Construct your own ELECTRE method

Miłosz Kadziński*[,a], Tomasz Mieszkowski[a], Michał Tomczyk[a], Sébastien Bigaret[b,c]

[a] *Institute of Computing Science, Poznań University of Technology, Piotrowo 2, 60-965 Poznań, Poland*
[b] *Institut Mines-Télécom, Télécom Bretagne, UMR CNRS 6285 Lab-STICC, Technopôle Brest Iroise, CS 83818, 29238 Brest Cedex 3, France*
[c] *Université Européenne de Bretagne*

Abstract

We postulate flexibility in constructing ELECTRE decision aiding methods so that they can be better suited for real-world decision problems, and implement this paradigm in practice. We provide a wide spectrum of elementary ELECTRE-based components that are able to interoperate, and make them available via the *diviz* platform. At the stage of construction of an outranking relation, we consider a variety of procedures for carrying out the concordance and (non-)discordance tests, computing the credibility degree, and checking the validity of a crisp relation. At the stage of its exploitation, we refer to several choice-, ranking-, and sorting-specific algorithms. By coupling together the ELECTRE-based construction and exploitation procedures, an analyst can reconstruct the existing methods, or develop her/his own ELECTRE without any mathematical or programming skills. For this reason, (s)he needs to combine the provided modules in one of several hundred ways that are discussed in the paper. The proposed approach is demonstrated with application of a few previously not considered variants of ELECTRE to evaluation of mass transit systems in the selected European cities and a set of products within a storage location assignment system.

Key words: Decision Analysis, Multiple criteria, Outranking methods, ELECTRE, Preference aggregation, *diviz*

1. Introduction

Over the last fifty years a significant body of research has demonstrated the benefits of using ELECTRE methods in real-world decision aiding [12, 13]. The usefulness of this family of Multiple Criteria Decision Aiding (MCDA) approaches comes from the fact that they employ an outranking-based preference model. Its main advantages consist in the non-compensatory character of the aggregation of multiple criteria, tolerating the imperfect knowledge of data, and operating on heterogeneous scales. Moreover, it has the abilities of dealing with both qualitative and quantitative performances as well as representing the situations of weak preference and incomparability (apart from the traditionally considered indifference and strict preference). An outranking relation is also appreciated for implementing an analogy to voting procedures by taking into account the reasons for and against an outranking [17].

The interpretability, faithfulness, flexibility, and descriptive character of an outranking model are essential in MCDA, since they encourage the participation of the Decision Makers (DMs) in the decision process

*Corresponding author: Institute of Computing Science, Poznań University of Technology, Piotrowo 2, 60-965 Poznań, Poland. Tel. +48-61 665 3022.

Email addresses: milosz.kadzinski@cs.put.poznan.pl (Miłosz Kadziński), t.mieszkowski@fastmail.fm (Tomasz Mieszkowski), michal.k.tomczyk@gmail.com (Michał Tomczyk), sebastien.bigaret@telecom-bretagne.eu (Sébastien Bigaret)

and lead to better understanding of the proposed recommendation [8]. Indeed, the ELECTRE methods had a considerable impact on human decisions in a variety of real-world multiple criteria problems. The application fields include finance, medicine, energy planning, military, project selection, agriculture, and environmental management (for a review, see [12, 13]).

All ELECTRE methods are based on the same rule: they first construct an outranking relation for all ordered pairs of compared objects, and then exploit this relation to deliver recommendation in function of the specific problem to solve. Several approaches have been designed for the three main types of multiple criteria problems: choice (e.g., ELECTRE I, Iv, and Is [4, 23, 25, 28, 30]), ranking (e.g., ELECTRE II, III, and IV [24, 30]) and sorting (ELECTRE Tri-B, Tri-C, and Tri-rC [2, 19, 39]). In fact, these methods are sequences of the elementary well-defined steps which contribute to some particular implementation of the construction and exploitation phases. When it comes to formalizing the general concepts of concordance and discordance, none of these approaches tolerates the freedom in their interpretation. As a result, each ELECTRE method is distinguished not only by its unique exploitation procedure, but also by the way it constructs an outranking relation. While these two phases are independent, paradoxically, a new approach can be obtained each time when coupling together the existing construction and exploitation procedures that have not been yet considered within a common methodological framework. Such developments are still considered valuable, because they are mainly application-driven, thus, allowing to deal with a specific multiple criteria problem at hand (see, e.g., [5]).

When it comes to the use of ELECTRE for practical decision aiding, it is seriously limited by the existing software. Firstly, the programs for ELECTRE Is [1], ELECTRE III/IV [36], and ELECTRE Tri-B [22] have been implemented in the previous century. Due to a rapid development of computer technologies, they fail to work correctly on the present-day operating systems. Secondly, several approaches that have been suggested in the literature (e.g., the effects of reinforced preference and counter-veto, or accounting for the interactions between criteria) have been implemented on an ad hoc basis just to illustrate the proposed concepts or to demonstrate the salient points of the propose methodology [38]. Although potentially useful, they have never been employed in the context of any real-world decision problem. Thirdly, while most software developed by academics is available free of charge [18], commercial packages implementing some ELECTRE-based procedures (e.g., DecisionCloud) cost hundreds of euro. Finally, the existing MCDA tools poorly expose the underlying sequential character of the method, sticking to a rather univocal implementation of the underlying elementary steps. The sole exception in this regard is the *diviz* software [20].

diviz is an open-source tool which allows to build complex MCDA algorithmic workflows from the elementary components. It postulates implementation of basic computation procedures as separate software pieces, which, if properly chained, would rebuild the original methods [20]. The main benefits of this proposal consist in removing the black box effect of certain software, avoiding repeated and unnecessary reimplementation of the same algorithms, and easily testing their variations. While the underlying idea is appealing, the selection of ELECTRE specific modules available on *diviz* [7] is very limited.

The contribution of this paper is of both methodological and software nature. It consists in postulating flexibility in constructing ELECTRE methods so that they are well suited for dealing with the specific real-world decision problems, and implementing this paradigm in practice. For this reason, we have designed a wide spectrum of elementary ELECTRE-based components that are able to interoperate, and make them available via the *diviz* platform. The proposed methodological bricks are useful for designing advanced approaches and supporting the analysts in both problem structuring and preference elicitation process.

At the stage of construction of an outranking relation, we consider a variety of procedures for carrying out the concordance and (non-)discordance tests, computing the credibility of an outranking relation,

and checking the validity of a crisp relation. These are derived, e.g., from ELECTRE Is, Tri-B, III, IV, and MR-Sort. We also account for the concepts which are not linked to any specific approach, such as modeling interactions between criteria, the effects of reinforced preference and counter-veto, using pre-veto (discordance) thresholds, or numerous procedures for aggregating concordance and discordance degrees into a valued or crisp outranking relation. We ensure universality of the implemented modules so that they admit comparison of alternatives either with each other or with class profiles (boundary or characteristic ones). In this way, we provide means for constructing an outranking relation that may be subsequently exploited to derive choice, ranking, or sorting recommendation.

At the stage of exploitation of an outranking relation, we consider the following approaches: algorithms for finding the graph kernel as in the ELECTRE I methods, distillation and ranking procedures of ELECTRE III/IV, Net Flow Score rules for exploiting valued and crisp outranking relation, and assignment rules of ELECTRE Tri-B, ELECTRE Tri-C, and ELECTRE Tri-rC. Taking advantage of such components, the user may construct her/his own ELECTRE in a few minutes without any mathematical or programming skills. This process boils down to combining the modules in one of several hundred ways that are possible with our proposal and discussed in the paper.

The remainder of this paper is organized in as follows. Section 2 is devoted to different ways of constructing an outranking relation. Section 3 is focused on procedures for exploiting this relation. In both sections, apart from reviewing the methodological advances, we present the functionality of the underlying software modules we have implemented. Section 4 provides a general view on how to construct one's own ELECTRE using *diviz*. Section 5 is devoted to the illustrative case studies which demonstrate the application of a few variants of ELECTRE that have never been considered before. Section 6 concludes the paper.

2. Construction of an Outranking Relation

We use the following notation:

- $A = \{a_1, a_2, \ldots, a_i, \ldots\}$ - a set of decision alternatives;
- $F = \{g_1, \ldots, g_j, \ldots, g_m\}$ - a consistent family of n criteria; without loss of generality, we assume both $g_j : A \to \mathbb{R}$ and that all criteria are maximized; $J = \{1, 2, \ldots, m\}$;
- $C_1, \ldots, C_h, \ldots, C_t$ with $t \geqslant 2$ - a set of pre-defined completely ordered (from the worst to the best) classes so that C_{h+1} is preferred to C_h, $h = 1, \ldots, t-1$;
- $P_C = \{p_1^c, \ldots, p_t^c\}$ - a set of characteristic reference profiles defining the most typical performance vector for each class C_h, $h = 1, \ldots, t$;
- $P_B = \{p_0^b, \ldots, p_t^b\}$ - a set of boundary profiles defining the limits between consecutive classes so that p_h^b is the upper limit of class C_h and the lower limit of class C_{h+1}, for $h = 1, \ldots, t$.

In this section, we review different ways of constructing an outranking relation. This construction is based on the concepts of concordance and non-discordance tests. Their role is to consider, respectively, the reasons for and the reasons against an outranking of one object (alternative or profile) over another. Then, their indications are aggregated into a valued outranking relation. We refer to various proposals on how this aggregation may be conducted. Finally, we discuss how to construct a crisp outranking relation based on the valued one.

At the same time, we present the functionality of the software modules we have implemented. They are accessible through the *diviz* platform. Some modules are responsible for conducting a well-defined

computation procedure with a single outcome, others can be parameterized to calculate result of the same type in slightly different ways, whereas yet another modules return multiple results. The modules are numbered from M1 to M11 so that they can be easily referred in Section 4.

2.1. Concordance Test

We consider each criterion g_j as a pseudo-criterion [33] that models per-criterion attractiveness with indifference $q_j(\cdot)$ and preference $p_j(\cdot)$ threshold functions [26]. The latter are most often defined as affine functions:

$$q_j(g_j(a)) = \alpha_{q_j} \cdot g_j(a) + \beta_{q_j} \text{ and } p_j(g_j(a)) = \alpha_{p_j} \cdot g_j(a) + \beta_{p_j}. \tag{1}$$

When comparing $g_j(a)$ and $g_j(b)$, interpretation of the values provided by $q_j(\cdot)$ and $p_j(\cdot)$ is the following:

- $q_j(g_j(b))$ is the maximal difference in performances for which the indifference relation (aI_jb) between a and b holds on criterion g_j; thus, if $|g_j(a) - g_j(b)|$ is neither greater than $q_j(g_j(a))$ nor greater than $q_j(g_j(b))$, this represents a non-significant advantage of a over b, and vice versa; let us denote the set of criteria for which aI_jb as:

$$F^I(a,b) = \{j : aI_jb \Leftrightarrow -q_j(g_j(a)) \leq g_j(a) - g_j(b) \leq q_j(g_j(b))\}; \tag{2}$$

- $p_j(g_j(b))$ is the minimal difference in performances for which the strict preference relation (aP_jb) between a and b holds on criterion g_j; thus, if $g_j(a) - g_j(b)$ is greater than $p_j(g_j(b))$, this represents a significant advantage of a over b; the set of criteria for which aP_jb is denoted with:

$$F^P(a,b) = \{j : aP_jb \Leftrightarrow g_j(a) - g_j(b) > p_j(g_j(b))\}. \tag{3}$$

Note that $q_j(g_j(b)) < g_j(a) - g_j(b) \leq p_j(g_j(b))$ represents an ambiguity zone between indifference and strict preference. In this case, a is weakly preferred to b (aQ_jb). The set of criteria for which aQ_jb is denoted with:

$$F^Q(a,b) = \{j : aQ_jb \Leftrightarrow q_j(g_j(b)) < g_j(a) - g_j(b) \leq p_j(g_j(b))\}. \tag{4}$$

Thus, by indicating which performance difference is negligible or significant, $q_j(\cdot)$ and $p_j(\cdot)$ allow accounting for the imperfect character of data [2, 29].

Marginal concordance. The marginal concordance index $c_j(a,b) \in [0,1]$ represents a degree to which g_j supports the hypothesis about outranking of a over b (aSb). If a is indifferent to b or a is either weakly or strictly preferred to b on g_j, then g_j is in favor of the assertion aSb with no reservation, and, thus, $c_j(a,b) = 1$. Let us denote an outranking relation on g_j by $S_j = I_j \cup Q_j \cup P_j$, and the corresponding criteria for which S_j holds with:

$$F^S(a,b) = \{j : aS_jb\} = F^I(a,b) \cup F^Q(a,b) \cup F^P(a,b). \tag{5}$$

If b is strictly preferred to a, then g_j is in the opposition to the assertion aSb, and, thus, $c_j(a,b) = 0$. Finally, if b is weakly preferred to a, g_j hesitates between the indifference and the opposition. Then, $c_j(a,b) \in (0,1)$ indicates which of these two options is prevailing. The marginal concordance index $c_j(a,b)$

is defined in the following way:

$$c_j(a,b) = \begin{cases} 1 & \text{if } g_j(b) - g_j(a) \leq q_j(g_j(a)), \\ 0 & \text{if } g_j(b) - g_j(a) > p_j(g_j(a)), \\ \frac{[p_j(g_j(a)) - (g_j(b) - g_j(a))]}{[p_j(g_j(a)) - q_j(g_j(a))]} & \text{if } q_j(g_j(a)) < g_j(b) - g_j(a) \leq p_j(g_j(a)). \end{cases} \tag{6}$$

Although we consider the indifference and preference thresholds as variables, they can be defined as constant values (in this case, $\alpha_{q_j} = \alpha_{p_j} = 0$ and $\beta_{p_j} \geq \beta_{q_j} \geq 0$). Remark that $q_j(g_j(a))$ can be equal to zero and/or equal to $p_j(g_j(a))$ for all a. If $p_j(g_j(a)) = q_j(g_j(a)) = 0$, g_j is a true-criterion, whereas if $p_j(g_j(a)) = q_j(g_j(a)) \geq 0$, g_j is considered as a quasi-criterion.

Comprehensive concordance. The comprehensive concordance index $C^S(a,b) \in [0,1]$ represents the strength of the coalition of criteria being in favor of the outranking relation aSb. Among these criteria we can distinguish these which support aSb with no hesitation, i.e., $F^S(a,b)$, and these which neither confirm the indifference nor the opposition, i.e., $F^Q(b,a)$. To compute $C^S(a,b)$, we need to consider intrinsic weights w_j assigned to $g_j \in F$. The coefficient w_j is interpreted as the voting power of g_j, i.e., the greater w_j, the more important g_j. Note that there exist some well-established procedures for assigning values to w_j [14]. The comprehensive concordance index is defined as follows:

$$C^S(a,b) = \frac{\sum_{j \in J} w_j \cdot c_j(a,b)}{\sum_{j \in J} w_j}. \tag{7}$$

This fraction is closer to one, when the strength of criteria $F^S(a,b)$ and $F^Q(b,a)$ outweighs the strength of criteria opposing to aSb, i.e., $F^P(b,a)$. In particular, $C^S(a,b) = 1$ indicates that all criteria strongly support aSb, whereas $C^S(a,b) = 0$ means that none of the criteria supports this assertion strongly or weakly.

Module M1: ElectreConcordance. This module computes comprehensive concordance indices $C^S(a,b)$ (out_1). Its structure is presented in Figure 1. It requires the user to specify a set of alternatives (in_1), their performances (in_4), a set of criteria along with the comparison thresholds (in_3), and criteria weights (in_6). The user may parameterize the module ($param_1$) to compare alternatives either with each other or with boundary or characteristic class profiles. If (s)he selected the comparison with some profiles, their definition (in_2) and performances (in_4) need to be provided at the input. The same possibility is offered by other modules designed for constructing an outranking relation, i.e., M2-M11. Thus, when presenting these modules, we will focus only on the features which are specific for them.

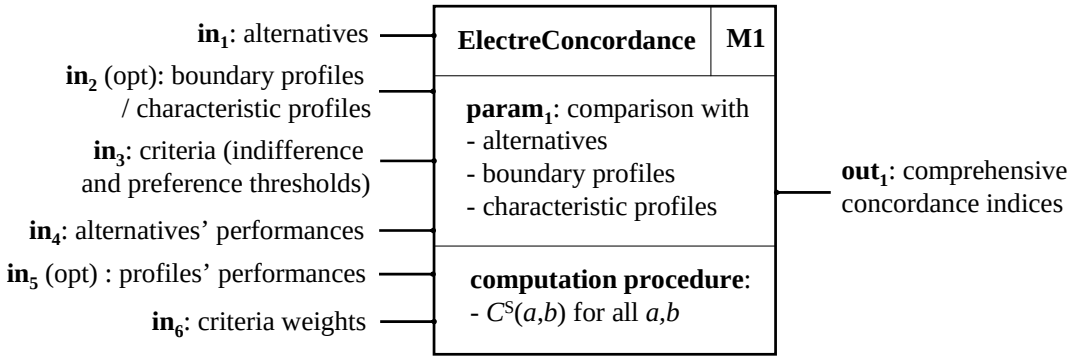

Figure 1: Structure of module M1 which computes comprehensive concordance indices.

