , and  $a_9$ . These alternatives attain positions in the bottom half of the ranking given their strength, which deteriorates the strength of  $a_1$ . When it comes to weakness,  $a_4$  has high ingoing preference degrees (over 0.4) when compared with 6 of 8 remaining alternatives. Among them,  $a_8$  and  $a_9$  are in the bottom three alternatives in terms of weakness. On the contrary, the comparison of  $a_1$  is unfavorable only against  $a_4$ , whose weakness is intermediate. However, the weakest alternatives ( $a_3$ ,  $a_8$ , and  $a_9$ ) are not or marginally preferred to  $a_1$ , implying its low weakness.

PrefRank II adopts an inverse weighting scheme when compared to PrefRank I. In this case, the strength is derived from being highly preferred to alternatives with great weakness, whereas the weakness is implied by comparing highly unfavorably against alternatives with great strength. Such an interpretation favors  $a_1$  against  $a_4$ . Indeed,  $a_1$  is highly preferred to  $a_3$ ,  $a_5$ ,  $a_7$ ,  $a_8$ , and  $a_9$ . These are the five alternatives with the greatest weaknesses, which implies high  $S^+(a_1)$ . At the same time, only one strong alternative ( $a_4$ ) is highly preferred to  $a_1$ . This guarantees the lowest weakness of  $a_1$  among all alternatives. In turn,  $a_4$  is highly preferred to  $a_1$ ,  $a_2$ ,  $a_3$ ,  $a_6$ , and  $a_8$ . Among them, only  $a_3$  and  $a_8$  can be judged as weak, and the remaining three alternatives' weakness is the least. This prevents the strength of  $a_4$  from attaining a very high level. Yet, numerous alternatives – including the strongest ones – are moderately preferred to  $a_4$ . This deteriorates  $S^-(a_4)$ , making it higher than weaknesses of  $a_2$  and  $a_6$ . A notable difference in the PrefRank II ranking compared to the orders imposed by PROMETHEE and PrefRank I is a more favorable comparison of  $a_9$  against  $a_8$ . It is implied by the greater strength of  $a_9$ , which comes from its relatively higher preference degrees when collated with the weakest alternatives ( $a_3$ ,  $a_5$ , and  $a_8$ ).

PrefRank III builds on the same idea as PrefRank I while accounting for the overall challenge of being preferred to different alternatives, as revealed by their relations with the remaining alternatives. This additional aspect implies some noticeable differences between the incomplete rankings derived from these two methods. In particular,  $a_1$  is preferred to  $a_4$ ,  $a_2$  is preferred to  $a_7$ , and  $a_9$  is preferred to  $a_5$ . The underlying reasons are the same, i.e., increased strength of  $a_1$ ,  $a_2$ , and  $a_9$  due to being more preferred than their counterparts to alternatives that are slightly more difficult to outrank.

The incomplete and complete rankings obtained with the five methods are presented in Figure 3. Let us first focus on quantifying the similarity between the rankings admitting incomparability. The top-ranked alternative, according to PROMETHEE I, PrefRank I, and PrefRank III is  $a_1$ . As a result, the NHR measure for these methods equals one. In turn, PrefRank II additionally admits  $a_4$  as a top option. Consequently, when compared with the remaining three methods, its agreement in recommending the most preferred alternative(s) is 0.5.

The incomplete rankings obtained with various methods are very alike. This is confirmed by the low values of Normalized Ranking Distances (see Table 8). The same rankings were constructed by PROMETHEE I and PrefRank III

Table 7: Results of the four methods for the fleet ranking problem.

Method	Result	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$
PROMETHEE	$\phi^+$	0.510	0.322	0.263	0.498	0.208	0.383	0.312	0.265	0.253
	$\phi^-$	0.136	0.223	0.506	0.374	0.408	0.195	0.315	0.450	0.407
	$\phi$	0.374	0.099	-0.243	0.123	-0.199	0.188	-0.003	-0.185	-0.154
PrefRank I	$S^+$	0.163	0.105	0.086	0.165	0.072	0.124	0.107	0.093	0.085
	$S^-$	0.051	0.076	0.168	0.125	0.129	0.069	0.100	0.146	0.134
	$S$	0.112	0.029	-0.082	0.040	-0.057	0.055	0.007	-0.053	-0.049
PrefRank II	$S^+$	0.168	0.112	0.080	0.152	0.076	0.131	0.106	0.085	0.091
	$S^-$	0.047	0.080	0.165	0.112	0.138	0.069	0.108	0.150	0.133
	$S$	0.121	0.032	-0.085	0.040	-0.062	0.063	-0.002	-0.065	-0.042
PrefRank III	$S^+$	0.169	0.107	0.087	0.165	0.069	0.127	0.103	0.088	0.084
	$S^-$	0.045	0.074	0.168	0.124	0.135	0.065	0.105	0.149	0.135
	$S$	0.124	0.033	-0.081	0.041	-0.066	0.062	-0.001	-0.061	-0.051

( $NRD = 0$ ). Then, highly similar incomplete rankings were obtained with PrefRank II and PROMETHEE I or PrefRank III ( $NRD = 0.042$ ). They differ only in terms of the relations imposed for  $a_9$  and  $a_8$  as well as  $a_9$  and  $a_3$ . The greatest differences are observed for the incomplete rankings derived with PrefRanks I and II ( $NRD = 0.104$ ). They concern the following pairs of alternatives:  $(a_1, a_4)$ ,  $(a_2, a_7)$ ,  $(a_9, a_8)$ ,  $(a_9, a_5)$ , and  $(a_9, a_3)$ . They are incomparable in the ranking imposed by PrefRank I while being related by the preference according to PrefRank II.

When considering the complete rankings, the similarities are more significant. Specifically, the rankings obtained with PROMETHEE II and PrefRanks I and III are the same (see Table 9). Regarding the ranking obtained with PrefRank II, it differs with respect to the remaining ones only in terms of placing  $a_5$  higher than  $a_8$ . As a result, Kendall's  $\tau$  is equal to 0.944. Still, all methods recommend the same alternative ( $a_1$ ) at the top. Consequently, the NHR values for all pairs of approaches based on the complete rankings are equal to one.

Table 8: Normalized Ranking Distances for the incomplete rankings for the fleet selection problem.

Method	PROMETHEE I	PrefRank I	PrefRank II	PrefRank III
PROMETHEE I	0.000	0.062	0.042	0.000
PrefRank I	0.062	0.000	0.104	0.062
PrefRank II	0.042	0.104	0.000	0.042
PrefRank III	0.000	0.062	0.042	0.000

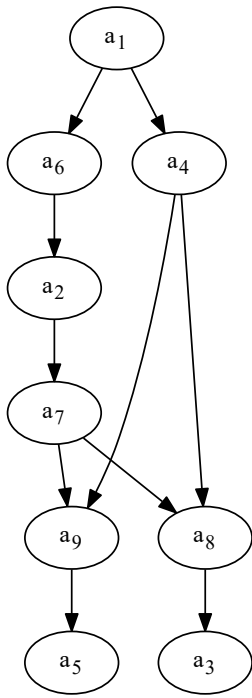
Table 9: Kendall's  $\tau$  based on the complete rankings obtained for the fleet selection problem.

Method	PROMETHEE II	PrefRank I	PrefRank II	PrefRank III
PROMETHEE II	1.000	1.000	0.944	1.000
PrefRank I	1.000	1.000	0.944	1.000
PrefRank II	0.944	0.944	1.000	0.944
PrefRank III	1.000	1.000	0.944	1.000

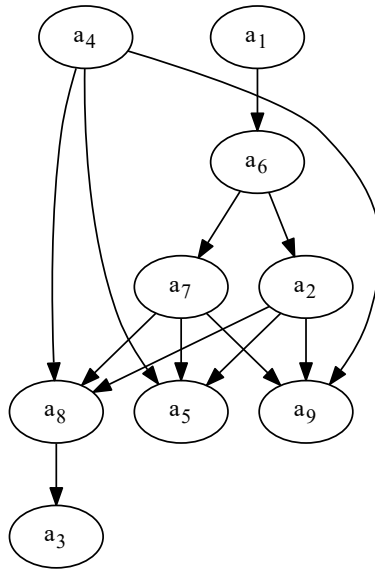
## 6. Experimental comparison between PROMETHEE and PrefRank

In this section, we report the results of an experimental comparison between PROMETHEE and the three variants of PrefRank. This is to investigate the similarity between the choice and rankings recommendations delivered by these approaches under a broad spectrum of problem characteristics. The considered number of alternatives ranges between 4 and 20 (with a step of two), and the number of criteria is between 3 and 8. For each problem size, we generated 100 instances with uniformly distributed performances and criteria weights. The indifference thresholds  $q_j$  were drawn from the interval between 0% and 20% of the performance range on a given criterion, whereas the preference thresholds  $p_j$  were drawn from the interval delimited by  $q_j$  and 50% of the performance range.

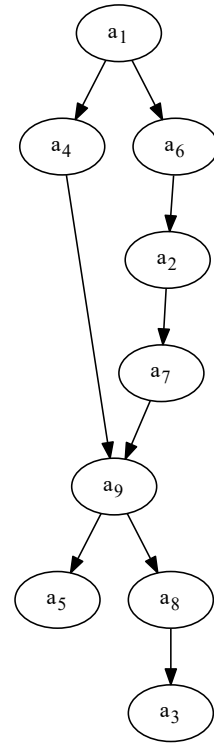
Then, we computed the comprehensive preference degrees and exploited them with the variants of PROMETHEE and PrefRank, leading to either incomplete or complete rankings. Finally, we quantified the similarity between these results for all pairs of methods given NHR, NRD, and Kendall's  $\tau$ . The average results over all problems instances



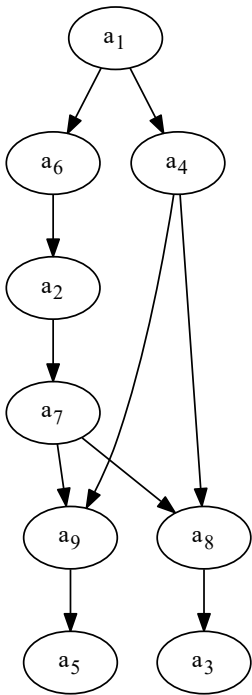
(a) PROMETHEE I



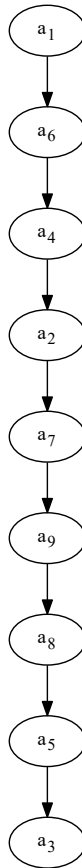
(b) PrefRank I



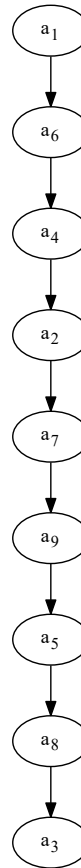
(c) PrefRank II



(d) PrefRank III



(e) PROMETHEE II, PrefRanks I and III



(f) PrefRank II

Figure 3: Incomplete (a – d) and complete (e – f) rankings derived with the four methods for the fleet selection problem.

are presented in Tables 10, 11, and 12. More detailed results for various problem sizes are exhibited in the form of heatmaps (for NHR, see Figure 4, whereas for the remaining two measures, see e-Appendix).

The NHR values are computed based on comparing top alternatives in the complete rankings delivered by each method. They are very high, ranging between 0.863 for PrefRanks I and II to 1 for PROMETHEE II and PrefRank III. On the one hand, these results confirm that the applied weighting scheme influences the methods' outcomes. On the other hand, the differences are minor. For example, PrefRank I recommends the selection of a different alternative than PROMETHEE II in about 9% of the considered scenario, whereas when comparing the indications of PROMETHEE II and PrefRank II, this difference is even slightly lesser. In general, the NHR values increase with lesser alternatives and more criteria. However, these trends are not strict. In particular, the greatest average NHR values (0.940) for PrefRanks I and II are observed for problems involving four alternatives and seven criteria. In turn, the least similarity (0.798) is noted for problems with three criteria and sixteen alternatives.

Table 10: The average NHR values for all considered problem instances and the four methods delivering complete rankings.

Method	PROMETHEE II	PrefRank I	PrefRank II	PrefRank III
PROMETHEE II	1.000	0.907	0.922	1.000
PrefRank I	0.907	1.000	0.863	0.907
PrefRank II	0.922	0.863	1.000	0.922
PrefRank III	1.000	0.907	0.922	1.000

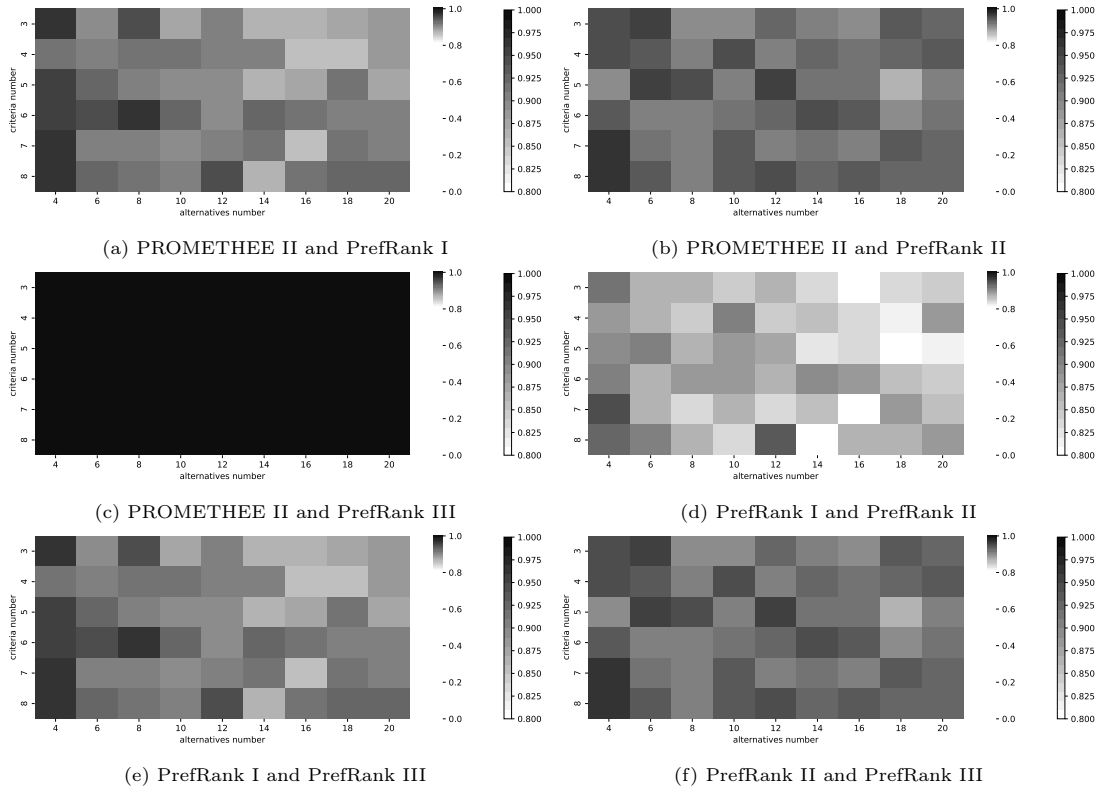


Figure 4: Heatmaps of average NHR values based on 100 runs for each problem size.

The analyses of similarities or distances between the complete and incomplete rankings confirm the same conclusions. First, the rankings delivered by PROMETHEE and PrefRank III were the same in all considered scenarios. Let us note that outside the experiment, we could generate instances for which some slight differences were observed. Nevertheless, such ideal similarity confirms that computing the strength and weakness while accounting for, respectively, strengths and weaknesses of other alternatives, as well as the difficulty in being preferred to them estimated based on their comparisons with the remaining options, leads to the orders that are highly similar to those derived with the original PROMETHEE. When it comes to the remaining pairs of methods, the average Kendall's  $\tau$  ranges between



0.901 for PrefRanks I and II to 0.947 for PrefRank I and PROMETHEE or PrefRank III (see Table 11). The extreme results are observed for the same pairs of approaches also for NRD (see Table 12). However, in this case, they range between 0.040 (the least distance) and 0.072 (the greatest distance). Thus, whether you account for the strengths or weaknesses of other alternatives when computing the results of PrefRank does matter. Still, the differences are not substantial as all these methods exploit the same matrices of comprehensive preference degrees. Also, the greatest differences with respect to PROMETHEE are observed for PrefRank I, which computes the strengths (weakness) of alternatives by analyzing other alternatives' strengths (weaknesses). Finally, the similarity trends for different numbers of alternatives and criteria are not definite. When comparing the rankings, a general conclusion is that greater similarities and lesser distances are observed for instances with fewer criteria and more alternatives. However, there are exceptions in this regard (e.g., very high similarities are observed for problems with the least considered number of alternatives). Moreover, the results differ from one pair of compared methods to another (see the heatmaps presented in the e-Appendix).

Table 11: The average Kendall's  $\tau$  values for all considered problems instances and the four methods delivering complete rankings.

Method	PROMETHEE II	PrefRank I	PrefRank II	PrefRank III
PROMETHEE II	1.000	0.937	0.947	1.000
PrefRank I	0.937	1.000	0.901	0.937
PrefRank II	0.947	0.901	1.000	0.947
PrefRank III	1.000	0.937	0.947	1.000

Table 12: The average NRD values for all considered problems instances and the four methods delivering incomplete rankings.

Method	PROMETHEE II	PrefRank I	PrefRank II	PrefRank III
PROMETHEE II	0.000	0.046	0.040	0.000
PrefRank I	0.046	0.000	0.072	0.046
PrefRank II	0.040	0.072	0.000	0.040
PrefRank III	0.000	0.046	0.040	0.000

## 7. Software and case study concerning special economic zones

### 7.1. Implementation of PrefRank

The methods proposed in this paper were implemented in a single programming module called *Outranking-PrefRank*. Its scheme is presented in Figure 5. The module accepts two inputs in the XML format, a set of alternatives and a matrix of preference degrees. Also, it requires setting the values of four parameters. The most important one is related to the variant of PrefRank used to compute the results. The remaining three parameters refer to the convergence and stopping conditions of the iterative computational procedure. In particular, one can specify the maximal number of iterations. The module provides four types of output in the XML format: the strengths (positive flows), weaknesses (negative flows), qualities (total flows), and the pairs of alternatives related by the weak preference according to the separate analysis of strengths and weaknesses (ranking).

The PrefRank module can be combined with others that operate on the standardized input and output in the XMCDA format (see Figure 6). The matrix of comprehensive preference degrees can be derived with various methods. In the standard setting, one applies *PrometheePreference* or one of its variants. All these modules require the specification of weights. They can be provided by the user or computed with one of the approaches that have been specifically designed for computing weights based on the user's incomplete input. These include the Simos-Roy-Figueira (SRF) procedure Figueira and Roy (2002), Best-Worst-Method (BWM) Rezaei (2015), surrogate weights de Almeida Filho et al. (2018), and Analytical Hierarchy Process (AHP) Saaty (1990). The output of PrefRank can be visualized using either *plotAlternativesHasseDiagram* in the case of an incomplete ranking or *plotAlternativesValuesPreorder* in the case of a complete ranking. Provided by the availability of the *diviz* platform Meyer and Bigaret (2012), the modules can be combined using its graphical user interface or outside the software as regular programming components.

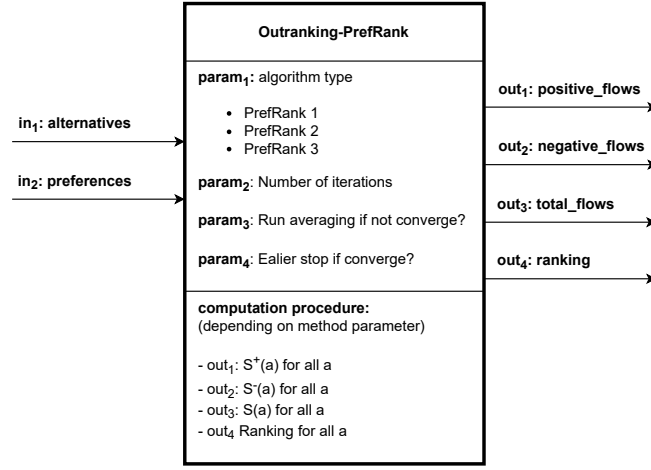


Figure 5: The scheme of the PrefRank module.

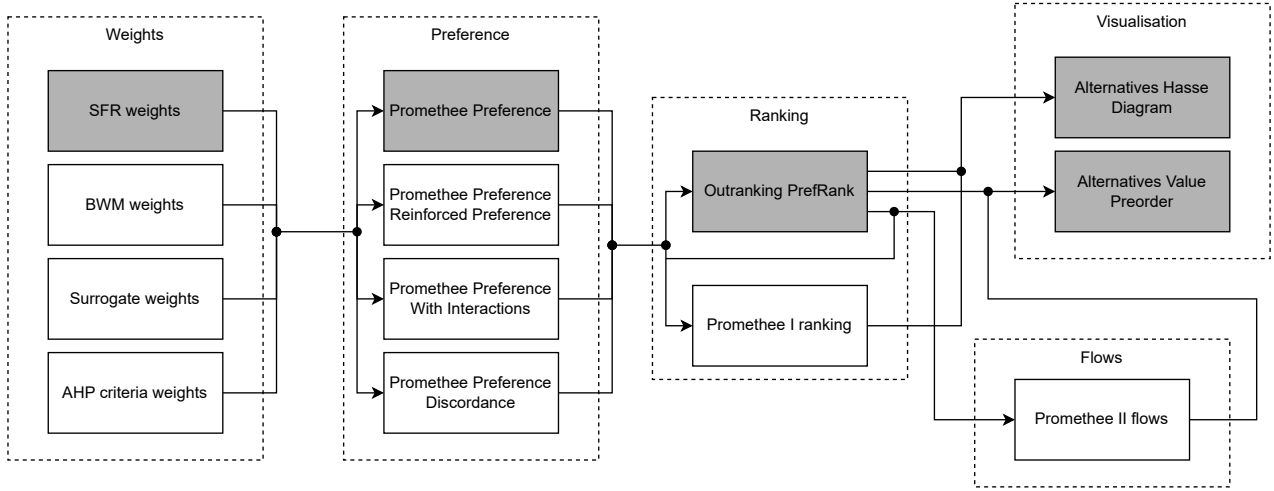


Figure 6: Possible connections between the PrefRank module and other related modules.

### 7.2. A case study concerning the evaluation of special economic zones in Poland

We apply PrefRank to the evaluation of Special Economic Zones in Poland. SEZs are designated areas offering favorable investment conditions, better infrastructure, and easier access to specialized staff. Moreover, running a business in SEZ is typically associated with highly advantageous tax regulations and reimbursement of some financial outlays for innovative projects. Hence such an area attracts investors and creates many businesses and job opportunities. We aim at ranking 10 SEZs that are located in Poland. They are evaluated in terms of the following five criteria:

- *Total area* (in ha; to be minimized) that each SEZ occupies; an efficient SEZ should occupy a relatively small area as larger areas naturally offer space for accommodating a more significant number of companies;
- *Capital expenditures* (in millions PLN; to be minimized), i.e., a cumulative value indicating how much money has been invested in a SEZ; a well-prospering SEZ should require a small cash contribution;
- *Total number of jobs* (to be maximized) in companies run in a given SEZ with a business permit; indeed, the creation of new jobs is one of the main goals of establishing a SEZ;
- *Business permits* (to be maximized), i.e., the number of issued business permits in a SEZ;
- *Financial result* (in millions PLN; to be maximized) is the total income generated by all businesses in a SEZ.

Table 13: The performances of ten Special Economic Zones in Poland in terms of five criteria.

Full name	SEZ Code	Total area	Capital expenditures	Number of jobs	Business permits	Financial result
Kamienna Góra	KAM	540.8285	2557.3	7530	60	555.1
Kostrzyn-Słubice	KOS	2201.2549	7133.4	32400	180	22984.9
Kraków	KRA	949.6604	4240.4	29580	189	1373.0
Legnica	LEG	1341.1473	5131.8	15294	86	7614.5
Łódź	LOD	1754.6376	13318.7	33401	209	7402.8
Mielec	MIE	1723.9743	7838.1	34992	268	4956.0
Pomorze	POM	2246.2929	10481.6	24893	173	1479.1
Słupsk	SLU	910.1585	1592.3	3478	79	761.5
Starachowice	STA	707.9814	1790.9	6829	56	701.0
Tarnobrzeg	TAR	1868.2066	7470.7	20740	195	18220.4

Table 14: The values of comparison thresholds, ranking of criteria, and weights derived from applying the SRF method for the problem of ranking Special Economic Zones.

Criterion	Indifference threshold $q_j$	Preference threshold $p_j$	Criteria ranks	Weight $w_j$
Total area	100	250	1	6.06
Capital expenditures	200	400	7	26.95
Number of jobs	2000	5000	5	20.02
Business permits	5	20	4	16.53
Financial result	50	150	8	30.44

The performances of all SEZs are reported in Table 13.

The preference model parameters are provided in Table 14. For each criterion, we used positive comparison thresholds. Moreover, we used the SRF method Figueira and Roy (2002) to elicit the criteria weights. The criteria were ranked from the worst to the best, and some empty cards were inserted between successive criteria, resulting in the ranks given in the table. Moreover, the ratio  $Z = 5$  between the most and the least important criteria, i.e., financial result and total area, was considered. The weights computed with SRF are given in Table 14.

The *diviz* workflow used to compute the results is presented in Figure 7. Its input consists of two CSV files that contain the performance matrix and the comparison thresholds. The dedicated modules transform them into XML objects that other algorithmic components can further process. The third input is an XML file containing the ranking of criteria required by the SRF procedure. It is exploited by the *SRF-weights* module to obtain the criteria weights. The suitably transformed input data are processed by the *PROMETHEE-preference* module to calculate the matrix of comprehensive preference degrees. It is shown in Table 15. This matrix is processed by the *Outranking-PrefRank* module to compute the strengths, weaknesses, and qualities of all alternatives. These values obtained for the three variants of PrefRank are presented in Table 16. The rankings (see Figure 8) are visualized using the *plotAlternativesValuesPreorder* and *plotAlternativesHasseDiagram* modules.

The complete rankings for the three methods, including PROMETHEE II and PrefRanks II and III, are the same. The order of SEZs obtained with these approaches is as follows:  $KOS \succ MIE \succ TAR \succ KRA \succ LOD \succ LEG \succ SLU \succ STA \succ POM \succ KAM$ . The only difference in the complete ranking produced by PrefRank I is that LEG is preferred to LOD. Overall, the best-ranked SEZ is KOS, which corresponds to a special economic zone around the cities of Kostrzyn and Słubice in western Poland. Indeed, its strength is the greatest, and its weakness is the least among all considered SEZs. This zone is relatively great, generating reasonable expenditures and creating a high number of jobs and the best financial outcome. As a result, KOS compares positively with eight out of the nine remaining zones. Only the preference degree of KRA compared to KOS is slightly greater than the inverse preference index. However, KRA compares poorly against LOD and MIE. As a result, it is ranked only fourth, preceded by MIE and TAR, both located in south-eastern Poland. On the contrary, KAM – located in south-western Poland – is ranked at the bottom. It is the least SEZ that generated a small number of jobs and attracted relatively few businesses. Moreover, it attained the worst financial results.

The differences in the incomplete rankings are more substantial (see Figure 8). Only PROMETHEE I and PrefRank III generated the same partial orders, with KOS being ranked at the top and many incomparabilities among

zones in the upper half of the ranking. For example, MIE is marginally better than KRA in strength (positive flow) but slightly worse given weakness (negative flow). The univocal position of KOS as the leading SEZ in Poland is confirmed by PrefRanks II and III. However, the remaining parts of incomplete rankings generated with these approaches are different. PrefRank II does not admit incomparability in the top part of the ranking while tolerating such ambiguity in indicating a more preferred zone for multiple pairs in the lower half of the ranking. It is the only method that does not rank KAM univocally at the bottom because KAM compares positively with POM in terms of strength. In turn, PrefRank II leaves many SEZs incomparable in the upper and lower parts of the ranking. In particular, KRA is incomparable with MIE and TAR. However, similarly to other approaches, it preserves the same division into zones that perform more and less favorably.

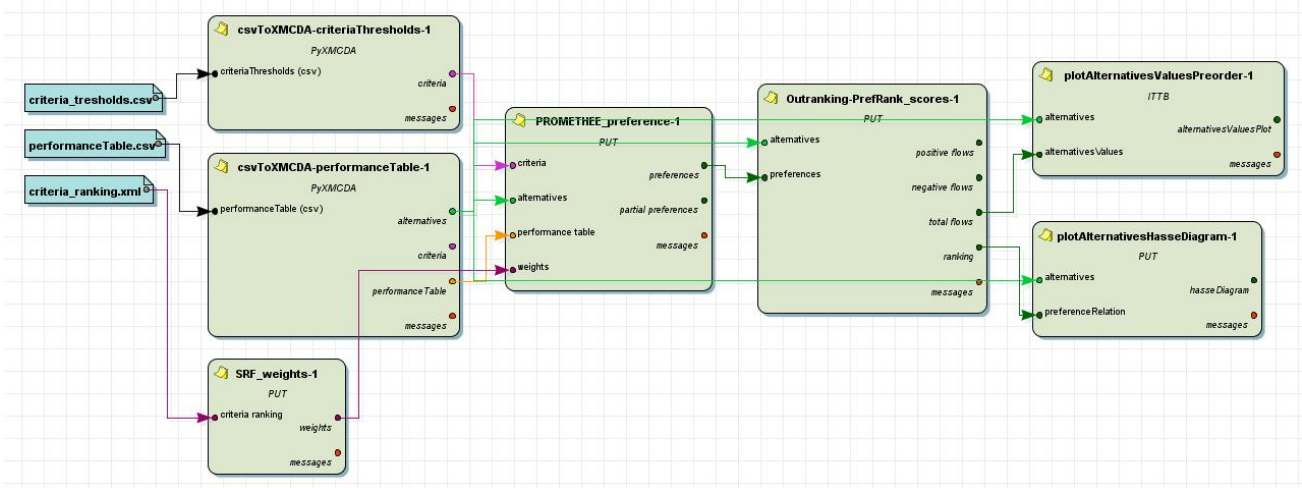


Figure 7: The *diviz* workflow used to compute the results of PrefRank for the problem of ranking the Special Economic Zones in Poland.

Table 15: The matrix of comprehensive preference degrees for the problem of ranking the Special Economic Zones in Poland.