Comprehensive concordance with reinforced preference. If aS_jb, the performance difference $g_j(a)-g_j(b)$ does not influence the marginal concordance index $c_j(a,b)$ and, in consequence, $C^S(a,b)$. However, if a is very strongly preferred to b, one may judge that it should obtain some bonus with respect to the case where the preference is not that strong. To satisfy this wish, we may refer to a reinforced preference threshold $rp_j(\cdot)$ [32]. Formally, $rp_j(g_j(a))$ corresponds to the difference in performances $g_j(a)-g_j(b)$ which is judged high enough for considering g_j as more significant in the coalition supporting aSb, comparing to the situation where the difference of performances is smaller than $rp_j(g_j(a))$ but not less than $p_j(g_j(a))$. When this threshold is crossed, whenever w_j is used in the formula for $C^S(a,b)$, it has to be replaced by $\omega_j \cdot w_j$, where $\omega_j > 1$ is a reinforcement factor. The set of criteria for which such reinforced preference aRP_jb occurs is denoted with:

$$F^{RP}(a,b) = \{j : aRP_jb \Leftrightarrow g_j(a) - g_j(b) > rp_j(g_j(a))\}. \tag{8}$$

Then, the new concordance index $C^{RP}(a,b)$ accounting for the reinforced preference is defined as follows [32]:

$$C^{RP}(a,b) = \frac{\sum_{j\in F^{RP}(a,b)} w_j \cdot \omega_j + \sum_{j\in J\setminus F^{RP}(a,b)} w_j \cdot c_j(a,b)}{\sum_{j\in F^{RP}(a,b)} w_j \cdot \omega_j + \sum_{j\in J\setminus F^{RP}(a,b)} w_j}. \tag{9}$$

Due to the reinforced preference effect, the contribution of criteria $F^{RP}(a,b)$ in the strength of the coalition of criteria supporting aSb is greater than their contribution when considering $C^S(a,b)$.

Module M2: ElectreConcordanceReinforcedPreference. This module (see Figure 2) computes comprehensive concordance indices $C^{RP}(a,b)$ with reinforced preference (out_1). When compared to M1, it has to be provided with two additional parameters: reinforced preference thresholds $rp_j(\cdot)$ (included in the specification of criteria in_3) and reinforcement factors ω_j (in_7).

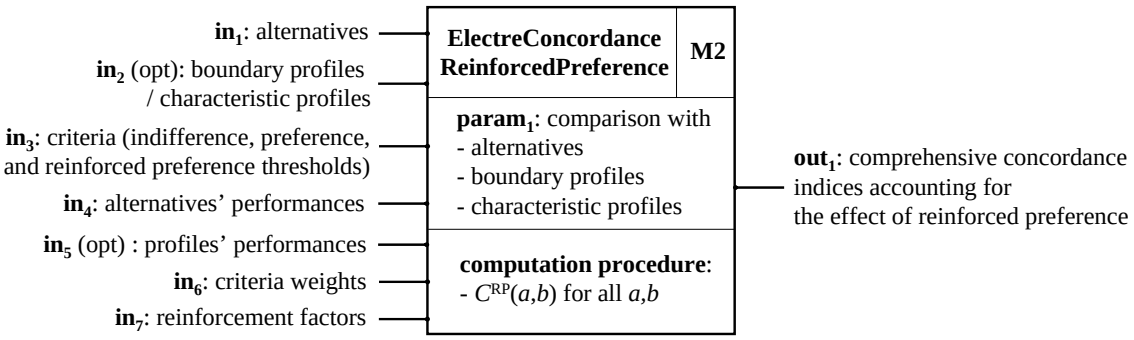

Figure 2: Structure of module M2 which computes comprehensive concordance indices with reinforced preference.

Comprehensive concordance with interactions between criteria. In the standard setting, we assume that a family of criteria F is constructed so that there are no interactions between criteria. To take them into account, the formula for computing comprehensive concordance index needs to be extended by considering the weights of the interaction coefficients [11]. Three types of interactions for pairs of criteria are considered [11]:

- mutual strengthening effect: if both criteria g_i and g_j support aSb, i.e., $i,j \in F^{NP}(a,b) = J\setminus F^P(b,a)$, their contribution to $C^S(a,b)$ must be greater than $w_i \cdot c_i(a,b) + w_j \cdot c_j(a,b)$; in this case, the joint presence of g_i and g_j in $F^{NP}(a,b)$ should justify an additional bonus; it can be imposed by using a mutual strengthening coefficient $w_{ij}^{MS} = w_{ji}^{MS} > 0$, which intervenes positively in $C^S(a,b)$; the set of

criteria pairs for which this interaction is considered meaningful for a pair (a, b) is denoted with:

$$F^{MS}(a,b) = \{i, j \in J : i, j \in F^{NP}(a,b) \text{ and } w_{ij}^{MS} > 0\}; \tag{10}$$

- mutual weakening effect: if both criteria g_i and g_j support aSb, i.e., $i, j \in F^{NP}(a,b)$, their contribution to $C^S(a,b)$ must be smaller than $w_i \cdot c_i(a,b) + w_j \cdot c_j(a,b)$; in this case, the joint presence of g_i and g_j in $F^{NP}(a,b)$ should contribute to $C^S(a,b)$ less than either g_i or g_j contributes in case only one of them is present in $F^{NP}(a,b)$; such a penalty can be modeled with a mutual weakening coefficient $w_{ij}^{MW} = w_{ji}^{MW} < 0$, which intervenes negatively in $C^S(a,b)$; the set of criteria pairs for which this interaction holds for a pair (a,b) is denoted with:

$$F^{MW}(a,b) = \{i, j \in J : i, j \in F^{NP}(a,b) \text{ and } w_{ij}^{MW} < 0\}; \tag{11}$$

- antagonistic effect: if criterion g_i supports aSb, i.e., $i \in F^{NP}(a,b)$, and g_j strongly opposes to aSb, i.e., $j \in F^P(b,a)$, the contribution of g_i to $C^S(a,b)$ must be smaller than $w_i \cdot c_i(a,b)$; thus, the strong opposition of g_j decreases the contribution of g_i to $C^S(a,b)$ when compared to the case when $j \notin F^P(b,a)$; this effect can be modeled by introducing an antagonism coefficient $w_{ij}^A > 0$, which intervenes negatively in $C^S(a,b)$; let us denote the set of criteria pairs for which this interaction holds for a pair (a,b) with:

$$F^A(a,b) = \{(i,j) \in J \times J : i \in F^{NP}(a,b),\ j \in F^P(b,a) \text{ and } w_{ij}^A > 0\}. \tag{12}$$

  The antagonistic effect for a pair $(i,j) \in J \times J$ neither implies nor excludes the reverse effect for (j,i).

Note that the mutual strengthening and weakening effects are mutually exclusive, i.e., for all a, b, $F^{MS}(a,b) \cap F^{MW}(a,b) = \emptyset$. The new concordance index accounting for the aforementioned types of interactions is defined as follows:

$$C^{INT}(a,b) = \frac{\sum\limits_{j \in J} w_j \cdot c_j(a,b) + \sum\limits_{\{i,j\} \in F^{MS}(a,b)} Z_{ij}^{ab} \cdot w_{ij}^{MS} + \sum\limits_{\{i,j\} \in F^{MW}(a,b)} Z_{ij}^{ab} \cdot w_{ij}^{MW} - \sum\limits_{(i,j) \in F^A(a,b)} Z_{ij}^{ab} \cdot w_{ij}^{A}}{\sum\limits_{j \in J} w_j + \sum\limits_{\{i,j\} \in F^{MS}(a,b)} Z_{ij}^{ab} \cdot w_{ij}^{MS} + \sum\limits_{\{i,j\} \in F^{MW}(a,b)} Z_{ij}^{ab} \cdot w_{ij}^{MW} - \sum\limits_{(i,j) \in F^A(a,b)} Z_{ij}^{ab} \cdot w_{ij}^{A}}, \tag{13}$$

where function Z_{ij}^{ab} is used to capture the interaction effects in the ambiguity zone. Among the multiple forms that can be chosen for Z_{ij}^{ab}, the two following ones have an intuitive and meaningful interpretation:

$$Z_{ij}^{ab,min} = Z(c_i(a,b), c_j(a,b)) = min\{c_i(a,b), c_j(a,b)\}, \text{ or} \tag{14}$$

$$Z_{ij}^{ab,multi} = Z(c_i(a,b), c_j(a,b)) = c_i(a,b) \cdot c_j(a,b). \tag{15}$$

Module M3: ElectreConcordanceWithInteractions. This module (see Figure 3) computes comprehensive concordance indices $C^{INT}(a,b)$ (out_1) accounting for the interactions between pairs of criteria. When compared to M1, it has to be provided with the specification of interactions of three different types (in_4). For each type, the user needs to indicate pairs of interacting criteria and a numerical value of an interaction coefficient. Furthermore, one can choose out of two pre-defined Z_{ij}^{ab} functions, i.e., either $Z_{ij}^{ab,min}$ or $Z_{ij}^{ab,multi}$.

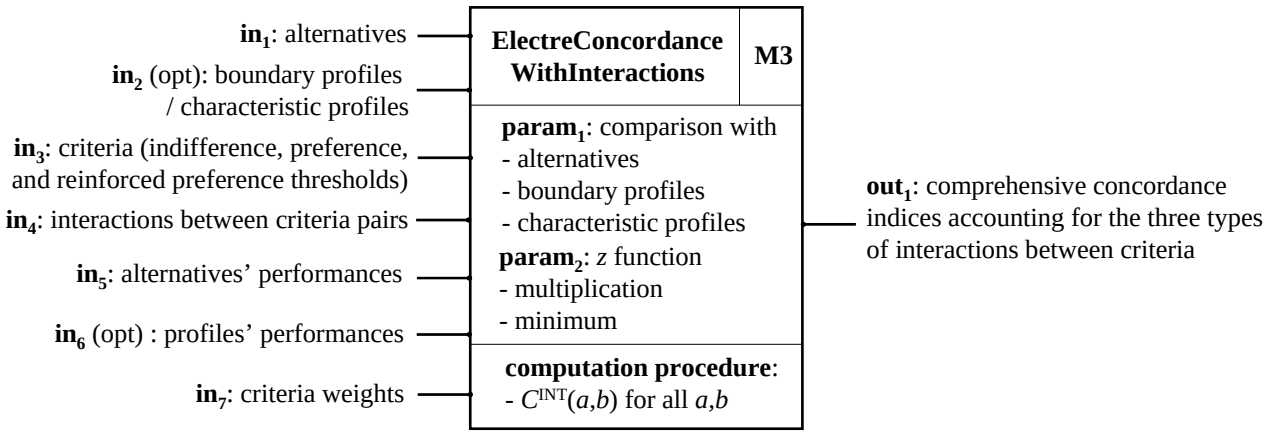

Figure 3: Structure of module M3 which computes comprehensive concordance indices accounting for the interactions between criteria.

2.2. Discordance (Non-Discordance) Test

Discordance refers to the criteria which oppose to aSb. Veto threshold $v_j(g_j(a))$ such that $v_j(g_j(a)) > p_j(g_j(a))$ for all a, can be used to model the effect of strong opposition g_j puts to aSb. Let us first discuss different ways of computing marginal discordance index $d_j(a,b)$ indicating to which degree g_j "vetoes" against the outranking.

Binary marginal discordance with veto threshold. In the basic scenario, the veto effect is modeled with a binary situation in which g_j imposes veto when $g_j(b)$ is better than $g_j(a)$ by at least the veto threshold $v_j(g_j(a))$, or not, otherwise. Then, $d_j(a,b)$ is defined as follows:

$$d_j^V(a,b) = \begin{cases} 1 & \text{if } g_j(b) - g_j(a) \geq v_j(g_j(a)), \\ 0 & \text{if } g_j(b) - g_j(a) < v_j(g_j(a)). \end{cases} \tag{16}$$

Comprehensive binary discordance. In the ELECTRE Is method [31], the comprehensive discordance occurred if at least one criterion vetoed an outranking relation. In this spirit, a comprehensive binary discordance $D^V(a,b)$ can be formalized as follows:

$$D^V(a,b) = \begin{cases} 1 & \text{if } \exists j \in J,\ d_j^V(a,b) = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

Module M4: ElectreIsDiscordanceBinary. This module (see Figure 4) computes comprehensive $D^V(a,b)$ and marginal $d_j^V(a,b)$ discordance indices (out_1 and out_2). When compared to M1, it has to be provided with veto rather than comparison thresholds (these need to be included in the specification of criteria in_3). It does not require, however, the criteria weights.

Marginal discordance with veto and preference thresholds. To weaken the veto effect, the marginal discordance index can be fuzzified. This can be achieved by deriving its value from the comparison of the performance difference $g_j(b) - g_j(a)$ with both $v_j(g_j(a))$ and $p_j(g_j(a))$ rather than with $v_j(g_j(a))$ only. If g_j supports aSb, i.e., $j \in F^{NP}(a,b)$, it cannot simultaneously oppose to this assertion, and, thus, $d_j(a,b) = 0$. If $g_j(b)$ is better than $g_j(a)$ by at least $v_j(g_j(a))$, then g_j imposes a strong veto, and, thus, $d_j(a,b) = 1$. If bP_ja, but the difference in favor of b is less than $v_j(g_j(a))$, then g_j weakly vetoes aSb, and, thus,

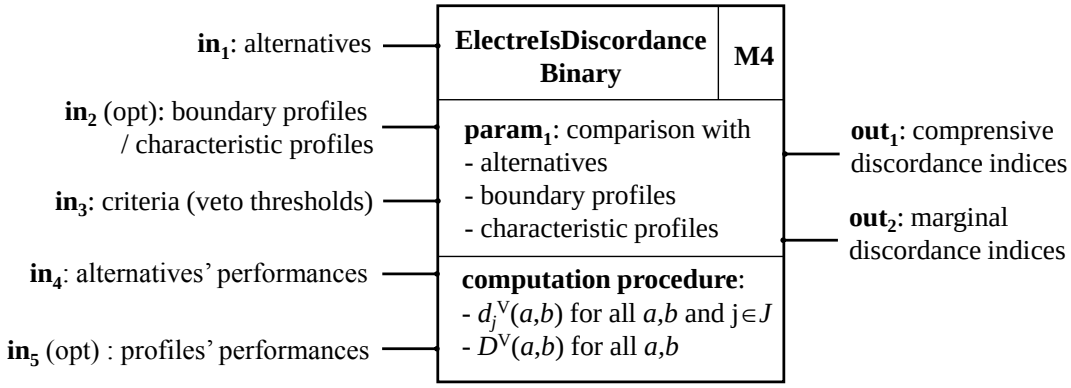

Figure 4: Structure of module M4 which computes marginal and comprehensive binary discordance indices.

$d_j(a,b) \in (0,1)$. Thus, the marginal discordance index is formally defined as follows:

$$d_j^{PV}(a,b) = \begin{cases} 1 & \text{if } g_j(b) - g_j(a) > v_j(g_j(a)), \\ 0 & \text{if } g_j(b) - g_j(a) \leq p_j(g_j(a)), \\ \frac{[v_j(g_j(a)) - (g_j(b) - g_j(a))]}{[v_j(g_j(a)) - p_j(g_j(a))]} & \text{if } p_j(g_j(a)) < g_j(b) - g_j(a) \leq v_j(g_j(a)). \end{cases} \tag{18}$$

Marginal discordance with veto and pre-veto thresholds. The definition of $d_j^{PV}(a,b)$ derived from the comparison of $g_j(b) - g_j(a)$ with $v_j(g_j(a))$ and $p_j(g_j(a))$ implies that the zone of neutrality which neither supports nor vetoes aSb is extremely scarce. Precisely, only if $g_j(b) - g_j(a) = p_j(g_j(a))$, then both $c_j(a,b)$ and $d_j^{PV}(a,b)$ are equal to zero. To enlarge this zone of neutrality, we may account for the pre-veto (discordance) threshold $pv_j(\cdot)$ [21], defined so that for all a, $p_j(g_j(a)) \leq pv_j(g_j(a)) \leq v_j(g_j(a))$. In this case, g_j contributes to the veto effect, i.e., $d_j(a,b) > 0$, iff $g_j(b) - g_j(a) > pv_j(g_j(a))$. Thus, the marginal discordance index is defined as follows:

$$d_j^{PVV}(a,b) = \begin{cases} 1 & \text{if } g_j(b) - g_j(a) > v_j(g_j(a)), \\ 0 & \text{if } g_j(b) - g_j(a) \leq pv_j(g_j(a)), \\ \frac{[v_j(g_j(a)) - (g_j(b) - g_j(a))]}{[v_j(g_j(a)) - pv_j(g_j(a))]} & \text{if } pv_j(g_j(a)) < g_j(b) - g_j(a) \leq v_j(g_j(a)). \end{cases} \tag{19}$$

Counter-veto effect. Analogously as for the reinforced preference, the counter-veto threshold $cv_j(\cdot)$ can be used to assign a special role to the criteria for which a very strong preference of one alternative over another occurs [32]. Let $cv_j(g_j(a))$ correspond to the difference in performances $g_j(a) - g_j(b)$ which is judged high enough for weakening the effect of veto against the credibility of outranking, comparing to the situation where the performance difference is smaller than $cv_j(g_j(a))$ but not less than $p_j(g_j(a))$. If this threshold is exceeded by $g_j(a) - g_j(b)$, g_j should be allowed to intervene directly in the definition of the credibility of an outranking (see Section 2.3). To make this intervention possible, the counter-veto effect is quantified with the number $cv(a,b)$ of criteria for which it occurs.