SEZ	KAM	KOS	KRA	LEG	LOD	MIE	POM	SLU	STA	TAR
KAM	0.000	0.330	0.330	0.330	0.330	0.330	0.330	0.197	0.027	0.330
KOS	0.670	0.000	0.359	0.670	0.574	0.574	0.797	0.670	0.670	0.690
KRA	0.670	0.374	0.000	0.695	0.330	0.330	0.630	0.670	0.670	0.530
LEG	0.670	0.330	0.305	0.000	0.634	0.634	0.634	0.527	0.670	0.330
LOD	0.670	0.225	0.592	0.366	0.000	0.305	0.730	0.670	0.670	0.305
MIE	0.670	0.265	0.670	0.366	0.435	0.000	1.000	0.670	0.670	0.383
POM	0.670	0.000	0.171	0.366	0.270	0.000	0.000	0.670	0.670	0.144
SLU	0.729	0.330	0.270	0.330	0.330	0.330	0.330	0.000	0.197	0.330
STA	0.562	0.330	0.326	0.330	0.330	0.330	0.330	0.131	0.000	0.330
TAR	0.670	0.170	0.316	0.670	0.574	0.530	0.800	0.670	0.670	0.000

Table 16: The results obtained for the problem of ranking Special Economic Zones in Poland with four methods.

Method	Result	KAM	KOS	KRA	LEG	LOD	MIE	POM	SLU	STA	TAR
PROMETHEE	$\phi^+$	0.281	0.631	0.544	0.526	0.504	0.570	0.329	0.353	0.333	0.564
	$\phi^-$	0.665	0.262	0.371	0.458	0.423	0.374	0.620	0.542	0.546	0.375
	$\phi$	-0.383	0.369	0.173	0.068	0.081	0.196	-0.291	-0.189	-0.213	0.189
PrefRank I	$S^+$	0.069	0.136	0.117	0.113	0.105	0.118	0.066	0.080	0.077	0.118
	$S^-$	0.145	0.062	0.083	0.099	0.093	0.081	0.128	0.115	0.114	0.082
	$S$	-0.075	0.074	0.034	0.014	0.012	0.037	-0.062	-0.035	-0.036	0.037
PrefRank II	$S^+$	0.051	0.137	0.119	0.112	0.112	0.127	0.077	0.072	0.066	0.126
	$S^-$	0.141	0.050	0.077	0.098	0.089	0.079	0.140	0.122	0.125	0.078
	$S$	-0.090	0.087	0.041	0.015	0.023	0.048	-0.063	-0.049	-0.059	0.047
PrefRank III	$S^+$	0.061	0.136	0.117	0.114	0.109	0.123	0.071	0.076	0.072	0.122
	$S^-$	0.143	0.056	0.080	0.099	0.091	0.081	0.134	0.117	0.118	0.081
	$S$	-0.083	0.080	0.037	0.015	0.017	0.042	-0.063	-0.041	-0.046	0.041

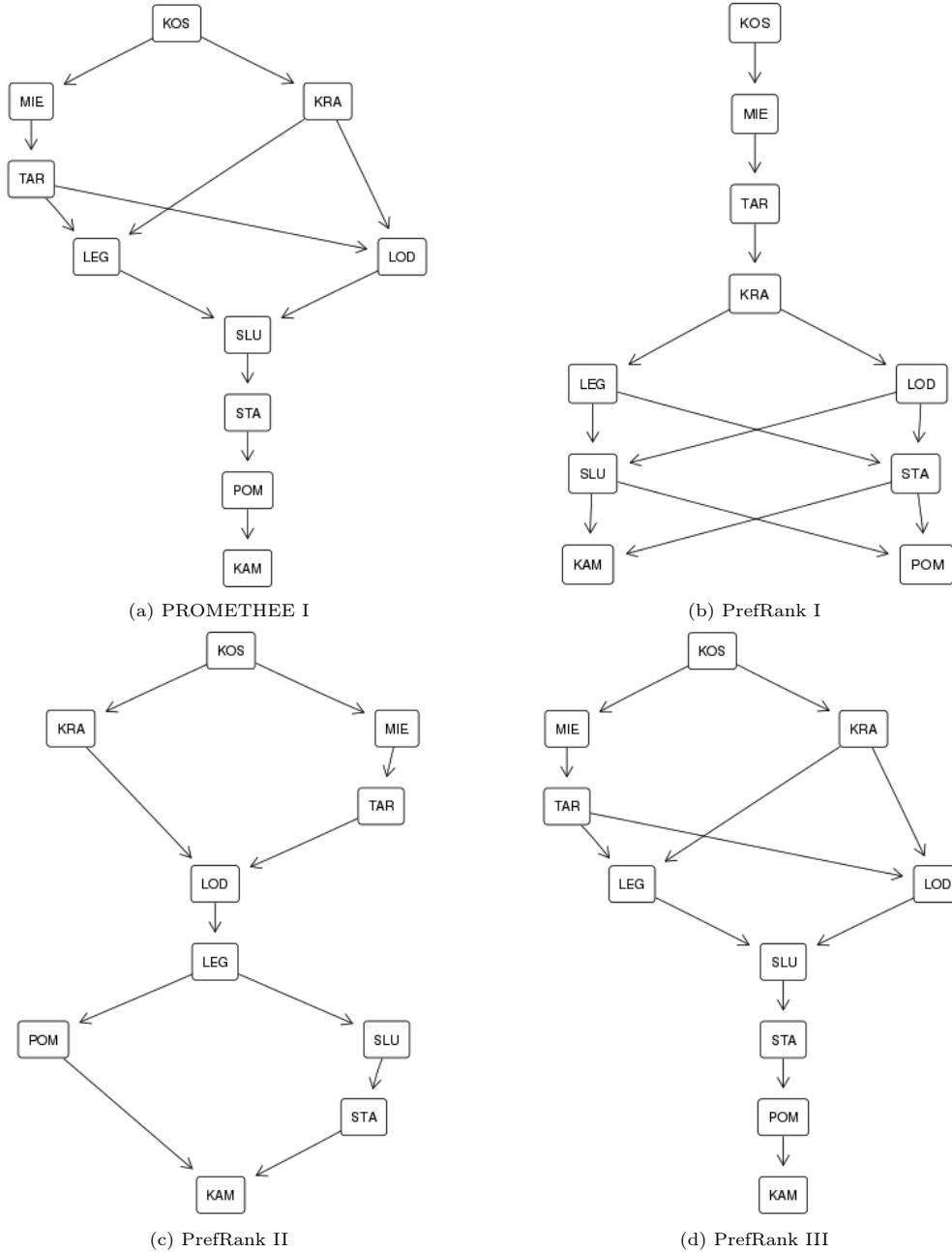


Figure 8: Incomplete rankings for the problem of ranking Special Economic Zones in Poland.

## 8. Summary and future research

In this paper, we proposed a novel family of approaches for exploiting a valued preference relation. It is inspired by both the Net Flow Rules that compute the strength and weakness of each alternative and algorithms originally conceived for scoring the websites. We described the three variants of the PrefRank method that evaluate each alternative by analyzing the preference degrees and the relative qualities of other alternatives. These variants differ concerning the implemented weighting schemes when aggregating the out- or in-going preference degrees. Nonetheless, each of them quantifies how strong each alternative is compared to others, how weak it is when confronted with the remaining ones, and what is the balance between strength and weakness. Finally, these factors are used to construct either an incomplete or a complete ranking.

We compared the results computed with the three variants of PrefRank and PROMTEHEE on a broad spectrum of simulated problems. The similarity in the generated recommendations was quantified in view of the top-ranked

alternatives, incomplete rankings, and complete orders. For all these perspectives, the consistency in the delivered results was significant. It was the greatest for PROMETHEE and PrefRank III while being the least for PrefRanks I and II. This confirms that the applied weighting scheme impacts the provided results.

We developed open-source software making the introduced methods available to other users. The module with the three variants of PrefRank was implemented in Java. Its source code is available at [https://bitbucket.org/Krzysztof\\_Martyn/prefrank](https://bitbucket.org/Krzysztof_Martyn/prefrank). To demonstrate its usability, we discussed the results of two studies. On the one hand, we illustrated the differences between various methods on the fleet selection problem. On the other hand, we considered an original problem of ranking Special Economic Zones in Poland. The task was to identify the zones that best use their area and funds to provide excellent financial profit and create many businesses or jobs. All methods identified a zone functioning in Kostrzyn and Słubice as the most preferred.

We envisage the following directions for future research. First, an appealing path consists of using other exploitation procedures for constructing complete or incomplete rankings. In this regard, the most natural proposal is to use an iterative procedure, which constructs the ranking in a top-bottom fashion. That is, the subset of most preferred alternatives at the current stage is added to the highest available positions, they are eliminated from further consideration, and the remaining alternatives are analyzed using the same procedure. The results obtained using such an approach for the previously considered fleet selection problem are presented in the e-Appendix. Interestingly, they confirm that the outcomes of PROMETHEE and PrefRank III can differ. Second, the proposed methods can be applied to preference or outranking matrices constructed with approaches other than PROMETHEE. The most natural direction involves investigating the use of PrefRank in the context of crisp outranking matrices produced by ELECTRE as well as results produced at different levels of hierarchically structured criteria Del Vasto-Terrientes et al. (2015). Third, we plan to consider additional preference information in the form of positive and negative indications of good or bad alternatives. Such information should influence the outcomes of PrefRank to, e.g., increase the strength and decrease the weakness of an alternative that is judged good by the DM. Finally, when using the methods from the same family to a given problem, it would be possible to combine their indications into a compromise one. The example algorithms applicable for this purpose have been proposed in Miebs and Kadziński (2021).

## Acknowledgments

Miłosz Kadziński and Krzysztof Martyn acknowledge financial support from the Polish National Science Center under the SONATA BIS project (grant no. DEC-2019/34/E/HS4/00045).

## References

- Angilella, S., Pappalardo, M. R. (2021). Assessment of a failure prediction model in the European energy sector: A multicriteria discrimination approach with a PROMETHEE based classification. *Expert Systems with Applications* 184, 115513.
- Arcidiacono, S. G., Corrente, S., and Greco, S. (2018). GAIA-SMAA-PROMETHEE for a hierarchy of interacting criteria. *European Journal of Operational Research*, 270(2):606–624.
- Behzadian, M., Kazemzadeh, R., Albadvi, A., and Aghdasi, M. (2010). PROMETHEE: A comprehensive literature review on methodologies and applications. *European Journal of Operational Research*, 200(1):198–215.
- Bottero, M., D’Alpaos, C., and Oppio, A. (2019). Ranking of adaptive reuse strategies for abandoned industrial heritage in vulnerable contexts: A multiple criteria decision aiding approach. *Sustainability*, 11(3).
- Brans, J.-P. and De Smet, Y. (2016). *PROMETHEE Methods*, pages 187–219. Springer New York, New York, NY.
- Brans, J. P., Mareschal, B., and Vincke, P. (1984). PROMETHEE: a new family of outranking methods in multicriteria analysis. In Brans, J., editor, *Operational Research, IFORS 84*, pages 477–490. North Holland, Amsterdam.
- Cinelli, M., Kadziński, M., Gonzalez, M., and Słowiński, R. (2020). How to support the application of multiple criteria decision analysis? Let us start with a comprehensive taxonomy. *Omega*, 96:102261.
- Cinelli, M., Kadziński, M., Miebs, G., Gonzalez, M., and Słowiński, R. (2022). Recommending multiple criteria decision analysis methods with a new taxonomy-based decision support system. *European Journal of Operational Research*, 302(2):633–651.
- Coppola, M., Guo, J., Gill, E., and de Croon, G. C. H. E. (2019). The PageRank algorithm as a method to optimize swarm behavior through local analysis. *Swarm Intelligence*, 13(3):277–319.

- Corrente, S., Figueira, J. R., and Greco, S. (2014a). Dealing with interaction between bipolar multiple criteria preferences in PROMETHEE methods. *Annals of Operations Research*, 217(1):137–164.
- Corrente, S., Figueira, J. R., and Greco, S. (2014b). The SMAA-PROMETHEE method. *European Journal of Operational Research*, 239(2):514–522.
- Corrente, S., Figueira, J. R., Greco, S., and Słowiński, R. (2017). A robust ranking method extending ELECTRE III to hierarchy of interacting criteria, imprecise weights and stochastic analysis. *Omega*, 73:1–17.
- de Almeida Filho, A. T., Clemente, T. R., Morais, D. C., and de Almeida, A. T. (2018). Preference modeling experiments with surrogate weighting procedures for the PROMETHEE method. *European Journal of Operational Research*, 264(2):453–461.
- Dejaegere, G., Boujelben, M. A., and De Smet, Y. (2022). An axiomatic characterization of Promethee II's net flow scores based on a combination of direct comparisons and comparisons with third alternatives. *Journal of Multi-Criteria Decision Analysis*, 29(5-6):364–380.
- Del Vasto-Terrientes, L., Valls, A., Slowinski, R., Zielniewicz, P. (2015). ELECTRE-III-H: An outranking-based decision aiding method for hierarchically structured criteria. *Expert Systems with Applications* 42 (11):4910–4926.
- Dias, L. C. and Lamboray, C. (2010). Extensions of the prudence principle to exploit a valued outranking relation. *European Journal of Operational Research*, 201(3):828–837.
- Figueira, J. and Roy, B. (2002). Determining the weights of criteria in the ELECTRE type methods with a revised Simos' procedure. *European Journal of Operational Research*, 139(2):317–326.
- Figueira, J. R., Greco, S., and Roy, B. (2009). ELECTRE methods with interaction between criteria: An extension of the concordance index. *European Journal of Operational Research*, 199(2):478–495.
- Figueira, J. R., Mousseau, V., and Roy, B. (2016). *ELECTRE Methods*, pages 155–185. Springer New York, New York, NY.
- Geldermann, J., Spengler, T., and Rentz, O. (2000). Fuzzy outranking for environmental assessment. Case study: iron and steel making industry. *Fuzzy Sets and Systems*, 115(1):45–65.
- Gleich, D. F. (2015). PageRank Beyond the Web. *SIAM Review*, 57(3):321–363.
- Govindan, K., Kadziński, M., Sivakumar, R. (2017). Application of a novel PROMETHEE-based method for construction of a group compromise ranking to prioritization of green suppliers in food supply chain. *Omega* 71: 129 – 145.
- Hu, Y.-C. and Chen, C.-J. (2011). A PROMETHEE-based classification method using concordance and discordance relations and its application to bankruptcy prediction. *Information Sciences*, 181(22):4959–4968.
- Jaszkiewicz, A. and Słowiński, R. (1994). The light beam search over a non-dominated surface of a multiple-objective programming problem. In Tzeng, G. H., Wang, H. F., Wen, U. P., and Yu, P. L., editors, *Multiple Criteria Decision Making*, pages 87–99, New York, NY. Springer New York.
- Kadziński, M., Greco, S., and Słowiński, R. (2012). Extreme ranking analysis in robust ordinal regression. *Omega*, 40(4):488 – 501.
- Kadziński, M. and Michalski, M. (2016). Scoring procedures for multiple criteria decision aiding with robust and stochastic ordinal regression. *Computers & Operations Research*, 71:54 – 70.
- Kadziński, M. and Tervonen, T. (2013). Robust multi-criteria ranking with additive value models and holistic pair-wise preference statements. *European Journal of Operational Research*, 228(1):169–180.
- Keeney, R. L. and Robillard, G. A. (1977). Assessing and evaluating environmental impacts at proposed nuclear power plant sites. *Journal of Environmental Economics and Management*, 4(2):153–166.
- Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.
- Kwak, H., Lee, C., Park, H., and Moon, S. (2010). What is Twitter, a Social Network or a News Media? In *Proceedings of the 19th International Conference on World Wide Web, WWW '10*, pages 591–600, New York, NY, USA. Association for Computing Machinery.
- Lempel, R. and Moran, S. (2000). The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks*, 33(1):387–401.
- Leyva-Lopez, J. C. and Aguilera-Contreras, M. A. (2005). A multiobjective evolutionary algorithm for deriving final ranking from a fuzzy outranking relation. In Coello Coello, C. A., Hernández Aguirre, A., and Zitzler, E., editors, *Evolutionary Multi-Criterion Optimization*, pages 235–249, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Liu, J., Liao, X., Kadziński, M., Mao, X., and Wang, Y. (2020). A preference learning framework for multiple criteria sorting with diverse additive value models and valued assignment examples. *European Journal of Operational Research*, 286(3):963–985.
- Lolli, F., Balugani, E., Ishizaka, A., Gamberini, R., Butturi, M. A., Marinello, S., Rimini, B. (2019). On the elicitation of criteria weights in PROMETHEE-based ranking methods for a mobile application. *Expert Systems with Applications* 120:217–227.
- Macharis, C., Brans, J. P., and Mareschal, B. (1998). The GDSS PROMETHEE procedure: a PROMETHEE-GAIA based procedure for group decision support. *Journal of Decision Systems*, 7:283–307.
- Mareschal, B. and De Smet, Y. (2009). Visual PROMETHEE: Developments of the PROMETHEE & GAIA multicriteria decision aid methods. In *2009 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 1646–1649.
- Meyer, P. and Bigaret, S. (2012). Diviz: A Software for Modeling, Processing and Sharing Algorithmic Workflows in MCDA. *Intelligent Decision Technologies*, 6(4):283–296.
- Miebs, G. and Kadziński, M. (2021). Heuristic algorithms for aggregation of incomplete rankings in multiple criteria group decision making. *Information Sciences*, 560:107–136.
- Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report,

Stanford InfoLab.

- Pelissari, R., Duarte, L. T. (2022). SMAA-Choquet-FlowSort: A novel user-preference-driven Choquet classifier applied to supplier evaluation. *Expert Systems with Applications* 207:117898.
- Perny, P. and Roy, B. (1992). The use of fuzzy outranking relations in preference modelling. *Fuzzy Sets and Systems*, 49(1):33–53.
- Rezaei, J. (2015). Best-worst multi-criteria decision-making method. *Omega*, 53:49–57.
- Roy, B. and Slowinski, R. (1993). Criterion of distance between technical programming and socio-economic priority. *RAIRO - Operations Research - Recherche Opérationnelle*, 27(1):45–60.
- Roy, B. and Słowiński, R. (2008). Handling effects of reinforced preference and counter-veto in credibility of outranking. *European Journal of Operational Research*, 188(1):185–190.
- Saaty, T. L. (1990). How to make a decision: The analytic hierarchy process. *European Journal of Operational Research*, 48(1):9–26.
- Siskos, E., Askounis, D., and Psarras, J. (2014). Multicriteria decision support for global e-government evaluation. *Omega*, 46:51–63.
- Szeląg, M., Greco, S., and Słowiński, R. (2014). Variable consistency dominance-based rough set approach to preference learning in multicriteria ranking. *Information Sciences*, 277:525–552.
- Wang, J. R. (2001). Ranking engineering design concepts using a fuzzy outranking preference model. *Fuzzy Sets and Systems*, 119(1):161–170.
- Wang, R. and Li, Y.-L. (2018). A Novel Approach for Group Decision-Making from Intuitionistic Fuzzy Preference Relations and Intuitionistic Multiplicative Preference Relations. *Information*, 9(3).
- Wątróbski, J., Jankowski, J., Ziemia, P., Karczmarczyk, A., Ziolo, M. (2019). Generalised framework for multi-criteria method selection. *Omega* 86:107–124.
- Wedlin, L. (2007). The role of rankings in codifying a business school template: classifications, diffusion and mediated isomorphism in organizational fields. *European Management Review*, 4(1):24–39.
- Żak, J. (2005). *Multiple criteria decision making in road transport (in Polish)*. Poznań University of Technology Press, Poznań.
- Ziemia, P. (2021). Multi-criteria approach to stochastic and fuzzy uncertainty in the selection of electric vehicles with high social acceptance. *Expert Systems with Applications* 173:114686.



# PrefRank: a family of multiple criteria decision analysis methods for exploiting a valued preference relation – eAppendix

Krzysztof Martyn<sup>a,\*</sup>, Magdalena Martyn<sup>a</sup>, Miłosz Kadziński<sup>a</sup>

<sup>a</sup>*Institute of Computing Science, Faculty of Computing and Telecommunications, Poznan University of Technology, Piotrowo 2, 60-965 Poznań, Poland. e-mails: krzysztof.martyn@cs.put.poznan.pl, magdalena.martyn@cs.put.poznan.pl, milosz.kadziński@cs.put.poznan.pl*

---

---

## 1. Computational procedure used in PrefRank

In this section, we discuss the method used for computing the strengths and weaknesses of different variants of PrefRank. We focus first on explaining the procedure in the context of deriving the strengths by PrefRank I (see Algorithm 1). It is inspired by the iterative methods used in the PageRank algorithm. It exploits matrix  $M$  of comprehensive preference degrees. In the beginning, strengths are equally distributed among all  $n$  alternatives, being set to  $\frac{1}{n}$ . Then, the computations are conducted iteratively until the difference between values obtained in the two successive iterations for any alternative is greater than the predefined threshold  $thr$  (we set it to 0.00001) or the maximum number of iterations  $maxstep$  has not been reached. In each iteration, the strengths from the previous iteration are stored, and the new strengths are calculated by multiplying matrix  $M$  by the vector of old strengths. If all strengths are equal to zero in some iteration, we terminate with the results from the previous iteration. At the end of each iteration, the strengths are normalized, to sum up to one.

---

**Algorithm 1:** Computation of the strengths  $S^+$  by PrefRank I based on the matrix  $M$  of comprehensive preference degrees.

---

```
1  $\forall_{i=1}^n S^+(a_i) = \frac{1}{n}$ 
2  $step = 0$ 
3 do
4    $S_{prev}^+ = S$ 
5    $S^+ = MS^+$ 
6   if  $\sum_{j=1}^n S^+(a_j) == 0$  then
7      $S^+ = S_{prev}^+$ 
8     break
9   end
10   $\forall_{i=1}^n S^+(a_i) = \frac{S^+(a_i)}{\sum_{j=1}^n S^+(a_j)}$ 
11   $step = step + 1$ 
12 while  $\max_i |S^+(a_i) - S_{prev}^+(a_i)| > thr$  and  $step < maxstep$ ;
```

---

It may happen that after reaching the maximum number of iterations  $maxstep$ , the results still do not converge, i.e., the greatest difference between strengths in the last two iterations is greater than the specified threshold  $thr$ . Such a situation may occur in the case of a preference cycle. Then, to terminate the computational procedure with the correct results, Algorithm 1 is modified to average the results from the previous two iterations ( $S = \frac{S+S_{prev}}{2}$ ).

---

\*Corresponding author: Institute of Computing Science, Faculty of Computing and Telecommunications, Poznan University of Technology, Piotrowo 2, 60-965 Poznań, Poland. e-mail: krzysztof.martyn@cs.put.poznan.pl; Tel. +48 61 665 3022; Fax: +48 61 8771525.

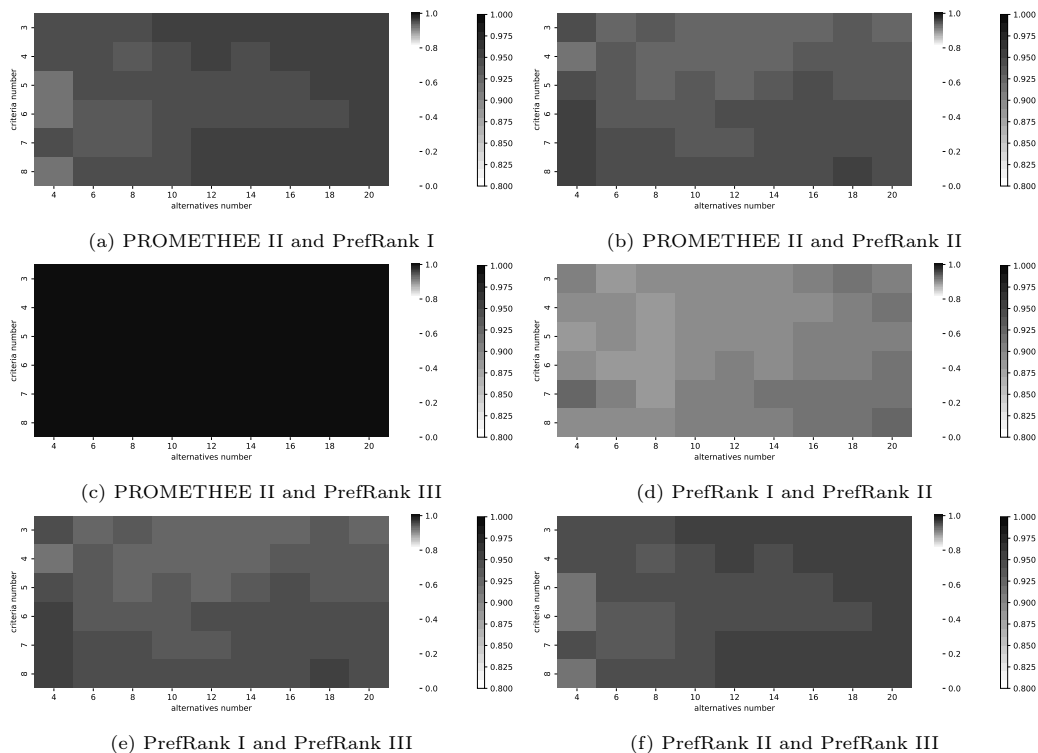
The computational procedure used by PrefRank to compute the weakness  $S^-$  is the same with the proviso that a transpose preference matrix  $M^T$  is exploited. Similarly, the remaining variants of PrefRank differ only in terms of matrices they exploit (see Table 1). In particular, PrefRank II computes the strengths based on  $MM^T$ . In turn, PrefRank III does the same by analyzing  $W_r W_c^T$ , where  $W_r$  and  $W_c$  are obtained by dividing the preference matrix  $M$  by the sum of values in the respective rows or columns, respectively.

Tabela 1: Matrices exploited to compute strengths and weaknesses by different variants of PrefRank.

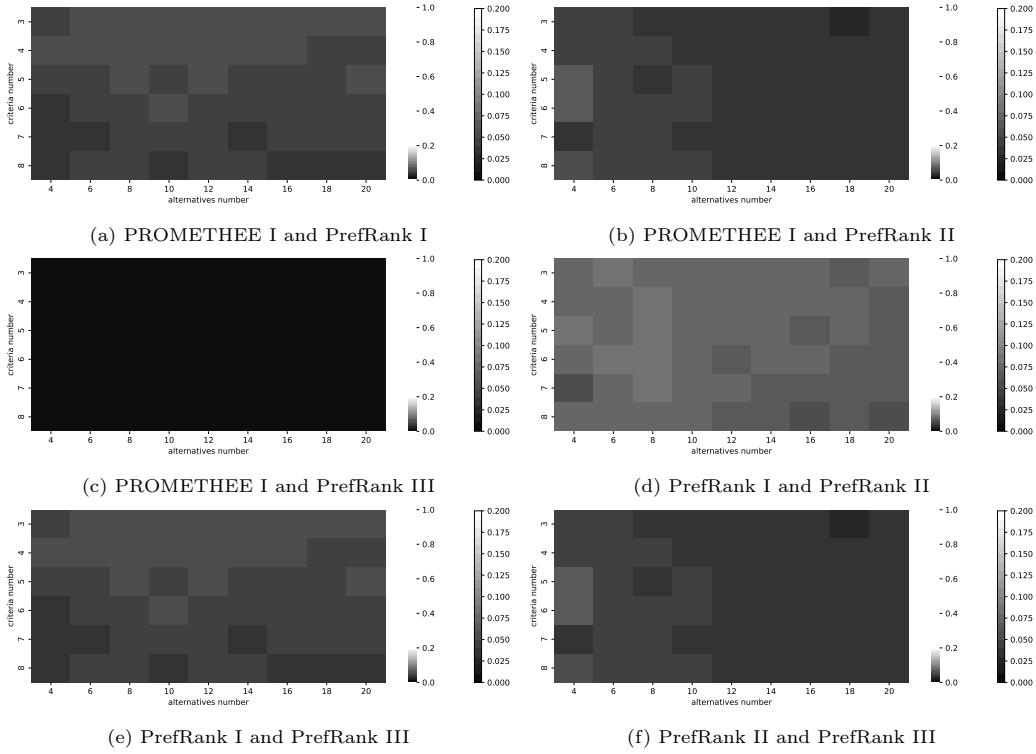
Method	Result	Matrix
PrefRank I	$S^+$	$M$
	$S^-$	$M^T$
PrefRank II	$S^+$	$MM^T$
	$S^-$	$M^T M$
PrefRank III	$S^+$	$W_r W_c^T$
	$S^-$	$W_c^T W_r$

## 2. Detailed similarity results between rankings based on the experimental comparison of four methods

In this section, we present heatmaps of average Kendall's  $\tau$  (see Figure 1) and  $NRD$  (see Figure 2) values based on 100 runs for each problem size. The number of alternatives ranges between 4 and 20, whereas the number of criteria is between 3 and 8. Six pairs of methods are considered. Kendall's  $\tau$  quantifies the similarity on the scale between  $-1$  (the least similarity) and  $1$  (the greatest similarity) based on complete rankings. In turn,  $NRD$  operates on the scale between  $0$  (the least difference) and  $1$  (the greatest difference) based on incomplete rankings.



Rysunek 1: Heatmaps of average Kendall's  $\tau$  values based on 100 runs for each problem size.



Rysunek 2: Heatmaps of average NRD values based on 100 runs for each problem size.

### 3. Results obtained for the fleet selection problem with the distillation procedure

An alternative procedure for constructing a complete ranking is inspired by the distillation method proposed as a part of ELECTRE III. First, we can consider the comprehensive qualities of alternatives to identify the subset of the most preferred ones at the current stage. Then, they are added to the highest available position and eliminated from further consideration. The remaining alternatives are analyzed in the same way until the subset of alternatives to be considered is empty.

In this section, we report the results obtained using such a procedure for the fleet selection problem considered in the main paper. The complete orders based on the four methods are presented in Table 2. They agree in terms of ranking  $a_1$  at the top. Only for PrefRank I,  $a_1$  shares the first position with  $a_4$ . The most interesting observation with respect to the outcomes described in the main paper consists of the difference in rankings produced by PROMETHEE II and PrefRank III. Specifically, the former ranks  $a_8$  better than  $a_3$ , whereas the latter judges these alternatives indifferent. This is because when these two alternatives are the only ones to be considered, the methods consider positive preference degrees of  $a_3$  over  $a_8$  and vice versa. However, PROMETHEE takes into account the difference between these degrees, hence favoring  $a_8$  because  $\pi(a_8, a_3)$  is greater than  $\pi(a_3, a_8)$ . In turn, PrefRank III considers a dual role of each alternative and applies the normalization of preference degrees. This leads to the same strengths and weaknesses for both  $a_3$  and  $a_8$ .

Tabela 2: Rankings obtained for the fleet selection problem using the iterative distillation procedure.

Rank	PROMETHEE I	PrefRank I	PrefRank II	PrefRank III
1	$a_1$	$a_1, a_4$	$a_1$	$a_1$
2	$a_4, a_6$	$a_6$	$a_4, a_6$	$a_4, a_6$
3	$a_2$	$a_2$	$a_2$	$a_2$
4	$a_7$	$a_7$	$a_7$	$a_7$
5	$a_5, a_9$	$a_5, a_9$	$a_9$	$a_5, a_9$
6	$a_8$	$a_8$	$a_5$	$a_3, a_8$
7	$a_3$	$a_3$	$a_8$	
8			$a_3$	



## Publication [P5]

K. Martyn, M. Martyn, and M. Kadziński. ScoreBin: scoring alternatives based on a crisp outranking relation with an application to the assessment of Polish technological parks. *Information Sciences*, 2023. Submitted



# ScoreBin: scoring alternatives based on a crisp outranking relation with an application to the assessment of Polish technological parks

Krzysztof Martyn<sup>a,\*</sup>, Magdalena Martyn<sup>a</sup>, Miłosz Kadziński<sup>a</sup>

<sup>a</sup>*Faculty of Computing and Telecommunications, Poznan University of Technology, Piotrowo 2, 60-965 Poznań, Poland*

---

## Abstract

We introduce a suite of multiple criteria methods that exploit a crisp outranking relation in the set of alternatives. For each option, we analyze the outranked and outranking alternatives to compute its strength, weakness, and comprehensive quality. We propose four variants that differ in weights assigned to outranking or being outranked by particular alternatives and how to quantify the difficulty or easiness of instantiating such relations. The scores derived from the analysis of an outranking graph can be enhanced by the Decision Makers' holistic judgments. They indicate subsets of alternatives deemed as comprehensively strong or weak. We apply the proposed methods to a case study concerning the performance of technological parks in Poland. We also compare the results obtained with the novel approaches with the state-of-the-art ELECTRE methods in an extensive experiment involving simulated decision problems.

*Keywords:* Multiple criteria decision aiding, Ranking, Choice, Scoring, Outranking relation, Technological parks

---

## 1. Introduction

Outranking methods in Multiple Criteria Decision Analysis (MCDA) involve comparing alternatives pairwise with respect to multiple criteria [20, 37]. Valued (fuzzy) and crisp (binary) outranking relations are different approaches used to characterize the preference structure in the set of alternatives. The former represents degrees of preference by assigning a numerical value to each pair [32, 34]. In this way, it provides nuanced and detailed information on the relationship's strength, allowing for finer distinctions. In turn, crisp outranking is a binary relation, either present or absent, without any degree of membership or uncertainty [11, 46]. If one alternative outranks another in overall performance, it is judged at least as good.

We focus on the exploitation of a crisp outranking relation. This is typically done by outranking-based approaches from the family of ELECTRE [15]. They assume the crisp relation holds if sufficient arguments support the outranking and there are no essential reasons to refute the assertion. ELECTRE methods are most helpful in handling decision problems that involve up to several criteria with heterogeneous performance scales, some of them involve qualitative or ordinal assessments, the knowledge related to the definition of criteria and respective performances is imperfect, and one wishes to avoid a full compensation effect [17]. Under such an environment, they help Decision Makers (DMs) make informed choices among different alternatives or rank them from the best to the worst [18].

Let us discuss the most popular choice and ranking techniques that exploit a crisp outranking relation. First, when dealing with a choice problem, one may select either the alternatives that outrank the greatest number of other options or are outranked by the least number of other actions [5]. This way of proceeding, implementing the plurality and anti-plurality rules, is known from the Social Choice Theory (SCT). Second, ELECTRE I identifies a kernel in an outranking graph as a subset of alternatives satisfying the properties of external and internal stabilities [19, 42]. That is, the alternatives contained in the kernel do not outrank each other, and those outside the kernel are outranked by at least one alternative from the kernel. Third, the Net Flow Score (NFS) procedure can be used to determine the

---

\*Corresponding author: Institute of Computing Science, Poznań University of Technology, Piotrowo 2, 60-965 Poznań, Poland. Tel. +48-61 665 3022.

*Email addresses:* krzysztof.martyn@cs.put.poznan.pl (Krzysztof Martyn), magdalena.martyn@cs.put.poznan.pl (Magdalena Martyn), milosz.kadzinski@cs.put.poznan.pl (Miłosz Kadziński)

strength and weakness of each alternative as the number of other options that are outranked by this alternative or outrank it [44]. These measures can be aggregated into a comprehensive quality measure by subtracting the weakness from the strength [6]. Alternatively, the rankings based on each option’s strong and weak points can be intersected to obtain a partial pre-order. Fourth, the rankings can be constructed iteratively using the descending and ascending distillation procedures [38] originally conceived in ELECTRE III to deal with a valued outranking relation [36]. In each iteration of this approach, either the best or the worst alternative is added to the order constructed in a top-down or a bottom-up manner. Then, the procedure is repeated until all alternatives are added to the ranking. Finally, ELECTRE-Score assigns a score range to each alternative by comparing it against reference alternatives [14]. They serve as limiting profiles associated with some precise scores derived from applying the deck of cards method.

Each method mentioned above poses some significant challenges in real-world decision aiding. As far as the approaches inspired by SCT are concerned, they do not consider the relations of recommended alternatives with other options. As a result, they may fail to indicate the alternatives that are strictly preferred over the selected ones. Regarding ELECTRE I, the cardinality of a graph kernel cannot be controlled. Hence, even if the DMs want to select a single best option, they may be left with a greater subset. Moreover, due to the kernel’s definition, it has some desirable properties considered as a whole; however, the individual alternatives contained in it can be very poor (e.g., being outranked by many alternatives not contained in the kernel and not outranking any other alternative). Also, ELECTRE I cannot rank alternatives in the presence of incomparabilities and intransitiveness [48]. As far as NFS is concerned, it does not differentiate between the quality of alternatives that outrank or are outranked by other options. In particular, being at least as good as some highly favorable option or inferior action is equally desirable. Similarly, being outranked by some advantageous alternative or bad option counts the same. Further, the distillation procedures fail to assign explicit, comprehensive scores or numerical values to alternatives. This is often needed when a cardinal ranking is expected at the method’s output. Also, the distillation does not consider the difficulty or easiness of outranking other alternatives. Moreover, ELECTRE-Score requires the specification of additional reference profiles and preference information that would allow to assign them some precise scores. Doing so, it is more cognitively demanding. Also, even if it assigns some scores to the alternatives, its operating procedure resembles a sorting method called ELECTRE Tri-nC that was originally proposed for assigning alternatives to preference-ordered classes rather than ranking them [2]. Finally, none of the methods mentioned above accepts additional indirect preference information to impact the ranking construction process once the outranking relation is already established. Consequently, influencing the quality (strength or weakness) of alternatives can be attained only by modifying the directly provided preference model parameters.

This paper proposes a suite of methods, ScoreBin, exploiting a crisp outranking relation. They assign a pair of scores – representing strength and weakness – to each alternative by comparing it with all remaining (existing) alternatives. Each score is a sum of two components; one derived from the outranking graph and the other based on the indirect DM’s feedback. These components form the most peculiar aspect of ScoreBin.

The graph component of a given alternative’s score considers which alternatives it outranks and which outrank it. However, instead of simply summarizing the numbers of such options, the strength and weakness depend on the strengths and weaknesses of other alternatives, particularly those related to it. This general idea is implemented differently in the four variants of ScoreBin. In the first variant, we increase the alternative’s strength when the alternatives it outranks are strong and increase the weakness if the alternatives that outrank it are weak. The second variant assumes that the alternative’s strength is greater when it outranks many weak alternatives, and its weakness is more significant when many strong alternatives outrank it. In the remaining two variants, the strength and weakness of each alternative depend on the options directly related to it as well as the difficulty or easiness characterizing these options when they outrank others and are outranked by others. In this spirit, the third variant of ScoreBin assumes an alternative is strong when outranking alternatives that are outranked by other strong alternatives. In turn, it is weak when being outranked by alternatives that outrank weak alternatives. On the contrary, the fourth variant supposes an alternative is strong when outranking alternatives that are hard to outrank (i.e., that are outranked by few alternatives) and it is weak when outranked by alternatives struggling to outrank (i.e., that outranks few alternatives).



The feedback component aims at adding a bonus to the strength or a penalty to the weakness based on the DM’s holistic statements. In the context of ranking and choice problems, such judgments typically have the form of pairwise comparisons [9], rank-related requirements [22], or preference intensities [16]. We introduce a novel form of comprehensive statements letting the DM indicate that an alternative is considered strong or weak. Such comprehensive assessments directly influence the strength or weakness of an assessed alternative. However, they also impact the scores attained by the remaining options indirectly via the graph component. By using this optional component, the DM may have increased control over the final ranking.

The trade-off between the two components is controlled by some intuitive parameters. Even if we propose their default values that allow deriving precise scores of alternatives, we enrich the basic framework with robustness analysis [1]. Specifically, we verify the stability of rankings attained for different parameter values using the Monte Carlo simulations [45]. The results are quantified with Rank Acceptability Indices estimating the share of possible parameter values that grant some rank to a given alternative [29].

The introduced methods are applied to a case study concerning the evaluation of technological parks in Poland [28]. These are specialized areas designed to support development, innovation, and research in various technology fields. They provide infrastructure and services to foster collaboration between business and academia. We analyze eleven parks evaluated on seven criteria with heterogeneous scales and preference directions. These concern sales costs, buildings’ surface, localization, generated profit, offered services, management assessment, and completed projects. We discuss the results of the four proposed variants of ScoreBin. We report both the partial rankings based on the separate consideration of strengths and weaknesses and the complete orders that build on the alternatives’ qualities defined as the difference between strengths and weaknesses. When discussing the study’s results, we also demonstrate the impact of indirect preference information on the strength or weakness of selected alternatives on the constructed rankings.

Our final contribution consists of conducting an extensive computational experiment. It quantifies the similarity between the recommendations provided by the four variants of ScoreBin and the existing methods over a large spectrum of simulated decision problems. The considered state-of-the-art approaches include the graph kernel approach in line with ELECTRE I [42], the simple NFS procedure referring to the number of outranked and outranking alternatives [44], and the orders established with Qualification Distillation (QD) of a crisp outranking [38]. We focus on the results relevant to choice and ranking problems. For the former, we treat the top-ranked alternatives by ScoreBin as recommended for the selection, whereas for the latter, we consider both partial and complete orders. The similarity measures are Kendall’s  $\tau$  for complete rankings [26], Rank Difference Measure (RDM) for partial pre-orders [41], Normalized Hit Ratio (NHR) [24], and average ranks of alternatives contained in the kernel.

The paper’s remainder is organized as follows. Section 2 introduces the notation and reminds the relevant state-of-the-art ELECTRE methods. In Section 3, we discuss the novel methods from the family of ScoreBin. Section 4 presents the measures used for comparing the recommendations obtained with various approaches. In Section 5, we illustrate the use of the proposed methods to assess Polish technological parks. Section 6 discusses the results of comparing the four variants of ScoreBin and three existing methods for exploiting a crisp outranking relation. The last section concludes the paper.

## 2. Notation and reminder on state-of-the-art methods exploiting a crisp outranking relation

We consider a finite set  $A = \{a_1, \dots, a_n\}$  of  $n$  alternatives evaluated on a family  $G = \{g_1, \dots, g_m\}$  of  $m$  criteria. We focus on ranking problems, aiming to order alternatives from the most to the least preferred given their performances  $g_j(a_i)$  on multiple criteria  $g_j : A \rightarrow \mathbb{R}$ ,  $j \in J = \{1, \dots, m\}$ . Without loss of generality, we consider gain-type criteria, hence preferring greater performances. The ELECTRE methods use an outranking relation  $S$  as the preference model [37]. When a crisp relation holds for a pair of alternatives  $a_i S a_k$   $a_i$  is comprehensively judged at least as good as  $a_k$ . To verify the truth of  $S$  for all pairs of alternatives, ELECTRE refers to various parameters [17].

Weight  $w_j$  associated with each criterion  $g_j$  determines its importance coefficients. To account for the uncertainty of alternatives’ performances, indifference  $q_j$  and preference  $p_j$  are associated with  $g_j$  [40]. The former expresses the

maximum difference between alternatives' performances on  $g_j$  that is negligible, implying indifference. The latter represents the minimum performance difference on  $g_j$  that implies the strict preference. Apart from the comparison thresholds, the veto threshold  $v_j$  expresses the minimal performance difference on  $g_j$ , which is so critical that it has the power to invalidate the outranking [34].

Outranking relation  $S$  is established via the concordance and discordance tests. The concordance index  $C(a_i, a_k)$  reflects the strength of the coalition of criteria supporting  $S$ :

$$C(a_i, a_k) = \sum_{j=1}^m w_j \cdot c_j(a_i, a_k) / \sum_{j=1}^m w_j, \quad (1)$$

where  $c_j(a_i, a_k)$  is a partial concordance index on  $g_j$  defined as follows:

$$c_j(a_i, a_k) = \begin{cases} 1 & \text{if } g_j(a_k) - g_j(a_i) \leq q_j, \\ \frac{g_j(a_i) - g_j(a_k) + p_j}{p_j - q_j} & \text{if } g_j(a_k) - g_j(a_i) > q_j \text{ and } g_j(a_k) - g_j(a_i) \leq p_j, \\ 0 & \text{if } g_j(a_k) - g_j(a_i) > p_j. \end{cases} \quad (2)$$

The partial discordance index  $D_j(a_i, a_k)$  captures the degree to which  $g_j$  opposes against  $a_i S a_k$ :

$$D_j(a_i, a_k) = \begin{cases} 1 & \text{if } g_j(a_k) - g_j(a_i) \geq v_j, \\ \frac{g_j(a_k) - g_j(a_i) - p_j}{v_j - p_j} & \text{if } g_j(a_k) - g_j(a_i) < v_j \text{ and } g_j(a_k) - g_j(a_i) \geq p_j, \\ 0 & \text{if } g_j(a_k) - g_j(a_i) < p_j. \end{cases} \quad (3)$$

The comprehensive concordance and partial discordances serve as the basis for computing the outranking credibility  $\sigma(a_i, a_k)$  [34]. At this stage, only sufficiently strong discordances are considered:

$$\sigma(a_i, a_k) = C(a_i, a_k) \prod_{j \in F} \frac{1 - D_j(a_i, a_k)}{1 - C(a_i, a_k)}, \quad (4)$$

where  $F = \{j : D_j(a_i, a_k) > C(a_i, a_k)\}$ . Outranking credibility can be interpreted as a valued preference relation. To transform it into a crisp one, we must compare it against the credibility threshold (cutting level)  $\lambda$ . Relation  $S$  can be represented in the form of an outranking graph where the alternatives are represented as nodes, and an arc from  $a_i$  to  $a_k$  is justified by  $a_i S a_k$ . In what follows, we will use the outranking function  $\mathbb{1}(a_i, a_k)$  defined as follows:

$$\mathbb{1}(a_i, a_k) = \begin{cases} 1 & \text{if } \sigma(a_i, a_k) \geq \lambda \text{ and } i \neq k, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Once a crisp outranking relation  $S$  is constructed, it can be exploited in the function of choice or ranking problem to provide an adequate recommendation. In what follows, we discuss three state-of-the-art methods serving this purpose. These approaches will be compared against the newly introduced ScoreBin methods.

When facing a choice problem, an outranking graph can be exploited to search for its kernel  $K \subseteq A$ , as proposed initially in ELECTRE I [17, 42]. Kernel  $K$  consists of alternatives that do not outrank any other alternative in  $K$ , whereas the alternatives outside  $K$  need to be outranked by at least one alternative contained in  $K$ . If the graph has cycles, they need to be eliminated. In this paper, we aggregate each cycle to an auxiliary node that inherits all in- and outgoing arcs of the alternatives contained in the cycle.

In turn, the NFS procedure computes the scores that can be used to order the alternatives from the best to the worst [44]. First, the strength  $S^+(a_i)$  and weakness  $S^-(a_i)$  of each alternative  $a_i$  are derived based on the numbers of

options which, respectively, are outranked by  $a_i$  and outrank  $a_i$ :

$$S^+(a_i) = \sum_{k=1}^n \mathbb{1}(a_i, a_k) \text{ and } S^-(a_i) = \sum_{k=1}^n \mathbb{1}(a_k, a_i). \quad (6)$$

Then, the comprehensive quality is calculated as the difference between strength and weakness:

$$S(a_i) = S^+(a_i) - S^-(a_i). \quad (7)$$

Alternatively, the quality measures mentioned above can be incorporated into the QD [38], inspired by ELECTRE III [36, 38]. The method constructs two complete preorders (descending and ascending). In the descending distillation, one orders the alternatives from the best to the worst, while in the ascending one, the ranking is constructed bottom-up. The distillations proceed iteratively, trying to determine a subset of alternatives with extreme quality, adding them to the constructed preorder, and eliminating them from further consideration. The ascending distillation focuses on the alternatives with the greatest quality and adds them to the currently lowest position (hence above all alternatives that have not been yet added to the ranking). In turn, the descending distillation identifies the alternatives with the least quality. It adds them to the current highest position (hence below all alternatives that have not yet been added to the ranking). In the case of a tie, the procedure tries to break it by running the internal distillation, whose scope is limited only to a subset of alternatives with the same quality score. The algorithm continues until all alternatives are added to the constructed order. Finally, the two rankings are intersected to obtain a final ranking, being a partial preorder.

### 3. ScoreBin: a new family of scoring methods exploiting a crisp outranking relation

This section presents a family of ScoreBin methods that exploit a crisp outranking relation defined by function  $\mathbb{1}(a_i, a_k)$  for  $a_i, a_k \in A$ . Similarly to NFS, we compute the strength  $S^+(a_i)$  and weakness  $S^-(a_i)$  of each alternative  $a_i \in A$ . They are subsequently combined into a comprehensive quality measure. However, the definition of elementary scores differs substantially from NFS. Specifically, the strength (weakness) is defined as a sum of a graph-based component  $G^+(a_i)$  ( $G^-(a_i)$ ) and a bonus (penalty) score  $b_i^+$  ( $b_i^-$ ):

$$S^+(a_i) = b_i^+ + \frac{G^+(a_i)}{\max_{a_k \in A} G^+(a_k)} \text{ and } S^-(a_i) = b_i^- + \frac{G^-(a_i)}{\max_{a_k \in A} G^-(a_k)}. \quad (8)$$

The graph component builds on the outranking graph. To increase its interpretability, it is normalized to the  $[0, 1]$  interval by dividing it through the maximal score attained by some alternative. Hence, an alternative with the greatest strength or weakness derived from analyzing its relations with the remaining alternatives has this component of the score equal to one. The graph-based strength of  $a_i$  is defined as a sum of weights  $\omega^+(a_k)$  associated with the options outranked by  $a_i$ . In turn, its graph-based weakness is a sum of weights  $\omega^-(a_k)$  linked to the options that outrank  $a_i$ , i.e.:

$$G^+(a_i) = \sum_{k=1}^n \mathbb{1}(a_i, a_k) \cdot \omega^+(a_k) \text{ and } G^-(a_i) = \sum_{k=1}^n \mathbb{1}(a_k, a_i) \cdot \omega^-(a_k). \quad (9)$$

The interpretation of weights  $\omega^+(a_k)$  and  $\omega^-(a_k)$  depends on the specific variant of ScoreBin. However, irrespective of their definition, alternatives that do not outrank any other option have  $G^+(a_i) = 0$ , while alternatives that are not outranked by any other option have  $G^-(a_i) = 0$ . If the comprehensive scores involved only the graph-based components, it might lead to undesirable situations. Precisely, outranking an option with  $\omega^+(a_k) = 0$  would not add anything to the alternative's strength, while being outranked by an option with  $\omega^-(a_k) = 0$  would not increase the alternative's weakness. To prevent such effects and, in addition, let the methods incorporate additional DM's preferences, the comprehensive score involves the other component.

To ensure the minimal impact of each option  $a_i \in A$  on the strength or weakness of alternatives it is related to, we include a base bonus  $\alpha^+$  or a penalty  $\alpha^-$  in its score. Since they need to be positive, the worst alternative in terms of

strength and the best alternative in terms of weakness always have a comprehensive score greater than zero. In this way, they impact the strengths and weaknesses of the remaining alternatives.

Also, we let the DMs provide indirect preference information. They may specify which alternatives are strong, being included in  $A_{strong}^* \subseteq A$ , and which are weak, being included in  $A_{weak}^* \subseteq A$ . No alternative can be simultaneously included in both sets, i.e.,  $A_{strong}^* \cap A_{weak}^* = \emptyset$ . Such additional information may be incorporated as a bonus  $\beta^+$  or a penalty  $\beta^-$  to the alternative's strength or weakness, respectively. Given the bonuses and penalties may serve two purposes, it is possible to consider them under a single variable:

$$b_i^+ = \begin{cases} \beta^+ & \text{if } a_i \in A_{strong}^*, \\ \alpha^+ & \text{else,} \end{cases} \quad \text{and } b_i^- = \begin{cases} \beta^- & \text{if } a_i \in A_{weak}^*, \\ \alpha^- & \text{else,} \end{cases} \quad (10)$$

where  $\alpha \in (0, 1)$  is a base bonus (penalty) indicating the alternative's minimal strength (weakness), hence letting it influence the scores of remaining options. In turn,  $\beta \in \mathbb{R}_{\geq \alpha}$  is the enhancement value based on the DM's indirect preferences.

Note that the positive (negative) feedback directly influences only the strength (weakness) of the alternative assessed by the DM. However, it might also indirectly change the scores of other options via the outranking graph's structure. The values of parameters  $\alpha$  and  $\beta$  need to be specified beforehand. The former is a share of the unitary score attained from the graph-based component by the best alternative. Hence, when  $\alpha = 0.1$ , the minimal value some alternative can obtain would be ten times lesser than the score obtained from the analysis of outranking relation by the most favorable option. In turn,  $\beta$  is a minimum strength (weakness) value assigned to an alternative judged by the DM as strong (weak). Overall, the strengths  $S^+(a_i)$  take values from the  $[\alpha^+, 1 + \beta^+]$  interval when  $A_{strong}^* \neq \emptyset$ , and  $[\alpha^+, 1 + \alpha^+]$ , otherwise. The analogous intervals for weakness  $S^-(a_i)$  are  $[\alpha^-, 1 + \beta^-]$  and  $[\alpha^-, 1 + \alpha^-]$ .

The strengths and weaknesses are computed using an iterative method. First, we assume that they are the same for all alternatives  $S^+(a_i) = S^-(a_i) = \frac{1}{n}$ . Then, in each iteration, they are transformed using Eqs. (8) and (9). The procedure is repeated until the greatest difference between scores obtained in two consecutive iterations is negligible (e.g., lower than some predefined threshold).

In the following subsections, we discuss the four variants of ScoreBin. They differ in assumptions in calculating weights  $\omega^+(a_i)$  and  $\omega^-(a_i)$ . Moreover, we illustrate their use on an example problem involving six alternatives ( $a_1 - a_6$ ) with outranking relation given in Table 1. We first consider the setting without additional preference information, i.e.,  $A_{strong}^* = A_{weak}^* = \emptyset$ . We assume that  $\alpha^+ = \alpha^- = 0.1$ . Intuitively,  $a_5$  should be relatively strong because it outranks three other alternatives while being outranked only by a single option. On the contrary,  $a_6$  should be rather weak as it does not outrank any other alternatives and is outranked by two options.

Table 1: Example outranking function for the problem involving six alternatives.

$\mathbf{1}(a_i, a_k)$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
$a_1$	0	0	1	1	1	0
$a_2$	0	0	0	1	0	1
$a_3$	0	0	0	1	0	0
$a_4$	0	0	0	0	0	0
$a_5$	0	1	0	1	0	1
$a_6$	0	0	0	0	0	0

### 3.1. ScoreBin I

The first variant of ScoreBin increases the strength of  $a_i$  when it outranks strong alternatives (i.e., with high  $S^+(a_k)$ ) and increases the weakness of  $a_i$  when it is outranked by weak alternatives (i.e., with high  $S^-(a_k)$ ). Hence, it assumes the following weights:

$$\omega^+(a_k) = S^+(a_k) \quad \text{and} \quad \omega^-(a_k) = S^-(a_k). \quad (11)$$

Hence, the strength increases with outranking more alternatives that are strong based on the outranking graph and/or the DM's direct feedback. In the same spirit, the weakness is more significant when being outranked by many

alternatives that are weak based on the outranking relations and/or the DM's indication of poor options.

The strengths (weaknesses) in ScoreBin I may be interpreted as a result of a weighted voting system, where each alternative votes for options that outrank it (are outranked by it) with the vote's strength equal to  $S^+$  ( $S^-$ ). We assume that the minimal value of such strength (weaknesses) is  $b^+$  ( $b^-$ ). The method is inspired by TrustRank [21], which considers a graph formed by the websites (nodes) and links (arcs) between them while assigning an additional bonus to the sites verified as trusted by an oracle.

The strengths, weaknesses, and qualities of all alternatives obtained with ScoreBin I for the example problem are given in Table 2. In addition, we report the unnormalized graph-based scores that facilitate the understanding of computations made by the method. Let us focus on alternatives  $a_5$  and  $a_6$ .

Table 2: Strengths, weaknesses, and qualities derived by the four variants of ScoreBin for the example problem.

ScoreBin	Value	Final score						Unnormalized graph component						
		$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	Value	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
I	$S^+$	1.100 (1)	0.311 (3)	0.206 (4)	0.100 (5)	0.640 (2)	0.100 (5)	$G^+$	0.946	0.200	0.100	0.000	0.511	0.000
	$S^-$	0.100 (1)	0.345 (4)	0.214 (2)	1.100 (6)	0.214 (2)	0.740 (5)	$G^-$	0.000	0.214	0.100	0.874	0.100	0.560
	$S$	1.000 (1)	-0.034 (4)	-0.009 (3)	-1.000 (6)	0.426 (2)	-0.640 (5)							
II	$S^+$	0.941 (2)	0.912 (3)	0.606 (4)	0.100 (5)	1.100 (1)	0.100 (5)	$G^+$	1.829	1.765	1.100	0.000	2.174	0.000
	$S^-$	0.100 (1)	0.409 (4)	0.364 (2)	1.100 (6)	0.364 (2)	0.665 (5)	$G^-$	0.000	1.100	0.941	3.559	0.941	2.012
	$S$	0.841 (1)	0.503 (3)	0.242 (4)	-1.000 (6)	0.736 (2)	-0.565 (5)							
III	$S^+$	0.901 (3)	0.936 (2)	0.632 (4)	0.100 (5)	1.100 (1)	0.100 (5)	$G^+$	5.372	5.605	3.569	0.000	6.705	0.000
	$S^-$	0.100 (1)	0.418 (4)	0.365 (2)	1.100 (6)	0.365 (2)	0.675 (5)	$G^-$	0.000	2.193	1.831	6.899	1.831	3.968
	$S$	0.801 (1)	0.518 (3)	0.267 (4)	-1.000 (6)	0.735 (2)	-0.575 (5)							
IV	$S^+$	1.100 (1)	0.296 (3)	0.172 (4)	0.100 (5)	0.578 (2)	0.100 (5)	$G^+$	21.562	4.225	1.562	0.000	10.298	0.000
	$S^-$	0.100 (1)	0.211 (4)	0.165 (2)	1.100 (6)	0.165 (2)	0.486 (5)	$G^-$	0.000	2.016	1.176	18.193	1.176	7.016
	$S$	1.000 (1)	0.085 (3)	0.007 (4)	-1.000 (6)	0.413 (2)	-0.386 (5)							

The unnormalized graph-based strength of  $a_5$  derives from the strengths of alternatives it outranks, i.e.,  $a_2$ ,  $a_4$ , and  $a_6$ :

$$\begin{aligned}
G^+(a_5) &= \mathbb{1}(a_5, a_1) \cdot S^+(a_1) + \mathbb{1}(a_5, a_2) \cdot S^+(a_2) + \mathbb{1}(a_5, a_3) \cdot S^+(a_3) + \\
&\quad + \mathbb{1}(a_5, a_4) \cdot S^+(a_4) + \mathbb{1}(a_5, a_5) \cdot S^+(a_5) + \mathbb{1}(a_5, a_6) \cdot S^+(a_6) = \\
&= 0 \cdot S^+(a_1) + 1 \cdot S^+(a_2) + 0 \cdot S^+(a_3) + 1 \cdot S^+(a_4) + 0 \cdot S^+(a_5) + 1 \cdot S^+(a_6) = 0.311 + 0.1 + 0.1 = 0.511
\end{aligned}$$

In turn, the unnormalized graph-based weakness of  $a_5$  builds on the weaknesses of alternatives that outrank it, i.e., only  $a_1$ :

$$\begin{aligned}
G^-(a_5) &= \mathbb{1}(a_1, a_5) \cdot S^-(a_1) + \mathbb{1}(a_2, a_5) \cdot S^-(a_2) + \mathbb{1}(a_3, a_5) \cdot S^-(a_3) + \\
&\quad + \mathbb{1}(a_4, a_5) \cdot S^-(a_4) + \mathbb{1}(a_5, a_5) \cdot S^-(a_5) + \mathbb{1}(a_6, a_5) \cdot S^-(a_6) = \\
&= 1 \cdot S^-(a_1) + 0 \cdot S^-(a_2) + 0 \cdot S^-(a_3) + 0 \cdot S^-(a_4) + 0 \cdot S^-(a_5) + 0 \cdot S^-(a_6) = 0.1
\end{aligned}$$

Overall, the graph-based strength of  $a_5$  is relatively high, whereas the respective weakness is very low. Then, the scores are normalized by the greatest score attained by some alternative and combined with the bonus or penalty components:

$$S^+(a_5) = b_5^+ + \frac{G^+(a_5)}{\max_{a_k \in A} G^+(a_k)} = 0.1 + \frac{0.511}{0.946} = 0.640,$$

$$S^-(a_5) = b_5^- + \frac{G^-(a_5)}{\max_{a_k \in A} G^-(a_k)} = 0.1 + \frac{0.1}{0.874} = 0.214.$$

The comprehensive quality of alternative  $a_5$  is equal to  $S(a_5) = S^+(a_5) - S^-(a_5) = 0.640 - 0.214 = 0.426$ .