Module M5: ElectreDiscordance. This module (see Figure 5) computes marginal discordance indices (out_1). If the user wishes to consider pre-veto thresholds ($param_2$), these indices are computed as $d_j^{PVV}(a,b)$. In this case, the pre-veto thresholds need to provided along with the preference and veto thresholds when specifying the criteria (in_3). Otherwise, the procedure returns $d_j^{PV}(a,b)$.

Additionally, this modules accounts for the counter-veto effect. If counter-veto thresholds are provided at its input (in_3), it indicates pairs of objects and criteria for which the counter-veto threshold was exceeded. This information may be subsequently used to derive $cv(a,b)$.

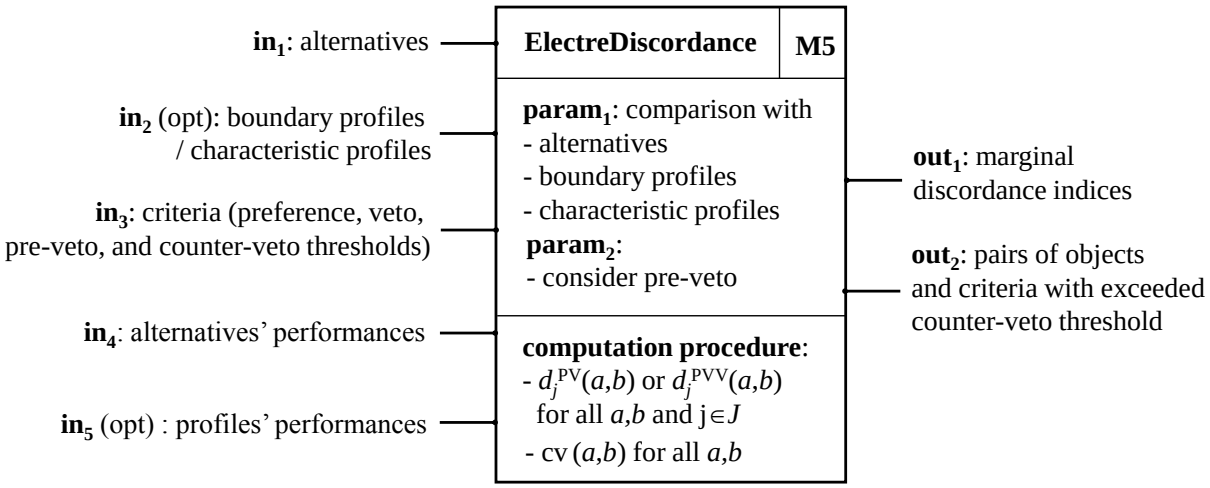

Figure 5: Structure of module M5 which computes marginal discordance indices while referring to veto and preference or pre-veto thresholds. It also provides information on the pairs of objects and criteria for which the counter-veto effect occurred.

Comprehensive discordance index. The marginal discordance indices can be aggregated into a comprehensive discordance index (called also coalitional veto) by taking into account the criteria importance coefficients [34]. It is computed analogously as the comprehensive concordance index $C^S(a,b)$, i.e.:

$$\Delta^{CD}(a,b) = \frac{\sum\limits_{j \in J} w_j^D \cdot d_j(a,b)}{\sum\limits_{j \in J} w_j^D}. \tag{20}$$

The weights w_j^D express the power of each criterion g_j to veto an outranking relation. In particular, all these weights may be the same (i.e., for $j \in J$, $w_j^D = 1$) or set equivalent to the weights w_j used in the concordance test. Further, $d_j(a,b)$ used in (20) could be $d_j^V(a,b)$, or $d_j^{PV}(a,b)$, or $d_j^{PVV}(a,b)$.

Module M6: ElectreComprehensiveDiscordance. This module (see Figure 6) computes comprehensive discordance indices $\Delta^{CD}(a,b)$ (out_1). It requires the user to specify marginal discordance indices $d_j(a,b)$ (in_3) and criteria weights w_j^D (in_4).

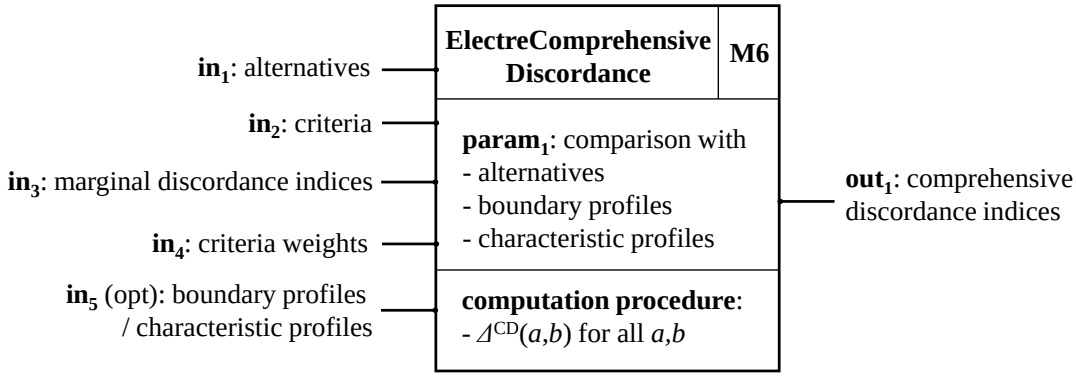

Figure 6: Structure of module M6 which computes comprehensive discordance indices.

2.3. Valued Outranking Relation

ELECTRE combines the comprehensive concordance index and discordance marginal indices in order to define a valued outranking relation $S(a,b)$. Note that $S(a,b)$ can be interpreted as a degree of credibility

of an outranking relation. In this subsection, whenever we refer to $C(a,b)$, it can be interpreted as one of its three versions discussed in Section 2.1, i.e., $C(a,b) \in \{C^S(a,b), C^{INT}(a,b), C^{RP}(a,b)\}$.

Credibility of outranking. The basic idea for computing the credibility of an outranking relation consists in synthesizing the strength of the coalition of criteria supporting aSb, i.e., $C(a,b)$, and the opposition of criteria being against this assertion, i.e., $\Delta(a,b)$. In the ELECTRE Tri-B method, $\Delta(a,b)$ is interpreted as a comprehensive non-discordance index Δ^{DC} which represents a degree to which the criteria that sufficiently strongly oppose to aSb (i.e., $d_j(a,b) > C(a,b)$), if any, collectively oppose a veto to this assertion. It is computed in the following way:

$$\Delta^{DC}(a,b) = \prod_{j \in J: d_j(a,b) > C(a,b)} \frac{1 - d_j(a,b)}{1 - C(a,b)}, \tag{21}$$

where $d_j(a,b) \in \{d_j^V(a,b), d_j^{PV}(a,b), d_j^{PVV}(a,b)\}$. $\Delta^{DC}(a,b) = 0$ means that some criteria are totally opposed to aSb, whereas $\Delta^{DC}(a,b) = 1$ indicates that none of the criteria vetoes aSb strongly enough. Now, to aggregate results of the concordance and non-discordance comprehensive indices into a credibility of an outranking relation, it is sufficient to multiply them:

$$S^{DC}(a,b) = C(a,b) \cdot \Delta^{DC}(a,b). \tag{22}$$

Thus, the greater the concordance and non-discordance, the greater the credibility.

Simplified credibility index. Two simpler variants for computing the comprehensive non-discordance have been proposed in [21]. The motivation for these proposals was to allow easier indirect inference of the discordance-related parameters than in case of Δ^{DC}, while still preserving the original discordance concept. This can be achieved by referring to the marginal discordance indices $d_j(a,b)$, $j \in J$, only, while neglecting the comparison with $C(a,b)$. Thus, the criteria that influence the value of a comprehensive non-discordance index are not limited to these for which $d_j(a,b) > C(a,b)$. Instead, all positive values of $d_j(a,b)$ impact the index value in the following way:

$$\Delta^{D}(a,b) = \prod_{j \in J} (1 - d_j(a,b)), \text{ or} \tag{23}$$

$$\Delta^{DM}(a,b) = 1 - max_{j \in J} d_j(a,b). \tag{24}$$

Now, the aggregation of the concordance and non-discordance comprehensive indices into a credibility of outranking is carried out analogously as in case of $S^{DC}(a,b)$, i.e.:

$$S^{D}(a,b) = C(a,b) \cdot \Delta^{D}(a,b), \text{ or} \tag{25}$$

$$S^{DM}(a,b) = C(a,b) \cdot \Delta^{DM}(a,b). \tag{26}$$

Module M7: ElectreCredibility. This module (see Figure 7) computes the credibility of an outranking relation (out_1). It requires the user to provide comprehensive concordance (in_3) and marginal discordance indices (in_4). In its basic variant, it uses formula for $S^{DC}(a,b)$. However, it can be parameterized to compute credibility using simplified formulations, i.e., either $S^{D}(a,b)$ ($param_2$ set to false) or $S^{DM}(a,b)$ ($param_3$ checked). Moreover, it admits ($param_4$) providing at the input comprehensive (aggregated) discordance indices instead of marginal ones. In this case, to derive a credibility of an outranking, it simply multiplies comprehensive concordance and non-discordance degrees. This can be used, e.g., to account for

D^V (one of the outputs of M4), but it also offers avenues for further developments.

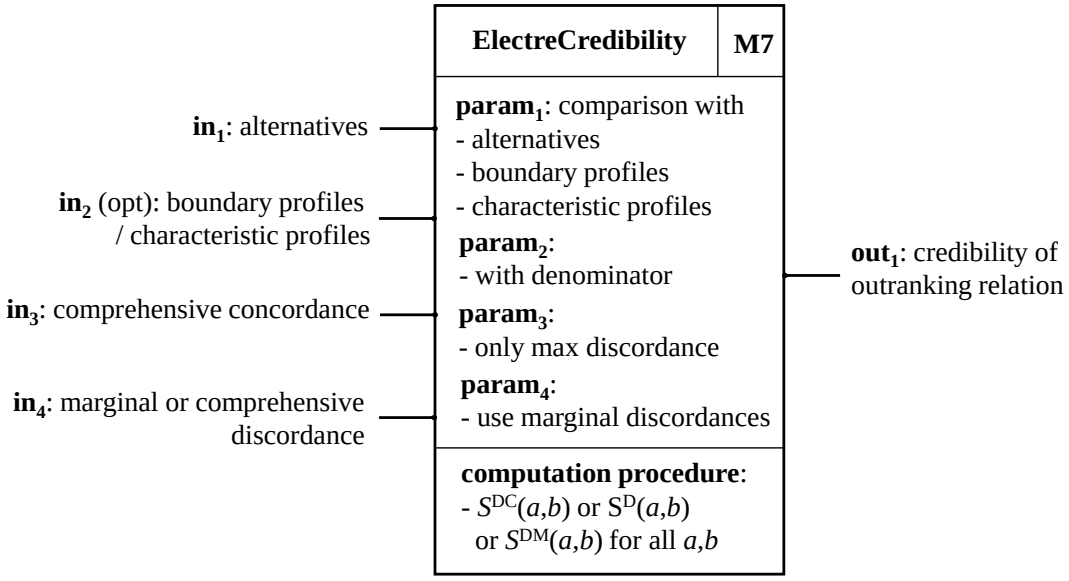

Figure 7: Structure of module M7 which computes credibility of an outranking relation.

Credibility with counter-veto. To weaken the effect of veto imposed by the discordant criteria on the credibility of outranking, we may consider the counter-veto effect [32]. It is materialized with the number $cv(a,b)$ of criteria for which the counter-veto threshold $cv_j(g_j(a))$ is exceeded by $g_j(a) - g_j(b)$. The proposed formula for the credibility index accounting for this phenomenon is [32]:

$$S^{CV}(a,b) = C(a,b) \cdot [\Delta(a,b)]^{(1-\frac{cv(a,b)}{m})}, \tag{27}$$

where $\Delta(a,b) \in \{\Delta^{DC}(a,b), \Delta^{D}(a,b), \Delta^{DM}(a,b)\}$. If $\Delta(a,b) \in (0,1)$ and $cv(a,b) > 0$, the effect of veto is weakened, and, thus, $S^{CV}(a,b)$ is higher than the credibility computed without considering the counter-veto.

Module M8: ElectreCredibilityWithCounterVeto. This module (see Figure 8) computes the credibility $S^{CV}(a,b)$ of an outranking relation (out_1) accounting for the counter-veto effect. When compared to M7, it requires the user to provide information on the pairs of objects and criteria for which the counter-veto effect occurred (in_4). This is used by the M8 module to compute $cv(a,b)$, which intervenes directly in the computation of a credibility degree. Besides, it offers the same functionality as M7 but tolerating comprehensive discordance at the input.

Credibility of outranking without considering criteria weights. In some decision making situations, we are not able or we do not want to specify weights of the criteria w_j, $j \in J$. ELECTRE IV replaces the use of importance coefficients with a definition of five embedded outranking relations. These are: quasi- (S_q), canonic- (S_c), pseudo- (S_p), sub- (S_s), and veto-dominance (S_v) relations. The quasi-dominance is the most credible and veto-dominance is the least credible among them. The conditions that need to be satisfied so that a given outranking relation holds are based on the following parameters: $m_P(a,b) = |F^P(a,b)|$, $m_Q(a,b) = |F^Q(a,b)|$, $m_{I>}(a,b) = |j \in F^I(a,b) : g_j(a) > g_j(b)|$, $m_I(a,b) = |j \in F^I(a,b) : g_j(a) = g_j(b)|$, $m_{I>}(b,a)$, $m_Q(b,a)$, and $m_P(b,a)$.

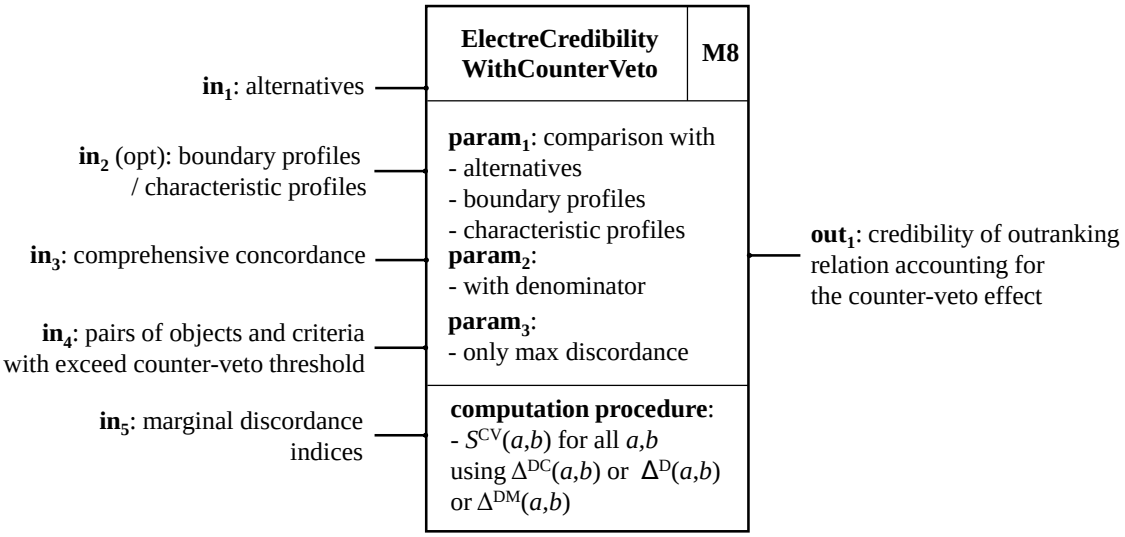

Figure 8: Structure of module M8 which computes credibility of an outranking relation accounting for the counter-veto effect.

The five embedded relations are defined as follows:

$$aS_qb \Leftrightarrow [m_P(b,a) + m_Q(b,a) = 0] \wedge [m_{I>}(b,a) < m_P(a,b) + m_Q(a,b) + m_{I>}(a,b)]; \quad (28)$$

$$aS_cb \Leftrightarrow m_P(b,a) = 0 \wedge m_Q(b,a) \leq m_P(a,b) \wedge [m_Q(b,a) + m_{I>}(b,a) < m_P(a,b) + m_Q(a,b) + m_{I>}(a,b)]; \quad (29)$$

$$aS_pb \Leftrightarrow m_P(b,a) = 0 \wedge [m_Q(b,a) < m_P(a,b) + m_Q(a,b)]; \quad (30)$$

$$aS_sb \Leftrightarrow m_P(b,a) = 0; \quad (31)$$

$$aS_vb \Leftrightarrow m_P(b,a) \leq 1 \wedge m_P(a,b) \geq m/2 \wedge [g_j(b) - g_j(a) < v_j(g_j(a)), j \in J]. \quad (32)$$

Thus, these relations are defined so that $S_q \subseteq S_c \subseteq S_p \subseteq S_s \subseteq S_v$. Clearly, the chosen values that can be assigned to $S(a,b)$ must be such that the more credible relation, the greater $S(a,b)$. Moreover, the transition from less credible relation to the more credible one needs to be perceived as a considerable gain. Even though there exist different sets of values satisfying these requirements, the most often employed option is the following: $aS_qb \Rightarrow S^{IV}(a,b) = 1.0$, $aS_cb \Rightarrow S^{IV}(a,b) = 0.8$, $aS_pb \Rightarrow S^{IV}(a,b) = 0.6$, $aS_sb \Rightarrow S^{IV}(a,b) = 0.4$, $aS_vb \Rightarrow S^{IV}(a,b) = 0.2$. If none of these relations holds, $S^{IV}(a,b) = 0.0$.

Module M9: ElectreIVCredibility. This module (see Figure 9) computes the credibility of an outranking relation $S^{IV}(a,b)$ as in ELECTRE IV (out_1). It requires the user to provide the indifference, preference, and veto thresholds (in_3).

2.4. Crisp Outranking Relation

The valued outranking relation $S(a,b)$ can be transformed into a crisp outranking relation S (let us denote it by S^{CUT}) by taking into account the cutting level $\lambda \in [0.5, 1.0]$, which represents the minimum value of $S(a,b)$ which implies $aS^{CUT}b$. Thus, if $S(a,b) \geq \lambda$, then $aS^{CUT}b$; otherwise, $\neg(aS^{CUT}b)$.

Module M10: cutRelationCrisp. This module (see Figure 10) transforms a valued relation into a crisp one (out_1). It requires the user to provide a valued relation (in_3) and a cutting level λ ($param_2$). If one provides $S(a,b)$ at the input (in_2), it would produce S^{CUT} at the output. However, if one provides $C(a,b)$ or $\Delta^{CD}(a,b)$ instead, it can be used to obtain the crisp concordance or discordance, respectively.

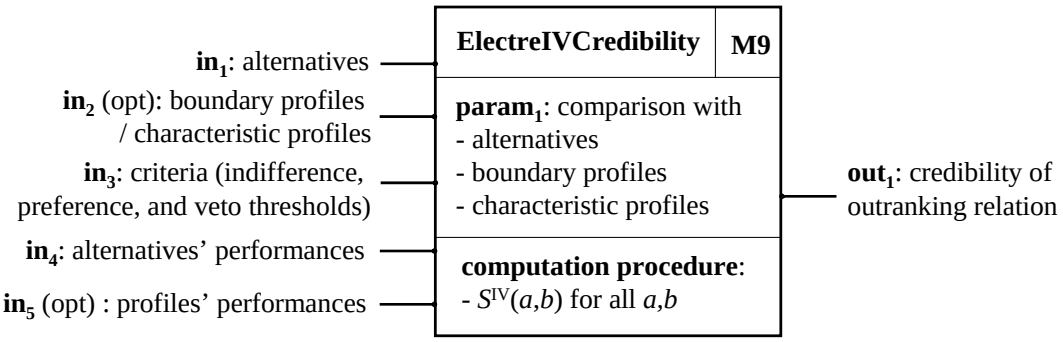

Figure 9: Structure of module M9 which computes credibility of an outranking relation without considering criteria weights.

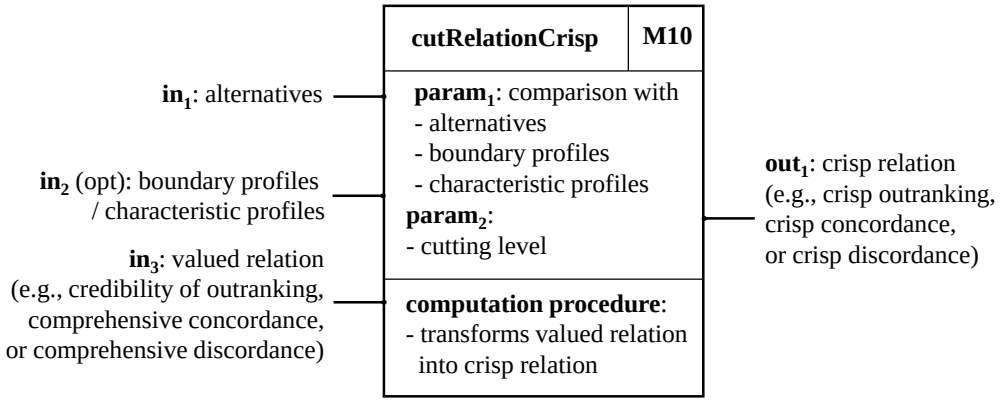

Figure 10: Structure of module M10 which transforms a valued relation into a crisp one.

Another way of constructing a crisp outranking relation is based on the separate consideration of the comprehensive concordance and discordance indices. In the ELECTRE Is method, the outranking relation S^{Is} holds if the coalition of criteria in favor of this assertion is strong enough, while there is no veto on any criteria. The former requirement can be verified by comparing $C(a,b)$ with λ^C representing the concordance majority threshold (cutting level), whereas the latter is satisfied if the value of comprehensive binary discordance $D^V(a,b)$ was equal to zero. Thus, S^{Is} can be defined as follows:

$$aS^{Is}b \iff C(a,b) \geq \lambda^C \text{ and } D^V(a,b) = 0; \tag{33}$$

$$\neg(aS^{Is}b) \iff C(a,b) < \lambda^C \text{ or } D^V(a,b) = 1. \tag{34}$$

Recently, [34] generalized this definition so that to consider the strength of the coalition of criteria against an outranking in the same spirit as the strength of the coalition supporting this assertion. This requires comparison of a comprehensive discordance index $\Delta^{CD}(a,b)$ with λ^D representing the discordance majority threshold (cutting level). Obviously, while the coalition in favor $aS^{COAL}b$ should be strong enough, the discordant coalition should be relatively weak. Thus, S^{COAL} is constructed in the following way:

$$aS^{COAL}b \iff C(a,b) \geq \lambda^C \text{ and } \Delta^{CD}(a,b) < \lambda^D; \tag{35}$$

$$\neg(aS^{COAL}b) \iff C(a,b) < \lambda^C \text{ or } \Delta^{CD}(a,b) \geq \lambda^D. \tag{36}$$

Module M11: ElectreCrispOutrankingAggregation. This module (see Figure 11) constructs a crisp outranking relation (out_1) based on the binary information concerning concordance and discordance tests. The user needs to provide information on whether the concordance and discordance tests were passed

or not (in_2 and in_3). Depending on how a crisp discordance has been previously computed, the module constructs a crisp outranking defined as S^{Is} or S^{COAL}. In any case, the outranking relation is true for some pair of objects only if the concordance test (1) was passed and the discordance test was failed (0).

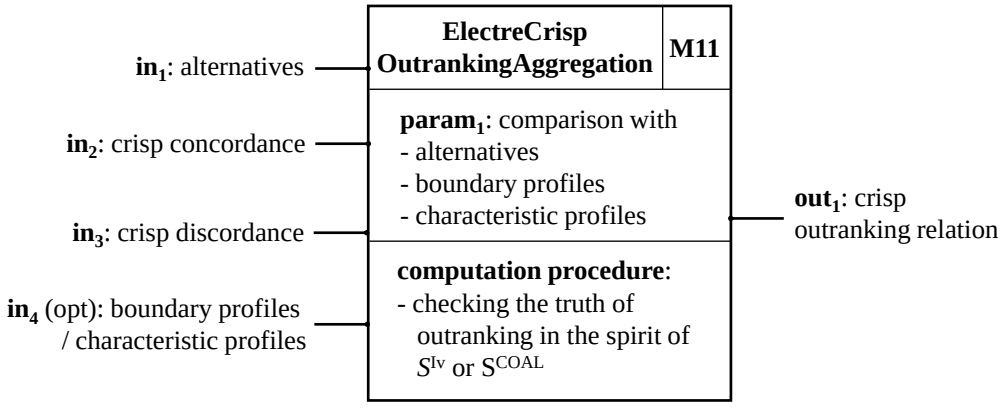

Figure 11: Structure of module M11 which constructs a crisp outranking on the basis of binary information on concordance and discordance.

S can be used also for representing the weak (Q) and strict (P) preference, indifference (I), and incomparability (R) as follows [26]:

$$aSb \wedge \neg(bSa) \iff aQb \vee aPb \iff a \succ b, \text{ where } \succ = \{Q \cup P\}, \tag{37}$$

$$aSb \wedge bSa \iff aIb, \tag{38}$$

$$\neg(aSb) \wedge \neg(bSa) \iff aRb. \tag{39}$$

3. Exploitation of an Outranking Relation

In this section, we review different ways of exploiting an outranking relation. These are divided into three groups depending on the type of considered problem: choice, ranking, or sorting. Analogously as in Section 2, we present the functionality of the underlying software modules we have implemented. Note that the modules for multiple criteria choice and ranking require specification of alternatives only, while the modules for sorting need to be additionally provided with information on decision classes and either boundary or characteristic class profiles. The modules discussed in this section are numbered from M12 to M18.

3.1. Choice Problems

Selecting a subset of the best alternatives while using an outranking relation as the preference model has been first considered in the family of ELECTRE I methods (including ELECTRE I, ELECTRE Iv, and ELECTRE Is; see, e.g., [4, 23, 25, 28, 30]). These approaches represent the outranking relation S imposed on set A with an outranking graph G_S whose nodes correspond to the alternatives and arcs reflect the truth of S. The best alternatives are assumed to be contained in the kernel K of graph G_S. Its definition involves the following two properties:

- an internal consistency of K: the alternatives in K are incomparable in terms of an outranking relation; thus, the nodes in K are not related by an arc;
- an external consistency of K: the alternatives not contained in K are outranked by at least one alternative belonging to K; thus, they correspond to the ending nodes of arcs whose starting nodes represent the alternatives from K.

If graph G_S is acyclic, there exists a unique kernel. The procedure for identifying K is based on an observation that in an acyclic graph, there exists at least one node without any predecessors. It corresponds to an alternative which is not outranked by any other alternative. Clearly, such nodes need to be included in K. Then, the procedure takes into account each node whose all predecessors have been already considered in terms of their presence in K. If none of their predecessors belongs to K, then such node is added to K. Such iterative checking is continued until all nodes are verified.

If a graph has a cycle, we need to reduce it before looking for a kernel K. The two basic techniques that can be used for reducing the graph's cycle consist in:

- aggregating all elementary nodes in a cycle into a singleton (an artificial node) inheriting all in- and out-arcs from its component nodes; consequently, all alternatives which form a cycle are considered indifferent;
- cutting an arc associated with the weakest credibility of an outranking $S(a, b)$; as a result, alternatives belonging to a cycle are no longer considered indifferent.

For more details, see [28, 37].

Module M12: ElectreIsFindKernel. This module (see Figure 12) finds a kernel (out_1) in a graph constructed on the basis of a crisp outranking relation (in_3). The user is allowed to choose a technique for reducing the potential graph's cycles ($param_1$). If (s)he wished to cut the weakest arc, the module needs to be additionally provided with a credibility of an outranking relation (in_2).

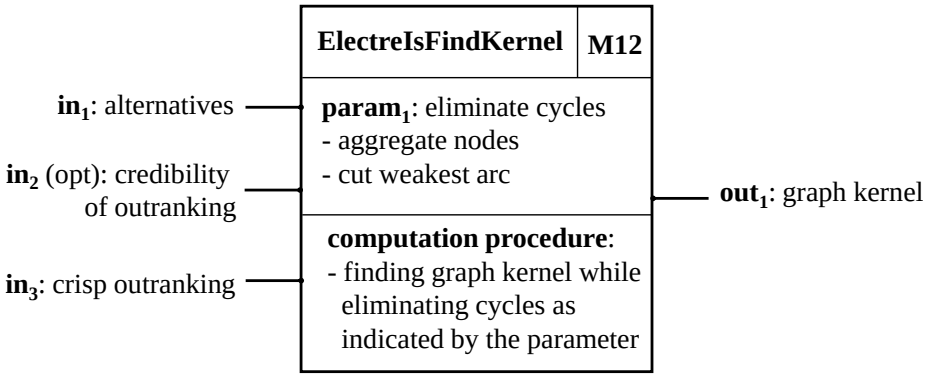

Figure 12: Structure of module M12 which finds a kernel in the outranking graph.

3.2. Ranking Problems

3.2.1. Net Flow Score procedure

In case of ranking problems, the alternatives may be ordered from the best to the worst with the Net Flow Score (NFS) procedure [6, 35]. For each alternative, NFS accounts for the arguments confirming its strength and/or weakness. In term of ELECTRE, for each $a \in A$ these arguments may be interpreted as the number of alternatives $b \in A$ which, respectively, are outranked by a (i.e., aSb) or do outrank a (i.e., bSa), i.e.:

$$NFS^S(a) = |b \in A : aSb| - |b \in A : bSa|. \tag{40}$$

Then, the final ranking is determined by $NFS(a)$ in A (the higher the score, the better).

Alternatively, instead of a crisp outranking relation, we can consider a valued one. In this case, for each $a \in A$:

$$NFS^{Sval}(a) = \sum_{b \in A} [S(a, b) - S(b, a)]. \tag{41}$$

Note that ranking and choice problems are closely related, because very often knowing a complete order of alternatives, we indicate these occupying top positions (in this case, having the maximal $NFS(a)$) as the best ones.

Module M13: ElectreNFSOutranking. This module (see Figure 13) computes net flow scores for all alternatives (out_1). It additionally provides information on the strength (out_2) and weakness (out_3). If the user indicates ($param_1$) that a crisp relation is provided at the input (in_2), the scores are computed as $NFS^S(a)$. Otherwise, a valued outranking relation needs to be provided (in_2) and $NFS^{Sval}(a)$ is used in the computation procedure. Moreover, we provide an avenue for further developments, because the module admits ($param_2$) to consider the strength and weakness derived from the non-outranking relation (in_3).

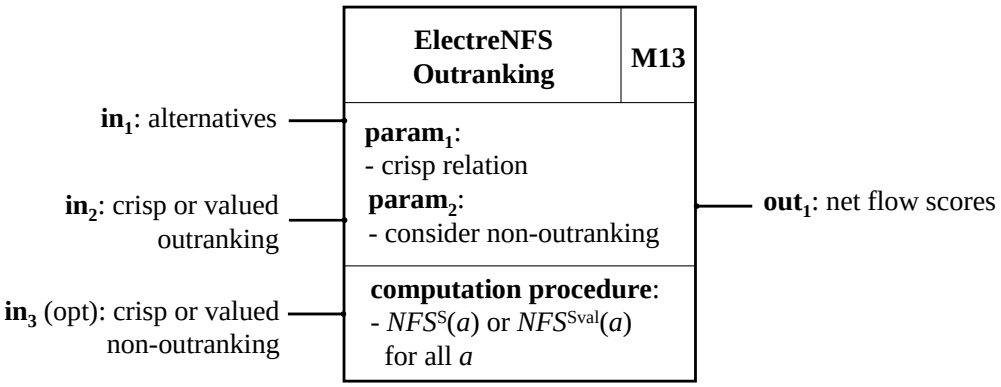

Figure 13: Structure of module M13 which computes net flow scores for the alternatives.

3.2.2. Distillation Procedures

To rank the alternatives, in ELECTRE methods one has traditionally applied distillation procedures which exploit a valued outranking relation (credibility matrix) $S(a,b)$ for $a,b \in A$. In ELECTRE III [24, 30], we construct two preorders P_D (downward) and P_U (upward) using, respectively, a descending and ascending distillation.

The partial preorder P_D is defined as partition on the set A into q ordered classes, $D_1,\ldots,D_h,\ldots,D_q$, where D_1 is the head-class in P_D. Each class deems together alternatives considered ex aequo according to P_D. The procedure for descending distillation is carried out in the following way:

1. Set the number of iteration to $k=0$, and fix the set of considered alternatives in k-th iteration to $B_k = A$.
2. Select the maximal credibility index: $\lambda_k = max_{a,b\in B_k, a\neq b} S(a,b)$.
3. Select the lower bound of the range of credibility indices that will be taken into account in the k-th iteration: $\lambda_{k+1} = max_{a,b\in B_k,\ S(a,b)<S_k^{max}-s(\lambda_k)}\{S(a,b),0\}$, where $s(x)=\alpha\cdot x+\beta$. The values of α and β are usually assumed to be equal to -0.15 and 0.3, respectively.
4. If $\lambda_k = 0$, then add all alternatives in B_k at the bottom of the descending preorder, and stop. Otherwise, set $k = k+1$.
5. Construct a crisp outranking relation $S_A^{\lambda_k}$ in the following way:

$$S_A^{\lambda_k} = \begin{cases} 1 & \text{if } S(a,b) > \lambda_k \text{ and } S(a,b) > S(b,a) + s(S(a,b)), \\ 0 & \text{otherwise.} \end{cases} \tag{42}$$

6. Compute the strength and weakness of each alternative $a \in B_k$ as the numbers of alternatives which are, respectively, outranked by a or outrank a. The quality of a, $Q_{B_k}^{\lambda_k}(a)$, is computed as the difference of its strength and weakness.
7. Select the set of alternatives D_k with the greatest quality $Q_{B_k}^{\lambda_k}(a)$. If this set contains a unique alternative, add it at the bottom of the descending preorder, and delete D_k from B_k. Otherwise, apply the same distillation procedure in D_k starting with $\lambda_k^{D_k} = \lambda_k$.
8. If A is empty, then stop. Otherwise, go to point 2.

The ascending distillation is conducted analogously with the proviso that the preorder is constructed bottom-up (rather than top-down), and the alternatives with the least (rather than the greatest) quality are retained first.

Module M14: ElectreDistillation. This module (see Figure 14) conducts either descending or ascending ($param_1$) distillation in the set of alternatives on the basis of a valued outranking relation (in_2). Depending on the selected option, it provides either upward or downward order. The module can be additionally parameterized with values for α and β used in $s(x)$ ($param_2$ and $param_3$).