When it comes to  $a_6$ , its unnormalized graph-based strength is equal to zero as it does not outrank any other option. As

a result, the strength  $S^+$  of  $a_6$  is set to the minimal possible value  $S^+(a_6) = 0.1$ . Then, its unnormalized graph-based weakness is derived from the weaknesses of  $a_2$  and  $a_5$  that outrank  $a_6$ :

$$\begin{aligned} G^-(a_6) &= \mathbb{1}(a_1, a_6) \cdot S^-(a_1) + \mathbb{1}(a_2, a_6) \cdot S^-(a_2) + \mathbb{1}(a_3, a_6) \cdot S^-(a_3) + \\ &\quad + \mathbb{1}(a_4, a_6) \cdot S^-(a_4) + \mathbb{1}(a_5, a_6) \cdot S^-(a_5) + \mathbb{1}(a_6, a_6) \cdot S^-(a_6) = \\ &= 0 \cdot S^-(a_1) + 1 \cdot S^-(a_2) + 0 \cdot S^-(a_3) + 0 \cdot S^-(a_4) + 1 \cdot S^-(a_5) + 0 \cdot S^-(a_6) = 0.345 + 0.214 = 0.560. \end{aligned}$$

The total weakness of alternative  $a_6$  is:

$$S^-(a_6) = b_6^+ + \frac{G^-(a_6)}{\max_{a_k \in A} G^-(a_k)} = 0.1 + \frac{0.560}{0.874} = 0.740,$$

and its comprehensive quality is  $S(a_6) = S^+(a_6) - S^-(a_6) = 0.1 - 0.74 = -0.64$ .

### 3.2. ScoreBin II

The second variant of ScoreBin assumes that the strength is derived from outranking many weak alternatives, and the weakness comes from being outranked by numerous strong alternatives. This requires setting the following weights:

$$\omega^+(a_k) = S^-(a_k) \text{ and } \omega^-(a_k) = S^+(a_k). \quad (12)$$

The underlying motivation is that the lack of outranking of weak options should question the strength of an alternative, and not being outranked by strong options should decrease its weakness. This idea is inspired by the HITS algorithm [27], employed for scoring websites based on hyperlinks. It uses the concepts of hubs and authorities to define the roles of websites. A good authority is linked by many good hubs, whereas a good hub links to many good authorities. Note that the original HITS method does not account for bonuses or penalties.

The results of ScoreBin II for an example problem are provided in Table 2. The unnormalized graph-based strength of  $a_5$  is:

$$\begin{aligned} G^+(a_5) &= \mathbb{1}(a_5, a_1) \cdot S^-(a_1) + \mathbb{1}(a_5, a_2) \cdot S^-(a_2) + \mathbb{1}(a_5, a_3) \cdot S^-(a_3) + \\ &\quad + \mathbb{1}(a_5, a_4) \cdot S^-(a_4) + \mathbb{1}(a_5, a_5) \cdot S^-(a_5) + \mathbb{1}(a_5, a_6) \cdot S^-(a_6) = \\ &= 0 \cdot S^-(a_1) + 1 \cdot S^-(a_2) + 0 \cdot S^-(a_3) + 1 \cdot S^-(a_4) + 0 \cdot S^-(a_5) + 1 \cdot S^-(a_6) = 0.409 + 1.1 + 0.665 = 2.174 \end{aligned}$$

Alternative  $a_5$  is the strongest because it outranks the three weakest options:  $a_4$ ,  $a_6$ , and  $a_2$ . Hence it has the highest possible strength  $S^+$ :

$$S^+(a_5) = b_5^+ + \frac{G^+(a_5)}{\max_{a_k \in A} G^+(a_k)} = 0.1 + \frac{2.174}{2.174} = 1.1.$$

The unnormalized graph-based weakness of  $a_5$  is:

$$\begin{aligned} G^-(a_5) &= \mathbb{1}(a_1, a_5) \cdot S^+(a_1) + \mathbb{1}(a_2, a_5) \cdot S^+(a_2) + \mathbb{1}(a_3, a_5) \cdot S^+(a_3) + \\ &\quad + \mathbb{1}(a_4, a_5) \cdot S^+(a_4) + \mathbb{1}(a_5, a_5) \cdot S^+(a_5) + \mathbb{1}(a_6, a_5) \cdot S^+(a_6) = \\ &= 1 \cdot S^+(a_1) + 0 \cdot S^+(a_2) + 0 \cdot S^+(a_3) + 0 \cdot S^+(a_4) + 0 \cdot S^+(a_5) + 0 \cdot S^+(a_6) = 0.941. \end{aligned}$$

Thus, its total weakness can be computed as follows:

$$S^-(a_5) = b_5^- + \frac{G^-(a_5)}{\max_{a_k \in A} G^-(a_k)} = 0.1 + \frac{0.941}{3.559} = 0.364.$$

The comprehensive quality of  $a_5$  is  $S(a_5) = S^+(a_5) - S^-(a_5) = 1.1 - 0.364 = 0.736$ . Furthermore, the strength of  $a_6$

is again  $S^+(a_6) = 0.1$  due to the lack of outranking any other option. In turn, its unnormalized graph-based weakness is:

$$\begin{aligned} G^-(a_6) &= \mathbb{1}(a_1, a_6) \cdot S^+(a_1) + \mathbb{1}(a_2, a_6) \cdot S^+(a_2) + \mathbb{1}(a_3, a_6) \cdot S^+(a_3) + \\ &\quad + \mathbb{1}(a_4, a_6) \cdot S^+(a_4) + \mathbb{1}(a_5, a_6) \cdot S^+(a_5) + \mathbb{1}(a_6, a_6) \cdot S^+(a_6) = \\ &= 0 \cdot S^+(a_1) + 1 \cdot S^+(a_2) + 0 \cdot S^+(a_3) + 0 \cdot S^+(a_4) + 1 \cdot S^+(a_5) + 0 \cdot S^+(a_6) = 0.912 + 1.100 = 2.012. \end{aligned}$$

It is the second highest weakness since  $a_6$  is outranked by the strongest alternative  $a_5$  and the third strongest option  $a_2$ . The total weakness of  $a_6$  is:

$$S^-(a_6) = b_6^- + \frac{G^-(a_6)}{\max_{a_k \in A} G^-(a_k)} = 0.1 + \frac{2.174}{3.559} = 0.665,$$

and its comprehensive quality is  $S(a_6) = S^+(a_6) - S^-(a_6) = 0.1 - 0.665 = -0.565$ .

### 3.3. ScoreBin III

The third variant of ScoreBin combines and extends the previous two by considering the difficulty and easiness of outranking alternatives. In this spirit, the strength of an alternative depends on the strengths of options outranking the same alternatives that are outranked by it. Hence an alternative is judged strong if it outranks alternatives that are also outranked by other strong alternatives. Analogously, the alternative's weakness depends on the weakness of options that are outranked by the same alternatives outranking it. Thus, an alternative is deemed weak if it is outranked by alternatives that outrank other weak options. Such effects can be accomplished using the following weights:

$$\omega^+(a_k) = \sum_{l=1}^n \mathbb{1}(a_l, a_k) \cdot S^+(a_l) \text{ and } \omega^-(a_k) = \sum_{l=1}^n \mathbb{1}(a_k, a_l) \cdot S^-(a_l). \quad (13)$$

ScoreBin III is inspired by SALSA [30], which ranks the websites by analyzing the websites it is linked to and the numbers of in- and outgoing links. In our adaptation, we consider a bipartite graph implied by the outranking relation with each alternative represented by a node in both parts of the graph. One of its parts stands for the graph-based strengths, and the other part for the graph-based weaknesses. The arc from node  $a_i$  in the strength part to node  $a_k$  in the weakness part is considered only if  $a_i S a_k$ .

The general idea underlying ScoreBin II and III is similar (e.g., an alternative is judged strong if it outranks weak options, i.e., the ones outranked by many strong alternatives). However, the two methods differ in the influence scope of indirect preference information, indicating some alternatives as strong or weak. In ScoreBin II, such information impacts both strengths and weaknesses of other alternatives. ScoreBin III is similar to ScoreBin I in limiting the impact of positive (negative) information only to the strengths (weaknesses) of alternatives.

The results of ScoreBin III for an example problem are given in Table 2. In addition, Table 3 reports weights  $\omega^+(a_k)$  and  $\omega^-(a_k)$  capturing the difficulty of outranking each  $a_k \in A$ . Hence, to increase the strength, it is most beneficial to outrank  $a_4$ ,  $a_6$ , and  $a_2$ , and to decrease the weakness significantly, one should not be outranked by  $a_5$ ,  $a_1$ , or  $a_2$ .

Table 3: Weights  $\omega^+(a_k)$  and  $\omega^-(a_k)$  for ScoreBin III and IV for the illustrative problem.

ScoreBin	Value	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$
III	$\omega^+$	0.000	1.100	0.901	3.569	0.901	2.036
	$\omega^-$	1.831	1.775	1.100	0.000	2.193	0.000
IV	$\omega^+$	-	6.073	10.000	1.562	10.000	2.663
	$\omega^-$	1.176	5.000	10.000	-	2.016	-

The unnormalized graph-based strength of  $a_5$  builds on outranking  $a_2$  (only outranked by  $a_5$ ),  $a_4$  (also outranked by  $a_1$ ,  $a_2$ , and  $a_3$ ), and  $a_6$  (also outranked by  $a_2$ ). When neglecting factors  $\mathbb{1}(a_i, a_k) = 0$ , it can be expressed as

follows:

$$\begin{aligned}
G^+(a_5) &= \mathbf{1}(a_5, a_2) \cdot \mathbf{1}(a_5, a_2) \cdot S^+(a_5) + \\
&+ \mathbf{1}(a_5, a_4) \cdot [\mathbf{1}(a_1, a_4) \cdot S^+(a_1) + \mathbf{1}(a_2, a_4) \cdot S^+(a_2) + \mathbf{1}(a_3, a_4) \cdot S^+(a_3) + \mathbf{1}(a_5, a_4) \cdot S^+(a_5)] + \\
&+ \mathbf{1}(a_5, a_6) \cdot [\mathbf{1}(a_2, a_6) \cdot S^+(a_2) + \mathbf{1}(a_5, a_6) \cdot S^+(a_5)] = \\
&= 1 \cdot S^+(a_1) + 2 \cdot S^+(a_2) + 1 \cdot S^+(a_3) + 3 \cdot S^+(a_5) = 0.901 + 2 \cdot 0.936 + 0.632 + 3 \cdot 1.100 = 6.705
\end{aligned}$$

Overall,  $a_5$  outranks alternatives which are outranked by other strong alternatives, which positively impacts its total strength:

$$S^+(a_5) = b_5^+ + \frac{G^+(a_5)}{\max_{a_k \in A} G^+(a_k)} = 0.1 + \frac{6.705}{6.705} = 1.1.$$

The unnormalized graph-based weakness of  $a_5$  derives from being outranked by  $a_1$  (which also outranks  $a_3$  and  $a_4$ ):

$$\begin{aligned}
G^-(a_5) &= \mathbf{1}(a_1, a_5) \cdot [\mathbf{1}(a_1, a_3) \cdot S^-(a_3) + \mathbf{1}(a_1, a_4) \cdot S^-(a_4) + \mathbf{1}(a_1, a_5) \cdot S^-(a_5)] = \\
&= 1 \cdot S^-(a_3) + 1 \cdot S^-(a_4) + 1 \cdot S^-(a_5) = 0.365 + 1.1 + 0.365 = 1.83.
\end{aligned}$$

Then, the total weakness of  $a_5$  is:

$$S^-(a_5) = b_5^- + \frac{G^-(a_5)}{\max_{a_k \in A} G^-(a_k)} = 0.1 + \frac{1.83}{6.899} = 0.365,$$

and its comprehensive quality amounts to  $S(a_5) = S^+(a_5) - S^-(a_5) = 1.1 - 0.365 = 0.735$ . The strength  $S^+$  of  $a_6$  is again 0.1 since  $G^+(a_6) = 0$ . Its unnormalized graph-based weaknesses derives from being outranked by  $a_2$  (that outranks also  $a_4$ ) and  $a_5$  (that outranks also  $a_2$  and  $a_4$ ):

$$\begin{aligned}
G^-(a_6) &= \mathbf{1}(a_2, a_6) \cdot [\mathbf{1}(a_2, a_6) \cdot S^-(a_6) + \mathbf{1}(a_2, a_4) \cdot S^-(a_4)] + \\
&+ \mathbf{1}(a_5, a_6) \cdot [\mathbf{1}(a_5, a_2) \cdot S^-(a_2) + \mathbf{1}(a_5, a_4) \cdot S^-(a_4) + \mathbf{1}(a_5, a_6) \cdot S^-(a_6)] = \\
&= 1 \cdot S^-(a_2) + 2 \cdot S^-(a_4) + 2 \cdot S^-(a_6) = 0.418 + 2 \cdot 1.1 + 2 \cdot 0.675 = 3.968.
\end{aligned}$$

Then, its total weakness is:

$$S^-(a_6) = b_6^- + \frac{G^-(a_6)}{\max_{a_k \in A} G^-(a_k)} = 0.1 + \frac{3.968}{6.899} = 0.675,$$

and the comprehensive quality amounts to  $S(a_6) = S^+(a_6) - S^-(a_6) = 0.1 - 0.675 = -0.575$ .

### 3.4. ScoreBin IV

The fourth variant of ScoreBin combines the first and third counterparts. An alternative is considered challenging to outrank if it is outranked by few other strong options, and it is easy to outrank when numerous weak options outrank it. The more challenging an alternative is to outrank, the more favorable it is for the quality of the alternative that does outrank it. The above idea is implemented using the following weights:

$$\omega^+(a_k) = \frac{1}{\sum_{l=1}^n \mathbf{1}(a_l, a_k) \cdot S^-(a_l)} \text{ and } \omega^-(a_k) = \frac{1}{\sum_{l=1}^n \mathbf{1}(a_k, a_l) \cdot S^+(a_l)}. \quad (14)$$

Unlike in ScoreBin I, the fourth variant lets additional preferences influence both strengths and weaknesses of other alternatives. For example, if  $a_k$  was judged weak, then  $a_i$  outranked by  $a_k$  is considered as easier to outrank. Hence all alternatives which also outrank  $a_i$  would have lower strengths.

The results for the illustrative problem are provided in Table 2, and the respective weights  $\omega^+$  and  $\omega^-$  are given in



Table 3. Some of them (e.g.,  $\omega^+(a_1)$ ) are undefined as no alternative outranks or is outranked by a given alternative. It is not an issue as this value is not required in the calculation of the strength or weakness of any alternative. For example, the difficulty of outranking, captured by  $\omega^+$ , is the greatest for  $a_3$  and  $a_5$ . In particular,  $a_5$  is outranked only by  $a_1$  which has the lowest weakness:

$$\omega^+(a_5) = \frac{1}{\mathbb{1}(a_1, a_5) \cdot S^-(a_1)} = \frac{1}{0.1} = 10.$$

The unnormalized graph-based strength of  $a_5$  is the second greatest among all alternatives mainly due to outranking  $a_2$ , which is relatively difficult to outrank:

$$G^+(a_5) = \mathbb{1}(a_5, a_2) \cdot \omega^+(a_2) + \mathbb{1}(a_5, a_4) \cdot \omega^+(a_4) + \mathbb{1}(a_5, a_6) \cdot \omega^+(a_6) = 6.073 + 1.562 + 2.663 = 10.298.$$

In turn, the contribution of outranking  $a_4$  to  $G^+(a_5)$  is rather low because  $a_4$  is outranked by many other options, which supports the easiness of being at least as good as  $a_4$ . The total strength of alternative  $a_5$  is:

$$S^+(a_5) = b_5^+ + \frac{G^+(a_5)}{\max_{a_k \in A} G^+(a_k)} = 0.1 + \frac{10.298}{21.562} = 0.578.$$

Then, the weight  $\omega^-$  of  $a_5$  equals:

$$\omega^-(a_5) = \frac{1}{\mathbb{1}(a_5, a_2) \cdot S^+(a_2) + \mathbb{1}(a_5, a_4) \cdot S^+(a_4) + \mathbb{1}(a_5, a_6) \cdot S^+(a_6)} = \frac{1}{0.296 + 0.1 + 0.1} = 2.016.$$

It is rather low, which speaks in favor of the easiness in outranking  $a_5$ . The unnormalized graph-based weakness of  $a_5$  is:

$$G^-(a_5) = \mathbb{1}(a_1, a_5) \cdot \omega^-(a_1) = 1.176,$$

and its total weakness can be computed as follows:

$$S^-(a_5) = b_5^- + \frac{G^-(a_5)}{\max_{a_k \in A} G^-(a_k)} = 0.1 + \frac{1.176}{18.193} = 0.165.$$

It is the second least weakness among all alternatives because  $a_5$  is outranked only by  $a_1$  whose  $\omega^-$  is the least. The comprehensive quality of  $a_5$  is  $S(a_5) = S^+(a_5) - S^-(a_5) = 0.578 - 0.165 = 0.413$ .

When it comes to  $a_6$ , the difficulty  $\omega^+$  in outranking it is:

$$\omega^+(a_6) = \frac{1}{\mathbb{1}(a_2, a_6) \cdot S^-(a_2) + \mathbb{1}(a_5, a_6) \cdot S^-(a_5)} = \frac{1}{0.211 + 0.165} = 2.663.$$

It is rather low because  $a_6$  is outranked by two other alternatives, including  $a_2$ , which has the third greatest weakness. The graph-based weakness of  $a_6$  is zero, and hence  $S^+(a_6) = 0.1$ . Weight  $\omega^-$  for  $a_6$  is undefined because it does not outrank any other option. Further, its unnormalized graph-based weakness is:

$$G^-(a_6) = \mathbb{1}(a_2, a_6) \cdot \omega^-(a_2) + \mathbb{1}(a_5, a_6) \cdot \omega^-(a_5) = 5 + 2.016 = 7.016.$$

It is the second greatest weakness among all alternatives. This is mainly due to being outranked by  $a_2$ , which has the second greatest  $\omega^-$  weight. The total weakness of  $a_6$  is:

$$S^-(a_6) = b_6^- + \frac{G^-(a_6)}{\max_{a_k \in A} G^-(a_k)} = 0.1 + \frac{7.016}{18.193} = 0.486,$$

and its comprehensive quality amounts to  $S(a_6) = S^+(a_6) - S^-(a_6) = 0.1 - 0.486 = 0.386$ .

### 3.5. Ranking construction

The strengths, weaknesses, and qualities can be used to impose an order on the set of alternatives. If a partial ranking is desired, one should intersect the two complete orders implied by the separate application of strengths and weaknesses, i.e. [4, 7]:

$$\begin{aligned}
 a_i P a_k \text{ (} a_i \text{ is preferred to } a_k \text{)} & \quad \text{iff } (S^+(a_i) > S^+(a_k) \text{ and } S^-(a_i) < S^-(a_k)) \\
 & \quad \text{or } (S^+(a_i) > S^+(a_k) \text{ and } S^-(a_i) = S^-(a_k)) \\
 & \quad \text{or } (S^+(a_i) = S^+(a_k) \text{ and } S^-(a_i) < S^-(a_k)); \\
 a_i I a_k \text{ (} a_i \text{ is indifference with } a_k \text{)} & \quad \text{iff } (S^+(a_i) = S^+(a_k) \text{ and } S^-(a_i) = S^-(a_k)); \\
 a_i R a_k \text{ (} a_i \text{ is incomparable with } a_k \text{)} & \quad \text{otherwise.}
 \end{aligned}$$

Thus rankings obtained for the illustrative problem with the ScoreBin variants, QD, and NFS are presented in Figure 1. In particular, they are the same for QD and NFS, not involving any incomparability. The same orders have also been obtained for ScoreBin I and IV, where  $a_2$  and  $a_3$  are incomparable. For ScoreBin II and III,  $a_2$  is preferred to  $a_3$ . The variants of ScoreBin also differ in the position of  $a_1$ . For the first and fourth variants,  $a_1$  is preferred to all remaining alternatives; for the second variant,  $a_1$  is incomparable with  $a_5$ , while in ScoreBin III,  $a_1$  is additionally incomparable with  $a_2$ .

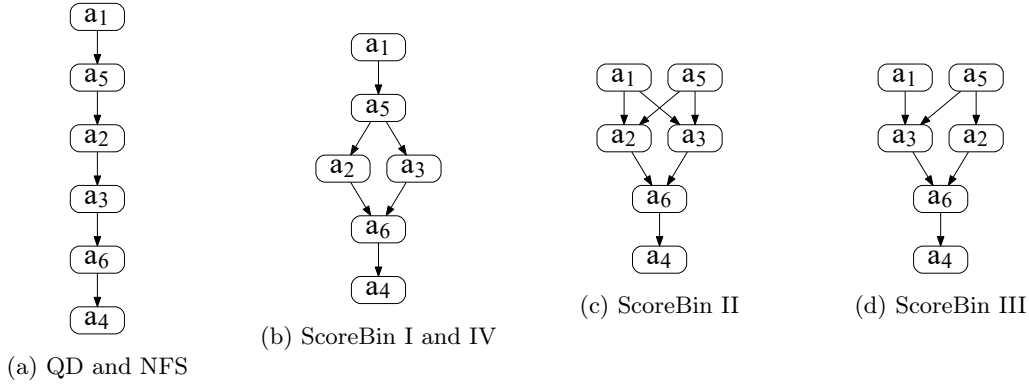


Figure 1: Partial rankings based on the strengths and weaknesses obtained with different methods for the illustrative problem.

When a complete ranking is desired, one should refer to the qualities of alternatives in the following way [6]:

$$\begin{aligned}
 a_i P a_k \text{ (} a_i \text{ is preferred to } a_k \text{)} & \quad \text{iff } S(a_i) > S(a_k); \\
 a_i I a_k \text{ (} a_i \text{ is indifferent with } a_k \text{)} & \quad \text{iff } S(a_i) = S(a_k).
 \end{aligned}$$

For the illustrative example, the complete rankings for all methods but ScoreBin I are the same:  $a_1 P a_5 P a_2 P a_3 P a_6 P a_4$ . For the first variant of ScoreBin, the ranks of  $a_2$  and  $a_3$  are inverse.

## 4. Measures used for comparing the choice or ranking recommendations

This section describes measures for comparing the rankings obtained with a pair of methods  $M'$  and  $M''$ . They refer to function  $rank(M, a_i)$  which determines the position of alternative  $a_i$  assigned to it by method  $M \in \{M', M''\}$  [24]:

$$rank(M, a_i) = 1 + |\{a_j \in A \setminus a_i : a_j P^M a_i\}|. \quad (15)$$

A subset of alternatives ranked  $r$ -th by  $M$  is defined as follows:

$$M(r) = \{a_i \in A : rank(M, a_i) = r\}. \quad (16)$$

To compare the subsets ranked  $r$ -th by methods  $M'$  and  $M''$ , we use the Rank Agreement measure:

$$RA(M', M'', r) = \frac{|M'(r) \cap M''(r)|}{|M'(r) \cup M''(r)|}. \quad (17)$$

Its particular case, called the Normalized Hit Ratio (NHR), is useful for quantifying the similarity between the choice recommendations, i.e., the subsets of alternatives ranked at the top [24]:

$$NHR(M', M'') = RA(M', M'', 1) = \frac{|M'(1) \cap M''(1)|}{|M'(1) \cup M''(1)|}. \quad (18)$$

It can also be used to compare a subset of top-ranked alternatives by method  $M$  with the most preferred subset of alternatives corresponding to, e.g., graph kernel  $K$ , i.e.:

$$NHR(M, K) = \frac{|M(1) \cap K|}{|M(1) \cup K|}. \quad (19)$$

In the same context, we can compute the average position in the ranking established with  $M$  of alternatives contained in the most preferred subset  $K$ :

$$AP(M, K) = \frac{\sum_{a_i \in K} \text{rank}(M, a_i)}{|K|}. \quad (20)$$

The remaining two measures serve for comparing a pair of rankings and build on the notion of distances between relations observed for the same pairs (see Table 4) [33, 41]. The Normalized Ranking Distance (NRD) is applicable when the two rankings are incomplete [24]:

$$NRD(M', M'') = \frac{\sum_{i=1}^n \sum_{k=1, k \neq i}^n RD(M', M'', a_i, a_k)}{2n \cdot (n-1)}. \quad (21)$$

It takes value in the  $[0, 1]$  interval, where value 0 means the two rankings are the same. When complete rankings need to be compared, we apply Kendall's  $\tau$  that considers only preference and indifference [26]:

$$\tau(M', M'') = 1 - 2 \cdot NRD(M', M''). \quad (22)$$

It takes values in the  $[-1, 1]$  interval, where 1 means the two rankings are the same.

Table 4: The distances  $RD(M', M'', a_i, a_k)$  between relations observed for pair  $(a_i, a_k)$  in the rankings determined by methods  $M'$  and  $M''$ .

$RD(M', M'', a_i, a_k)$	$a_i P^{M''} a_k$	$a_i I^{M''} a_k$	$a_i R^{M''} a_k$	$a_i P^{-, M''} a_k$
$a_i P^{M'} a_k$	0	2	3	4
$a_i I^{M'} a_k$	2	0	2	2
$a_i R^{M'} a_k$	3	2	0	3
$a_i P^{-, M'} a_k$	4	2	3	0

## 5. Evaluation of technological parks in Poland

This section reports the results of a case study concerning an assessment of technological parks in Poland [28]. The parks have been created to offer favorable conditions for the development of innovative businesses, particularly in the high-tech sector, by providing access to modern infrastructure, scientific expertise, and financial resources. Technological parks typically provide a range of services to tenant companies, such as research and development facilities, technology transfer centers, consulting services, and training programs. Over the years, they have successfully attracted both domestic and foreign investors and contributed significantly to the growth of Poland's innovation ecosystem. We aim to rank eleven technological parks in Poland. They are evaluated in terms of the following seven criteria (see Table 5):

- *Sales costs* [mln PLN] ( $g_1$ ; to be minimized): total incurred costs for sales of products and services.
- *Park buildings' surface* ( $g_2$ ; to be minimized).
- *Park's localization* ( $g_3$ ; to be minimized): a distance from the communication railroads, roads, airports, universities, and industrial plants.
- *Total sales* [mln PLN] ( $g_4$ ; to be maximized): profit generated by the park.
- *Number of services types* ( $g_5$ ; to be maximized) offered by the park.
- *Overall evaluation of park's management* ( $g_6$ ; to be maximized), as expressed by the tenants.
- *Number of completed projects* ( $g_7$ ; to be maximized) realized by the park in partnership with other institutions.

Table 5: The performance matrix for the problem of ranking technological parks in Poland.

Park	Sales costs [mln PLN]	Buildings' surface	Park's localization	Total sales [mln PLN]	Number of services	Evaluation of park's management	Completed projects
$a_1$	3.873125	2259.40	17	1.410718	11.07	4.23	2
$a_2$	0.772961	3536.74	24	1.104072	15.12	4.24	2
$a_3$	1.300742	3428.92	23	1.855023	15.00	4.64	2
$a_4$	2.683445	3163.00	25	4.617871	13.11	4.31	2
$a_5$	2.340402	5980.50	24	0.704410	18.00	4.21	3
$a_6$	1.848784	2853.92	22	3.343191	2.94	3.58	3
$a_7$	2.474156	2161.63	19	0.290557	3.99	3.68	1
$a_8$	2.626398	8100.00	23	33.823535	12.96	3.67	18
$a_9$	2.349051	13203.31	21	2.271590	4.18	4.41	1
$a_{10}$	2.448512	7396.00	21	2.326377	14.04	4.34	4
$a_{11}$	8.679106	21682.09	25	9.798043	9.70	3.14	4

The criteria weights were derived using the Simos-Roy-Figueira (SRF) method [13, 10] that is often coupled with outranking methods [43]. It requires ranking the criteria from the least to the most important. In addition, it is possible to insert blank cards between groups of criteria judged indifferent to increase the preference intensity. This led to the following raw ranks assigned to particular criteria (the lower the rank, the less important the criterion):  $g_7 - 1$ ,  $g_6 - 2$ ,  $g_2 - 4$ ,  $g_3 - 6$ ,  $g_5 - 6$ ,  $g_1 - 9$ , and  $g_4 - 10$ . Hence, e.g.,  $g_3$  and  $g_5$  are judged indifferent, whereas the desired difference between the weights assigned to  $g_1$  and  $g_5$  is three times greater than between  $g_6$  and  $g_7$ . Moreover, coefficient  $Z = 7$  specified the ratio between weights assigned to the most and the least preferred groups.

The obtained criteria weights and three types of thresholds needed to construct an outranking relation are given in Table 6. We set the credibility threshold to  $\lambda = 0.74$ . It corresponds to the sum of weights of criteria contained in the following subset  $\{g_1, g_2, g_4, g_5\}$  that has been judged sufficient for validating the outranking. The outranking function  $\mathbb{1}(a_i, a_k)$  is provided in Table 7. The respective outranking graph is given in Figure 2a). We exploit it using ELECTRE I, QD, NFS, and the four variants of ScoreBin. The value of the base bonus and penalty for ScoreBin is set to  $\alpha = 0.1$ .

### 5.1. Scenario without indirect preference information

Let us first discuss the results obtained without any additional preferences of the DM. A graph kernel is composed of five alternatives:  $K = \{a_8, a_{10}, a_5, a_4, a_7\}$ . In particular, it includes two options,  $a_8$  and  $a_{10}$ , that are not outranked by any other alternative, and three other options that allow ensuring internal and external stabilities. The strengths, weaknesses, and comprehensive qualities obtained with NFS and four ScoreBin methods are shown in Table 8. For QD, we report the ranks in the ascending and descending distillations. The respective rankings are presented in Figures 2 and 3.

The most preferred alternative in the rankings obtained with all methods is  $a_{10}$ . Consequently, NHR between all rankings equals one. Their comparison to the kernel leads to  $NHR$  equal to  $1/5$  as  $a_{10}$  whose selection they recommend is one of the five alternatives indicated by ELECTRE I. The favorable evaluation of  $a_{10}$  derives from its high strength implied by outranking four other alternatives, including a relatively strong  $a_2$  and three weaker alternatives  $a_2$ ,  $a_9$ ,

Table 6: Criteria weights and comparison thresholds for the problem of ranking technological parks in Poland.

$g_j$	Indifference threshold $q_j$	Preference threshold $p_j$	Veto threshold $v_j$	Weight $w_j$
$g_1$	0,5	3	25	0.22
$g_2$	200	1000	inf	0.11
$g_3$	0	1	inf	0.16
$g_4$	0,25	1	10	0.25
$g_5$	1	4	10	0.16
$g_6$	0	0	inf	0.06
$g_7$	0	1	2	0.04

Table 7: Outranking function  $\mathbb{1}(a_i, a_k)$  for the problem of ranking technological parks in Poland.

$\mathbb{1}(a_i, a_k)$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$	$a_{11}$
$a_1$	0	0	0	0	0	0	1	0	0	0	0
$a_2$	0	0	0	0	1	0	0	0	0	0	0
$a_3$	0	1	0	1	1	0	0	0	1	0	0
$a_4$	1	0	0	0	0	1	0	0	1	0	0
$a_5$	0	0	0	0	0	0	0	0	0	0	0
$a_6$	0	0	0	0	0	0	0	0	1	0	0
$a_7$	0	0	0	0	0	0	0	0	0	0	0
$a_8$	0	0	0	0	0	0	0	0	0	0	1
$a_9$	0	0	0	0	0	0	0	0	0	0	0
$a_{10}$	0	1	1	0	0	0	0	0	1	0	1
$a_{11}$	0	0	0	0	0	0	0	0	0	0	0

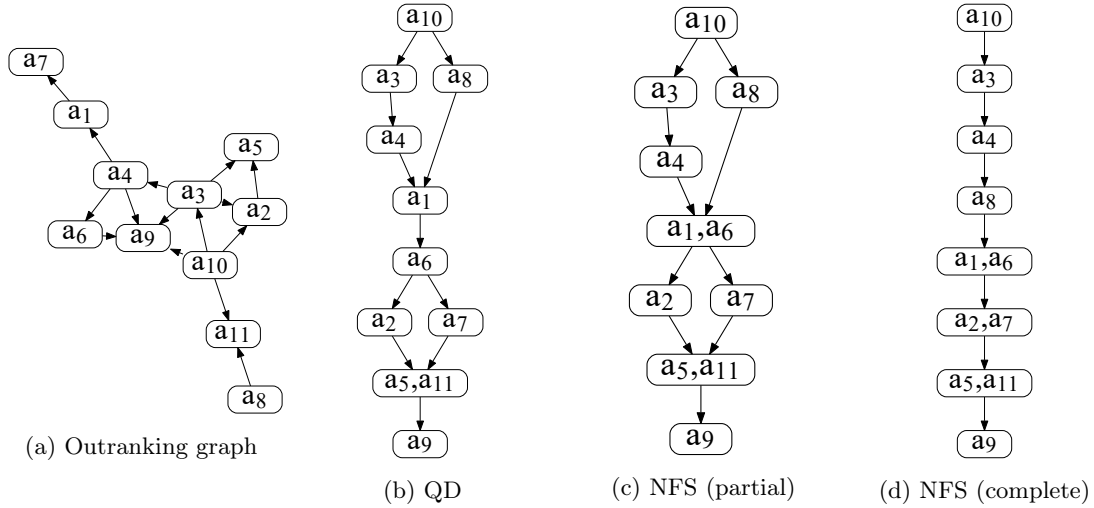


Figure 2: Outranking graph (a) and rankings obtained with QD (b) and NFS – partial (c) and complete (d) for the problem of assessing technological parks in Poland.

and  $a_{11}$ . Also, its weakness is minimal for all methods because  $a_{10}$  is not outranked by any other option. On the contrary, the least preferred alternative in all rankings is  $a_9$ . Such a poor assessment derives from its minimal strength implied by the lack of outranking any other alternative. Moreover, its weakness is very high because it is outranked by the three strongest alternatives ( $a_{10}$ ,  $a_3$ , and  $a_4$ ) and a moderately ranked option  $a_6$ .

However, the rankings exhibit some differences in their intermediate parts. We explain them by referring to two example alternatives,  $a_8$  and  $a_2$ . Alternative  $a_8$  outranks only a single option while not being outranked by any other option. Its minimal weakness puts it high in all partial rankings due to its most favorable position in the order implied by the alternatives' weak points. However, its position in the complete rankings is lower, i.e., fourth or fifth. This is because its strength is relatively low, being derived only from outranking  $a_{11}$ . Since the strength of  $a_{11}$  is very low for ScoreBin I, the strength of  $a_8$  is the lowest. It is slightly higher for ScoreBin II because the weakness of  $a_{11}$  is intermediate. Then, the strength of  $a_8$  in ScoreBin III is even greater because  $a_{11}$  is also outranked by a very strong option  $a_{10}$ . Finally, ScoreBin IV assigns a higher strength to  $a_8$  because outranking  $a_{11}$  proves beneficial as it is outranked only by the two options with the least weakness ( $a_8$  and  $a_{10}$ ).

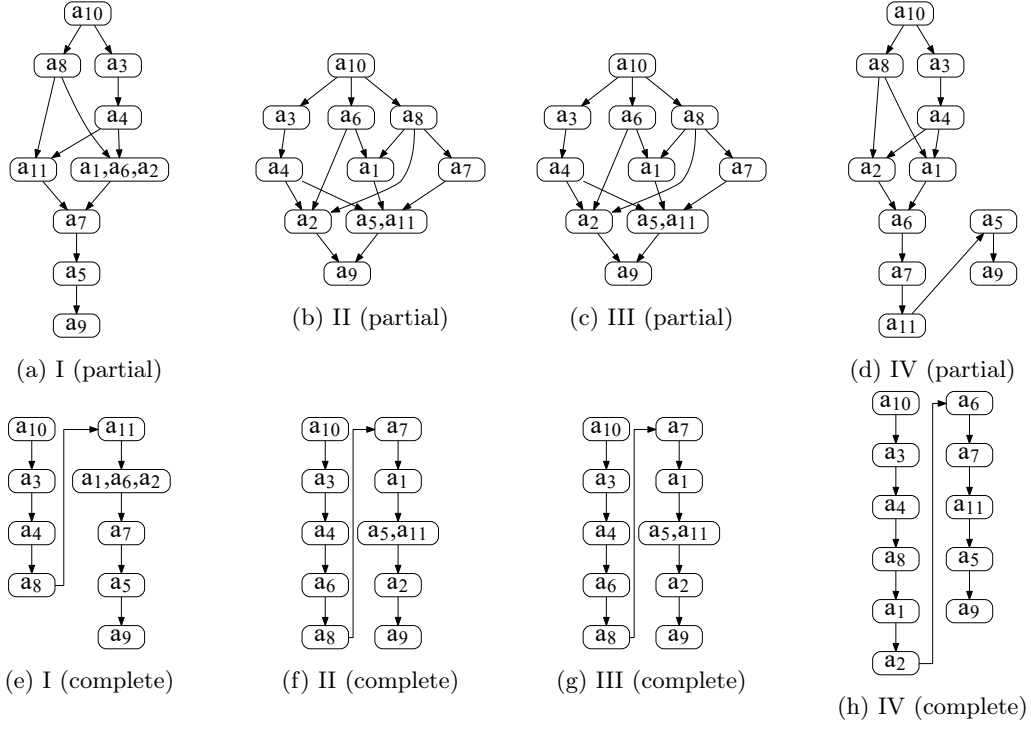


Figure 3: Incomplete (a – d) and complete (e – h) rankings derived with the four variants of ScoreBin for the problem of assessing technological parks in Poland.

As far as  $a_2$  is concerned, it is outranked by  $a_3$  and  $a_{10}$ , and it outranks  $a_5$ . This makes it ranked in the lower half for most methods mainly due to its relatively high weakness. It is exceptionally high in ScoreBin II and III. For the former approach, it is the consequence of being outranked by the two strongest alternatives. In turn, for ScoreBin III, the high weakness of  $a_2$  derives from the high weakness of other alternatives outranked by  $a_3$  and  $a_{10}$ . In ScoreBin I, the weakness of  $a_2$  is intermediate because even if two other options outrank it, they have very low weaknesses. This is even more evident for ScoreBin IV because  $a_2$  is outranked by the extremely strong alternatives, further decreasing its weakness.

Table 8: Results of six ranking methods for the problem of assessing technological parks in Poland.

Method	Value	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$	$a_{11}$
ScoreBin I	$S^+$	0.183 (4)	0.183 (4)	0.822 (2)	0.487 (3)	0.100 (8)	0.183 (4)	0.100 (8)	0.183 (4)	0.100 (8)	1.100 (1)	0.100 (8)
	$S^-$	0.400 (6)	0.400 (6)	0.200 (3)	0.300 (4)	0.700 (10)	0.400 (6)	0.500 (9)	0.100 (1)	1.100 (11)	0.100 (1)	0.300 (4)
	$S$	-0.217 (6)	-0.217 (6)	0.622 (2)	0.187 (3)	-0.600 (10)	-0.217 (6)	-0.400 (9)	0.083 (4)	-1.000 (11)	1.000 (1)	-0.200 (5)
ScoreBin II	$S^+$	0.152 (7)	0.282 (5)	1.100 (1)	0.722 (3)	0.100 (8)	0.497 (4)	0.100 (8)	0.282 (5)	0.100 (8)	1.100 (1)	0.100 (8)
	$S^-$	0.311 (4)	0.743 (10)	0.422 (6)	0.422 (6)	0.504 (8)	0.311 (4)	0.145 (3)	0.100 (1)	1.100 (11)	0.100 (1)	0.504 (8)
	$S$	-0.159 (7)	-0.461 (10)	0.678 (2)	0.300 (3)	-0.404 (8)	0.186 (4)	-0.044 (6)	0.182 (5)	-1.000 (11)	1.000 (1)	-0.404 (8)
ScoreBin III	$S^+$	0.114 (7)	0.269 (5)	1.100 (1)	0.694 (3)	0.100 (8)	0.523 (4)	0.100 (8)	0.269 (5)	0.100 (8)	1.100 (1)	0.100 (8)
	$S^-$	0.304 (4)	0.765 (10)	0.433 (6)	0.433 (6)	0.491 (8)	0.304 (4)	0.114 (3)	0.100 (1)	1.100 (11)	0.100 (1)	0.491 (8)
	$S$	-0.190 (7)	-0.496 (10)	0.667 (2)	0.262 (3)	-0.391 (8)	0.219 (4)	-0.013 (6)	0.169 (5)	-1.000 (11)	1.000 (1)	-0.391 (8)
ScoreBin IV	$S^+$	0.317 (5)	0.228 (6)	0.795 (2)	0.782 (3)	0.100 (8)	0.176 (7)	0.100 (8)	0.346 (4)	0.100 (8)	1.100 (1)	0.100 (8)
	$S^-$	0.227 (6)	0.223 (5)	0.161 (3)	0.162 (4)	0.912 (10)	0.227 (6)	0.850 (8)	0.100 (1)	1.100 (11)	0.100 (1)	0.911 (9)
	$S$	0.090 (5)	0.004 (6)	0.634 (2)	0.620 (3)	-0.812 (10)	-0.051 (7)	-0.750 (8)	0.245 (4)	-1.000 (11)	1.000 (1)	-0.811 (9)
NFS	$S^+$	1.000 (4)	1.000 (4)	4.000 (1)	3.000 (3)	0.000 (8)	1.000 (4)	0.000 (8)	1.000 (4)	0.000 (8)	4.000 (1)	0.000 (8)
	$S^-$	1.000 (3)	2.000 (8)	1.000 (3)	1.000 (3)	2.000 (8)	1.000 (3)	1.000 (3)	0.000 (1)	4.000 (11)	0.000 (1)	2.000 (8)
	$S$	0.000 (5)	-1.000 (7)	3.000 (2)	2.000 (3)	-2.000 (9)	0.000 (5)	-1.000 (7)	1.000 (4)	-4.000 (11)	4.000 (1)	-2.000 (9)
QD	Desc.	(4)	(4)	(2)	(3)	(8)	(4)	(8)	(4)	(8)	(1)	(8)
	Asc.	(5)	(8)	(3)	(4)	(9)	(6)	(6)	(1)	(11)	(1)	(9)

Table 9 summarizes the  $NRD$  values for the partial rankings constructed by the six methods. For ScoreBin II and III, the rankings are the same, resulting in  $NRD = 0$ . They are the least similar to the partial preorder obtained with ScoreBin I ( $NRD = 0.232$ ). The first variant of ScoreBin involves fewer incomparabilities while giving relatively higher priority to  $a_2$  and  $a_{11}$ , and relatively lower positions to  $a_5$ ,  $a_6$ , and  $a_7$ . Moreover, the partial rankings for ScoreBin IV and  $QD$  are profoundly matching ( $NRD = 0.055$ ), varying only in terms of relations established for the following pairs:  $(a_2, a_6)$ ,  $(a_2, a_7)$ , and  $(a_5, a_{11})$ . Finally, the incomplete rankings for  $QD$  and  $NFS$  are very similar, differing only for the relation assigned to the pair  $(a_2, a_6)$ .

Kendall's  $\tau$  values for the complete rankings constructed by  $NFS$  and four ScoreBin variants are given in Table 10.

Table 9: Normalized Ranking Distances between the partial rankings for the problem of assessing technological parks in Poland.

Method	ScoreBin I	ScoreBin II	ScoreBin III	ScoreBin IV	NFS	QD
ScoreBin I	0.000	0.232	0.232	0.086	0.100	0.109
ScoreBin II	0.232	0.000	0.000	0.209	0.173	0.182
ScoreBin III	0.232	0.000	0.000	0.209	0.173	0.182
ScoreBin IV	0.086	0.209	0.209	0.000	0.064	0.055
NFS	0.100	0.173	0.173	0.064	0.000	0.009
QD	0.109	0.182	0.182	0.055	0.009	0.000

Again, these rankings are the same for ScoreBin II and III. Then, the greatest similarity can be observed for rankings obtained with ScoreBin IV and *NFS* ( $\tau = 0.909$ ) that differ in relations assigned to the following pairs:  $(a_1, a_6)$ ,  $(a_2, a_6)$ ,  $(a_2, a_7)$ , and  $(a_5, a_{11})$ . The most distinctive ranking among the four variants was obtained with ScoreBin I. Its similarity to the rankings of ScoreBin II and III is 0.673, indicating differences in the relations observed for many pairs involving mainly the alternatives with the intermediate ranks such as  $a_8$ ,  $a_1$ ,  $a_{11}$ , and  $a_5$ .

Table 10: Kendall's  $\tau$  based on the complete rankings obtained for the problem of assessing technological parks in Poland.

Method	ScoreBin I	ScoreBin II	ScoreBin III	ScoreBin IV	NFS
ScoreBin I	1.000	0.673	0.673	0.800	0.782
ScoreBin II	0.673	1.000	1.000	0.727	0.818
ScoreBin III	0.673	1.000	1.000	0.727	0.818
ScoreBin IV	0.800	0.727	0.727	1.000	0.909
NFS	0.782	0.818	0.818	0.909	1.000

Finally, let us consider an average position in the complete rankings of the five alternatives contained in the kernel. For example, the ranks of these options according to ScoreBin II and III are:  $a_{10} - 1$ ,  $a_4 - 3$ ,  $a_8 - 5$ ,  $a_7 - 6$ , and  $a_5 - 8$ . Their average rank is  $AP = 23/5 = 4.6$ . It is the best of all methods. The highest value of  $AP$  is observed for ScoreBin I mainly because of very low positions of  $a_7$  (9) and  $a_5$  (10). In general, such relatively low  $AP$  values confirm the disadvantage of using the graph kernel for indicating the most preferred subset of alternatives under some decision scenarios. In this case, the kernel contains many alternatives. Even if it includes very strong alternatives, some others – being incomparable with the strong ones – are weaker, offering arguments for eliminating the remaining options via outranking them.

Table 11: Average position  $AP$  of alternatives contained in the outranking graph kernel based on the complete rankings obtained for the problem of assessing technological parks in Poland.

Method	ScoreBin I	ScoreBin II	ScoreBin III	ScoreBin IV	NFS
$AP$	5.4	4.6	4.6	5.2	4.8

### 5.2. Scenario with indirect preference information

In this section, we consider the problem of ranking technological parks in Poland with the same crisp outranking relation. However, we account for additional indirect preference information. We assume that the DM indicated  $a_4$  as a strong alternative ( $A_{strong}^* = \{a_4\}$ ), mainly due to being at least as good as three other options. In addition,  $a_1$  and  $a_2$  were pointed as weak alternatives ( $A_{weak}^* = \{a_1, a_2\}$ ). For the former, this is due to being outranked by  $a_4$ , while for the latter, the claim is motivated by being outranked by  $a_{10}$  and  $a_2$ . We set the value of an enhancement bonus/penalty to  $\beta = 0.8$ . We discuss only the results for the four variants of ScoreBin. The scores are provided in Table 12, and the respective partial and complete are shown in Figure 4.

For ScoreBin I, accounting for additional preferences does not change the upper part of the ranking, with  $a_{10}$  still being the most preferred alternative, followed by  $a_3$ ,  $a_8$ , and  $a_4$ . In fact, indicating  $a_4$  as a strong option also increased the strengths of  $a_3$  (directly) and  $a_{10}$  (indirectly). In turn, the negative information increases the impact of the two alternatives judged as weak ( $a_1$  and  $a_2$ ) and those which are outranked by them ( $a_5$  and  $a_7$ ). Consequently, these four options are now all ranked at the very bottom or in the lower ranking half.

When considering the rankings implied by ScoreBin II,  $a_3$  and  $a_{10}$  still have the highest graph-based strengths equal to one. However, the bonus obtained by  $a_4$  impacted its total strength, which appeared to be greater than for  $a_3$

and  $a_{10}$ . This bonus also implied the increase of weaknesses for  $a_1$ ,  $a_6$ , and  $a_9$  (e.g.,  $a_3$  became preferred to  $a_6$  in the partial ranking, whereas these two alternatives were incomparable when no additional preferences were considered). Indicating  $a_1$  and  $a_2$  as weak options deteriorated their weaknesses. However, it did not significantly impact the weakness of alternatives outranked by them, as observed for ScoreBin I.

In ScoreBin III, additional reinforcement of  $a_4$  increased its strength to the greatest value among alternatives and, thus, to being ranked at the top in the complete ranking. This statement also indirectly increased the strengths of  $a_6$ ,  $a_3$ , and  $a_{10}$  as, similarly to  $a_4$ , they outrank  $a_9$ . However, since  $a_3$  and  $a_{10}$  have the greatest graph-based strength, they do not take much advantage of this relation. Pointing out  $a_1$  as weak led to increasing its weakness over the level of  $a_3$ ,  $a_4$ ,  $a_5$ , and  $a_{11}$ . As a result,  $a_3$  and  $a_4$  are preferred to  $a_1$  in both rankings, whereas  $a_5$  and  $a_{11}$  prove to be better than  $a_1$  in the complete order and incomparable with it in the partial ranking. In addition, this statement indirectly increased the weakness  $a_6$  because it is outranked by  $a_4$ , which also outranks  $a_1$ .

The impact of additional preferences in ScoreBin IV is most visible for the alternatives judged by the DM. Judging  $a_4$  as strong impacts the quality of alternatives that are outranked by the same alternatives as  $a_4$ . In particular, it led to a decrease in the weaknesses of  $a_4$  from 0.162 to 0.137 and  $a_5$  from 0.912 to 0.860. Overall, this guaranteed  $a_4$  the highest place in the complete ranking and being incomparable with  $a_{10}$  at the top of the partial ranking. Declaring  $a_1$  as a poor option led to the increase of its weakness from 0.226 to 0.970 and indirectly to the rise of the weakness of  $a_6$ . The weakness of  $a_9$  was not affected greatly because its graph-based weakness was already the highest (i.e., equal to one). Moreover, claiming that  $a_2$  is poor decreased the difficulty of outranking  $a_5$  and then decreased the strengths of  $a_2$  and  $a_3$ . Overall, both  $a_1$  and  $a_2$  deteriorated their positions compared to the scenario without indirect preferences.

Table 12: Results of the four variants of ScoreBin for the problem of assessing technological parks in Poland when accounting for indirect preference information.

ScoreBin	Value	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$	$a_{11}$
I	$S^+$	0.168 (4)	0.168 (4)	1.099 (2)	1.097 (3)	0.100 (8)	0.168 (4)	0.100 (8)	0.168 (4)	0.100 (8)	1.100 (1)	0.100 (8)
	$S^-$	1.006 (9)	1.032 (10)	0.182 (3)	0.250 (4)	1.100 (11)	0.306 (6)	0.928 (8)	0.100 (1)	0.790 (7)	0.100 (1)	0.265 (5)
	$S$	-0.838 (9)	-0.864 (10)	0.917 (2)	0.847 (3)	-1.000 (11)	-0.138 (5)	-0.828 (8)	0.068 (4)	-0.690 (7)	1.000 (1)	-0.165 (6)
II	$S^+$	0.142 (7)	0.228 (5)	1.100 (2)	1.675 (1)	0.100 (8)	0.447 (4)	0.100 (8)	0.228 (5)	0.100 (8)	1.100 (2)	0.100 (8)
	$S^-$	1.188 (10)	1.309 (11)	0.355 (4)	0.355 (4)	0.407 (6)	0.488 (8)	0.133 (3)	0.100 (1)	1.100 (9)	0.100 (1)	0.407 (6)
	$S$	-1.046 (10)	-1.081 (11)	0.745 (3)	1.320 (1)	-0.307 (7)	-0.041 (6)	-0.033 (5)	0.128 (4)	-1.000 (9)	1.000 (2)	-0.307 (7)
III	$S^+$	0.112 (7)	0.248 (5)	1.100 (2)	1.652 (1)	0.100 (8)	0.589 (4)	0.100 (8)	0.248 (5)	0.100 (8)	1.100 (2)	0.100 (8)
	$S^-$	1.035 (9)	1.460 (11)	0.430 (5)	0.430 (5)	0.475 (7)	0.335 (4)	0.111 (3)	0.100 (1)	1.100 (10)	0.100 (1)	0.475 (7)
	$S$	-0.923 (9)	-1.212 (11)	0.670 (3)	1.222 (1)	-0.375 (7)	0.254 (4)	-0.011 (6)	0.148 (5)	-1.000 (10)	1.000 (2)	-0.375 (7)
IV	$S^+$	0.151 (6)	0.146 (7)	0.695 (3)	1.595 (1)	0.100 (8)	0.173 (5)	0.100 (8)	0.348 (4)	0.100 (8)	1.100 (2)	0.100 (8)
	$S^-$	0.970 (10)	0.907 (9)	0.169 (4)	0.137 (3)	0.860 (7)	0.270 (5)	0.823 (6)	0.100 (1)	1.100 (11)	0.100 (1)	0.892 (8)
	$S$	-0.819 (10)	-0.761 (8)	0.526 (3)	1.458 (1)	-0.760 (7)	-0.097 (5)	-0.723 (6)	0.248 (4)	-1.000 (11)	1.000 (2)	-0.792 (9)

Let us comment only on the pairs of the most and the least similar recommendations obtained with different variants of ScoreBin. When it comes to NRD, the greatest similarity (0.068) between the partial ranking is noted for ScoreBin II and III, while the least consistency is observed for ScoreBin I and III. Then comparing the complete rankings in terms of Kendall's  $\tau$ , the pair of methods leading to the most similar orders (0.891) is the same as for NRD. In turn, the least similar rankings (0.564) were generated by ScoreBin I and IV. Finally, NHR is equal to one for the following three pairs of ScoreBin variants: (II, III), (II, IV), and (III, IV), as they indicate both  $a_4$  and  $a_{10}$  at the top. In turn, comparing the top-ranked alternative ( $a_{10}$ ) by ScoreBin I with the choice-oriented recommendations of other variants leads to NHR equal to 0.5. The differences are generally more significant than for the recommendations obtained in the scenario without DM's additional preference information.

### 5.3. Robustness analysis

Applying ScoreBin requires setting the parameters of the minimal score influence  $\alpha$  and strength/weakness enhancement  $\beta$ . The choice of their values impacts the obtained comprehensive qualities and the ranking. To investigate the stability of results for various admissible values of  $\alpha$  and  $\beta$  in the feasible space (A, B), we need to conduct a robustness analysis [39]. Its results can be quantified as Rank Acceptability Indexs (RAIs) defined as the share of feasible parameter values that grant an alternative a given rank [29, 25]:

$$RAI(a, r) = \int_{(\alpha, \beta) \in (A, B)} m(\alpha, \beta, a, r) d(\alpha, \beta), \quad (23)$$



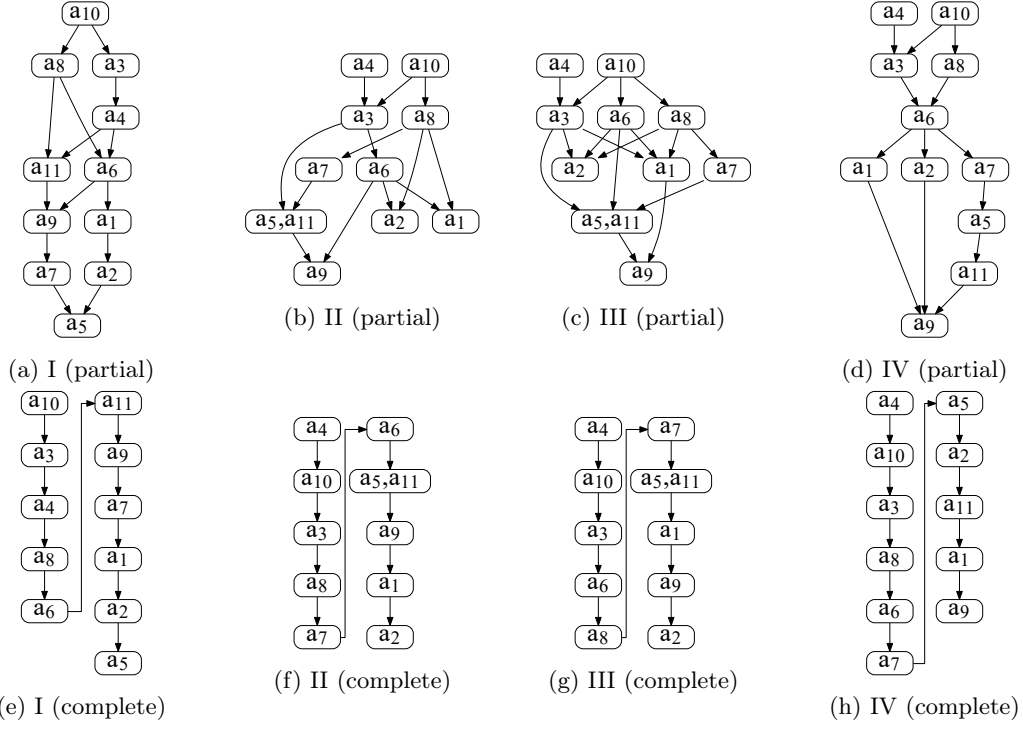


Figure 4: Incomplete (a – d) and complete (e – h) rankings derived with the four variants of ScoreBin for the problem of assessing technological parks in Poland when accounting for indirect preference information.

where  $m(\alpha, \beta, a, r)$  is the rank membership function:

$$m(\alpha, \beta, a, r) = \begin{cases} 1, & \text{if } \text{rank}(M, a) = r, \\ 0, & \text{otherwise.} \end{cases}$$

To estimate  $RAIs$ , we may apply the Monte Carlo simulation and compute the ratio of feasible parameters for which  $a$  is ranked  $r$ -th [8, 45]. We demonstrate the results of such an analysis for the scenario with indirect preference information and ScoreBin I. We simulated uniformly distributed values of  $\alpha \in (0.005, 1]$  with a step 0.005 and  $\beta \geq \alpha$ .

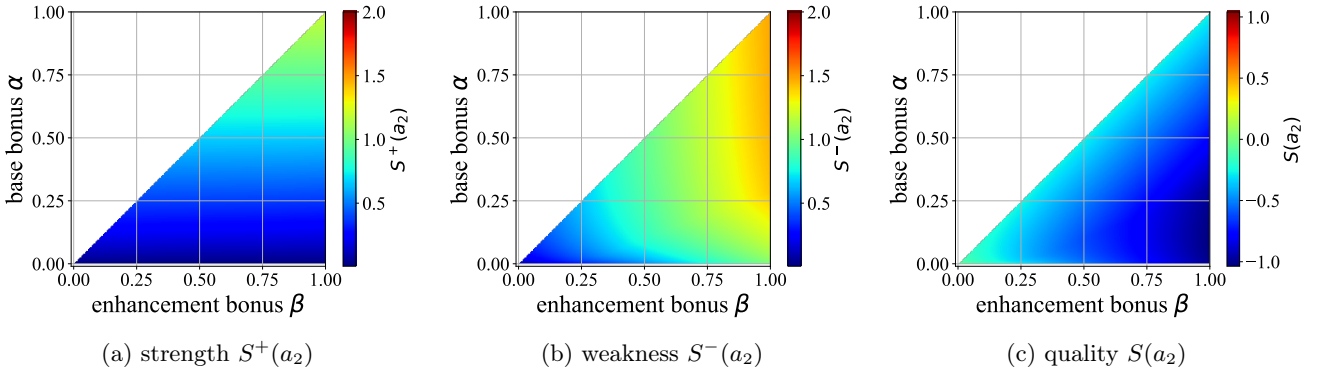


Figure 5: Strengths (a), weaknesses (b), and comprehensive qualities (c) of alternative  $a_2$  for uniformly distributed values of parameters  $\alpha$  and  $\beta$  for the problem of assessing technological parks in Poland.