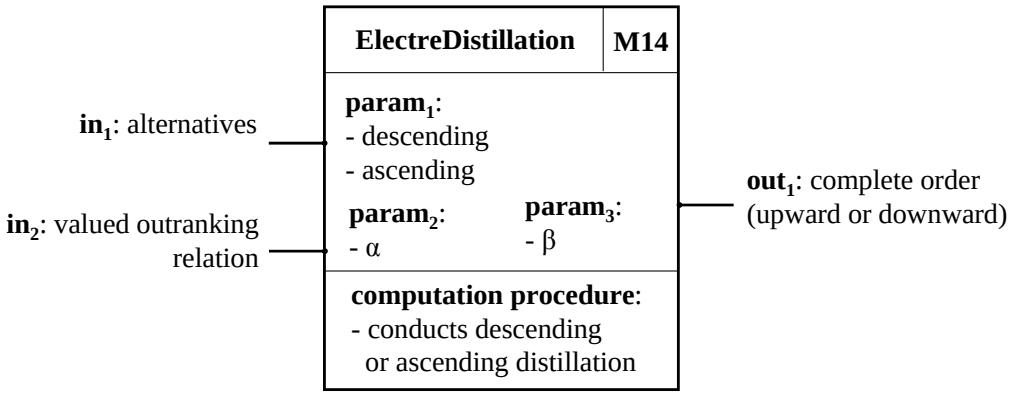

Figure 14: Structure of module M14 which conducts ascending/descending distillation in the set of alternatives.

The upward and downward preorders are subsequently combined to produce a partial final preorder $P = P_D \cap P_A$. Additionally, we may compute ranks of the alternatives which are derived from the number of nodes on the longest path to the top of the final preorder (increased by one). Finally, a median preorder may be constructed by ordering the alternatives using their ranks. For these with the same rank, one investigates the differences between their positions in P_D and P_A, and ranks higher these alternatives for which this comparison is more advantageous.

Module M15: ElectreDistillationRank. This module (see Figure 15) computes a final partial preorder (out_1), ranks of the alternatives (out_2) in the final preorder, and a median preorder (out_3). It requires the user to provide downward (in_2) and upward (in_3) orders of alternatives derived from the descending and ascending distillation procedures, respectively.

3.3. Sorting Problems

3.3.1. Multiple Criteria Sorting with Boundary Class Profiles

In the ELECTRE Tri-B method [39], the assignment of an alternative $a \in A$ to a certain class results from the comparison of a with the boundary profiles P_B which define the class limits. To determine

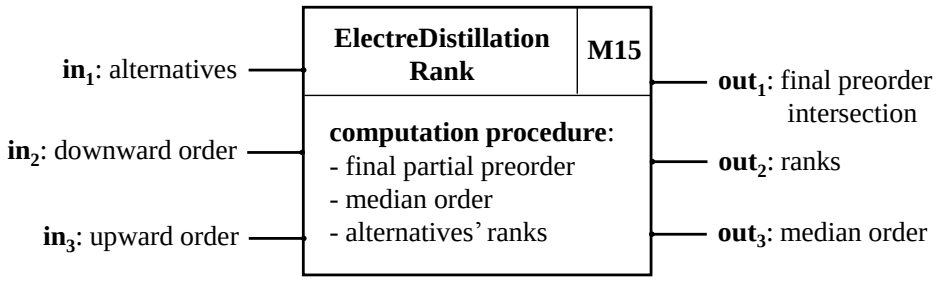

Figure 15: Structure of module M15 which computes a final partial preorder, a median preorder, and ranks of the alternatives based on the distillation outcomes.

an assignment for $a \in A$, two disjunctive assignment procedures (called pessimistic (pseudo-conjunctive) and optimistic (pseudo-disjunctive) procedures) exploit the binary relations ($\succ$, I, and R) between a and boundary profiles in P_B.

Pessimistic assignment procedure. To indicate the class in which a can be assigned to according to the pessimistic rule, compare a successively to p_h^b, for $h = t-1, \ldots, 0$, seeking the first boundary profile p_h^b such that aSp_h^b. Then, select C_{h+1}.
Optimistic assignment procedure. To indicate the class in which a can be assigned to according to the optimistic rule, compare a successively to p_h^b, for $h = 1, \ldots, t-1$, seeking the first boundary profile p_h^b such that $p_h^b \succ a$. Then, select C_h.

Module M16: ElectreTriClassAssignments. This module (see Figure 16) computes the pessimistic (out_1) and optimistic (out_2) assignments for the set of alternatives. It needs to be provided with a crisp outranking relation (in_4) for the set of alternatives and boundary class profiles.

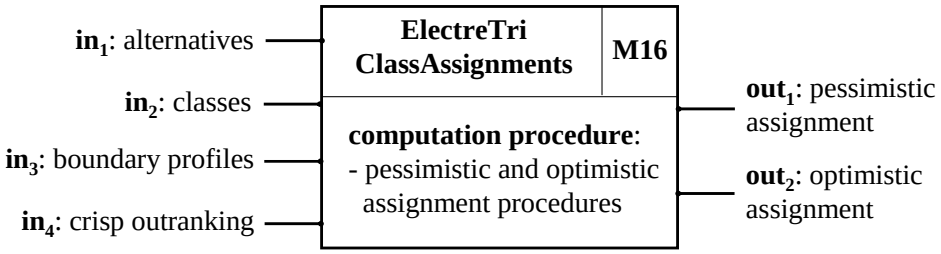

Figure 16: Structure of module M16 which computes pessimistic and optimistic class assignments for the alternatives using ELECTRE Tri-B assignment procedures.

3.3.2. Multiple Criteria Sorting with Characteristic Class Profiles

In the ELECTRE Tri-C method [2], alternatives are compared with the characteristic reference profiles P_C rather than the boundary ones. ELECTRE Tri-C applies the descending and ascending assignment rules to indicate the lower and upper classes to which an alternative could be assigned for a particular outranking model.

Descending assignment procedure. The descending rule compares a successively to p_h^c, for $h = t+1, \ldots, 0$, searching for the first profile p_h^c, such that $S(a, p_h^c) \geqslant \lambda$, i.e.:

(i) If $h = t$, select C_t as a class to assign alternative a.

(ii) If $0 < h < t$: if $S(p_h^c, a) > S(a, p_{h+1}^c)$, then select C_h; otherwise, choose C_{h+1}.

(iii) If $h = 0$, select C_1.

Ascending assignment procedure. The ascending rule compares a successively to b_h, for $h = 1, \ldots, t+1$, searching for the first characteristic profile p_h^c, such that $S(p_h^c, a) \geqslant \lambda$, i.e.:

(i) If $h = 1$, select C_1 as a class to assign alternative a.

(ii) If $1 < h < (t+1)$: if $S(a, p_h^c) > S(p_{h-1}^c, a)$, then select C_h; otherwise, choose C_{h-1}.

(iii) If $h = (t+1)$, select C_t.

The order of outcomes of the descending and ascending assignment rules vary, i.e. with some outranking models the ascending rule indicates an assignment to a better class than the descending rule, whereas with other models the order is inverse (for details, see [19]). Thus, these two procedures need to used conjointly to indicate a possibly imprecise class interval to which $a \in A$ is assigned.

Module M17: ElectreTri-CClassAssignments. This module (see Figure 17) computes an interval class assignment (out_1) for the set of alternatives by combining the outcomes of the descending and ascending procedures of ELECTRE Tri-C. It needs to be provided with a crisp outranking relation (in_5) for the set of alternatives and boundary class profiles as well as the underlying credibility degrees (in_4).

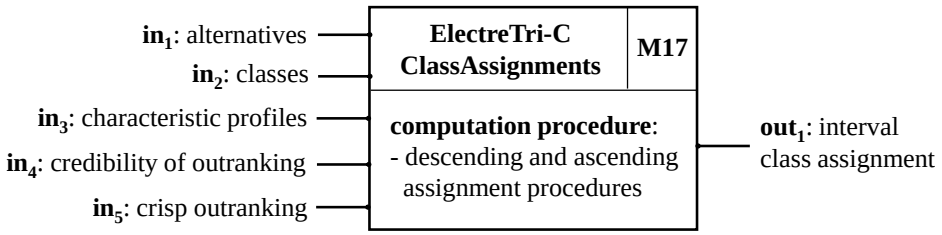

Figure 17: Structure of module M17 which computes interval class assignments for the alternatives using descending and ascending rules of ELECTRE Tri-C.

Similarly to ELECTRE Tri-C, the ELECTRE Tri-rC method derives its recommendation from the comparison of alternatives with the characteristic class profiles [19]. It employs, however, the revised assignment procedures which indicate the extreme class assignments unambiguously (i.e., using univocal and precise conditions) and allow easier indirect inference of the parameters of an outranking model. The following procedures indicate the worst and the best classes for the possible assignment of $a \in A$:

Procedure for indicating the worst class. To indicate the worst class in which a can be assigned to, compare a successively to b_h, for $h = t-1, \ldots, 0$, seeking the first characteristic profile b_h such that:

$$a \succ b_h \text{ and } c(a, b_{h+1}) > c(b_h, a). \tag{43}$$

Select C_{h+1}.

Procedure for indicating the best class. To indicate the best class in which a can be assigned to, compare a successively to b_h, for $h = 2, \ldots, t+1$, seeking the first characteristic profile b_h such that:

$$b_h \succ a \text{ and } c(b_{h-1}, a) > c(a, b_h). \tag{44}$$

Select C_{h-1}.

Module M18: ElectreTriClassAssignments. This module (see Figure 18) computes an interval class assignment (out_1) for the set of alternatives by applying the assignment procedures of ELECTRE Tri-rC. It needs to be provided with the same inputs as M17.

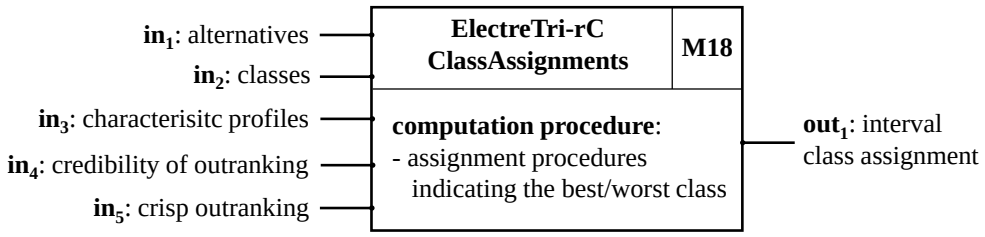

Figure 18: Structure of module M18 which computes interval class assignments for the alternatives using joint assignment rules of ELECTRE Tri-rC.

4. Construct Your Own ELECTRE Method in *diviz*

4.1. diviz

diviz is an open-source which allows to design, execute, and share complex workflows implementing procedures of decision analysis [20]. The software infrastructure consists of:

- a Java client for algorithmic workflow design and visual analysis of the outcomes,
- distant servers for executing the workflows, i.e., computing the results.

MCDA procedures as well as visualization or reporting tools are available in *diviz* via XMCDA web-services. They need to read inputs and write outputs formatted using the XMCDA standard. XMCDA represents MCDA concepts (e.g., alternative, class, outranking relation, or veto threshold) using general data structures referred to as MCDA types which are coded in XML with tags and attributes (e.g., *alternativesComparisons*, *categoriesProfiles*, or *criteriaValues*). In this way, the standard ties together all the web-services, making their interoperability possible. Indeed, the web-services can be combined into complex workflows even if they are written by different programmers in various programming languages.

4.2. Workflow Design

Procedures for construction and exploitation of an outranking relation in the spirit of ELECTRE have been implemented and made available on *diviz* as a collection of individual modules (components). They have been described in detail in Sections 2 and 3. In this section, we will refer to the specific modules using their numbers, i.e., M1-M18.

All our modules are written in Python, which is a high-level, general-purpose, and multi-paradigm programming language. Its design enhances expressiveness, code readability, and quick prototyping. It is also perceived as a language which is perfectly suited for systems integration. All aforementioned features make it suitable for preparing the modules that are meant to inter-operate within a framework such as *diviz*.

The design of decision analysis workflows in *diviz* is performed via an intuitive graphical user interface. Each component is represented by a rectangular box which can be linked to data files or other computation modules. Thus, the design of the workflow does not require any programming skills, but rather understanding the role of each module [20]. To construct a workflow, the user chooses the modules (s)he is interested in from the list of available elements. Using a "drag-and-drop" function, (s)he adds them to the workspace along with the data files. Subsequently, the inputs and outputs of different components can be

linked using connectors to define the structure of the workflow. Once the design is finished, it is possible to execute the workflow. As already mentioned, the underlying calculations are performed on computing servers through the use of the XMCDA web-services. Thus, *diviz* requires connection to the Internet.

The outputs and inputs of different modules need to be connected so that the underlying data types are the same. Only in this way, one can effectively turn chains of components into a complete method. In Figure 19, we present a general scheme for construction of one's own ELECTRE from the modules we provide. This scheme consists of two main parts corresponding to the construction and exploitation of an outranking relation. The former one is additionally divided into four parts corresponding to carrying out concordance and (non-)discordance tests, and constructing a valued or crisp outranking relation. In the scheme, we refer to different concepts using the notation introduced in Sections 2 and 3. For each concept (separate box), we indicate which module should be used for its implementation referring to the symbols M1-M18. Any possible path in Figure 19 corresponds to a valid ELECTRE method. Some of them allow reconstruction of the existing approaches. For example, by combining:

- M1 ($C^S(a,b)$), M4 ($D^V(a,b)$), M11 (S^{Iv}), and M12 (graph kernel), one obtains ELECTRE Is;
- M1 ($C^S(a,b)$), M5 ($d^{VP}(a,b)$), M7 ($S^{DC}(a,b)$), M14, and M15 (distillation), one obtains ELECTRE III,
- M1 ($C^S(a,b)$), M5 ($d^{VP}(a,b)$), M7 ($S^{DC}(a,b)$), and M16, one obtains ELECTRE Tri-B.

Nevertheless, there are only few paths corresponding to the existing methods, while the rest can be followed to construct new approaches possibly better suited for dealing with a particular decision problem.

Note that the possibly multiple outcomes of each component can be viewed either in *diviz* or in an external web-browser. This allows better understanding of the steps involved in the workflow as well as more effective troubleshooting, when designing a new method. Another important feature of *diviz* is history of past executions, which also contributes to the better understanding of methods execution, but more importantly, it allows to precisely calibrate the parameters of individual components. Moreover, *diviz* allows to construct several methods (possibly sharing the same components) within a single workflow. In this way, the software enhances the comparison of results they provide. Finally, *diviz* enables to export any workflow as an archive (i.e., single file containing all necessary information including input data). This archive can be subsequently shared with other users, who can then import it (by loading the archive) into their software and continue the development of the workflow or execute it on the original data.

5. Illustrative Case Studies

In this section, we illustrate how construction of one's own ELECTRE method can be incorporated in a decision aiding process. We approach two real-world problems with the original combinations of ELECTRE-based components that so far have not been considered in the literature. Firstly, we reconsider the problem of evaluating mass transit systems [10] in view of multiple criteria ranking and choice. Secondly, we sort a set of products into classes of specific inventory management policy [16] using either boundary or characteristic profiles. For all problems, we have constructed dedicated *diviz* workflows which are available online[1]:

- *electreProductsRanking.dvz* for Section 5.1.1;

[1] `http://www.cs.put.poznan.pl/mkadzinski/diviz/electre`; when the paper is accepted for publication, the workflows will be made available at the official website of the project: `http://www.decision-deck.org/diviz/workflows.html`

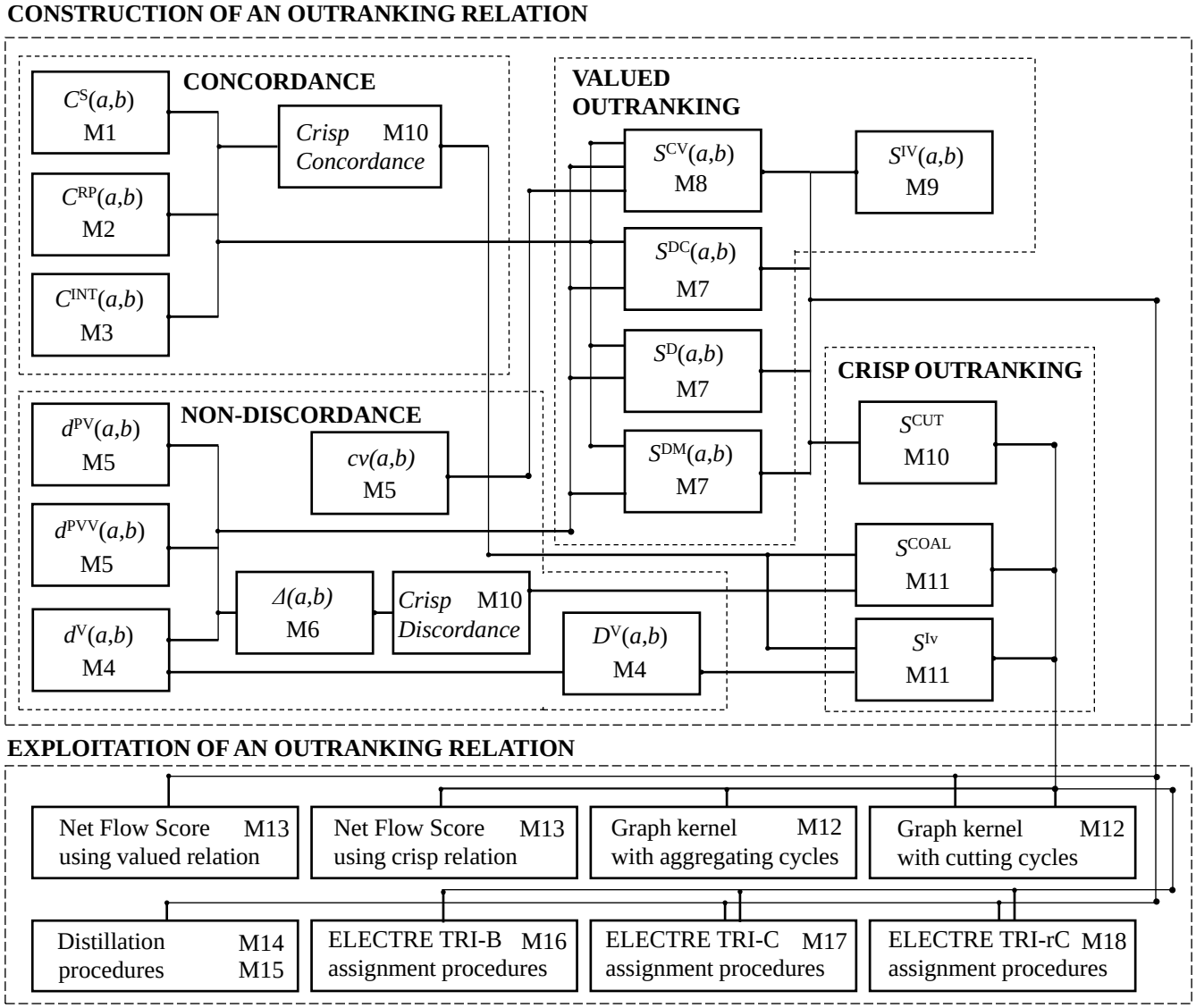

Figure 19: Construct your own ELECTRE in *diviz*.