The strengths, weaknesses, and comprehensive scores of  $a_2$  for different values of  $\alpha$  and  $\beta$  are presented in Figure 5. The respective ranks are illustrated in Figure 6. Since  $a_2$  was not directly judged by the DM and did not outrank any of such alternatives, its strength depends only on  $\alpha$ , growing with the increase of  $\alpha$ . However, this happens for the remaining alternatives too, so the fourth rank of  $a_2$  is stable across all studied parameter values. However, the weakness of  $a_2$ , being outranked by  $a_3$  and  $a_{10}$  increases faster with the raise of  $\alpha$  than for alternatives outranked by

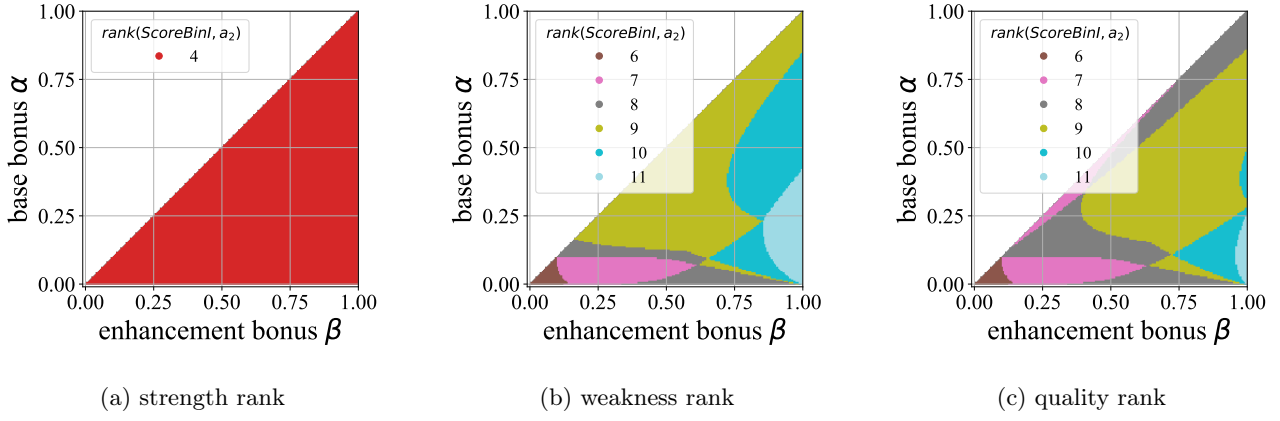


Figure 6: Ranks attained by alternative  $a_2$  for uniformly distributed values of parameters  $\alpha$  and  $\beta$  for the problem of assessing technological parks in Poland.

only one other option (e.g.,  $a_1$ ,  $a_6$ ,  $a_7$ , and  $a_{11}$ ). Moreover,  $a_2$  was indicated as one of the poor alternatives by the DM. Hence its weakness increase with the increase of  $\beta$  too. Overall, the weakness rank of  $a_2$  is sixth for only 1.3% samples (with  $\alpha, \beta < 0.1$ ). The most common weakness rank of  $a_2$  is ninth (49.0%). For large values of  $\beta > 0.9$  and large values of  $\alpha > 0.3$ ,  $a_2$  attained the greatest  $S^-$ . The distribution of qualities and respective ranks for  $a_2$  are illustrated in Figures 5c) and 6c), respectively.

Such distributions can be conveniently summarized with *RAIs*. For all alternatives, the stochastic acceptabilities based on the comprehensive quality measures are presented in Table 13. The most common rank of  $a_2$  is ninth (50.4%), followed by eighth (27.8%) and seventh (11.6%). Its expected rank, obtained by averaging the ranks observed for all samples, is 8.55. The top-ranked options attained the most stable ranks:  $a_{10} - 1$  for 96.4% (interval  $[1, 3]$ ),  $a_3 - 2$  for 93% (interval  $[2, 3]$ ),  $a_4 - 3$  for 91.6% (interval  $[1, 3]$ ), and  $a_8 - 4$  for 100%. In turn, the least preferred alternative for most cases (77.7%) was  $a_9$ . The expected ranking is:

$$a_{10} P a_3 P a_4 P a_8 P a_6 P a_{11} P a_1 P a_7 P a_2 P a_5 P a_9.$$

Table 13: Rank Acceptability Indices (in %) and expected ranks *ER* obtained with ScoreBin I for technological parks in Poland in the scenario accounting for additional preference information.

Rank	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$	$a_{11}$
1	-	-	-	3.6	-	-	-	-	-	96.4	-
2	-	-	93.0	4.8	-	-	-	-	-	2.1	-
3	-	-	7.0	91.6	-	-	-	-	-	1.4	-
4	-	-	-	-	-	-	-	100.0	-	-	-
5	0.9	-	-	-	-	83.8	-	-	-	-	16.2
6	30.6	1.3	-	-	-	15.0	2.3	-	-	-	50.0
7	20.5	11.6	-	-	-	1.3	48.3	-	10.9	-	7.5
8	37.6	27.8	-	-	-	-	20.1	-	1.8	-	12.7
9	10.1	50.4	-	-	2.5	-	19.2	-	4.0	-	13.7
10	0.3	7.2	-	-	78.4	-	8.5	-	5.6	-	-
11	-	1.6	-	-	19.1	-	1.5	-	77.7	-	-
<i>ER</i>	7.26	8.55	2.07	2.88	10.17	5.18	7.87	4.0	10.37	1.05	6.58

## 6. Experimental comparison of results attained by different methods exploiting a crisp outranking relation

This section compares the recommendation attained by the four variants of ScoreBin, Net Flow Score, Qualification Distillation, and ELECTRE I. We simulated decision problems involving from 8 to 20 alternatives (with a step of 2), from 3 to 8 criteria, and performances generated from a uniform distribution from the  $[0, 1]$  range. To investigate the results for various densities of the outranking graph, we considered  $\lambda \in \{0.5, 0.6, 0.7, 0.8\}$  and three different sets

of thresholds: low ( $q_j = 0.05$ ,  $p_j = 0.15$ ,  $v_j = 0.25$ ), medium ( $q_j = 0.15$ ,  $p_j = 0.3$ ,  $v_j = 0.5$ ), and high ( $q_j = 0.25$ ,  $p_j = 0.45$ ,  $v_j = 0.75$ ). For each problem size and parameter setting, we conducted 100 independent runs. If the outranking relation was empty, the problem was regenerated. For ScoreBin, we assumed  $\alpha = 0.1$  and did not simulate indirect preference information.

The similarity of recommendations obtained with ScoreBin and the state-of-the-art methods was quantified using four measures. The average *NRD*, *NHR*, Kendall’s  $\tau$ , and *AP* values are reported in Tables 14 – 17. The more detailed results for various numbers of alternatives and criteria are presented as heatmaps in Figures 7 – 13. The detailed results revealing the impact of thresholds ( $q_j$ ,  $p_j$ ,  $v_j$ ) and credibility threshold  $\lambda$  are presented in the e-Appendix (supplementary material available online). The experiments confirm that even if the results provided by the considered methods are similar, the ranking and choice recommendations they suggest differ.

When considering the differences between partial rankings (see Table 14), the least average *NRD* values are observed for ScoreBin II and III (0.006), and the greatest values hold for ScoreBin IV and *QD* (0.153). The latter two methods construct the most distinctive rankings because *NRD* is greater than 0.1 when they are compared with other approaches. The similarities for pairs of methods based on similar ideas are greater. These include ScoreBin I and IV, ScoreBin II and II, and *NFS* and *QD*. When analyzing the results for various problem sizes (see Figures 7 and 8), the greater number of alternatives and criteria implies greater *NRD* values between the partial rankings obtained with ScoreBin I–III and *QD* or *NFS*. For example, the greatest average *NRD* for ScoreBin I (0.145), ScoreBin II (0.155), and ScoreBin III (0.160) when compared to *QD* are observed for 20 alternatives and 6 criteria. In turn, the least differences are observed for the problems with 6 alternatives and 3 criteria (*NRD* equal to 0.081, 0.072, and 0.071). For ScoreBin IV and *QD*, the greatest differences are observed for problems involving from 8 to 12 alternatives and 3 criteria. When comparing ScoreBin IV and *NFS*, the greatest differences hold for problems with numerous alternatives and few criteria.

The trends observed for comparing complete rankings are very similar (see Kendall’s  $\tau$  values in Figure 9). The rankings obtained with five methods are highly similar (see Table 15), with the greatest Kendall’s  $\tau$  values observed for ScoreBin II and III (0.989) and the least value holding for ScoreBin III and IV (0.8). The rankings obtained with ScoreBin and *NFS* for various problem sizes are the most similar for small problems with 6 alternatives and 3 criteria. The values of Kendall’s  $\tau$  are similar for the remaining sizes. For example, for ScoreBin I and *NFS*, they range between 0.893 and 0.921.

When comparing the agreement in indicating the subset of the most preferred alternatives (see Table 16), the observations on the most and the least similar pairs of methods are the same. However, the absolute similarity values are lower. For example, ScoreBin IV recommends the same alternatives as other methods in about 60% cases. The evident exception is formed by ScoreBin II and III, for which the choice-based agreement is close to 99%. When comparing the top-ranked alternatives with the kernel, we can note substantial differences (*NHR* between 0.223 and 0.323). This is caused by the kernel’s specific nature, which implies that apart from the evidently favorable alternatives, it can also contain other options which are outranked by many other alternatives but are needed to satisfy the kernel’s properties. When considering the results for various problem sizes, the trends for *NHR* are similar to those for *NRD* (see Figures 10 and 11). In particular, the choice-oriented recommendation made by ScoreBin is the most similar to the one by *NFS* for small problems with 6 alternatives. For ScoreBin I–III, the *NHR* values are the greatest for problems with few criteria, and for ScoreBin IV – for instances involving more criteria. When comparing the alternatives ranked at the top by ScoreBin with the graph kernel, the *NHR* values decrease with the more significant numbers of alternatives and criteria (see Figure 12).

The *AP* measure quantified the average position of alternatives in the kernel in the ranking generated with a given method. The average results are presented in Table 17. The least (i.e., the best) value of *AP* is attained by ScoreBin IV. However, the differences are not substantial, with the least *AP* value observed for ScoreBin II equal to 2.824. Trends visible in Figure 13 confirm that when considering problems with fewer alternatives, the average position of the options contained in the graph kernel is higher (better). This means that when more alternatives are involved, the graph kernel tends to include some options that are ranked low by ScoreBin. For ScoreBin I–III, an

additional trend is observed: with the increase in the number of criteria,  $AP$  gets better. For example, for ScoreBin I, 20 alternatives, and 3 criteria,  $AP$  is 4.122 but for 8 criteria in drops to 3.497. An opposite trend can be observed for ScoreBin IV with the respective measure values equal to 3.081 (for 3 criteria) and 3.658 (for 8 criteria).

Table 14: The average  $NRD$  values for the partial rankings obtained with the six methods for all considered problem instances.

Method	ScoreBin I	ScoreBin II	ScoreBin III	ScoreBin IV	QD	NFS
ScoreBin I	0.000	0.084	0.089	0.108	0.128	0.059
ScoreBin II	0.084	0.000	0.006	0.148	0.127	0.056
ScoreBin III	0.089	0.006	0.000	0.153	0.130	0.061
ScoreBin IV	0.108	0.148	0.153	0.000	0.156	0.103
QD	0.128	0.127	0.130	0.156	0.000	0.086
NFS	0.059	0.056	0.061	0.103	0.086	0.000

Table 15: The average Kendall's  $\tau$  for the complete rankings obtained with the five methods for all considered problem instances.

Method	ScoreBin I	ScoreBin II	ScoreBin III	ScoreBin IV	NFS
ScoreBin I	1.000	0.873	0.864	0.873	0.908
ScoreBin II	0.873	1.000	0.989	0.810	0.918
ScoreBin III	0.864	0.989	1.000	0.800	0.908
ScoreBin IV	0.873	0.810	0.800	1.000	0.865
NFS	0.908	0.918	0.908	0.865	1.000

Table 16: The average  $NHR$  values comparing graph kernel and partial rankings for all considered problem instances.

Method	ScoreBin I	ScoreBin II	ScoreBin III	ScoreBin IV	QD	NFS	Kernel
ScoreBin I	1.000	0.824	0.816	0.685	0.714	0.830	0.278
ScoreBin II	0.824	1.000	0.986	0.656	0.735	0.860	0.270
ScoreBin III	0.816	0.986	1.000	0.647	0.730	0.848	0.269
ScoreBin IV	0.685	0.656	0.647	1.000	0.587	0.679	0.323
QD	0.714	0.735	0.730	0.587	1.000	0.836	0.223
NFS	0.830	0.860	0.848	0.679	0.836	1.000	0.254
Kernel	0.278	0.270	0.269	0.323	0.223	0.254	1.000

Table 17: The average  $AP$  values comparing the graph kernel with the complete rankings generated by five methods for all considered problem instances.

Method	ScoreBin I	ScoreBin II	ScoreBin III	ScoreBin IV	NFS
AP	2.763	2.824	2.823	2.502	2.799

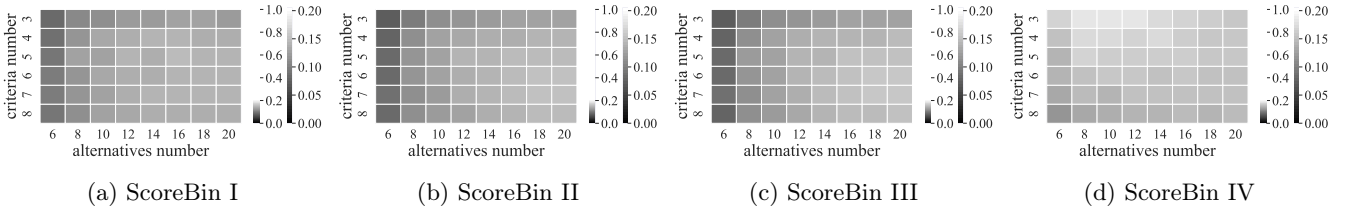


Figure 7: Normalized Ranking Distance for partial rankings obtained with ScoreBin and  $QD$  for different numbers of criteria and alternatives.

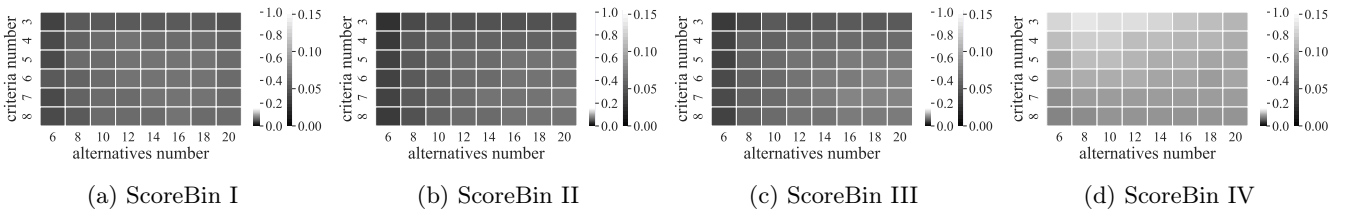


Figure 8: Normalized Ranking Distance for partial rankings obtained with ScoreBin and  $NFS$  for different numbers of criteria and alternatives.

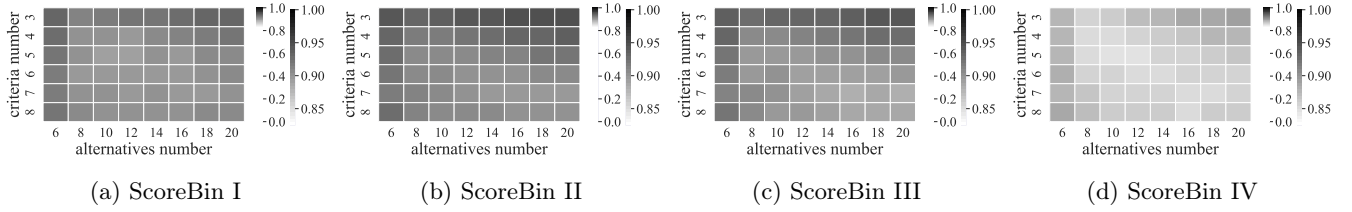


Figure 9: Kendall's  $\tau$  for complete rankings obtained with ScoreBin and *NFS* for different numbers of criteria and alternatives.

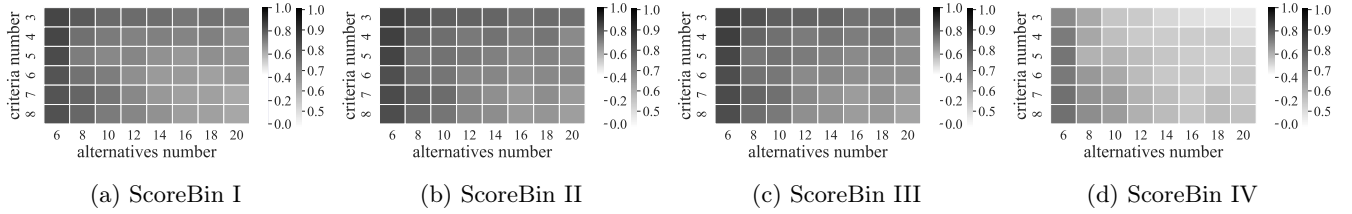


Figure 10: Normalized Hit Ratio for the choice-based recommendations derived from the partial rankings obtained with ScoreBin and *QD* for different numbers of criteria and alternatives.

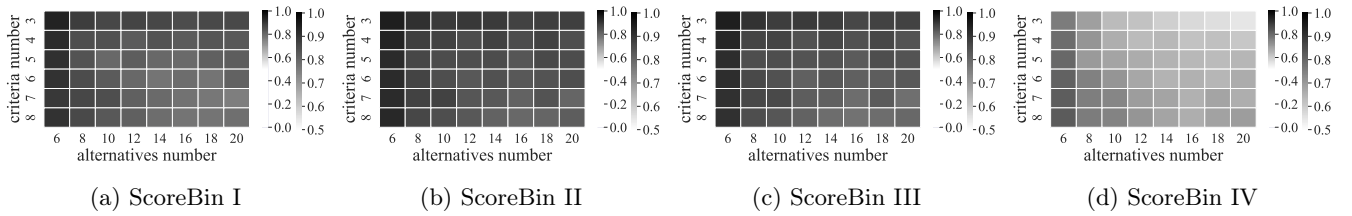


Figure 11: Normalized Hit Ratio for the choice-based recommendations derived from the partial rankings obtained with ScoreBin and *NFS* for different numbers of criteria and alternatives.

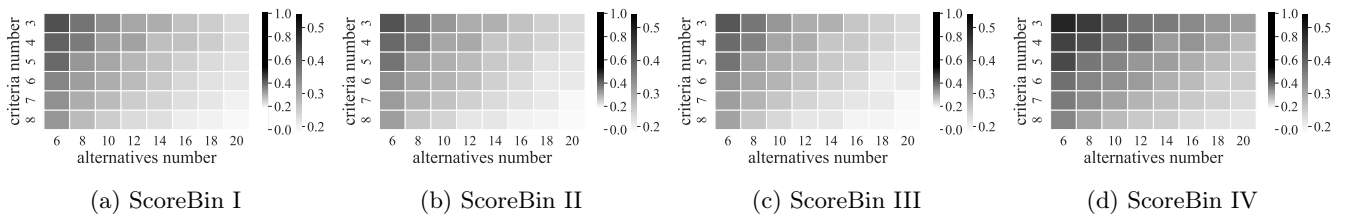


Figure 12: Normalized Hit Ratio for the choice-based recommendations derived from the complete rankings obtained with ScoreBin and graph kernels for different numbers of criteria and alternatives.

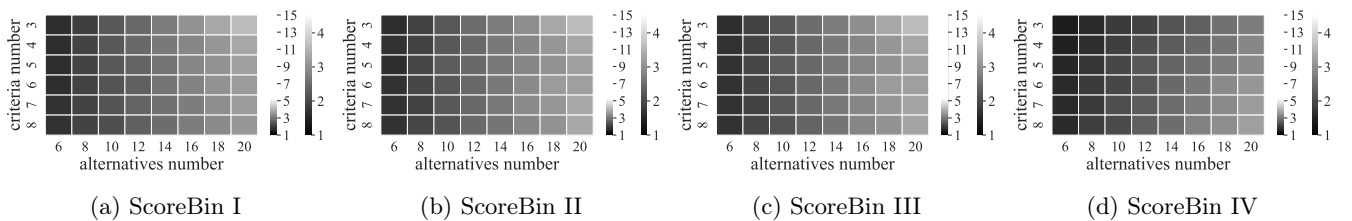


Figure 13: Average position *AP* of alternatives contained in the kernel in the complete rankings obtained with ScoreBin for different numbers of criteria and alternatives.

## 7. Summary

This paper introduced a family of ScoreBin methods for ranking and choice problems. They exploit a crisp outranking relation to deliver each alternative's strength, weakness, and comprehensive quality. We defined four method variants that score alternatives by analyzing which options they outrank and which are preferred to them. Moreover, we consider the relative easiness or difficulty of outranking each alternative by analyzing its relations with the remaining options. The four variants differ in the weights they associate with alternatives, defining their contribution to the

strengths and weaknesses of alternatives they are related to. We enriched ScoreBin to tolerate optional indirect preference information indicating subsets of strong and weak alternatives according to DM. The method allows the construction of either a partial preorder based on the separate consideration of strengths and weaknesses or a complete ranking derived from the analysis of comprehensive qualities.

We applied ScoreBin to assess technological parks in Poland, aiming to indicate parks that are managed efficiently, bring substantial economic profits, and support innovation development. We identified  $a_{10}$  as the most promising option and  $a_4$  as its backup. We enriched the primary analysis in a two-fold way. On the one hand, we included indirect preference information, revealing the intuitive changes in the ranking given the DM's indication of strong and weak options. On the other hand, we conducted the robustness analysis and quantified the shares of feasible parameter values that grant each alternative a given position. This way, we created a more robust ranking that is less sensitive to the selection of specific values for the parameters of the base and enhancement bonuses or penalties.

We compared the recommendations offered by ScoreBin, Net Flow Score, Qualification Distillation, and ELECTRE I on many simulated decision problems. We quantified the similarity between the partial and complete rankings and top-ranked alternatives. The results for all measures were consistent. They confirmed the most significant similarity for ScoreBin II and III, I and IV, and *NFS* and *QD*. The most distinctive results among the versions of ScoreBin are offered by its fourth variant.

The choice of a particular variant of ScoreBin depends on which way of computing the strengths and weakness is the most appealing for a given problem. Intuitively, we find the logic of ScoreBin I and IV as most relevant for real-world problems. On the one hand, ScoreBin I assumes that a strong alternative needs to outrank strong alternatives. On the other hand, ScoreBin IV is based on the idea that a strong alternative needs to outrank alternatives that are difficult to outrank.

The most attractive directions for future research are three-fold. First, we may adapt the method to decision contexts where a crisp preference relation is constructed differently. For example, it may be helpful to infer the weights of DMs in a group decision problem based on the analysis of relations in the committee [12]. Obviously, the valued outranking relation that is subsequently binarized may also be established otherwise. In particular, it may represent the share of feasible parameter values that support the preference for one alternative over another [3, 23] or the performances of alternatives do not need to be deterministic, hence reflecting some uncertainty [31, 47]. Such uncertainty can also be related to indirect judgments indicating some alternatives as strong or weak. Second, we computed the strengths and weaknesses defined consistently. However, it is possible to combine their definitions from different variants of ScoreBin under the same methodological framework. Clearly, it is also possible to use a few variants at once and aggregate their results either by averaging them or investigating the spaces of consensus and disagreement [33]. Third, the essential direction concerns using ScoreBin to support the solution of real-world decision problems. A crisp outranking relation has already been used as a preference model in such various studies as personnel, supplier, site, project, or investment portfolio selections [18, 35]. Hence we find applications of such types as the most promising ones.

## Acknowledgments

Krzysztof Martyn and Miłosz Kadziński acknowledge support from the Polish National Science Center under the SONATA BIS project (grant no. DEC-2019/34/E/HS4/00045). Magdalena Martyn is grateful for the support from the Polish Ministry of Education and Science (grant no. 0311/SBAD/0735).

- [1] Aissi, H. and Roy, B. (2010). Robustness in multi-criteria decision aiding. In Ehrgott, M., Figueira, J. R., and Greco, S., editors, *Trends in Multiple Criteria Decision Analysis*, pages 87–121. Springer.
- [2] Almeida-Dias, J., Figueira, J., and Roy, B. (2012). A multiple criteria sorting method where each category is characterized by several reference actions: The Electre Tri-nC method. *European Journal of Operational Research*, 217(3):567–579.
- [3] Angilella, S., Catalfo, P., Corrente, S., Giarlotta, A., Greco, S., and Rizzo, M. (2018). Robust sustainable development assessment with composite indices aggregating interacting dimensions: The hierarchical-SMAA-Choquet integral approach. *Knowledge-Based Systems*, 158:136–153.

- [4] Behzadian, M., Kazemzadeh, R., Albadvi, A., and Aghdasi, M. (2010). PROMETHEE: A comprehensive literature review on methodologies and applications. *European Journal of Operational Research*, 200(1):198–215.
- [5] Brans, S. J. and Fishburn, P. C. (1978). Approval voting. *American Political Science Review*, 72(3):831–847.
- [6] Brans, J. and Vincke, P. (1985). A preference ranking organization method. *Management Science*, 31(6):647–656.
- [7] Brans, J.-P. and Mareschal, B. (2005). *PROMETHEE Methods*, pages 163–186. Springer, New York, NY.
- [8] Ciomek, K. and Kadziński, M. (2021). Polyrun: A Java library for sampling from the bounded convex polytopes. *SoftwareX*, 13:100659.
- [9] Corrente, S., Greco, S., Kadziński, M., and Słowiński, R. (2013). Robust ordinal regression in preference learning and ranking. *Machine Learning*, 93:381–422.
- [10] Corrente, S., Greco, S., and Słowiński, R. (2016). Multiple criteria hierarchy process for ELECTRE Tri methods. *European Journal of Operational Research*, 252(1):191–203.
- [11] Dias, L. C. and Rocha, H. (2023). A stochastic method for exploiting outranking relations in multicriteria choice problems. *Annals of Operations Research*, 321(1):165–189.
- [12] Eden, C. and Kilgour, D. M. (2010). *Handbook of group decision and negotiation*. Springer, Cham.
- [13] Figueira, J. and Roy, B. (2002). Determining the weights of criteria in the ELECTRE type methods with a revised Simos’ procedure. *European journal of operational research*, 139(2):317–326.
- [14] Figueira, J. R., Greco, S., and Roy, B. (2022). Electre-Score: A first outranking based method for scoring actions. *European Journal of Operational Research*, 297(3):986–1005.
- [15] Figueira, J. R., Greco, S., Roy, B., and Słowiński, R. (2013). An Overview of ELECTRE Methods and their Recent Extensions. *Journal of Multi-Criteria Decision Analysis*, 20(1-2):61–85.
- [16] Figueira, J. R., Greco, S., and Słowiński, R. (2009). Building a set of additive value functions representing a reference preorder and intensities of preference: GRIP method. *European Journal of Operational Research*, 195(2):460–486.
- [17] Figueira, J. R., Mousseau, V., and Roy, B. (2016). *ELECTRE Methods*, pages 155–185. Springer, New York.
- [18] Govindan, K. and Jepsen, M. B. (2016). ELECTRE: A comprehensive literature review on methodologies and applications. *European Journal of Operational Research*, 250(1):1–29.
- [19] Govindan, K., Kadziński, M., Ehling, R., and Miebs, G. (2019). Selection of a sustainable third-party reverse logistics provider based on the robustness analysis of an outranking graph kernel conducted with ELECTRE I and SMAA. *Omega*, 85:1–15.
- [20] Greco, S., Ehrgott, M., and Figueira, J. (2016). *Multiple Criteria Decision Analysis: State of the Art Surveys*. International Series in Operations Research & Management Science. Springer New York.
- [21] Gyongyi, Z., Garcia-Molina, H., and Pedersen, J. (2004). Combating web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*.
- [22] Kadziński, M., Greco, S., and Słowiński, R. (2013). RUTA: a framework for assessing and selecting additive value functions on the basis of rank related requirements. *Omega*, 41(4):735–751.
- [23] Kadziński, M., Greco, S., and Słowiński, R. (2015). Multiple criteria ranking and choice with all compatible minimal cover sets of decision rules. *Knowledge-Based Systems*, 89:569–583.
- [24] Kadziński, M. and Michalski, M. (2016). Scoring procedures for multiple criteria decision aiding with robust and stochastic ordinal regression. *Computers & Operations Research*, 71:54–70.
- [25] Kadziński, M. and Tervonen, T. (2013). Robust multi-criteria ranking with additive value models and holistic pair-wise preference statements. *European Journal of Operational Research*, 228(1):169–180.
- [26] Kendall, M. G. (1938). A New Measure of Rank Correlation. *Biometrika*, 30(1/2):81–93.
- [27] Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632.
- [28] Kowaluk, B. (2010). Benchmarking of technological parks in Poland (in Polish). Technical report, Polska Agencja Rozwoju Przedsiębiorczości, Warsaw, Poland.
- [29] Lahdelma, R. and Salminen, P. (2001). SMAA-2: Stochastic Multicriteria Acceptability Analysis for Group Decision Making. *Operations Research*, 49(3):444–454.
- [30] Lempel, R. and Moran, S. (2000). The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks*, 33(1):387–401.
- [31] Liao, H., Yang, L., and Xu, Z. (2018). Two new approaches based on ELECTRE II to solve the multiple criteria decision making problems with hesitant fuzzy linguistic term sets. *Applied Soft Computing*, 63:223–234.
- [32] Martel, J.-M., D’Avignon, G. R., and Couillard, J. (1986). A fuzzy outranking relation in multicriteria decision making. *European Journal of Operational Research*, 25(2):258–271.
- [33] Miebs, G. and Kadziński, M. (2021). Heuristic algorithms for aggregation of incomplete rankings in multiple criteria group decision making. *Information Sciences*, 560:107–136.
- [34] Mousseau, V. and Dias, L. (2004). Valued outranking relations in ELECTRE providing manageable disaggregation procedures. *European Journal of Operational Research*, 156(2):467–482.
- [35] Rogers, M., Bruen, M., Maystre, L.-Y., Rogers, M., Bruen, M., and Maystre, L.-Y. (2000). The ELECTRE methodology. *ELECTRE and Decision Support: Methods and Applications in Engineering and Infrastructure Investment*, pages 45–85.
- [36] Roy, B. (1978). ELECTRE III : Un algorithme de classements fondé sur une représentation floue des préférences en présence de critères multiples. *Cahiers du CERO*, 20(1):3–24.
- [37] Roy, B. (1991). The Outranking Approach and the Foundations of ELECTRE Methods. *Theory and Decision*, 31(1):49–73.
- [38] Roy, B. (1996). *Multicriteria Methodology for Decision Aiding*. Kluwer Academic, Dordrecht.