- *electreProductsChoice.dvz* for Section 5.1.2;
- *electreProductsSortingBoundary.dvz* for Section 5.2.1;
- *electreProductsSortingCharacteristic.dvz* for Section 5.2.2.

They can be used to reproduce the results discussed in this section. For this purpose: 1) download *diviz*[2], 2) launch it, 3) import the workflow ("Workflow - Import as new"), and 4) run it on *diviz* ("Execution - Run"). Moreover, the workflows illustrate how to prepare the input data and put together the *diviz* modules in very diverse settings so that they can be later easily adapted to other problems at hand.

5.1. Multiple Criteria Ranking and Choice: Evaluation of Mass Transit Systems

We reconsider the problem of evaluating mass transit systems (MTSs) operating in nine major European cities [10]. Each of these systems can be perceived as a set of organized components that carries out

[2] `http://www.cs.put.poznan.pl/mkadzinski/diviz/`; after the official release of the new version of *diviz* (scheduled for mid-May 2015) the software will be available at `http://www.decision-deck.org/diviz/download.html`

passenger transportation services within the urbanized area. The multiple criteria evaluation of MTSs aims to support the DMs in selecting the most desirable public transportation solutions for metropolitan areas. The constructed family of criteria includes characteristics of technical, economical, and social character as well as the interests, requirements, and expectations of passengers, operators, and local authorities. It is composed of the following seven factors [10]:

- Accessibility of the MTS (C1; km/km^2; to be maximized) - density of the public transportation network in the metropolitan area;
- Degree of crowdedness (C2, %; to be minimized) - an overall level of the capacity utilization of vehicles used in the MTS in the peak hours;
- Commercial speed of transportation means (C3, km/h; to be maximized) - a weighted average of operational speed of all transportation modes used in the MTS;
- Quality of the fleet (C4; pts; to be maximized) - an aggregated criterion composed of several measures such as: average age of vehicles, percentage of low floor vehicles in the fleet, technical reliability of transportation means, and special equipment used in the vehicles to increase the comfort of travel;
- Safety of the MTS (C5; to be minimized) - a ratio of the total number of accidents caused by MTS per total number of inhabitants of the considered metropolitan area;
- Financial efficiency (C6; %; to be maximized) - a percentage share of subsidies in total operational costs generated by the particular MTS;
- Waiting time (C7; minutes; to be minimized) - an aggregated waiting time of passengers traveling by MTS in the peak hours.

The performance matrix is provided in Table 1.

Table 1: Performance matrix for the problem of evaluating mass transit systems in nine European cities.

City	Abbrev.	C1	C2	C3	C4	C5	C6	C7
Barcelona	BCN	4.15	45	21.9	9	44.0	49.9	2
Brussels	BRU	1.22	26	21.6	2	54.0	55.35	3.5
Helsinki	HEL	3.09	32	26.5	10	25.0	43.49	4
Lisbon	LIS	4.23	31	26.1	10	12.0	56.5	4.5
London	LON	2.51	82	26.3	11	31.0	61.61	2.5
Oslo	OSL	3.17	19	23.1	14	7.0	44.54	3.5
Paris	PAR	1.84	46	21.8	6	25.0	37.8	2.5
Prague	PRA	2.04	40	26.1	6	40.0	62.91	3.5
Warsaw	WAR	1.51	70	20.1	3	10.6	53.85	5

In the following subsections, we will consider this problem in terms of multiple criteria ranking or choice. In both settings, the intra- and inter-criterion parameters are the same. When conducting the concordance test, we refer to the indifference and preference thresholds, whereas in case of verifying the discordance, we employ both pre-veto and veto thresholds. Depending on the characteristics of criteria, we refer to the thresholds which are either constant or defined with an affine function. The assumed values for all thresholds and importance coefficients used in the concordance test are provided in Table 2.

5.1.1. Multiple Criteria Ranking by Exploiting a Valued Outranking Relation

In this section, we will first construct a valued outranking relation and then exploit it so that to order the MTSs from the best to the worst. At the phase of construction of an outranking relation, we take into

Table 2: The indifference, preference, pre-veto, and veto thresholds, and weights used in the concordance test for all criteria for the problem of evaluating mass transit systems in nine European cities.

g_j	q_j	p_j	pv_j	v_j	w_j
C1	0.2	0.5	2.0	3.0	6
C2	$0.1g_2(a)$	$0.2g_2(a)$	$0.35g_2(a)$	$0.5g_2(a)$	5
C3	$0.02g_3(a)$	$0.05g_3(a)$	$0.1g_3(a)$	$0.2g_3(a)$	3
C4	1.0	3.0	7.0	10.0	4
C5	$0.2g_5(a)$	$0.3g_5(a)$	$0.5g_5(a)$	$0.7g_5(a)$	2
C6	$0.05g_6(a)$	$0.1g_6(a)$	$0.25g_6(a)$	$0.35g_6(a)$	4
C7	0.5	1.0	1.5	2.5	7

account interactions between the criteria. For example, we assume that the relative weight of quality of the fleet (C4) and financial efficiency (C6) considered together should be larger than 8 ($= w_4 + w_6 = 4 + 4$) as a result of a mutual strengthening effect. The interaction coefficients are provided in Table 3. To capture the interaction effects in the ambiguity zone, we use $Z_{ij}^{ab,multi}$.

Table 3: Interaction coefficients for the problem of ranking mass transit systems.

Type	Criteria	Coefficient
mutual strengthening	C4, C6	$w_{46}^{MS} = 2.0$
mutual weakening	C1, C3	$w_{13}^{MW} = -1.5$
antagonistic	C7, C2	$w_{72}^{A} = 3.0$

When aggregating the results of the concordance and discordance tests into a credibility of outranking relation, for each pair of alternatives $(a, b) \in A \times A$ we account for the maximal marginal discordance index $d_j^{PVV}(a, b)$. As a result, a valued outranking relation is materialized with $S^{DM}(a, b)$. When exploiting these credibility indices, we employ both distillation and NFS procedures.

The workflow that we have constructed for this purpose is presented in Figure 20. For clarity, we will discuss its structure in detail. To save space, such explanation will be omitted for other illustrative studies subsequently presented in this section. To support understanding of the connections between different modules, we have enriched Figure 20 with a clear structure of the workflow while referring to a notation used throughout the paper. It makes evident the specific implementation of both construction and exploitation phases of an outranking relation.

The workflow contains the following input files: *cities.xml* (specification of all MTSs), *criteria.xml* (specification of all criteria along with the indifference, preference, pre-veto, and veto thresholds), *interactions.xml* (listing of interaction effects and coefficients), *performances.xml* (performance table), and *weights.xml* (weights used in the concordance test). All ELECTRE modules are parameterized so that to compare alternatives against themselves rather than with (boundary or characteristic) class profiles. Further parameterization ensures that:

- ElectreConcordanceWithInteractions employs $Z_{ij}^{ab,multi}$ rather than $Z_{ij}^{ab,min}$ to capture interaction effects in the ambiguity zone;
- ElectreCredibility aggregates the results delivered by ElectreConcordanceWithInteractions and ElectreDiscordance using S^{DM} rather than S^{DC} or S^{D};
- ElectreNFSOutranking aggregates arguments in favor and against each MTS derived from a valued outranking relation rather than from the crisp one;
- one of the ElectreDistillation modules performs a downward distillation, whereas the other conducts an upward distillation.

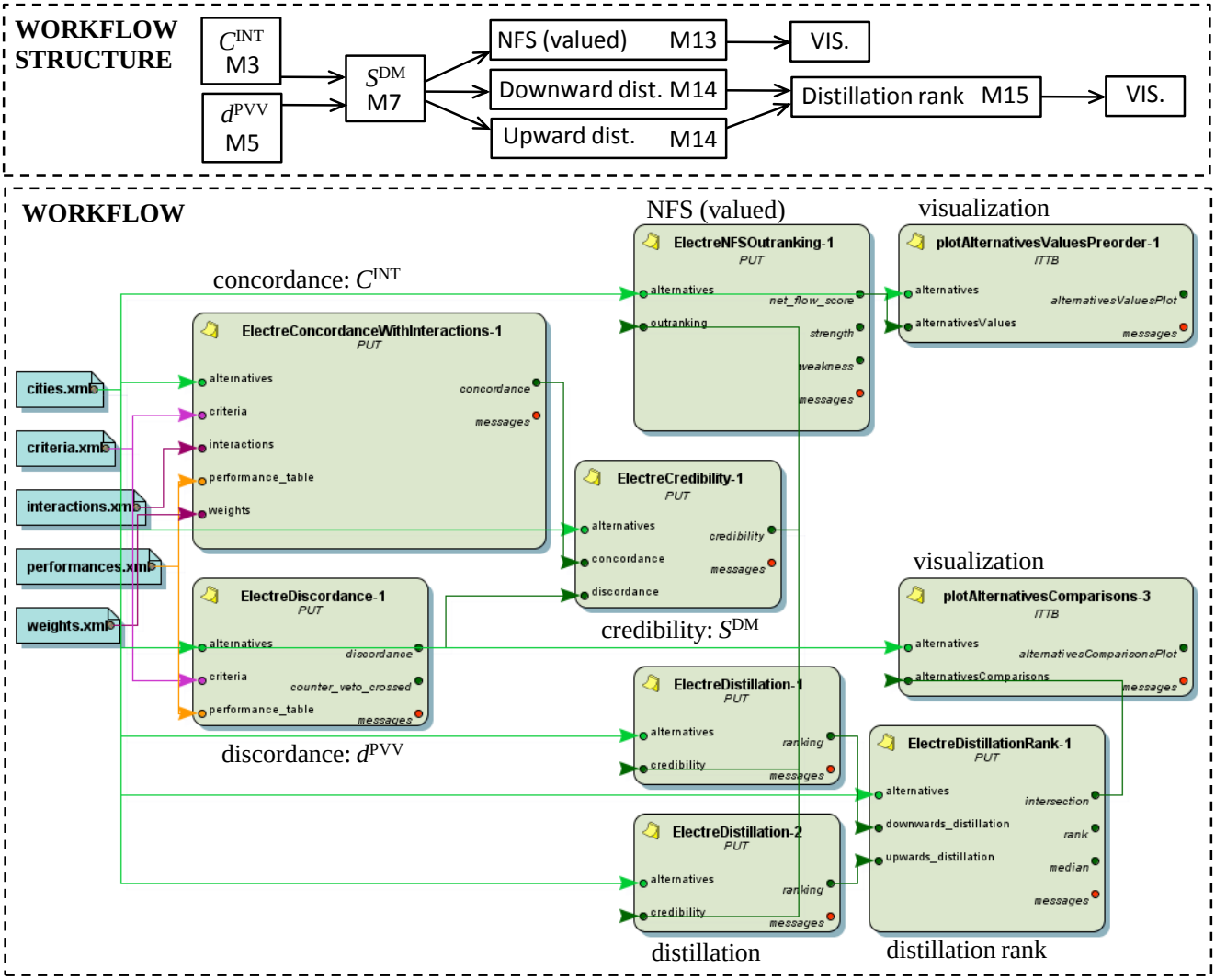

Figure 20: Workflow for multiple criteria ranking of mass transit systems.

For graphical presentation of the outcomes of ranking procedures, we use a pair of visualization modules (plotAlternativesValuesPreorder and plotAlternativesComparisons) which are available in *diviz*.

diviz exhibits the elementary results obtained at each phase of the workflow. With such modular structure of the workflow, it is easy to verify the impact of different effects or assumptions on the partial results. For example, for a pair (BCN,BRU):

- the comprehensive concordance indices with or without considering interactions between criteria are, respectively, C^{INT}(BCN,BRU) = 0.6604 and C^{S}(BCN,BRU) = 0.7009;
- the credibility degrees of outranking taking into account all partial discordance indices or only the maximal discordance index are, respectively, S^{D}(BCN,BRU) = 0.1649 and S^{DM}(BCN,BRU) = 0.1712.

In what follows, we focus on the final rankings. When it comes to the results obtained with NFS, the strength, weakness, and scores are provided in Table 4. The ranking determined by NFS^{Sval} is depicted in Figure 21a) (the graph was obtained with plotAlternativesValuesPreorder module). Comprehensively, LIS with the greatest strength and OSL with the least weakness prove to be the best, whereas WAR with the least strength and PAR with the greatest weakness occupy the bottom ranks.

Table 4: Results of the Net Flow Score procedure (strength, weakness, and scores), ranks and median order derived from the distillation procedure for the problem of evaluating mass transit systems in nine European cities.

	NFS (valued relation)			ELECTRE III	
City	Strength	Weakness	NFS^{Sval}	Rank	Median
BCN	1.834	1.787	0.047	3	3
BRU	1.170	2.846	-1.676	4	4
HEL	2.244	2.627	-0.383	4	5
LIS	4.898	1.305	3.594	1	1
LON	1.359	1.542	-0.183	3	3
OSL	3.446	1.146	2.300	2	2
PAR	1.059	4.084	-3.025	5	6
PRA	2.782	1.633	1.149	3	3
WAR	1.000	2.822	-1.823	6	7

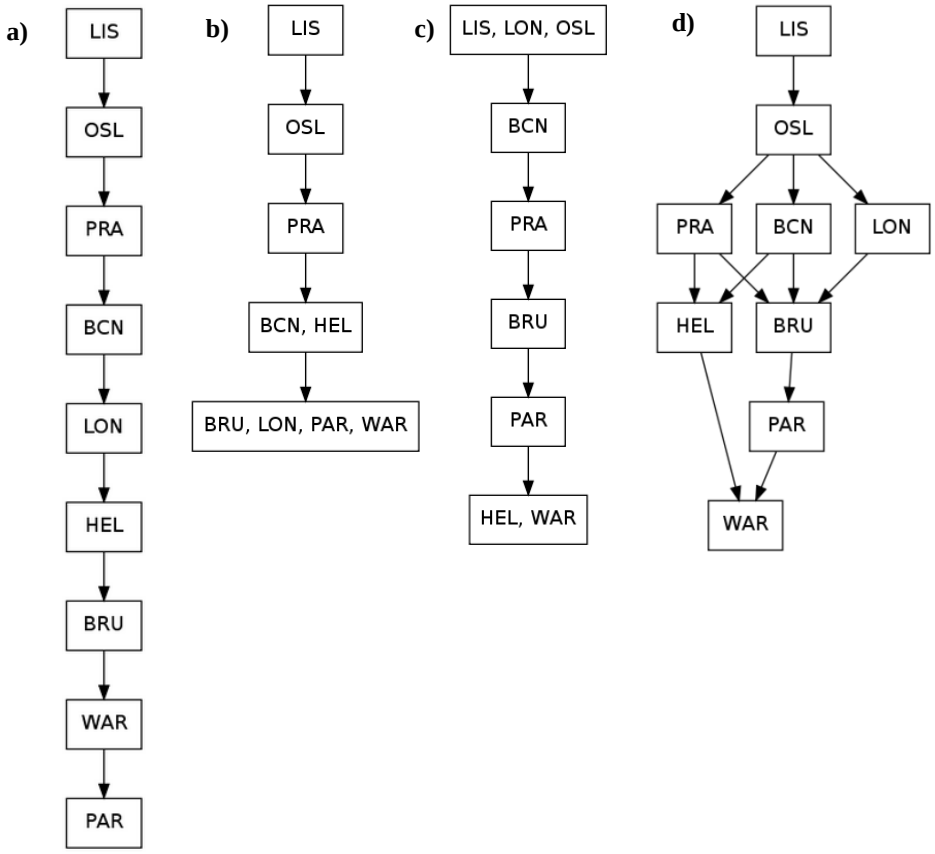

Figure 21: Orders of MTSs derived from a) NFS procedure, b) downward distillation, c) upward distillation, and d) final ranking in ELECTRE III.

As far as outcomes typical for ELECTRE III are concerned, the downward and upward orders are presented in Figures 21b) and c), respectively. On one hand, when constructing the ranking from the top, LIS is judged the best, while BRU, LON, PAR, and WAR turn out to be the worst. On the other hand, when starting from the bottom, HEL and WAR are clearly worse than other MTSs, whereas LIS, LON, and OSL are considered superior to the remaining cities. These two orders are combined into a partial final preorder (see Figure 21d); the graph was obtained with plotAlternativesComparisons module parameterized so that to reduce arcs that can be obtained with transitivity). Note that some pairs of MTSs are incomparable, because their relation in the downward and upward preorders is not univocal. The ranks derived from the final preorder are provided in Table 4 (column "Rank"). When compared to the ranking obtained with NFS, the order of some cities is inverse (see, e.g., PAR and WAW). Even though the ranks of HEL and

BRU are the same, the superiority of BRU over HEL in the upward preorder outweighs its loss in the downward preorder. This is reflected in the median order (see Table 4, column "Median") where BRU is preferred to HEL.

5.1.2. Multiple Criteria Choice by Exploiting a Crisp Outranking Relation

In this section, we will first construct a crisp outranking relation and then exploit it so that to select a subset of the best of MTSs. When constructing a crisp relation, we consider separately the strength of the coalition of criteria supporting the assertion and against an outranking. For the purpose of illustration, when computing the comprehensive concordance index, we use the same thresholds and weights as in the previous subsection, but neglect the interactions between criteria. When aggregating marginal discordance indices into a comprehensive discordance index Δ^{CD}, we use the same weights w_j^D for all criteria ($w_j^D = 1$, $j = 1, \ldots, 7$), thus, assigning them equal powers to veto an outranking relation. The concordance λ^C and discordance λ^D majority thresholds are set to 0.7 and 0.2, respectively. When exploiting thus constructed crisp outranking relation, we search for the kernel in the underlying graph and select the best MTSs with NFS.

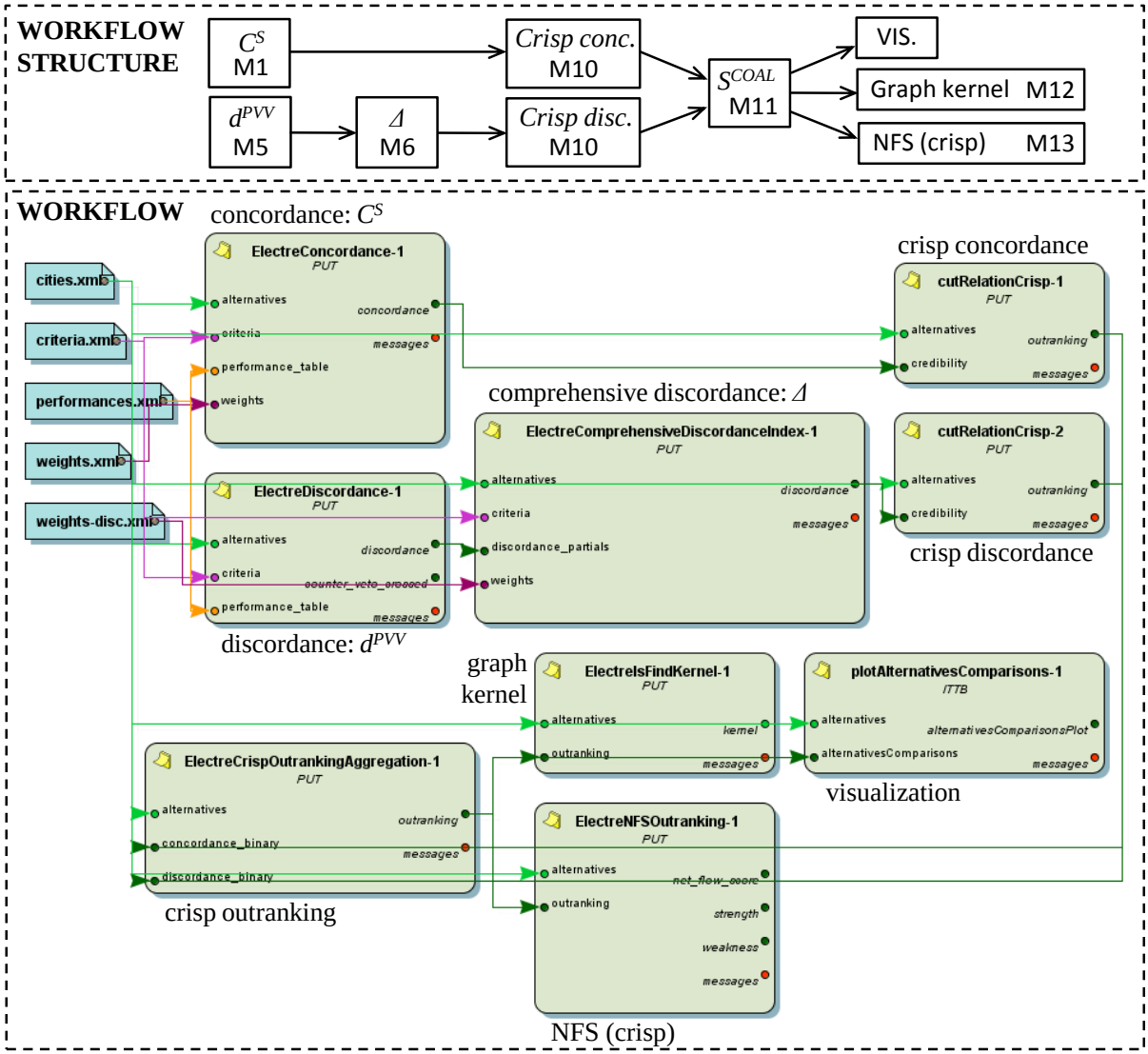

Figure 22: Workflow for multiple criteria choice of mass transit systems.

The workflow constructed for this purpose along with its intuitive structure is presented in Figure 22.

We use the same input files as in the previous subsection with the proviso that listing of the interaction effects is replaced with specification of the discordance weights w_j^D (*weights-disc.xml*). The sequential character of the workflow is best visible when ElectreCrispOutrankingAggregation puts together the outcomes of concordance and discordance tests so that to determine a crisp outranking relation.

The graph of an outranking relation is presented in Figure 23 (the graph was obtained with plotAlternativesComparisons module parameterized so that to draw all arcs). Its kernel identified with ElectreIsFindKernel is composed of LIS, LON, and OSL. The same cities proved to be the best in terms of NFS with $NFS^S = 4$. On the contrary, WAR and PAR attain the worst scores equal to -7 and -6, respectively.

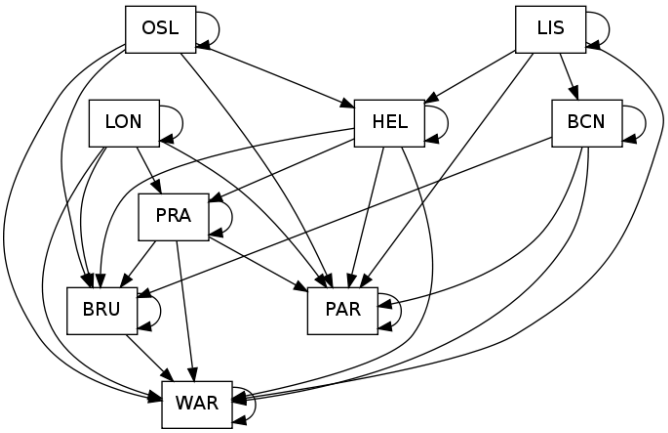

Figure 23: Graph of an outranking relation for the problem of evaluating MTSs.

5.2. Multiple Criteria Sorting: Evaluation of Products

We reconsider the problem of evaluating a set of products (items) within a storage location assignment system [16]. The products are compared with respect to the following features which influence the company's inventory management policy [16]:

- demand (C1; in units between 0 and 1000; to be maximized) - the average frequency of orders by clients;
- size (C2; m^2; to be minimized) - the product's density in the range between 0 and 1;
- profitability (C3; in %; to be maximized) - the financial return of each unit of product;
- consumer's sensitivity (C4; linguistic variables coded in the range between 0 (very little) and 4 (much); to be maximized) - the product's sensitiveness to the level of service, e.g., the slow delivery of orders.

The task consists in assigning each product to a class-based storage. The three considered classes Cl_1-Cl_3 (with Cl_1 being the worst one) represent different levels of operational, storage, and cost efficiency. Thus, the products with great demand, small size, high profitability, and high sensitivity to the level of customer service should be assigned to class Cl_3, and subsequently allocated to the most favorable storage area in the warehouse (close to input/output) [16]. The performance matrix is provided in Table 5. To save space, we consider 25 randomly drawn products out of 50 originally considered in [16].

In the following subsections, we will consider this problem in terms of multiple criteria sorting with either boundary or characteristic class profiles. In both settings, we assume the same values for the indifference, preference, and veto thresholds as well as criteria weights (see Table 6). Additionally, when

Table 5: Performance matrix for the problem of evaluating products, and class assignments obtained with the boundary (Tri-B) and characteristic (Tri-rC) class profiles.

					Assignments		
	Criteria				ELECTRE Tri-B		Tri-rC
	C1	C2	C3	C4	Pessimistic	Optimistic	Class interval
a1	50	0.060	70	4	2	3	1-2
a2	675	0.006	33	1	2	3	2-3
a4	350	0.014	100	4	3	3	3-3
a5	120	0.054	10	3	1	1	1-1
a9	450	0.026	90	3	3	3	3-3
a11	100	0.135	5	2	1	1	1-1
a13	55	0.282	75	2	1	2	1-2
a15	35	0.514	76	1	1	2	1-2
a17	190	0.111	53	2	2	2	2-2
a19	80	0.288	63	3	1	1	1-1
a23	420	0.069	47	0	2	2	2-2
a24	130	0.231	30	3	1	1	1-1
a26	670	0.051	100	2	3	3	3-3
a27	255	0.149	41	1	1	1	1-1
a29	590	0.076	65	4	3	3	3-3
a31	420	0.117	85	2	2	2	3-3
a33	280	0.189	86	1	1	2	2-3
a36	840	0.071	34	1	2	3	2-3
a39	380	0.171	65	3	2	2	2-2
a40	130	0.500	50	2	1	1	1-1
a42	915	0.077	5	0	1	3	1-3
a45	230	0.357	53	2	1	1	1-1
a46	730	0.116	60	4	3	3	3-3
a48	970	0.093	46	2	3	3	3-3
a50	145	0.620	32	1	1	1	1-1

comparing the products with class profiles, we acknowledge a bonus in case of very strong preference of one object over another. Precisely, we consider the effect of reinforced preference on $C1$ and $C4$ (see Table 6 for the reinforced preference thresholds rp_j and reinforcement factors ω_j). The cutting level, representing the minimum credibility degree which implies the truth of a crisp outranking relation, is set to $\lambda = 0.55$.

Table 6: The indifference, preference, veto, counter-veto, and reinforced-preference thresholds, reinforcement factors, and weights used in the concordance test for all criteria for the problem of evaluating products within a storage assignment location system.

g_j	q_j	p_j	v_j	cv_j	rp_j	ω_j	w_j
C1	10	20	300	500	100	1.2	3
C2	0.01	0.03	0.20	0.40	–	–	2
C3	2	5	25	40	–	–	1
C4	0	1	3	–	3	1.5	1

5.2.1. Multiple Criteria Sorting with Boundary Class Profiles

Let us first approach the problem of assigning the products to class-based storage by comparing them against the class limits. Since we consider three decision classes, it is sufficient to define two boundary profiles: p_1^b (the boundary between Cl_1 and Cl_2) and p_2^b (the boundary between Cl_2 and Cl_3). Their performances on all criteria are provided in Table 7.

Table 7: Boundary class profiles for the problem of evaluating products within a storage assignment location system.

p_j^b	C1	C2	C3	C4
p_1^b	300	0.200	50	1.5
p_2^b	450	0.130	65	2.5

The workflow for this particular problem is presented in Figure 24. All *diviz* modules are parameterized to compare alternatives with the boundary class profiles. Their definition and specification of the performances are provided in separate input files, respectively, *boundary.xml* and *boundary-perf.xml*. While accounting for the reinforced preference within the concordance test, it might be of interest to the DM to check the influence of this effect on the comprehensive concordance index. For example, for a pair $(p_2^b, a1)$ its value increases from 0.4286 (when not considering the reinforced preference) to 0.4737, while when comparing p_2^b with $a2$ the effect of reinforced preference does not occur, and, thus, $C^{RP}(p_2^b, a2) = C^S(p_2^b, a2)$.

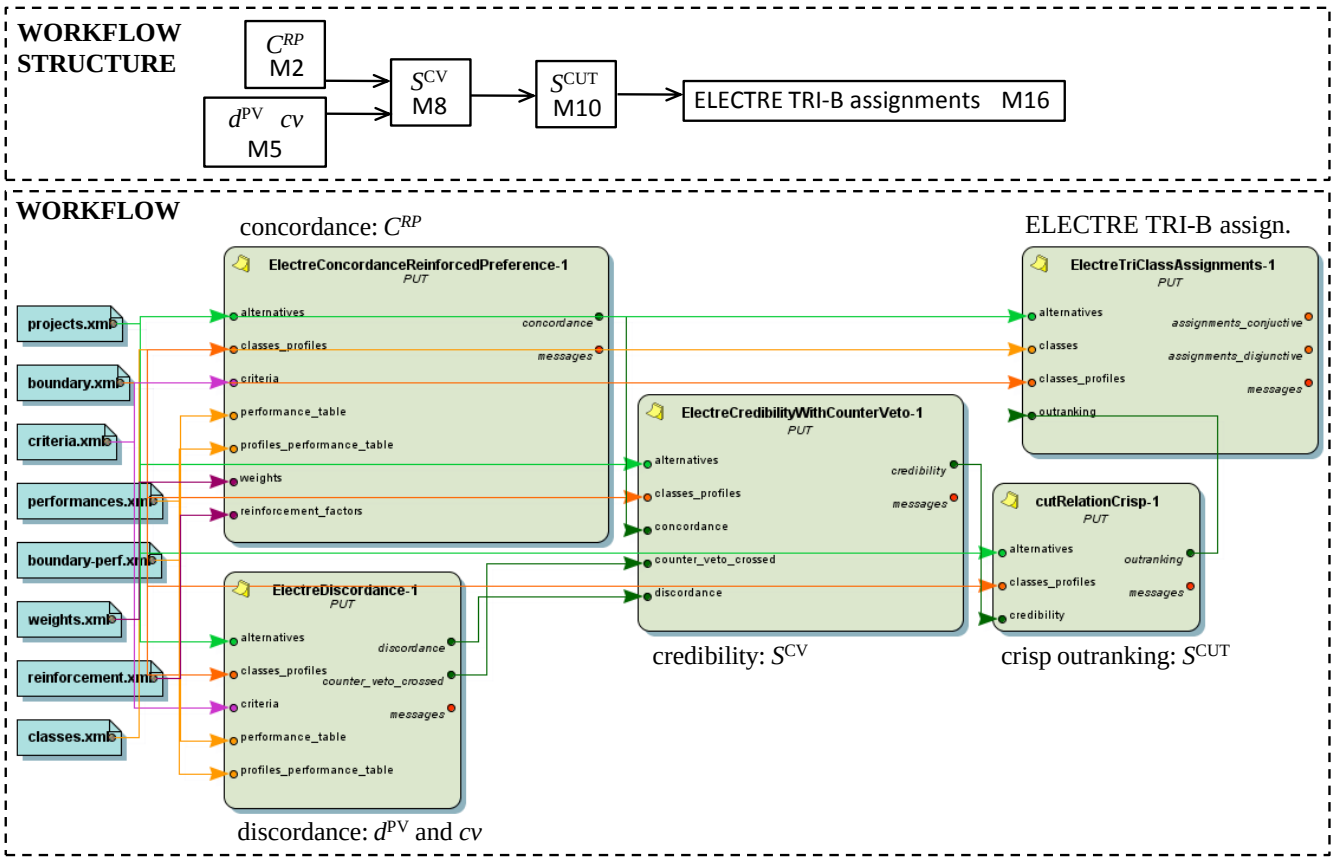

Figure 24: Workflow for multiple criteria sorting of products with boundary class profiles.

When further constructing a valued outranking relation, to weaken the potential impact of veto, we account for the counter-veto effect on three selected criteria (see Table 6 for the counter-veto thresholds cv_j). For each pair of objects, the number of criteria for which the counter-veto threshold is crossed is returned at the output of the ElectreDiscordance module. This happens once for, e.g., $a4$ and p_1^b or $a48$ and p_2^b. These numbers intervene in the computation of credibility S^{CV} of an outranking relation with ElectreCredibilityWithCounterVeto. We assume that a comprehensive non-discordance index is interpreted as Δ^{DC} so that to take into account only the criteria which sufficiently strongly oppose to an outranking. The ElectreTriClassAssignments module which is responsible for computing class assignments with the pessimistic and optimistic procedures of ELECTRE Tri-B takes into account only a crisp outranking relation (i.e., the output of the cutRelationCrisp module) for all pairs composed of a project and a boundary profile. Moreover, only at this final stage of the workflow, one requires formal specification of the decision classes which is provided in *classes.xml*.