- [39] Roy, B. (2010). Robustness in operational research and decision aiding: A multi-faceted issue. *European Journal of Operational Research*, 200(3):629–638.
- [40] Roy, B., Figueira, J., and Almeida-Dias, J. (2014). Discriminating thresholds as a tool to cope with imperfect knowledge in multiple criteria decision aiding: Theoretical results and practical issues. *Omega*, 43:9–20.
- [41] Roy, B. and Slowinski, R. (1993). Criterion of distance between technical programming and socio-economic priority. *RAIRO - Operations Research - Recherche Opérationnelle*, 27(1):45–60.
- [42] Roy, B. (1968). Classement et choix en présence de points de vue multiples. *R.I.R.O.*, 2(8):57–75.
- [43] Shanian, A., Milani, A., Carson, C., and Abeyaratne, R. (2008). A new application of ELECTRE III and revised Simos procedure for group material selection under weighting uncertainty. *Knowledge-Based Systems*, 21(7):709–720.
- [44] Szlag, M., Greco, S., and Słowiński, R. (2014). Variable consistency dominance-based rough set approach to preference learning in multicriteria ranking. *Information Sciences*, 277:525–552.
- [45] Tervonen, T. and Lahdelma, R. (2007). Implementing stochastic multicriteria acceptability analysis. *European Journal of Operational Research*, 178(2):500–513.
- [46] Vincke, P. (1992). Exploitation of a crisp relation in a ranking problem. *Theory and Decision*, 32(3):221–240.
- [47] Wu, X. and Liao, H. (2023). Managing uncertain preferences of consumers in product ranking by probabilistic linguistic preference relations. *Knowledge-Based Systems*, 262:110240.
- [48] Zahid, K., Akram, M., and Kahraman, C. (2022). A new ELECTRE-based method for group decision-making with complex spherical fuzzy information. *Knowledge-Based Systems*, 243:108525.



# ScoreBin: scoring alternatives based on a crisp outranking relation with an application to the assessment of Polish technological parks – e-Appendix

Krzysztof Martyn<sup>a,\*</sup>, Magdalena Martyn<sup>a</sup>, Miłosz Kadziński<sup>a</sup>

<sup>a</sup>*Faculty of Computing and Telecommunications, Poznan University of Technology, Piotrowo 2, 60-965 Poznań, Poland*

## 1. Detailed similarity results between rankings obtained with ScoreBin and the state-of-the-art methods

This section presents heatmaps of average *NRD*, *NHR*, Kendall's  $\tau$ , and *AP* values based on 100 runs for each problem size.

### 1.1. The impact of the credibility threshold and the number of alternatives on the similarity between recommendations

The detailed results for various numbers of alternatives and different values of credibility threshold  $\lambda$  are in Figures 1–7. The similarity for partial rankings (*NRD*) obtained with ScoreBin, QD, and NFS is high. For example, when taking NFS and QD as the reference, the greatest values are observed for ScoreBin IV (0.116 and 0.171, respectively). The complete rankings are also very similar, as captured by Kendall's  $\tau$ . The least similarity is observed for ScoreBin IV and QD. When comparing the recommendations of the most preferred alternatives (*NHR*), the most similar methods are NFS and ScoreBin I–III, and the least agreement is observed ScoreBin IV and QD. When considering all measures, the credibility threshold  $\lambda$  has a negligible impact on the similarity between the rankings. There is a trend across all metrics indicating that the similarity of rankings decreases with the increase in the number of alternatives.

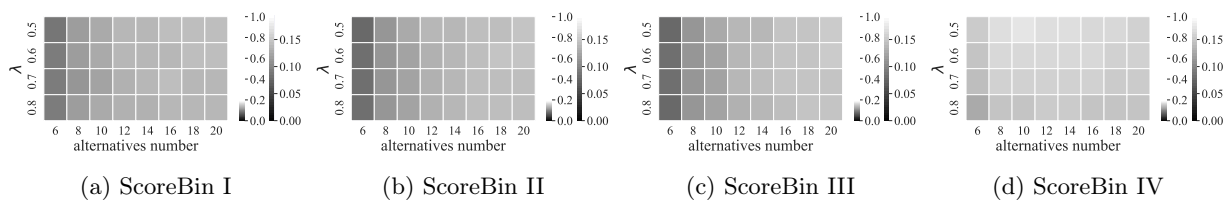


Figure 1: Normalized Ranking Distance for partial rankings obtained with ScoreBin and QD for different numbers of alternatives and values of credibility threshold  $\lambda$ .

\*Corresponding author: Institute of Computing Science, Poznań University of Technology, Piotrowo 2, 60-965 Poznań, Poland. Tel. +48-61 665 3022.

*Email addresses:* [krzysztof.martyn@cs.put.poznan.pl](mailto:krzysztof.martyn@cs.put.poznan.pl) (Krzysztof Martyn), [magdalena.martyn@cs.put.poznan.pl](mailto:magdalena.martyn@cs.put.poznan.pl) (Magdalena Martyn), [milosz.kadziński@cs.put.poznan.pl](mailto:milosz.kadziński@cs.put.poznan.pl) (Miłosz Kadziński)

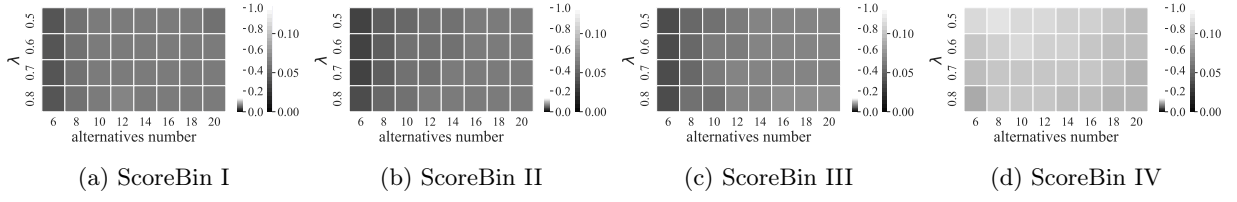


Figure 2: Normalized Ranking Distance for partial rankings obtained with ScoreBin and NFS for different numbers of alternatives and values of credibility threshold  $\lambda$ .

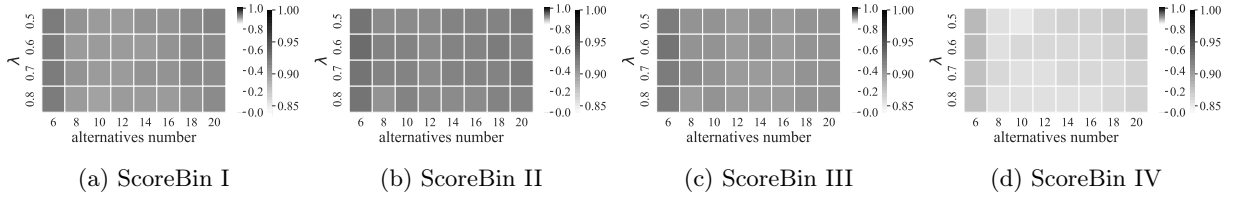


Figure 3: Kendall's  $\tau$  for complete rankings obtained with ScoreBin and NFS for different numbers of alternatives and values of credibility threshold  $\lambda$ .

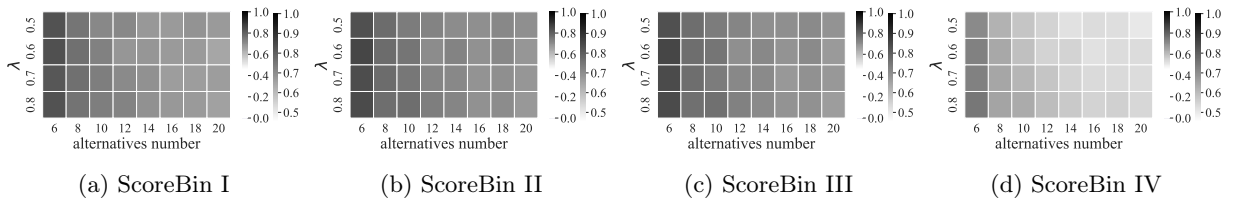


Figure 4: Normalized Hit Ratio for the choice-based recommendations derived from the partial rankings obtained with ScoreBin and QD for different numbers of alternatives and values of credibility threshold  $\lambda$ .

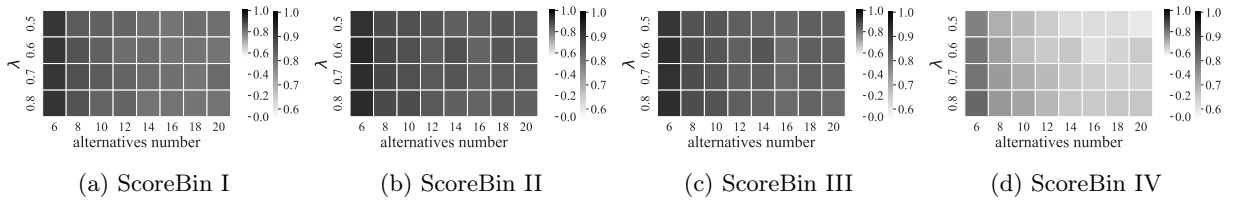


Figure 5: Normalized Hit Ratio for the choice-based recommendations derived from the partial rankings obtained with ScoreBin and NFS for different numbers of alternatives and values of credibility threshold  $\lambda$ .

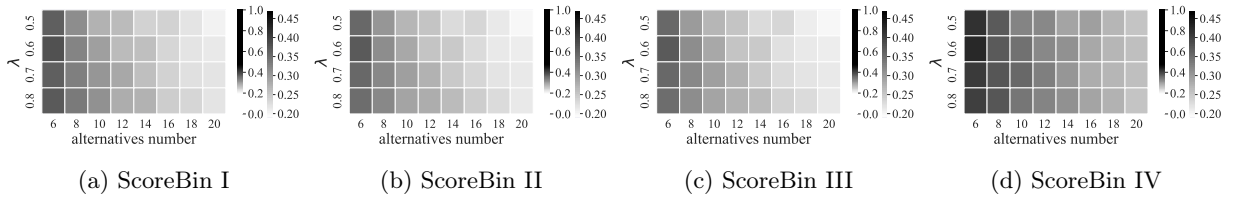


Figure 6: Normalized Hit Ratio for the choice-based recommendations derived from the complete rankings obtained with ScoreBin and graph kernels for different numbers of alternatives and values of credibility threshold  $\lambda$ .

### 1.2. The impact of the criteria thresholds and the number of alternatives on the similarity between recommendations

Heatmaps for different numbers of alternatives and three different sets of thresholds: low ( $q_j = 0.05$ ,  $p_j = 0.15$ ,  $v_j = 0.25$ ), medium ( $q_j = 0.15$ ,  $p_j = 0.3$ ,  $v_j = 0.5$ ), and high ( $q_j = 0.25$ ,  $p_j = 0.45$ ,  $v_j = 0.75$ ) are

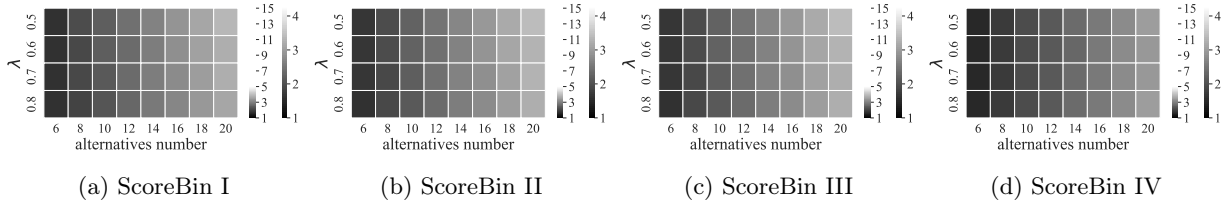


Figure 7: Average position AP of alternatives contained in the kernel in the complete rankings obtained with ScoreBin for different numbers of alternatives and values of credibility threshold  $\lambda$ .

in Figures 8– 14.

When comparing both the similarity between the rankings and the recommended alternatives between ScoreBin and QD and NFS, the most similar are for low values of thresholds. Additionally, it can be seen that the biggest difference in NHR is between ScoreBin IV and NFS for high parameter values. When we consider the suggested options given by the kernel, they are more similar for medium and less similar for high parameters.

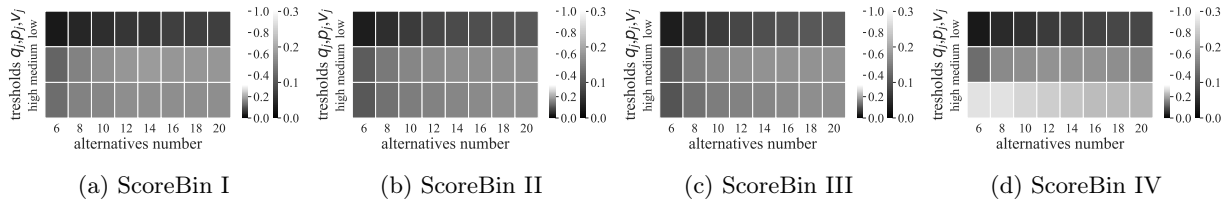


Figure 8: Normalized Ranking Distance for partial rankings obtained with ScoreBin and QD for different numbers of alternatives and three different sets of thresholds  $q_j, p_j, v_j$ .

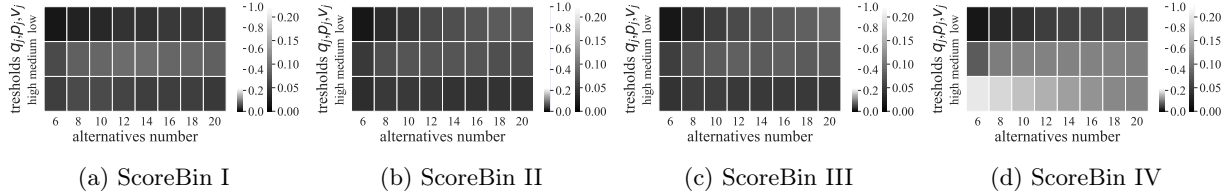


Figure 9: Normalized Ranking Distance for partial rankings obtained with ScoreBin and NFS for different numbers of alternatives and three different sets of thresholds  $q_j, p_j, v_j$ .

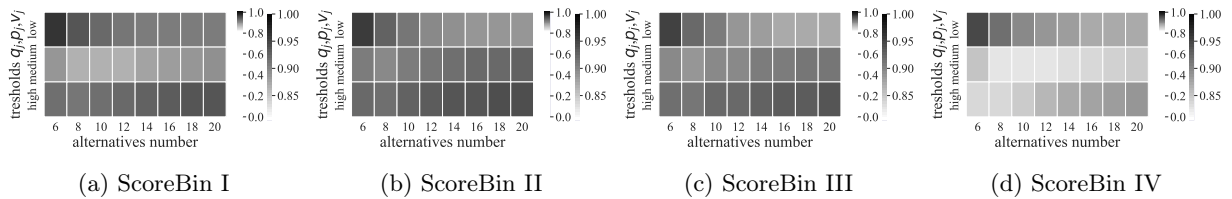


Figure 10: Kendall's  $\tau$  for complete rankings obtained with ScoreBin and NFS for different numbers of alternatives and three different sets of thresholds  $q_j, p_j, v_j$ .

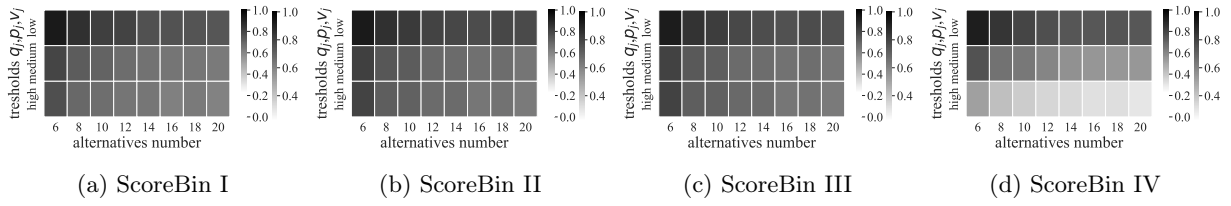


Figure 11: Normalized Hit Ratio for the choice-based recommendations derived from the partial rankings obtained with ScoreBin and QD for different numbers of alternatives and three different sets of thresholds  $q_j, p_j, v_j$ .

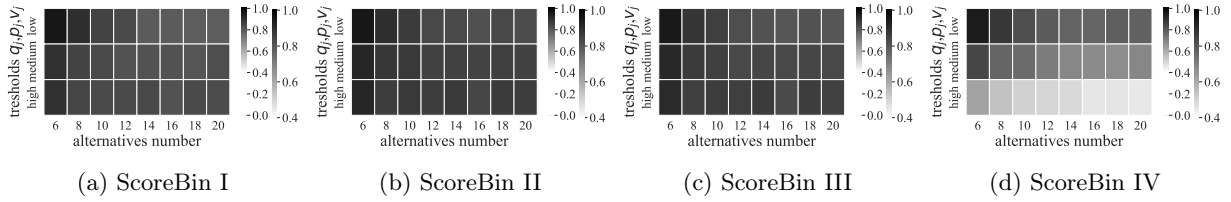


Figure 12: Normalized Hit Ratio for the choice-based recommendations derived from the partial rankings obtained with ScoreBin and NFS for different numbers of alternatives and three different sets of thresholds  $q_j, p_j, v_j$ .

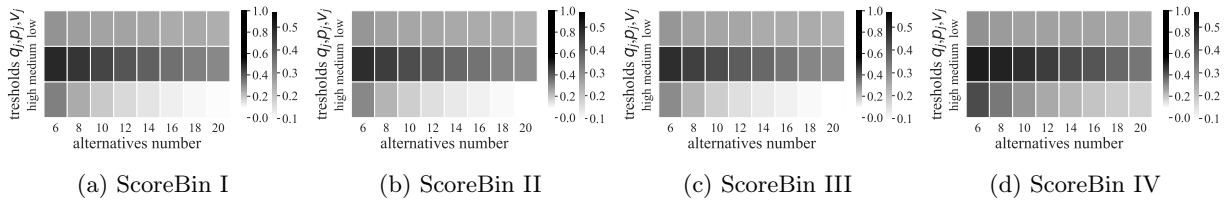


Figure 13: Normalized Hit Ratio for the choice-based recommendations derived from the complete rankings obtained with ScoreBin and graph kernels for different numbers of alternatives and three different sets of thresholds  $q_j, p_j, v_j$ .

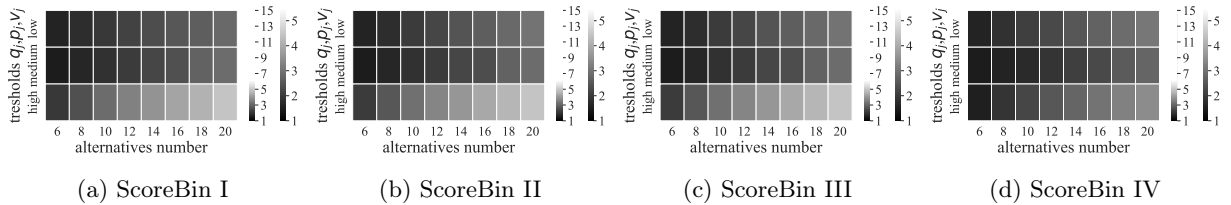


Figure 14: Average position AP of alternatives contained in the kernel in the complete rankings obtained with ScoreBin for different numbers of alternatives and three different sets of thresholds  $q_j, p_j, v_j$ .

# Extended abstract in Polish

## Metody wielokryteriowego wspomagania decyzji inspirowane innymi poddyscyplinami sztucznej inteligencji

### Wprowadzenie

Problemy decyzyjne uwzględniają różne punkty widzenia na jakość rozważanych wariantów. Te punkty są sformalizowane jako kryteria oceny. W rzeczywistych problemach zazwyczaj nie istnieje opcja, która posiada najlepsze oceny na wszystkich kryteriach. Prowadzi to do sytuacji, w której istnieje wiele potencjalnie najlepszych rozwiązań i od preferencji decydenta zależy, które z nich uzna za najbardziej korzystne. Decydent musi więc wzbogacić przebieg procesu decyzyjnego o informację preferencyjną, która odzwierciedla jego system wartości. Głównym zadaniem inteligentnych systemów wspomagania decyzji jest wykorzystania takich preferencji w celu wypracowania spójnej rekomendacji dla danego problemu.

Problemy decyzyjne można podzielić na trzy główne typy: wybór, ranking (porządkowanie) oraz sortowanie (klasyfikacja porządkowa). Wybór polega na wskazaniu podzbioru najbardziej preferowanych opcji. Ranking dotyczy szeregowania wariantów od najlepszego do najgorszego. Z kolei sortowanie polega na przypisaniu wariantów do predefiniowanych klas, które są uporządkowane pod względem preferencji.

W ramach różnych poddyscyplin sztucznej inteligencji zostało zaproponowanych wiele narzędzi wspierających użytkowników w przetwarzaniu i analizie danych. W ramach wielokryteriowego wspomagania decyzji (ang. *Multi-Criteria Decision Aiding* – MCDA) zaproponowano metody i techniki, które wypracowują wiarygodną rekomendację w oparciu o dobrze ugruntowane podstawy matematyczne. Z kolei uczenie maszynowe (ang. *Machine Learning* – ML) koncentruje się na rozwijaniu algorytmów uczących się na podsta-

wie danych. Służą one do wykrywania występujących w nich wzorców oraz predykcji dla nowych, niewidzianych na etapie uczenia danych. W szczególności modele głębokich sieci neuronowych są w stanie przetworzyć duże zbiory danych i na ich podstawie rozwiązywać złożone problemy. Wreszcie metody eksploracji Internetu pozwalają na ocenianie stron internetowych, bazując na znajdujących się na nich informacjach oraz hiperłączach.

Wielokryteriowe wspomaganie decyzji oraz uczenie maszynowe pozwalają na przeanalizowanie różnych opcji i zarekomendowanie decydentowi sposobu rozwiązania problemu decyzyjnego. Główne cele i założenia tych dyscyplin, a co za tym idzie metody i ich możliwości, różnią się. Po pierwsze, MCDA w pełni skupia się na użytkowniku, jego wiedzy i preferencjach. Rozwiązanie problemu jest zależne od jego osądów, dotyczących wariantów oraz dostarczonej informacji preferencyjnej. Przez ich eksploatację podejścia wspomaganie decyzji odkrywają priorytety użytkownika. Z kolei ML jest głównie nastawione na model, skupiając się na eksploracji danych poprzez ich analizę i odkrywaniu występujących w nich wzorców. Głównym celem uczenia maszynowego jest rozwiązanie problemu, optymalizującego jakąś cechę rozwiązania, np. w postaci minimalizacji funkcji straty. Te różne cele mają swoje przełożenie na charakterystyką metod, wielkości rozważanych problemów oraz udziału decydenta w rozwiązaniu problemów.

Wykorzystywane w MCDA modele preferencji są inspirowane rzeczywistymi sposobami podejmowania decyzji przez ludzi. Z tego względu metody MCDA są łatwo interpretowalne; ich rekomendacje są wyjaśnialne, a proces przetwarzania można uzasadnić. ML skupia się na nieliniowych modelach, pozwalających na odkrywanie abstrakcyjnych oraz złożonych wzorców i zależności w danych. Pozwala to na uzyskanie wysokiej skuteczności predykcyjnej, ale ogranicza możliwości w analizie wpływu danych wejściowych na ostateczną decyzję.

Ograniczenia i możliwości różnych obszarów sztucznej inteligencji były motywacją do przeprowadzenia badań w ramach tej rozprawy doktorskiej. Zaobserwowano, że wykorzystanie w metodach MCDA inspiracji z uczenia maszynowego, głębokich sieci neuronowych czy eksploracji zasobów Internetu mogą pozwolić na rozwiązywanie nowych, bardziej złożonych problemów decyzyjnych. Przeprowadzone badania odbywały się w trzech obszarach badawczych, które zostały opisane w pięciu publikacjach, z czego na dzień 31 maja 2023 roku trzy zostały zaakceptowane do druku w międzynarodowych czasopismach.

## **Metody wspomaganie decyzji inspirowane głębokimi sieciami neuronowymi**

W ciągu ostatnich lat nastąpił znaczący wzrost ilości gromadzonych i przetwarzanych danych. Dostępne są obszerne zbiory danych, które zawierają zarówno dane historyczne dotyczące problemów decyzyjnych, jak i informacje o podjętych w przeszłości decyzjach.

Kluczowym czynnikiem dla przedsiębiorstw gromadzących te dane jest możliwość przeanalizowania ich w sposób zrozumiały, tj. umożliwiający zweryfikowanie poprawności wyciągniętych wniosków. Od lat metody MCDA dostarczają narzędzi do analizy i wspomagania procesu podejmowania decyzji. Te podejścia są łatwe w interpretacji i zapewniają wiarygodne wyjaśnienie swoich rekomendacji.

Zdobywanie informacji preferencyjnej w postaci bezpośredniego dialogu z decydentem powoduje, iż tradycyjne metody MCDA zostały zaprojektowane, aby uczyć się z niewielkich zbiorów danych. Te informacje preferencyjne odzwierciedlają rzeczywisty system wartości decydenta, który często charakteryzuje się wysoką spójnością. W takiej sytuacji zazwyczaj preferencje są dezagregowane do wartości parametrów metody za pomocą programowania matematycznego. Metody te zawodzą jednak w sytuacjach, gdy informacja preferencyjna jest bogata i silnie niespójna.

Natomiast głębokie sieci neuronowe od początku miały na celu radzenie sobie z dużymi zbiorami danych uczących, obarczonych szumem i niespójnościami. Dużą uwagę poświęcono tu kwestii optymalizacji procesu treningu oraz zmniejszenia czasu obliczeń. Do treningu wykorzystywane są zaawansowane techniki statystyczne oraz optymalizacyjne, pozwalające w efektywny sposób przeszukać przestrzeń parametrów w celu znalezienia najbardziej pasującego modelu. Większość obliczeń w sieciach neuronowych polega na operacjach na macierzach, przez co mogą być wykonywane równoległe na dedykowanych układach sprzętowych takich jak jednostki przetwarzania graficznego (GPU) czy tensorowego (TPU). Natomiast techniki takie jak distributed learning pozwalają na uczenie się na zbiorach danych niemieszczących się na jednej maszynie obliczeniowej.

W ostatnim okresie znacząco wzrósł wolumen kolekcjonowanych danych i decyzji, które muszą zostać przeanalizowane w sposób automatyczny i zrozumiały dla użytkownika. To spowodowało rozwój badań w obszarze uczenia preferencji, które czerpie z obu dziedzin. Pozwalają one na łatwe skalowanie wraz z rosnącą liczbą informacji, zapewniając przy tym możliwość interpretacji modelu.

W ramach niniejszej rozprawy doktorskiej zaproponowano schemat uczenia się parametrów metod na podstawie dużej ilości niespójnych danych referencyjnych. Opracowane techniki rozwiązują problem sortowania, korzystając z procedury opartej na progach rozdzielających klasy. Przedstawiono osiem metod uczenia preferencji w postaci sieci neuronowych, które bazują na wysoce interpretowalnych metodach MCDA. Obejmują one operator OWA, całkę Choquet, addytywną funkcję wartości, odległość od idealnej i antyidealnej opcji oraz, metody bazujące na relacji przewyższania i preferencji istniejącej pomiędzy parami wariantów.

Przykładowo zaproponowane metody ANN-Ch-Pos., ANN-Ch-Constr. oraz ANN-Ch-Uncons. bazują na modelu preferencji w postaci całki Choquet. Ta pierwsza dopuszcza jedynie na pozytywne interakcje między kryteriami oraz dodatnie wagi kryteriów. W pozostałych dwóch możliwe są zarówno interakcje pozytywne, jak i negatywne. W ANN-

Ch-Constr są one jednak ograniczone w taki sposób, aby suma współczynnika interakcji pomiędzy parą kryteriów oraz wagami każdego kryterium z tej pary była nieujemne. Dodatkowo wagi kryteriów również muszą być dodatnie. Natomiast w ANN-Ch-Uncons. nie ma ograniczeń na wartości wag oraz współczynników interakcji. Metody te są reprezentowane jako sieci neuronowe zawierające od jednej do dwóch warstw liniowych specjalnie dostosowanych do wyżej wymienionych wymogów metod.

Bardziej skomplikowane architektury zostały zaproponowane choćby dla metod ANN-UTADIS oraz ANN-PROMETHEE. Ta pierwsza zawiera łącznie pięć warstw, zaś ta druga – sześć. Warstwy te zostały zaprojektowane w taki sposób, aby móc odtworzyć dowolny monotoniczny kształt cząstkowych funkcji wartości lub preferencji. Zaproponowane architektury przestrzegają ograniczeń na monotoniczność kryteriów, co pozwala na uzyskiwanie interpretowalnego modelu preferencji. Ponadto, możliwe jest dostarczanie wyjaśnień decyzji oraz informacji o wpływie poszczególnych kryteriów. Opracowane metody pozwalają uniknąć definiowania hiperparametrów takich jak punkty charakterystyczne czy kształt funkcji preferencji. Zamiast tego zastosowano bardziej ogólne funkcje pozwalające na lepsze dopasowanie się do danych wejściowych, jednocześnie zachowując oryginalną ideę metod MCDA.

Aby efektywnie przetwarzać duże zbiory danych w akceptowalnym czasie, wykorzystane są algorytmy optymalizacji dedykowane dla uczenia głębokiego. Przykładowo, w celu przyspieszenia procesu trenowania i uwolnienia go od zależności od kolejności rozważanych wariantów, zastosowano algorytm Batch Gradient Descent. Następnie zastosowano technikę augmentacji danych poprzez dodawanie szumu gaussowskiego do danych treningowych w każdej epoce uczenia. Dzięki temu uzyskano poprawę odporności modelu na zakłócenia, jego zdolności do generalizacji oraz redukcję nadmiernego dopasowania do danych uczących.