Although the pessimistic and optimistic procedures have been mainly used separately, for illustrative purpose, we present their outcomes together in Table 5. Obviously, the assignment indicated by the optimistic rule is always at least as good as indication of the pessimistic one. For our problem, for 17

out of 25 projects the assignments suggested by both rules are the same, and only for one project ($a42$) the recommended classes are not consecutive. For example, since $a1$ is preferred to p_1^b and incomparable with p_2^b, its pessimistic (optimistic) assignment is Cl_2 (Cl_3). The best (worst) class is suggested for 6 and 10 (12 and 8) projects by the pessimistic and optimistic rules, respectively. As noted in [16], the stock management policy calls for prudence, and, thus, the pessimistic assignments are more appropriate for this particular study.

5.2.2. Multiple Criteria Sorting with Characteristic Class Profiles

In this section, the three considered decision classes are defined by a set of characteristic profiles such that p_j^c corresponds to Cl_j, $j = 1, 2, 3$. Their performances are provided in Table 8. These profiles can be used with the assignment rules of ELECTRE Tri-C or ELECTRE Tri-rC. We present the results of applying the latter one.

Table 8: Characteristic class profiles for the problem of evaluating products within a storage assignment location system.

p_j^c	C1	C2	C3	C4
p_1^c	250	0.250	40	1
p_2^c	350	0.150	55	2
p_3^c	500	0.100	70	3

The workflow implementing such scenario is presented in Figure 24. Now, all modules are parameterized so that to compare alternatives with the characteristic profiles. When conducting the discordance test, we use the ElectreIsDiscordanceBinary, thus, computing the binary discordance indices d_j^V and D^V as in the ELECTRE Is method. These ensure that the veto effect occurs iff the loss of one object with respect to another is at least as great as the veto threshold. This can be observed, e.g., for comparison of $a1$ with p_2^c on $C1$. When computing the credibility degree, we account only for the comprehensive discordance index D^V. Thus, the credibility degree is equal to the comprehensive concordance if no veto occurs (e.g., for comparison of $a1$ with p_1^c), or to zero, otherwise (e.g., for comparison of $a1$ with p_2^c). Finally, the assignment rules of ELECTRE Tri-rC require information on both crisp and valued outranking relation for all pairs composed of a project and a characteristic profile. Thus, these two relations need to be provided as the inputs of the ElectreTri-rCClassAssignments module.

The class intervals suggested for all products by the joint assignment rules of ELECTRE Tri-rC are provided in Table 5. For 18 out of 25 products the extreme classes are the same. For example, $a17$ is preferred to p_1^c, indifferent with p_2^c, and preferred by b_2^c, which implies its precise assignment to Cl_2. For the remaining 7 items ($a1$, $a2$, $a13$, $a15$, $a33$, $a36$, and $a42$), the assignment remains ill-determined taking into account the way that the set of characteristic profiles defines the classes [2]. In particular, $a42$ performs relatively well on $C1$ and $C2$, and rather bad on $C3$ and $C4$. When collated with the characteristic profiles which are composed of typical performances for each class, this diversity in the performance vector implies incomparability of $a42$ with all these profiles and results in the imprecise assignment $Cl_1 - Cl_3$.

6. Conclusions

In this paper, we reviewed a wide spectrum of ELECTRE-based concepts and postulated flexibility in their joint consideration. At the stage of construction of an outranking relation, we accounted for different ways of conducting the concordance and (non-)discordance tests, computing the credibility of an outranking relation, and checking the validity of a crisp relation. These procedures were derived from the existing ELECTRE methods (e.g., ELECTRE Is, III, IV, Tri-B, and MR-Sort) as well as proposals which are not linked to any specific approach (e.g., reinforced preference, counter-veto, pre-veto, or simplified

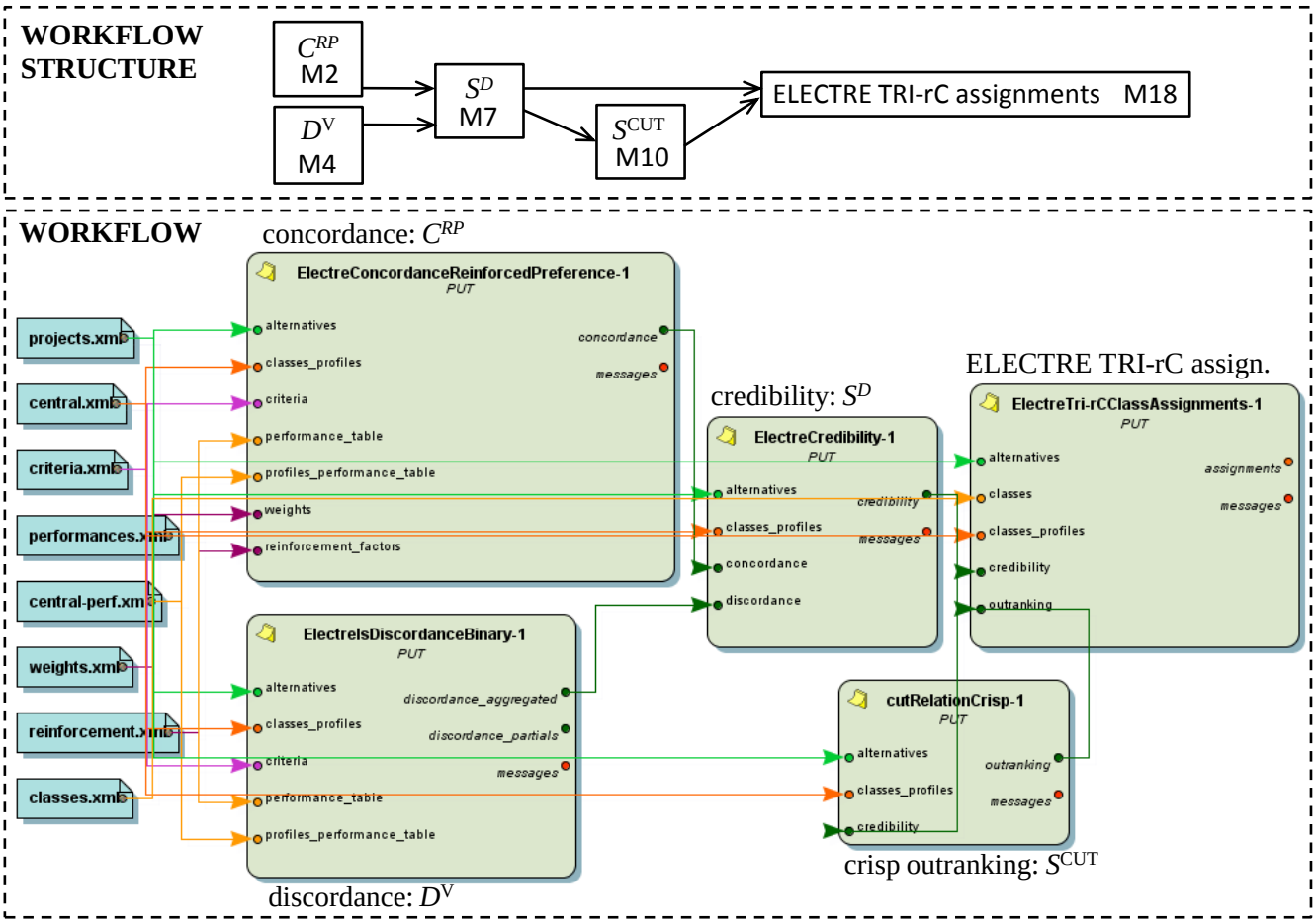

Figure 25: Workflow for multiple criteria sorting of products with characteristic class profiles.

comprehensive discordance). At the stage of exploitation of an outranking relation, we considered several algorithms which can be used to decide the ranking (e.g., distillation of a valued outranking relation as in ELECTRE III or Net Flow Score procedure), or presence in the subset of the best alternatives (e.g., identifying the graph kernel as in ELECTRE Is), or the assignment into the pre-defined and ordered classes (e.g., sorting rules of ELECTRE Tri-B, Tri-C, and Tri-rC).

More importantly, we implemented the postulate of flexibility in designing the ELECTRE methods in practice. We designed several highly parameterized computational modules, each responsible for conducting some construction or exploitation procedure. These modules are universal in a sense that they admit comparison of alternatives either with each other or with class profiles (boundary or characteristic ones). We made them available via the *diviz* platform, which offers the infrastructure for coupling the modules together so that to reconstruct the existing methods or develop one's own ELECTRE which is best suited for a particular decision context. Overall, with our proposal one can consider several hundred valid combinations of modules which can be obtained without any mathematical or programming skills.

We illustrated how a decision aiding process can be supported by construction of new ELECTRE methods by reconsidering two real-world problems. These problems concerned evaluation of either mass transit systems in cities or products within a storage location assignment system. We approached them by designing a few previously not considered variants of ELECTRE. For all accounted scenarios, we have constructed dedicated *diviz* workflows which are attached as the e-Appendices. They can be used to reproduce the results discussed in the paper, but they also serve as examples which can be easily adapted for dealing with other problems. Let us emphasize, however, that when applying any variant of ELECTRE,

an analyst needs to verify if its properties correspond to the characteristics of a decision context [12].

Although the spectrum of ELECTRE-based approaches considered in this paper is very wide, it is by no means exhaustive. Nonetheless, the source code of all modules is available and can be extended by other researchers so that to meet their specific requirements. Moreover, the new modules for *diviz* can be created in any programming language and only adjusted so that to read input and write output in the XMCDA format. The useful extensions of the proposed framework include, but are not limited to:

- accounting for the inverse thresholds [15] in the concordance and discordance tests; currently, when the threshold is defined with an affine function, we consider the direct thresholds which are defined from the worst of the two performances;
- considering group (categorical) credibility indices which allow joint comparison of a single object (e.g., alternative) with a set of other objects (e.g., a set of characteristic profiles defining the same class) as proposed in the ELECTRE Tri-nC method [3];
- adding other exploitation procedures implementing, e.g., the assignment rules of THESEUS [9] or partitioning procedure and constructing the complete pre-orders in the spirit of ELECTRE II [27].

Acknowledgments

The authors thank Patrick Meyer from Telecom Bretagne for helping us to make the software available on the *diviz* platform, and to the whole Decision Deck community for stimulating this work. Miłosz Kadziński and Michał Tomczyk acknowledge financial support from the National Science Center (grant no. DEC-2013/11/D/ST6/03056).

References

[1] Ait Younes, A., Azibi, R., Roy, B., 2000. ELECTRE Is : Manuel d'utilisation. Document du LAMSADE no 111, Université Paris-Dauphine.

[2] Almeida Dias, J., Figueira, J., Roy, B., 2010. Electre Tri-C: A multiple criteria sorting method based on characteristic reference actions. European Journal of Operational Research 204 (3), 565 – 580.

[3] Almeida-Dias, J., Figueira, J., 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.

[4] Benayoun, R., Roy, B., Sussman, B., 1966. ELECTRE: une méthode pour guider le choix en présence des points de vue multiples. Note de travail 49, SEMA-METRA International, Direction Scientifique.

[5] Bottero, M., Ferretti, V., Figueira, J., Greco, S., Roy, B., 2015. Dealing with a multiple criteria environmental problem with interaction effects between criteria through an extension of the Electre III method. European Journal of Operational Research (in press), –.

[6] Bouyssou, D., Marchant, T., Pirlot, M., Tsoukiàs, A., Vincke, P., 2006. Evaluation and decision models with multiple criteria: Stepping stones for the analyst, 1st Edition. International Series in Operations Research and Management Science, Volume 86. Springer, Boston.

[7] Cailloux, O., 2010. ELECTRE and PROMETHEE MCDA methods as reusable software components. In: Proceedings of the 25th Mini-EURO Conference on Uncertainty and Robustness in Planning and Decision Making (URPDM 2010). Coimbra, Portugal, pp. 359–368.

[8] Corrente, S., Greco, S., Kadziński, M., Słowiński, R., 2013. Robust ordinal regression in preference learning and ranking. Machine Learning 93 (2-3), 381–422.

[9] Fernandez, E., Navarro, J., 2011. A new approach to multi-criteria sorting based on fuzzy outranking relations: The THESEUS method. European Journal of Operational Research 213 (2), 405 – 413.

[10] Fierek, S., Żak, J., Solecka, K., Kruszyński, M., 2012. Multiple criteria evaluation of the mass transit systems in European cities. Logistyka 2, 509–522.

[11] Figueira, J., Greco, S., Roy, B., 2009. ELECTRE methods with interaction between criteria: an extension of the concordance index. European Journal of Operational Research 199, 496–505.

[12] 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, 61–85.

[13] Figueira, J., Mousseau, V., Roy, B., 2005. ELECTRE methods. In: Greco, S., Ehrgott, M., Figueira, J. (Eds.), Multiple criteria decision analysis: State of the art surveys. Springer, pp. 133–153.

[14] Figueira, J., Roy, B., 2002. Determining the weights of criteria in the ELECTRE type methods with a revised Simos procedure. European Journal of Opera- tional Research 139, 317326.

[15] Figueira, J., Roy, B., 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 (1), 9 – 20.

[16] Fontana, M., Cavalcante, C., 2013. Electre Tri method used to storage location assignment into categories. Pesquisa Operacional 33, 283 – 303.

[17] Greco, S., Kadziński, M., Mousseau, V., Słowiński, R., 2011. ELECTRE^{GKMS}: Robust ordinal regression for outranking methods. European Journal of Operational Research 214 (1), 118–135.

[18] Ishizaka, A., Nemery, P., 2013. Multi-criteria Decision Analysis: Methods and Software. Wiley, New York.

[19] Kadziński, M., Tervonen, T., Figueira, J., 2014. Robust multi-criteria sorting with the outranking preference model and characteristic profiles. Omega (in press), –.

[20] Meyer, P., Bigaret, S., 2012. diviz: a software for modeling, processing and sharing algorithmic workflows in MCDA. Intelligent Decision Technologies 6, 283–296.

[21] Mousseau, V., Dias, L., 2006. Valued outranking relations in ELECTRE providing manageable disaggregation procedures. European Journal of Operational Research 156, 467–482.

[22] Mousseau, V., Słowiński, R., P.Zielniewicz, 1999. ELECTRE TRI 2.0a: Methodological guide and users documentation. Document du LAMSADE no 118 and 188bis, Université Paris-Dauphine.

[23] Roy, B., 1968. Classement et choix en presence de points de vue multiples (la methode ELECTRE). La Revue d'Informatique et de Recherche Opérationelle (RIRO) 8, 57–75.

[24] 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.

[25] Roy, B., 1990. The outranking approach and the foundations of ELECTRE methods. In: Bana e Costa, C. (Ed.), Readings in Multiple Criteria Decision Aid. Springer-Verlag, Berlin, pp. 155–183.

[26] Roy, B., 1996. Multicriteria Methodology for Decision Aiding. Kluwer Academic, Dordrecht.

[27] Roy, B., Bertier, P., 1971. La methode ELECTRE II. Note de travail 142, SEMA-METRA International, Direction Scientifique.

[28] Roy, B., Bouyssou, D., 1993. Aide Multicritère à la Décision : Méthodes et Cas. Economica, Paris.

[29] Roy, B., Figueira, J., 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 (0), 9 – 20.

[30] Roy, B., Present, M., Silhol, D., 1986. A programming method for determining which Paris metro stations should be renovated. European Journal of Operational Research 24, 318–334.

[31] Roy, B., Skalka, J., 1984. ELECTRE IS : Aspects méthodologiques et guide d'utilisation. Tech. rep., Document du LAMSADE N^o 30, Université Paris-Dauphine, Paris.

[32] Roy, B., Słowiński, R., 2008. Handling effects of reinforced preference and counter-veto in credibility of outranking. European Journal of Operational Research 188, 185–190.

[33] Roy, B., Vincke, P., 1984. Relational systems of preference with one or more pseudo-criteria: some new concepts and results. Management Science 39, 1323–1335.

[34] Sobrie, O., Pirlot, M., Mousseau, V., 2014. New veto rule for a sorting model, 20th Conference of the International Federation of Operational Research Societies.

[35] Szelag, M., Greco, S., Słowiński, R., 2014. Variable consistency dominance-based rough set approach to preference learning in multicriteria ranking. Information Sciences 277 (1), 525 – 552.

[36] Vallee, D., Zielniewicz, P., 1994. ELECTRE III-IV, version 3.x, aspects methodologiques (tome 1), guide dutilisation (tome 2). Document du LAMSADE no 85 and 85bis, Université Paris-Dauphine.

[37] Vincke, P., 1999. Robust solutions and methods in decision-aid. Journal of Multi-Criteria Decision Analysis 8 (3), 181–187.

[38] Weistroffer, H., Smith, C., Narula, S., 2005. Multiple criteria decision support software. In: Multiple Criteria Decision Analysis: State of the Art Surveys. Vol. 78 of International Series in Operations Research & Management Science. Springer New York, pp. 989–1009.

[39] Yu, W., 1992. ELECTRE TRI : Aspects méthodologiques et manuel d'utilisation. Document du LAMSADE no 74, Université Paris-Dauphine.