Przedstawione metody prezentują w pełni wyjaśnialny model preferencji, co zostało zilustrowane na przykładzie problemu Employee Rejection / Acceptance. Opracowane modele pozwalają na określenie roli poszczególnych kryteriów oraz podzbiorów kryteriów. Ponadto dostarczają wglądu w to, jak wpływ oceny wariantów na poszczególnych kryteriach wpływa na ostateczną decyzję. Dodatkowo możemy ocenić, które różnice w ocenach są pomijalnie małe, a które są znaczące a nawet krytyczne. Następnie modele umożliwiają określenie, jak silna powinna być koalicja kryteriów, aby można było stwierdzić, że jedna opcja jest co najmniej tak dobra jak inna. W ramach tych metod stosowana jest również łatwa do zrozumienia i przejrzysta procedura sortowania oparta na progach, która umożliwi klasyfikację, porównując całościowe wyniki wariantów z progami rozdzielającymi klasy.

Sprawdzono konkurencyjność rozwiązań zaproponowanych w ramach rozprawy, przeprowadzając szereg eksperymentów. Dotyczą one dziewięciu referencyjnych zbiorów danych, które są typowo wykorzystywane w problemach uczenia preferencji. Zbiory te za-



wierają ponad tysiąc opcji oraz problemy wymagające porównania kilku milionów par wariantów. Przeprowadzono analizę tych zbiorów w celu oceny spójności informacji preferencyjnych. Analizy wykazały, że wszystkie badane zbiory danych zawierały niespójne preferencje, ale liczba tych niespójności znacznie się różniła między różnymi problemami. W celu dostrojenia wartości hiperparametrów dla zaproponowanych metod sprawdzono dla każdego rozważanego problemu, jaki jest optymalny ich zestaw za pomocą eksperymentów wykorzystujących technikę grid search.

Do określenia skuteczności algorytmów, użyto dwóch miar jakości klasyfikacji. Pierwszą jest standardowy błąd klasyfikacji (błąd zero-jedynkowy (0/1)) odnoszący się do liczby wariantów, które model sklasyfikował nieprawidłowo. Drugą jest pole pod wykresem krzywej ROC (ang. Area Under Curve – AUC), ujmująca, ile zmian w rankingu, powstałym na podstawie globalnych ocen, należy dokonać, aby uzyskać w pełni spójne rozwiązanie.

Dodatkowo, przetestowano trzy różne scenariusze rozwiązywania postawionego problemu dla każdego ze zbiorów danych, aby ocenić, jak dobrze różne metody radzą sobie z uogólnianiem wiedzy. Scenariusze te obejmowały niewielką liczbę danych treningowych w porównaniu do danych testowych, równą wielkość obu zbiorów danych oraz sytuację, w której zbiór treningowy był znacząco większy od zbioru testowego.

W ramach tych eksperymentów najlepsze wyniki z zaproponowanych metod pod względem błędu 0/1 zostały osiągnięte dla modeli ANN-UTADIS oraz ANN-Ch-Uncons. Natomiast dla miary AUC dodatkowo wysoką skuteczność wykazała metoda ANN-PROMETHEE. Wysoka jakość dla miary AUC wynika z faktu, iż metody bazujące na relacjach preferencji i przewyższania poprawnie odtwarzają większość relacji między parami wariantów, natomiast gorzej sobie radzą w przypadku klasyfikacji. Biorąc pod uwagę trudność problemów, wynikającą z niespójności informacji preferencyjnej, zaobserwowano, iż wszystkie metody osiągają niższe skuteczności dla problemów bardziej niespójnych. Przy porównaniu jakości metody UTADIS zaproponowanej w tej pracy doktorskiej z metodami opartymi na programowaniu matematycznym, zastosowany w tej pracy sposób znajdowania parametrów metody wykazuje statystycznie lepszą skuteczność dla danych niereferencyjnych niż rozwiązania oparte na programowaniu matematycznym. Ponadto, porównano wariant metody UTADIS zaproponowany w tej pracy z metodami opartymi na programowaniu matematycznym. Zastosowany w tej rozprawie sposób znajdowania parametrów metody wykazuje statystycznie lepszą skuteczność dla danych niereferencyjnych niż pozostałe.

## **Metody wspomaganie decyzji inspirowane uczeniem maszynowym**

Wspólną cechą metod uczenia maszynowego jest możliwość odtwarzania bardzo skomplikowanych przekształceń danych wejściowych w celu uzyskania jak największej skuteczności

ści predykcyjnej. W rzeczywistych sytuacjach mogą istnieć kryteria, dla których nie ma jednoznacznego kierunku preferencji. Często zdarza się, iż istnieje zakres ocen preferowanych, a te powyżej i poniżej są mniej istotne dla użytkownika. Co więcej, wykorzystanie wysoce złożonych przekształceń do odwzorowania danych zmniejsza jego interpretowalność i może prowadzić do przeuczenia się modelu. W związku z powyższym, wiele modeli wykorzystuje techniki regularyzacji, mające na celu ograniczenie złożoności modelu.

Powyższe obserwacje były motywacją do stworzenia dwóch metod modelowania kryteriów niemonotonicznych dla potrzeb addytywnej funkcji wartości. Pierwsza z nich kontroluje złożoność modelu poprzez minimalizację zmian kierunku monotoniczności. Zaproponowano różne typy kryteriów monotonicznych takie jak zysk, koszt, kryteria monotoniczne z obszarem wypłaszczenia preferencji, a także kryteria niemonotoniczne A- i V-kształtne oraz o dowolnym przebiegu. W celu określenia kształtu cząstkowych funkcji wartości wykorzystywane są zmienne binarne, które sterują m. in. ich kierunkiem monotoniczności, normalizacją oraz złożonością. Aby znaleźć parametry modelu, należy rozwiązać dedykowany problem mieszanego całkowitoliczbowego programowania liniowego.

Drugim sposobem modelowania kryteriów niemonotonicznych jest wykorzystania złożenia dwóch komponentów, niemalejącego i nierosnącego. W trakcie optymalizacji możliwe jest wykorzystanie tylko jednego z nich lub ich kombinacji. Pozwala to na zaprezentowanie dowolnej funkcji monotonicznej i niemonotonicznej, dostarczając przy tym wyjaśnienia jej kształtu. Do zamodelowania takiego typu kryterium nie ma potrzeby wykorzystywania zmiennych binarnych, co prowadzi do prostszego problemu programowania liniowego niż w poprzednim sposobie. Jednakże powoduje to możliwość powstawania dowolnie skomplikowanych funkcji, jeśli będzie tego wymagała złożoność rozwiązywanego problemu.

Wynikiem rozwiązania problemu programowania matematycznego jest pojedyncza reprezentatywna instancja. Taki model dostarcza jednoznacznych przypisań do klas wraz z uzasadnieniem wpływu każdej oceny na wynikową decyzję. Dodatkowo pozwala on na analizę i interpretację modelu, dostarczając informacji, jakie wartości musiałyby ulec zmianie tak, aby klasyfikacja również się zmieniła.

Przypisania uzyskane przy użyciu reprezentatywnego modelu są konfrontowane z wynikami analizy odporności. Pozwala ona sprawdzić, jak zmienia się rekomendacja dla spójnych instancji modelu, gdy złożoność modelu jest ograniczona do minimalnej możliwej wartości. Wyniki tej analizy przyjmują postać możliwych przypisań wariantów niereferencyjnych do klas. Oznaczają one zbiór klas, do których dany wariant może być przypisany przez co najmniej jedną spójną instancję modelu sortowania. Możliwość takiego przypisania sprawdza się poprzez rozwiązanie oryginalnego problemu programowania matematycznego wraz z dodatkowymi ograniczeniami, wymuszającymi minimalną możliwą złożoność oraz przydziału wariantu do konkretnej klasy przez założoną metodę sorto-

wania. Jeśli istnieje rozwiązanie dla tak sformułowanego problemu, oznacza to, że dany wariant może być przypisany do określonej klasy.

W niniejszej rozprawie doktorskiej został przedstawiony także nowy problem sortowania z wieloma powiązаныmi ze sobą decyzjami. W tym problemie każdy wariant jest oceniany pod względem wielu atrybutów decyzyjnych, które obejmują klasy uporządkowane według preferencji. Decydent przypisuje zbiór wariantów referencyjnych do klas na każdym atrybucie decyzyjnym. Klasy te oznaczają poziom jakości lub ryzyka na wcześniej zdefiniowanej skali dla wszystkich decyzji. Problem ten jest zainspirowany problemami klasyfikacji wieloetykietowej, polegającymi na przypisaniu do obiektu podzbioru etykiet.

Zaproponowany dla tego problemu sposób rozwiązania polega na zbudowaniu zbioru powiązanych ze sobą modeli preferencji po jednym dla każdej decyzji. Wykorzystują one zbiór ograniczeń wewnątrz każdej decyzji oraz ograniczenia łączące modele dla różnych decyzji. Te pierwsze zapewniają odpowiednie relacje pomiędzy wartościami wariantów wykorzystywanych do klasyfikacji na pojedynczym atrybucie decyzyjnym. Natomiast te drugie odpowiadają relacjom między całkowitymi wartościami tego samego wariantu dla różnych decyzji.

Użyteczność zaproponowanych metod zademonstrowano na przykładzie rzeczywistego problemu dotyczącego zarządzania ryzykiem podczas produkcji i przetwarzania nanomateriałów. Rozważanymi wariantami były różne scenariusze ekspozycji na nanomateriał, dla których należało zdecydować, w jakim stopniu jest wymagany dany środek ostrożności. Zostały wzięte pod uwagę dwa przypadki. W pierwszym z nich skoncentrowano się na problemie sortowania związanych z noszeniem maski oddechowej, gdzie cząstkowe funkcje wartości powinny być jak najmniej złożone. Model reprezentatywny wskazał, iż kryteria, które dotyczą limitów wykrywania nanomateriału, ich zdolności do przenoszenia przez powietrze oraz czas ekspozycji mają na większy udział w użyteczności globalnej wariantów natomiast kryteria mówiące o ilości nanomateriału, częstotliwość ekspozycji oraz kontroli inżynierskiej miały niewielki wpływ na wymagalność maski oddechowej.

W drugim przypadku uwzględniono również decyzje odnośnie używania wyciągu laboratoryjnego z oraz bez filtra HEPA oraz używania odkurzacza z filtrem HEPA. W tym wypadku kryteria niemonotoniczne zostały zaprezentowane jako złożenie monotonicznych składowych. Kryteria mówiące o zdolności do przenoszenia przez powietrze, limitów wykrywania nanomateriału oraz czasie ekspozycji na niego miały największy wpływ na klasyfikację. Uzyskane cząstkowe funkcje wartości były podobne dla wszystkich atrybutów decyzyjnych w szczególności dla tych, w których wykorzystywany jest filtr HEPA. Najbardziej różniły się dla decyzji dotyczących używania wyciągu laboratoryjnego z oraz bez filtra HEPA potwierdzając ich komplementarność.

## Metody wspomaganie decyzji inspirowane eksploracją zasobów Internetu

Podczas podejmowania decyzji często nie ocenia się każdej z opcji niezależnie, a bierze się pod uwagę, jak dobra jest ona w zestawieniu z innymi. Zazwyczaj odbywa się to poprzez porównywanie wariantów parami i badanie relacji występujących między nimi, a następnie zagregowanie tych przesłanek do ostatecznego rozwiązania problemu. W MCDA powstało wiele metod wykorzystujących tę ideę. W tym kontekście najbardziej popularne są rodziny metod ELECTRE oraz PROMETHEE, bazujące odpowiednio na relacji przewyższania oraz preferencji.

Relacje te można podzielić na dwie grupy, wartościowaną (rozmytą) oraz binarną. Zbiór wszystkich relacji pomiędzy wariantami można przedstawić w postaci grafu skierowanego, gdzie wierzchołkami są warianty, a łuki odpowiadają relacjom. W rzeczywistych przypadkach zarówno relację przewyższania, jak i preferencji rzadko bezpośrednio wskazują jakiś wariant jako najlepszy lub pozwalają na uszeregowanie wariantów w jednoznaczny sposób, spełniający własności porządku. Z tego powodu konieczne jest wykorzystanie dodatkowych technik eksploatujących te relacje w celu uzyskania rekomendacji wariantów najbardziej preferowanych lub rankingu.

Istniejące metody eksploatacji relacji uznają wszystkie warianty za jednakowo istotne i bycie lepszym niż relatywnie słaby wariant jest uwzględniane w takim samym stopniu jak przewyższanie wariantu dobrego czy trudnego do przewyższania. Dodatkowo, techniki te nie pozwalają na kontrolę ich wyniku poprzez zastosowanie pośrednich preferencji. W tym celu należy dokonać zmian we wcześniejszych etapach wspomaganie decyzji.

Te spostrzeżenia były motywacją dla zaproponowania dwóch rodziny metod, PrefRank oraz ScoreBin, służących do analizy różnych typów relacji. Były one inspirowane zarówno metodą Net Flow Score, która uwzględnia zarówno siły i słabości wariantów oraz algorytmami analizy grafów, zaproponowanych oryginalnie w ramach eksploracji zasobów Internetu.

Metody PrefRank służą do analizy wartościowanej (rozmytej) relacji preferencji, zaś ScoreBin do przetwarzania binarnej relacji przewyższania. Poszczególne metody w ramach obu tych rodzin różnią się schematem ważenia podczas agregacji porównań parami. Pozwalają one ująć różne aspekty wariantu takie jak trudność i łatwość w przewyższaniu lub preferencji grafie. PrefRank I (ScoreBin I) jest inspirowany metodą PageRank (TrustTank) i uznaje wariant za silny, jeżeli jest on preferowany nad (przewyższa) inne silne opcje. Drugi wariant obu tych rodzin wzoruje się na algorytmie HITS i uznaje, że wariant jest dobry, jeżeli jest preferowany nad (przewyższa) wiele opcji słabych. W PrefRank III (ScoreBin III), którego inspiracją jest metoda Salsa, wariant jest silny, jeżeli jest on preferowany nad (przewyższa) takie opcje, które są gorsze od innych silnych opcji. Ostatecznie Scorbin IV przyjmuje, że wariant jest dobry, jeżeli przewyższa warianty, które trudno jest

przewyższyc.

Dodatkowo rodzina metod ScoreBin uwzględnia opcjonalną informację preferencyjną pozwalającą na podanie przez decydenta podzbioru wariantów silnych i słabych. Taka informacja przekłada się na wartość bonusu lub kary wariantu, którego dotyczy oraz wpływa również na inne warianty poprzez zależności w grafie przewyższania. Oprócz tego decydent może zdecydować o wielkości minimalnego wpływu, jaki warianty będą miały na siły i słabości innych wariantów. Wartości tych parametrów wpływają na ranking wariantów, pozwalając decydentowi na większą kontrolę nad ostatecznym wynikiem. Z tego powodu zaproponowano metodę analizy odporności, sprawdzającą możliwe pozycje wariantów w rankingu w zależności od wartości bonusów lub kar. Wykorzystuje ona symulacje Monte Carlo do określenia procentu instancji, w których wariant był na określonej pozycji w rankingu. Dodatkowo analiza ta dostarcza informacji o tym, jakie przedziały wartości parametrów powinny zostać wybrane, aby wariant znalazł się na danej pozycji w rankingu

Zaproponowane metody zostały porównane pod względem podobieństwa zwracanego przez nie wyniku z innymi metodami eksploatacji relacji takimi jak NFS, ELECTRE I oraz Qualification Distillation, która bazuje na procedurze destylacji znanej z ELECTRE III. Analiza ta została wykonana dla różnej wielkości symulowanych problemów. Wyniki pokazały, że rankingi w przypadku metod PrefRank oraz NFS są do siebie bardzo podobne. Natomiast w przypadku metod eksploatacji relacji przewyższania najbardziej podobne do siebie były metody bazujące na podobnych koncepcjach czyli ScoreBin I z IV, ScoreBin II z III oraz NFS z Qualification Distillation. Natomiast rekomendacje uzyskiwane dla ELECTRE I znacząco różniły się od pozostałych metod.

Obie rodziny metod zostały również przetestowane w rzeczywistych problemach. PrefRank zastosowano do oceny specjalnych stref ekonomicznych w Polsce. Wszystkie warianty tej metody wskazały, że strefa Kostrzyn i Słubice jest najbardziej preferowana pod względem wzrostu finansowego i tworzenia nowych miejsc pracy. Metody ScoreBin zostały zastosowane do identyfikacji najlepiej zarządzanego parku technologicznego w Polsce, który przynosi największe zyski oraz wspiera rozwój przemysłu, oraz badań.

## Podsumowanie

Niniejsza rozprawa doktorska dotyczy nowych metod wielokryteriowego wspomaganie decyzji, które są inspirowane innymi poddyscyplinami sztucznej inteligencji. Określono trzy główne obszary badawcze związane z metodami, łączącymi MCDA z uczeniem maszynowym, głębokimi sieciami neuronowymi oraz eksploracją zasobów Internetu. Efektem tych badań było powstanie pięciu oryginalnych publikacji. Prace te wykazały, iż wykorzystanie technik z różnych obszarów może pozwolić na tworzenie nowych metod radzących sobie z coraz większymi i bardziej złożonymi problemami decyzyjnymi.

Na główny wkład tej rozprawy składa się kilka elementów. Po pierwsze, zaproponowano algorytmy uczenia preferencji do znajdowania wartości parametrów wybranych metod MCDA na podstawie dużego, wysoce niespójnego zbioru przykładowych decyzji. Modele te zostały zaimplementowane w formie wysoce interpretowalnych sieci neuronowych, które rozwiązują problem sortowania. Dodatkową zaletą tego rozwiązania jest to, iż model jest w stanie lepiej dostosować cząstkowe funkcje metod do danych bez konieczności ich arbitralnego definiowania przez decydenta.

Po drugie, dla modelu preferencji w postaci addytywnej funkcji wartości, przedstawiono dwa nowe sposoby modelowania kryteriów niemonotonicznych o dowolnym kształcie. Pierwszy z nich pozwala na kontrolowanie złożoności funkcji poprzez minimalizację liczby zmian monotoniczności. Zaproponowano sposób przedstawienia różnych typów kryteriów, zarówno monotonicznych, jak i niemonotonicznych w postaci ograniczeń problemu mieszanego całkowitoliczbowego programowania matematycznego. Natomiast drugi sposób zapewnia interpretowalność funkcji niemonotonicznej poprzez jej rozkład na dwie monotoniczne składowe. Dzięki temu możliwe jest zamodelowanie dowolnego kształtu bez ograniczeń na jego złożoność.

Następnie zaproponowano sposób modelowania problemu sortowania z wieloma wzajemnie powiązаныmi atrybutami decyzyjnymi. Problem ten polega na przypisaniu wariantu do jednej z wcześniej zdefiniowanych klas dla każdego atrybutu decyzyjnego. Zaproponowany model konstruuje osobne modele sortowania dla każdego atrybutu decyzyjnego, uwzględniając zarówno zależności wewnątrz decyzji, jak i między nimi. W rezultacie pozwala on na dobranie dla każdego wariantu najbardziej adekwatnej kombinacji decyzji.

Kolejnym elementem rozprawy były dwie nowe rodziny eksploatacji (PrefRank oraz ScoreBin) rozmytej i binarnej relacji preferencji lub przewyższania. Rozwiązują one problem rankingu oraz wyboru, wyznaczając siły i słabości wariantów. Bazują przy tym na algorytmach oceniających strony internetowe na podstawie hiperłączy. W ramach każdej z rodzin zaproponowano kilka wariantów różniących się od siebie wagami, jakie przypisują alternatywom, definiując ich wkład w mocne i słabe strony opcji, z którymi są powiązane. Dodatkowo został zaproponowany sposób uwzględnienia holistycznej informacji preferencyjnej dotyczącej tego, czy dana opcja jest uznana za silną lub słabą. Pozwala to na podwyższenie lub obniżenie jakości wybranego podzbiory wariantów, a przez ich relacje z pozostałymi wariantami także na wywarcie wpływu na osiągnięte przez nie wyniki.

W ramach rozprawy doktorskiej przeprowadzono eksperymenty mające na celu ocenę jakości zaproponowanych rozwiązań oraz porównanie ich wyników z innymi metodami powszechnie stosowanymi w rozważanych problemach. W szczególności, konkurencyjność zaproponowanych rozwiązań uczenia preferencji została przetestowana na dziewięciu zbiorach danych różniących się zarówno liczbą wariantów, jak i ich trudnością. W ramach eksperymentów wykazano, iż niektóre zaproponowane metody osiągają wyższą skutecz-

ność predykcji pod względem miary AUC i błędu 0/1 niż zaproponowane wcześniej metody uczenia preferencji.

Następnie użyteczność metod modelowania niemonotonicznych kryteriów zostały przedstawione na przykładzie problemu analizy ryzyka podczas produkcji nanomateriałów. Rozważono ten problem w dwóch scenariuszach rozważając pojedynczą decyzję oraz wiele atrybutów decyzyjnych. W opinii specjalistów dziedzinowych zaangażowanych w te zastosowania zaproponowane metody pozwoliły na znalezienie satysfakcjonujących rozwiązań.

Wreszcie metody eksploatacji relacji preferencji i przewyższania zostały przetestowane pod względem podobieństwa wyników z innymi metodami eksploatacji. Dodatkowo przydatność tych metod została zaprezentowana na dwóch rzeczywistych problemach oceny specjalnych stref ekonomicznych oraz parków technologicznych w Polsce.

Przedstawione w tej rozprawie doktorskiej badania mogą stanowić punkt wyjścia do potencjalnych przyszłych badań. Po pierwsze, chociaż rozważane problemy były już znacznej wielkości, obejmujące tysiące wariantów lub miliony porównań parami, w wielu obszarach uczenia maszynowego rozważa się znacznie większe problemy. Warto byłoby więc przetestować zaproponowane rozwiązania dla problemów o rozmiarach typowych dla takich dziedzin zastosowań.

Po drugie, wykorzystanie głębokich sieci neuronowych daje możliwości przetestowania skuteczności wielu technik zaproponowanych w tymże obszarze. Metody takie jak transfer learning, active learning, federated learning lub blockchain mogą znaleźć zastosowanie w problemach decyzyjnych, związanych z wieloma powiązаныmi decyzjami lub decyzjami grupowymi. Możliwe jest też opracowanie metod uczenia preferencji z wykorzystaniem sieci neuronowych, bazując na innych metodach MCDA, choćby stosujących profile charakterystyczne lub graniczne do zdefiniowania klas.

Zarówno metody oparte na sieciach neuronowych, jak i zaproponowane sposoby modelowania kryteriów niemonotonicznych mogłyby być wykorzystane w problemach rankingu. W tym celu informacja preferencyjna przyjmowałaby postać porównań parami wariantów i nie byłoby konieczności stosowania sortowania opartego na progach. Rozważając modele kontrolujące stopień złożoności kryteriów niemonotonicznych, istnieje możliwość rozszerzenia ich na transformacje wielomianowe, które są istotne w rzeczywistych problemach.

Wreszcie w metodach PrefRank i ScoreBin obliczanie wartości sił i słabości odbywa się w podobny sposób. Możliwe jest jednak wykorzystywanie różnych kombinacji tych metod jednocześnie i zbadanie różnych typów agregowania takich wyników. Podobnie w przypadku metod inspirowanych sieciami neuronowymi możliwe byłoby połączenie wielu architektur w jedną i agregowanie wyników do jednego miary jakości. Ostateczne decyzje mogłyby być podejmowane na podstawie głosowania większościowego lub ważonego, gdzie wagi byłyby ustalane w trakcie procesu uczenia.





# Declarations

**Marco Cinelli**

Faculty Governance and Global Affairs  
Leiden University College  
Leiden, The Netherlands

**Declaration**

I hereby declare the following contribution as an author of the paper:

Kadziński, M., Martyn, K., Cinelli, M., Słowiński, R., Corrente, S., & Greco, S. (2020). Preference disaggregation for multiple criteria sorting with partial monotonicity constraints: Application to exposure management of nanomaterials. *International Journal of Approximate Reasoning*, 117, 60-80.

- Co-authorship of the concept of applying multiple criteria value model for exposure management of engineered nanomaterials;
- Collection of data for the concerned case study (Section 3);
- Co-authorship of the text of the publication in the application-oriented parts.

Kadziński, M., Martyn, K., Cinelli, M., Słowiński, R., Corrente, S., & Greco, S. (2021). Preference disaggregation method for value-based multi-decision sorting problems with a real-world application in nanotechnology. *Knowledge-Based Systems*, 218, 106879.

- Co-authorship of the concept of applying multiple criteria value model for multi-decision sorting in the context of exposure management of engineered nanomaterials;
- Collection of data for the concerned case study (Section 4);
- Co-authorship of the text of the publication in the application-oriented parts.



Marco Cinelli, PhD

Poznań, 31 maja 2023 r.

**Roman Słowiński**  
Instytut Informatyki  
Politechnika Poznańska  
ul. Piotrowo 2  
60-965 Poznań

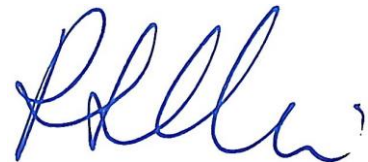
### **OŚWIADCZENIE DOTYCZĄCE WKŁADU W POWSTANIE PRACY**

Kadziński, M., Martyn, K., Cinelli, M., Słowiński, R., Corrente, S., & Greco, S. (2020). Preference disaggregation for multiple criteria sorting with partial monotonicity constraints: Application to exposure management of nanomaterials. *International Journal of Approximate Reasoning*, 117, 60-80.

- **Conceptualization, Methodology:** współautor koncepcji modelowania kryteriów niemonotonicznych z minimalizacją liczby zmian monotoniczności oraz wykorzystującej ją metody dezagregacji preferencji (Rozdział 2.3);
- **Writing - review & editing:** poprawa wstępnych wersji artykułu oraz pomoc przy opracowaniu odpowiedzi na uwagi zgłaszane przez recenzentów.

Kadziński, M., Martyn, K., Cinelli, M., Słowiński, R., Corrente, S., & Greco, S. (2021). Preference disaggregation method for value-based multi-decision sorting problems with a real-world application in nanotechnology. *Knowledge-Based Systems*, 218, 106879.

- **Conceptualization, Methodology:** współautor koncepcji modelowania kryteriów niemonotonicznych jako złożenie dwóch składowych niemalejącej i nierosnącej oraz wykorzystującej ją metody dezagregacji preferencji (Rozdział 2.1);
- **Writing - review & editing:** poprawa wstępnych wersji artykułu oraz pomoc przy opracowaniu odpowiedzi na uwagi zgłaszane przez recenzentów.



prof. dr hab. inż. Roman Słowiński

Catania, May 31, 2023

**Salvatore Corrente**

Department of Economics and Business  
University of Catania  
Catania, Italy

**Declaration**

I hereby declare the following contribution as an author of the paper:

Kadziński, M., Martyn, K., Cinelli, M., Słowiński, R., Corrente, S., & Greco, S. (2020). Preference disaggregation for multiple criteria sorting with partial monotonicity constraints: Application to exposure management of nanomaterials. *International Journal of Approximate Reasoning*, 117, 60-80.

- **Conceptualization, Methodology:** Co-authorship of the idea underlying the paper in the scope of considering various types of non-monotonicity, conducting robustness analysis, and formulating the proof on the “no jump property” (Section 2.4.2)
- **Writing - review & editing** of the manuscript.

Kadziński, M., Martyn, K., Cinelli, M., Słowiński, R., Corrente, S., & Greco, S. (2021). Preference disaggregation method for value-based multi-decision sorting problems with a real-world application in nanotechnology. *Knowledge-Based Systems*, 218, 106879.

- **Conceptualization, Methodology:** Co-authorship of the idea underlying the paper in the scope of modeling non-monotonicity as a sum of gain- and cost-type components;
- **Writing - review & editing** of the manuscript.

Salvatore Corrente, PhD



Catania, May 31, 2023

**Salvatore Greco**

Department of Economics and Business  
University of Catania  
Catania, Italy

### Declaration

I hereby declare the following contribution as an author of the paper:

Kadziński, M., Martyn, K., Cinelli, M., Słowiński, R., Corrente, S., & Greco, S. (2020). Preference disaggregation for multiple criteria sorting with partial monotonicity constraints: Application to exposure management of nanomaterials. *International Journal of Approximate Reasoning*, 117, 60-80.

- **Conceptualization, Methodology:** Co-authorship of the idea underlying the paper in the scope of considering various types of non-monotonicity, conducting robustness analysis, and formulating the proof on the “no jump property” (Section 2.4.2)
- **Writing - review & editing** of the manuscript.

Kadziński, M., Martyn, K., Cinelli, M., Słowiński, R., Corrente, S., & Greco, S. (2021). Preference disaggregation method for value-based multi-decision sorting problems with a real-world application in nanotechnology. *Knowledge-Based Systems*, 218, 106879.

- **Conceptualization, Methodology:** Co-authorship of the idea underlying the paper in the scope of modeling non-monotonicity as a sum of gain- and cost-type components;
- **Writing - review & editing** of the manuscript.



Salvatore Greco

Poznań, 31 maja 2023 r.

**Magdalena Martyn**  
Instytut Informatyki  
Politechnika Poznańska  
ul. Piotrowo 2  
60-965 Poznań

### **OŚWIADCZENIE DOTYCZĄCE WKŁADU W POWSTANIE PRACY**

K. Martyn, M. Martyn, and M. Kadziński. PrefRank: a family of multiple criteria decision analysis methods for exploiting a valued preference relation.

- **Conceptualization, Methodology:** współautorka koncepcji wykorzystania technik eksploracji internetu do eksploatacji relacji preferencji (Rozdział 3);
- **Writing - review & editing:** poprawa wstępnych wersji artykułu

K. Martyn, M. Martyn, and M. Kadziński. ScoreBin: scoring alternatives based on a crisp outranking relation with an application to the assessment of Polish technological parks.

- **Conceptualization, Methodology:** współautorka koncepcji wykorzystania technik eksploracji internetu do eksploatacji relacji przewyższania (Rozdział 3);
- **Writing - review & editing:** poprawa wstępnych wersji artykułu

Magdalena Martyn







© 2023 Krzysztof Martyn

Poznan University of Technology  
Faculty of Computing and Telecommunications  
Institute of Computing Science  
Typeset using L<sup>A</sup>T<sub>E</sub>X in Computer Modern.

Bib<sub>T</sub>E<sub>X</sub>:

```
@phdthesis{ Martyn2023,  
  author = "Krzysztof Martyn",  
  title = "{MCDA methods inspired by other subdisciplines of artificial  
intelligence}",  
  school = "Poznan University of Technology",  
  address = "Pozna{\n}, Poland",  
  year = "2023",  
}
```